INTERNATIONAL STANDARD

ISO 11783-10

Second edition
2015-09-15

Tractors and machinery for agriculture and forestry — Serial control and communications data network —

Part 10: Task controller and management information system data interchange

Tracteurs et matériels agricoles et forestiers — Réseaux de commande et de communication de données en série —

Partie 10: Contrôleur de tâches et échange de données des systèmes d'information de gestion

Reference number
ISO 11783-10:2015(E)

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT) see the following URL: Foreword - Supplementary information

The committee responsible for this document is ISO/TC 23, *Tractors and machinery for agriculture and forestry*, Subcommittee SC 19, *Agricultural electronics*.

This second edition cancels and replaces the first edition (ISO 11783-10:2009) which has been technically revised.

ISO 11783 consists of the following parts, under the general title *Tractors and machinery for agriculture and forestry — Serial control and communications data network*:

— *Part 1: General standard for mobile data communication*

— *Part 2: Physical layer*

— *Part 3: Data link layer*

— *Part 4: Network layer*

— *Part 5: Network management*

— *Part 6: Virtual terminal*

— *Part 7: Implement messages application layer*

— *Part 8: Power train messages*

— *Part 9: Tractor ECU*

— *Part 10: Task controller and management information system data interchange*

— *Part 11: Mobile data element dictionary*

— *Part 12: Diagnostics services*

— *Part 13: File server*

— *Part 14: Sequence control*

Introduction

This International Standard specifies a communications system for agricultural equipment based on the ISO 11898-1 protocol. SAE J1939[1] documents, on which parts of ISO 11783 are based, were developed jointly for use in truck and bus applications and for construction and agriculture applications. Joint documents were completed to allow electronic units that meet the truck and bus SAE J1939[1] specifications to be used by agricultural and forestry equipment with minimal changes.

General information on this International Stamdard is to be found in ISO 11783-1. The purpose of this International Standard is to provide an open, interconnected system for on-board electronic systems. It is intended to enable electronic control units (ECUs) to communicate with each other, providing a standardized system.

The International Organization for Standardization (ISO) draws attention to the fact that it is claimed that compliance with this part of ISO 11783 may involve the use of a patent concerning the controller area network (CAN) protocol referred to throughout the document.

ISO takes no position concerning the evidence, validity, and scope of this patent.

The holder of this patent has ensured ISO that he is willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with ISO. Information may be obtained from:

Robert Bosch GmbH

Wernerstrasse 51

Postfach 30 02 20

D-70442 Stuttgart-Feuerbach

Germany

Attention is drawn to the possibility that some of the elements of this part of ISO 11783 may be the subject of patent rights other than those identified above. ISO shall not be held responsible for identifying any or all such patent rights.

Tractors and machinery for agriculture and forestry — Serial control and communications data network —

Part 10: **Task controller and management information system data interchange**

1 Scope

This International Standard as a whole specifies a serial data network for control and communications on forestry or agricultural tractors and mounted, semi-mounted, towed, or self-propelled implements. Its purpose is to standardize the method and format of transfer of data between sensors, actuators, control elements, and information storage and display units, whether mounted on, or part of, the tractor or implement. This part of ISO 11783 describes the task controller applications layer, which defines the requirements and services needed for communicating between the task controller (TC) and electronic control units. The data format to communicate with the farm-management computer, the calculations required for control, and the message format sent to the control function are defined in this part of ISO 11783.

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 11783-1, *Tractors and machinery for agriculture and forestry — Serial control and communications data network — Part 1: General standard for mobile data communication*

ISO 11783-3, *Tractors and machinery for agriculture and forestry — Serial control and communications data network — Part 3: Data link layer*

ISO 11783-5, *Tractors and machinery for agriculture and forestry — Serial control and communications data network — Part 5: Network management*

ISO 11783-6, *Tractors and machinery for agriculture and forestry — Serial control and communications data network — Part 6: Virtual terminal*

ISO 11783-7, *Tractors and machinery for agriculture and forestry — Serial control and communications data network — Part 7: Implement messages application layer*

ISO 11783-11, *Tractors and machinery for agriculture and forestry — Serial control and communications data network — Part 11: Mobile data element dictionary*

ISO 11783-12, *Tractors and machinery for agriculture and forestry — Serial control and communications data network — Part 12: Diagnostics services*

ISO 11898-1, *Road vehicles — Controller area network (CAN) — Part 1: Data link layer and physical signalling*

ISO/IEC 10646, *Information technology — Universal Coded Character Set (UCS)*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 11783-1 apply.

3.1
client
control function (CF) that establishes a connection with a task controller (TC) or a data logger (DL) and provides data for data logging by the DL or TC and/or accepts control commands from the TC

Note 1 to entry: This term has been introduced in version 4 to replace the term working set for the CF that connects to a TC or DL. The multi member working set concept is not applied to TC or DL clients. The communication between a TC or DL and their clients is limited to transmission of messages between TC or DL and the working set master

3.2
coding data
data that changes infrequently, such as machine or chemical data, or data which is referenced in a task for the purpose of allocating tasks administratively

3.3
data dictionary entity
description of the data dictionary identifier, definition, value range, value resolution, and units specifications used by process data variables

3.4
data logger
DL
control function (CF) defined specifically to perform data logging functionality

3.5
data logger (DL) client
CF that establishes a connection with a data logger (DL) and provides data for data logging by the DL

3.6
data transfer file set
collection of files in the extensible markup language format and binary formats, which are used for the data transfer between the farm management information system and the task controller of an ISO 11783 network

3.7
device descriptor object pool
DDOP
collection of device-related objects and their relationships which together describe the functionality and structure of a device for the purposes of task control and data logging

3.8
device element
any addressable item on a device

EXAMPLE Nozzle on sprayer boom where the nozzle has individually addressable process data variables.

3.9
field
area of land managed by a farmer, represented by either a single partfield or a collection of more than one partfield

Note 1 to entry: The field is only of importance within the farm management information system for business management considerations and is not necessarily related to a single crop.

3.10
grid cell
rectangular areas defined by overlaying a grid on a partfield

3.11
management computer gateway
CF that interfaces to the management computer system and to the ISO 11783 network

Note 1 to entry: A management computer gateway can store data for transmission at a later time.

3.12
partfield
area characterized by the cultivation of only one agricultural crop

Note 1 to entry: Partfield is the XML element to which tasks are allocated to obtain smallest granularity.

3.13
polygon
planar surface defined by one exterior boundary and by zero or more interior boundaries

Note 1 to entry: Each interior boundary describes a hole in the surface.

Note 2 to entry: A single or group of polygons can be used to define a treatment zone.

3.14
process data variable
element of information that has a value that describes the state of a process

Note 1 to entry: Process data variables consist of the attributes range, resolution, and units, as defined in the data dictionary.

3.15
setpoint value source
CF that can determine a setpoint value during a task for use by another CF and contains one or more Device Process Data object(s) with the property attribute set to the “control source” value

3.16
setpoint value user
CF that accepts a setpoint value from either the TC or another source to modify its real-time performance such as rate control and contains one or more Device Process Data object(s) with the property attribute set to the “settable” value

3.17
task controller (TC) client
CF that establishes a connection with a task controller (TC) and provides data for data logging and/or accepts control commands from the TC

3.18
task controller (TC) number
identification number that is derived from the function instance of the task controller (TC)

3.19
XML element
representation of the data of a domain object and consists at least of an opening tag, a number of attributes, and a closing tag

4 Abbreviations

For the purposes of this document, the abbreviations given in ISO 11783-1 apply.

5 General description

5.1 Task management

There are two main purposes of task management in the mobile implement control system.

The first is the management of the farm resources, including tractors, implements, sensor systems, workers, and the products used. It is possible for the farmer to plan and evaluate the use of the resources. He is then able to automatically control the use of his inventory of products and can keep track of the status and conditions of his machinery. Resource designators are globally transferred as coding data and are part of the data transfer file set as detailed in Clause 7.

The second purpose is the management of the farm activities carried out in the fields. These activities are described by tasks to distinguish all the work that is planned, is in execution, or has been done by the farmer or by a contractor for one customer in one partfield.

The data transfer is possible in two directions. The planned tasks are sent to the task controller (TC) on the mobile implement control system (MICS) and the results of the work are sent back to the farm-management information system (FMIS). Tasks can be generated both on the FMIS and on the MICS.

Task management has the following workflow.

a) Planning field tasks and maintaining coding data using the software of an FMIS computer operated by the farmers or contractors, as detailed in 5.2.

b) Converting the task data into XML format.

c) Assigning the task data produced by the planning software to the data required for the implements or sensor systems to be used to complete the planned tasks. This step is optional.

d) Transferring the task data from the FMIS system to the TC of the MICS, as detailed in 5.4.

e) The TC uses the task data to transmit process data messages to the ECUs on the implement.

f) The TC collects data according to the DataLogTriggers specified in the task data.

g) Transferring the collected data to the FMIS. Collected data can be in the XML format or in a proprietary format. When a proprietary format is used, this step involves converting the proprietary format into the XML format.

h) Reading the XML files and converting into the FMIS format for storage and evaluation of the data.

Figure 1 illustrates the interfaces between the software on the FMIS computer and the ECUs mounted on an ISO 11783-configured implement.

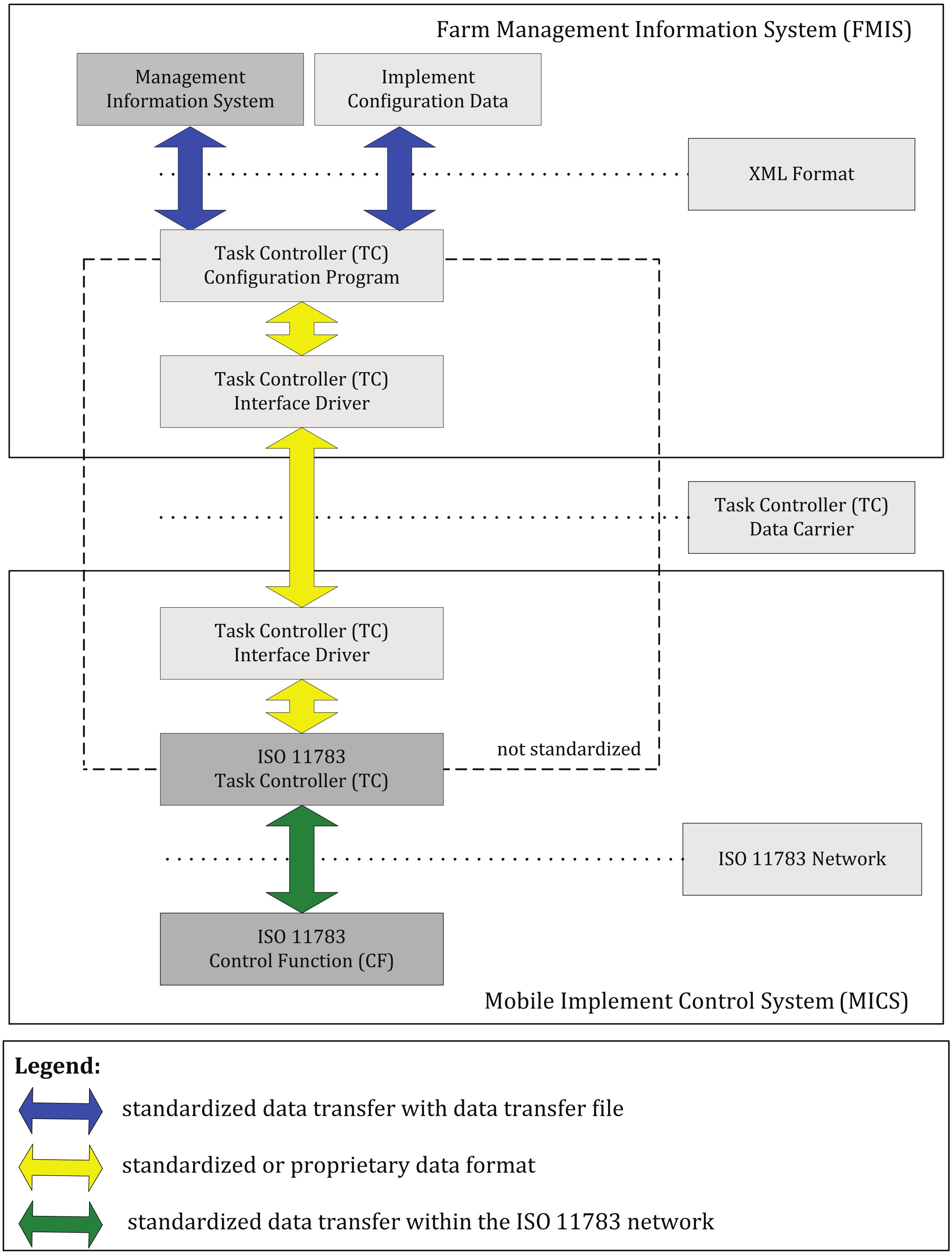


Figure 1 — Task management entities and interfaces

5.2 Task management on FMIS computer

Task management is defined as a part of an FMIS, responsible for planning and evaluation of field work. Tasks specify what, where, how, by whom, and when work is planned to be carried out.

The amount of data transferred between the FMIS and the MICS is dictated by the administrative requirements of a farming enterprise. For the recording of field activities only, the task management can be used to file the data in a working journal. For this purpose, only the coding data need be transferred from FMIS to MICS, and the tasks are created on the MICS by selection of the involved resources. In this case, only the data transfer file set from MICS to FMIS contain tasks. In enterprises where tasks are planned on the FMIS, these will be included, together with the coding data, in the data transfer file set from FMIS to MICS. These planned tasks can range from mere planned allocations of resources to geographical information for site-specific field operations.

5.3 Preselection and assignment of Devices

Any client device in the mobile system can only be identified uniquely by its CF NAME. For FMIS purposes, the client CF NAME needs to be unique worldwide. This implies that the device manufacturer has the responsibility to ensure that the Identity Number in combination with the other fields in the NAME uniquely identifies the device.

At the FMIS, the preselection of devices depends on the planned task. It can be necessary to assign a type of device or function, a specific device, or even the device of a particular manufacturer. The DeviceAllocation XML elements may include planned assignments about the devices to be used. This information ranges between specific and indistinct.

The XML attribute ClientNAMEValue contains the eight-byte NAME of a control function as it is defined in ISO 11783-5. Not all parts of a NAME need be specified and only certain elements of a NAME need to be defined to determine a device on the mobile network. Those parts of ClientNAMEValue containing information to be used on the mobile system to select a proper device are masked by a bitset structure stored in the XML attribute, ClientNAMEMask. All different combinations of elements of the NAME structure can be marked as valid for device selection (logical AND). On the FMIS, these masks could be coded as symbols. Once the preselection information is set in a task on the FMIS, it is not overwritten on the mobile system because the device that is used during task processing on the mobile system is stored as the XML attribute DeviceIdRef.

5.4 Task controller interface driver

After generating the interface files, the TC driver of the manufacturer of the TC is activated on the farm computer. This driver is responsible for the data transfer to the TC, which is part of the MICS using its proprietary data format or the ISO 11783-10 XML format and data carrier, such as any type of memory card or radio link. The translation of the data from the data transfer file set in messages on the ISO 11783 network, as well as the kind of transfer between the mobile system and the FMIS, is not subject to a standardized procedure as part of this International Standard. To make implement-specific set point data, this driver can add and use device descriptor data supplied by the manufacturers.

5.5 Task controller user interface

A TC can provide a means for the user to interact with the TC. User interaction can be through a virtual terminal (VT) or other interface. Operator interfaces can range from very simple to elaborate, depending on the designer’s intent. For example, a simple TC that only supports single tasks that run automatically might not require any user interaction. More advanced TCs can offer additional operator interface capabilities, such as

— select a task from a list,

— start/stop/resume/complete a task,

— modify a task,

— create a task, and

— add new coding data.

Through the user interface, an operator can react to special circumstances or events in order to execute tasks in a reasonable way. The operator can also be informed about the status and results of the tasks and their components. For example, the operator could print a work confirmation for a farmer.

5.6 Data logger function

A separate data logger (DL) can be installed on a network, e.g. to function as a telemetry data logger. The task controller protocol is used by such a telemetry data logger which enables this data logging only control function to use device descriptors and process data messages for collecting data in addition to logging data from any other parameter groups that are broadcast or can be requested on the ISO 11783 network.

This DL CF is identified on the network by a specific data logger function, as specificied in ISO 11783-1. The DL functionality is a subset of the TC functionality and uses the same connection mechanism as the TC control function. Reusing the TC to TC client communication protocol makes the data logging functionality and the process data definitions of the TC available for use with the DL CF, without interfering with the TC client controlling functionalities of the TC.

The data logger function can be used for general non-task-related data logging. The use of the data logger function is not restricted to a telemetry environment.

The data logger function is introduced in ISO 11783-10 version 4. The ISO 11783-10 version is communicated on the network in the Version message (B.5.3) and in the data transfer file set in the VersionMajor attribute of the ISO11783_TaskData XML element (D.32).

6 Task controller requirements

6.1 Task selection and execution

The TC can provide a mechanism for the selection and shall provide a mechanism for the execution of a task contained in the data transfer file set. Selection of an individual task can be done either by the operator through an operator interface or automatically by the TC. The means of task selection is not specified by this part of ISO 11783. The TC designer is free to decide how task selection is implemented. The method provided for starting and stopping a task is not subject to standardization. This functionality is also to be determined by the TC designer.

On a MICS, a task may or may not be selected. If a task has not been selected by the operator, then the TC may prompt the operator for a task selection or may select a task automatically.

The status of a task is defined in Table 1.

Table 1 — Task status

Planned	Task is prepared at FMIS or MICS but not yet processed on MICS.
Running	Task is currently being processed on MICS. Only one task can be active per TC at the same time.
Paused	Task was previously running, is not currently running, and is not yet completed. This state can be a task final state from a TC perspective. Tasks with this state in data transfer file set with a MICS origin are required to be processed by the FMIS.
Completed[a]	Task is finished. This status can only be set by the operator and cannot be set automatically by the MICS. Tasks with this state in data transfer file set with a MICS origin are required to be processed by the FMIS.
Template[b]	Task is a template task. When a template task is started, a copy of the template is made and the copy is started as a new task. Template tasks that are defined by the FMIS shall not be modified by the MICS. Template tasks that are created on the MICS shall be identified as new tasks in the data transfer file set. Once a template task that is created on the MICS is instantiated in a non-template task, then the template task shall not be modified.

Table 1 *(continued)*

Canceled[b]	The purpose of this state is to be able to withdraw a task. The FMIS or MICS can use this status to signal the MICS or FMIS to withdraw a task.

[a] The completed task status is optional. Some TCs may not support this status.

[b] This definition is introduced in ISO 11783-10 version 4.

6.2 Logging of time and position

There can be a need to assign some optional information, such as date, time, and GPS position data, to events that occur during task processing. This could be an event reflecting the interconnection of XML elements, such as the allocation of GPS position-related comments and/or flags or the assignment or exchange of a worker, for example. Other events are based on logging of received process data variable values from task-related clients, which shall be supplied with information of date, time, and position.

The XML elements AllocationStamp and Time enable the allocation of time and date values to several XML elements where needed. These XML elements can be of the type planned or effective, for specifying whether an event was planned or accomplished. Additionally, at a task level within the Time XML element only, more detailed time types, such as preparation time, ineffective time, repair time, or clearing time, are defined. Prior to ISO 11783-10 version 4, all time and date values had to be specified in local time and date values inside this XML element. In version 4 and later, time zone information may be added to the time and data values.

Within a Task XML element, multiple instances of the Time XML element can be included. Examples of when this occurs are when a task with a Time XML element of type planned is started and a new Time XML element of type effective is added or when a task is resumed and a new Time XML element of type effective is added. Time XML elements shall only be added to a Task XML element upon task status changes, upon resource assignment changes to tasks, and upon the TC start up or shut down events. A product allocation is also an example of a resource assignment change. Previously recorded Time XML elements shall not be modified.

Optionally, the AllocationStamp and Time XML elements can include up to two position XML elements that contain GNSS-supplied information. The first position XML element is added when the AllocationStamp or Time XML element is created and the second position XML element is added when the AllocationStamp or Time XML element is completed. If there is only a single position XML element present, then it specifies the position at the start of the AllocationStamp or Time XML element. The intermediate positions along a continuous CommentAllocation are not specified in the AllocationStamp XML element itself but are recorded in the binary log files specified by the TimeLog XML element. See D.10 for the position recording definition of continuous comments.

To be able to allocate multiple process data values, the Time XML element can include several DataLogValue XML elements. The number of these DataLogValues stored inside the task shall be limited to totals or single instance values. For a large number of DataLogValues, the use of the TimeLog XML element and binary log file is defined.

All process data-related logging data can be stored in a binary format as a separate file. The reference to the binary file is to be set by the XML element TimeLog. More TimeLog XML elements can exist per task, referring to external files with unique names inside the name space of the set of tasks belonging to the data transfer file set. The uniqueness of file name prefixes shall be guaranteed by the TC. Each TimeLog file definition results in two separate files — one to contain the binary data and the other to contain the XML coded header structure of the binary data set. The header structure defines the maximum data per binary record and enables the correct interpretation of the binary data. The filename extension of the binary file shall be ".bin"; the filename extension of the XML header structure file shall be ".xml".

6.3 Logging parameters from parameter groups

In addition to the ProcessDataVariable values, the values or parameters from other parameter groups can be logged by a TC. The XML elements DataLogTrigger and DataLogValue contain attributes to specify from which parameter group a value is to be logged. These attributes are optional. When these attributes are specified, the DataLogDDI attribute in DataLogTrigger or DataLogValue shall be set to the ParameterGroupNumberValue (DDI = $DFFE_{16}$). Each parameter group may contain multiple values, and therefore both a parameter group number and a start and stop bit to obtain a single value from the data field of the CAN data frame shall be specified when the TC logs data from parameter groups other than the ProcessDataVariable. The size of the value is at a maximum 32 bits and is stored as the DataLogValue attribute in the XML element DataLogValue.

When the TC logs data from parameter groups, a reference to a DeviceElement is required. If this log data originates from parameter groups other than the ProcessDataVariable, a device with a control function NAME with a filled-in ClientNAME XML attribute and a DeviceElement referring to this device shall be defined. These device and DeviceElement XML elements may be generated by the TC or can be supplied by the FMIS.

6.4 Logging of task events

The planned or effective allocation of resources like Workers, Devices, Products, Comments, and Controls to a Task XML element is specified through an allocation XML element pattern. This allocation pattern is instantiated in the XML elements WorkerAllocation, DeviceAllocation, ProductAllocation, CommentAllocation, GuidanceAllocation, and ControlAssignment. Within a single Task XML element, multiple instances of allocations of these resources can occur. Examples of when multiple allocations are recorded within a single task are when a planned allocation becomes effective upon a task start or when a resource is disconnected and reconnected to a task during execution of that task. Previously recorded resource allocations shall not be modified but new resource allocations can be added to a task.

6.5 Language, formats, and measurement units selection

On an ISO 11783 network, the language, formats, and measurement units which are set and used by a VT control function can be different from the locale settings used by TC or DL control functions. In order for clients to configure their device descriptors to the language, formats, and units used on the TC or DL user interfaces, clients can query these control functions for their standard setup and can adapt the device descriptor text attributes and value presentations to the language, formats, and units of the TC or DL. The device descriptor text attributes and DeviceValuePresentations shall be set such that the TC or the DL can use these values to display configuration and operating information about the client to the operator correctly on the user interface used by the TC or DL.

The following requirements are applicable to TCs and DLs that operate in an ISO 11783 network in which a VT is present.

a) The TC shall send the standard language, format, and measurement units messages defined in ISO 11783-7, hereafter referred to as "standard setups" (Note this term is defined in ISO 11783-6). The TC shall use the standard setup as defined and received from the primary connected VT (VT that will display values from the clients). If a standard setup has not been determined by a connection to a VT (as would be the case in a factory-new TC), the TC shall use its default standard setup defined by the manufacturer until a connection to the VT is established.

b) The TC shall report standard setup during client initialization and at any time when there is a change. The change may occur due to an operator change of the settings of the VT to which the TC is connected or the TC being connected to a different VT with a different standard setup. Standard device descriptor modification methods and rules shall apply to allow the client to modify its device descriptor to match the TCs VT operator-selected language — by updating text fields and DeviceValuePresentation (DVP) values.

c) The TC shall store the standard setups in non-volatile storage and restore the values during initialization prior to connection to a VT. Once a connection with the VT is established, then the TC follows the setup of the VT that it connects to.

d) The TC shall respond to ISO 11783-7 "Language Command" requests sent to the global address (GA). This requirement is introduced in ISO 11783-10 version 4.

e) The TC shall respond to ISO 11783-7 "Language Command" requests directed to the TC. This requirement is introduced in ISO 11783-10 version 4.

If the TC does not make use of a VT, then the TC shall provide a proprietary means to configure its language and units.

The following requirements are applicable to clients:

a) Clients shall configure their standard setup according to the TC to which they are publishing the device descriptor. This requirement is introduced in ISO 11783-10 version 4. ISO 11783-10 version 3 and prior clients followed the VT or T-ECU setup.

b) Clients shall use their default language if the selected language is not supported by the client.

6.6 Connection management

Upon power-up, a specific sequence of events shall occur, in order to ensure proper initialization of the TC and clients, as described in 6.6.1 and 6.6.2 and depicted in Figure 2.

6.6.1 Task controller initialization

The TC shall complete the following initialization.

a) The TC shall complete the address claim procedure in accordance with ISO 11783-5 and shall also send a request for an address claimed to the global destination address (255).

b) The TC shall wait for 6 s after completing the address claim procedure.

c) The TC shall begin transmission of the Task Controller Status message.

d) The TC shall allow clients to upload and initialize their device descriptor object pools.

e) The TC shall request the version of the client after a client has requested the version of the TC. This requirement has been introduced in version 4. Prior to version 4, the TC was not required to request the version of the client.

6.6.2 Client initialization with the task controller

The client shall complete the following initialization.

a) The client shall complete the address claim procedure in accordance with ISO 11783-5.

b) The client shall wait for 6 s after completing the address claim procedure.

c) The client shall wait until its selected TC begins transmission of the Task Controller Status message.

d) The client shall identify itself as a Working Set Master with its members to its selected TC, using the Working Set Master and Working Set Member messages given in ISO 11783-7. Prior to version 3, communication between a TC and a Working Set was not limited to TC to Working Set Master communication only. In version 4 and later, the client is still required to identify itself via the Working Set Master message but the communication with a TC is limited to the Working Set Master as the client CF only.

The client may send these messages for other purposes (e.g. VT initialization).

e) The client shall begin transmission of the Client Task message.

f) A version 4 and later client shall query the version of its selected TC to determine its capabilities with a Request Version message (B.5.2). Prior to version 4, this request version step was recommended when the client intended to use version 3 or later features.

   If the TC does not support the required ISO 11783-10 version or features, then the client shall limit its version and features to those that are available in the TC.

g) A version 4 and later client shall wait for and respond to a Request Version message transmitted by a version 4 and later TC. This enables the TC to process information from the client correctly.

h) The client may request the language and format messages from its selected TC.

i) The client shall query its selected TC to determine if its device descriptor object pool already exists.

j) The client shall either

   1) activate the device descriptor object pool at the TC when the device descriptor object pool already exists in the TC, or

   2) commence and complete a transfer of the device descriptor object pool to its selected TC and activate the device descriptor at its selected TC, accomplished using the messages defined in Annex B and either the transport protocol (see ISO 11783-3) or extended transport protocol (see ISO 11783-3), depending on the size of the device descriptor object pool.

6.6.3 Connection maintenance

The TC shall transmit a cyclic Task Controller Status message at an interval of 2 s. The TC shall also send a Task Controller Status message immediately whenever the task status changes or the value in any of the other Task Controller Status message bytes changes, although at least 200 ms shall elapse between Task Controller Status messages (the maximum transmit rate for Task Controller Status message is 5 Hz). This message includes an indication of the current task status and is sent to the destination-specific global address (GA). If the client does not receive this message for at least 6 s, it assumes a possible uncontrolled shutdown of the TC and stops sending the Client Task message. The client may re-establish connection to the TC by restarting the initialization procedure.

All clients that maintain a connection with a TC shall indicate their presence by transmitting to the TC a cyclic Client Task message at an interval of 2 s. The client shall wait at least 6 s after finishing the address claim procedure before transmitting the Client Task message. This timeout enables the TC to detect a restart of a client. If the TC does not receive this message for at least 6 s, it assumes a possible uncontrolled shutdown of the client.

All time values mentioned in ISO 11783-10, e.g. the 200 ms and 2 s intervals and 6 s time-out are the time values that shall be used for the implementation of this communication protocol. The accuracy of the implementation of these time values is subject to test requirements defined by AEF.[[2]]

The Task Controller Status message and the Client Task message are defined in Annex B.

When a client is restarted, or starts and connects to the TC during an active task, the TC shall accept the upload and activation of the client's device descriptor and issue the measurement commands applicable to this client.

The initialization and shutdown procedures of the connection with the TC are specified in Figure 2. The step "Request Localization Label" shall be skipped when the TC has indicated in the "Structure Label" answer that no device descriptor for the originating ECU is present in the TC.

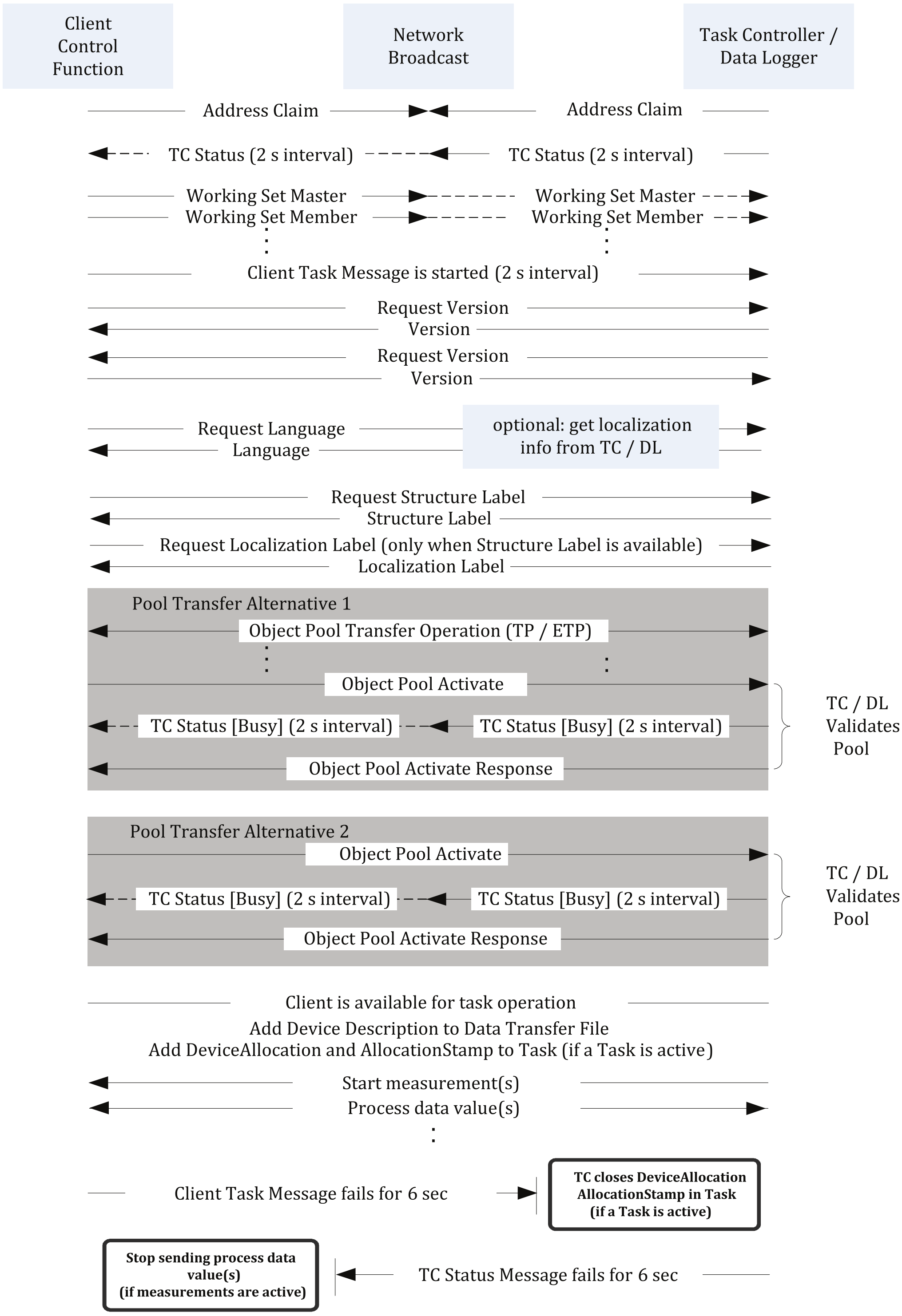


Figure 2 — Initialization and shutdown of connections

6.6.4 Task controller connection shutdown

The system shutdown is defined as the period of time when the Key Switch state indicates the tractor key is off and the ECU Power remains on. Actuator Power may or may not remain on concurrent with ECU Power (see ISO 11783-7).

When the Key Switch state indicates that the tractor key has been turned off, and while ECU Power has not been terminated, the CFs in a system are able to transition to a shutdown state that is appropriate for that system. In some CFs, this can cause immediate termination of all network communications, where other CFs can request power to be maintained for a more orderly shutdown. Others could ignore the Key Switch state and continue normal operation until power is interrupted.

The following practices are introduced in version 4 and are mandatory for version 4 compatible CF implementations. Version 3 and prior CF implementations can be encountered in a system and these are identified and described in the following practices.

6.6.4.1 Task controller shutdown behaviour

The TC can expect that version 3 and prior clients can terminate communications without warning. The mandatory behaviour of the TC is to monitor the Key Switch state and take the following actions as a result of the transition from ”Key switch not Off” to “Key switch Off”.

a) The TC shall disable its “unexpected client shutdown detection logic” to avoid unnecessary notification to the operator as a result of one client shutting down immediately while another client maintains the ECU Power beyond the normal TC 6 s timeout (see 6.6.3).

b) The TC shall maintain services for a minimum of 2 s following the last “Maintain Power” request from those clients that have device descriptor object pools active in the TC. This ensures that data being collected from clients can be finalized correctly during the system shutdown period.

c) The TC shall continue to monitor the Key Switch state and shall reinitialize if “Key Switch Off” changes to ”Key Switch Not Off”. In this case, the TC shall ensure that the Task Controller Status message is discontinued and the standard Initialization process is performed (see 6.6.1).

NOTE ISO 11783-10 version 2 did not specify shutdown behaviour and ISO 11783-10 version 3 did not specify the mandatory requirements, therefore, a version 3 or prior TC may discontinue all communications with the network, including discontinuing the Task Controller Status message.

6.6.4.2 Client shutdown behaviour

Client behaviour can vary significantly depending on the design of the specific client. One variation of a client design implemented according to ISO 11783-10 version 3 and prior is not to monitor the Key Switch state and continue as normal until power is lost.

The mandatory behaviour of a client is to monitor the Key Switch state and take the following actions as a result of the transition from “Key switch not Off” to “Key switch Off”:

a) The client shall send a “Maintain Power“ message (see ISO 11783-7) to inform the system of the state of the client and can use this message as the means to request that ECU Power and/or Actuator Power be maintained.

b) The client monitors the “Maximum time of tractor power“ parameter (see Wheel-based speed and distance message in ISO 11783-7) and uses this information during any power management processes it executes.

c) The client shall send a Connection Deactivate command to the TC to eliminate the possibility of an unexpected shutdown indication (see B.6.10).

d) The client shall not consider the lack of the Task Controller Status message or other TC to client messages as an unexpected shutdown of the TC, and therefore shall not attempt a connection to any other TC as can be available.

e) The client shall continue to monitor the Key Switch state and reinitialize if it changes from ”Key Switch Off” to ”Key Switch Not Off”.

6.6.4.3 Task Resume upon Power cycle

When the TC starts after a power cycle, it shall resume the status that a Task was in before the power cycle occurred. It is recommended that all TC functions are resumed after the power cycle. Data logging and position-based control shall be activated automatically or the operator shall be informed to acknowledge further operation of these functions if these were active before the power cycle occurred.

If the TC implementation cannot resume the task functions automatically, it shall prompt the operator that a task functionality was active before power down and that reactivation is requested.

Note that the clients are responsible for allowing or disallowing execution of control commands.

6.7 Task controller number

The Connection Management rules require that there shall always be a TC with function instance zero (0) on the network if a TC is present on that network. TCs shall be factory set to function instance zero (0), but shall retain the function instance as configured by the operator. A mechanism is required to conveniently resolve conflicts in the case where there are multiple TCs with the same function instance and in the case where there is no TC with function instance zero connected to the network. The TC shall be responsible for providing a proprietary means for setting the function instance from the TC. This means shall ensure that duplicate function instances between TCs will not be created. A newly set function instance shall not be used until a reconnect of the TC to the network is performed.

The TC with function instance zero (0) is defined as the “primary TC”. The proprietary means to set the function instance shall represent to the operator a TC number (See Clause 3). This ensures that TCs from all manufacturers will present a consistent numbering scheme for the operator to choose the primary (and secondary) TC(s). The TC number shall be in the range of 1 to 32, corresponding to function instances 0 to 31. TCs shall then be referenced as TC number 1, TC number 2, etc. A TC number equals TC function instance plus 1. The offset of 1 is for the support of operators, which may not be familiar with a zero-based numbering system.

The TC with function instance not zero (>0) shall follow the same connection procedure as specified in 6.6.

NOTE In case the TC is a component inside an ECU which has the ability to fulfil more than one control function, it is recommended that the TC control function can be configured by the operator to be active or inactive.

6.7.1 Client initialization on networks with multiple task controllers

Normally a client shall connect to a TC with function instance 0. If a client has a means to perform a “Move to another Task Controller” function on networks with multiple TCs, it can connect to a TC with a function instance >0. This function shall allow movement of the client to each of the available TCs in sequence. For example, this function could be accomplished with a “Next Task Controller” soft key or button in the user interface. The function behaves as follows:

a) “Move to another Task Controller” is enabled if the client detects more than one TC on the network.

b) When “Move to another Task Controller” is activated, the client shall perform the following:

   1) set to a safe state, or prevent activation of this feature unless it is in a safe state;

   2) send the Connection Deactivate message to the TC and wait for the response;

   3) stop sending the Client Task message to the TC;

   4) start the initialization process with another TC on the network.

The client shall save the new TC as the preferred TC for a next power cycle. If the preferred TC is not available within a certain time period after startup, the client can initialize connection to any other TC on the network. The client can provide a means for the operator to set the maximum wait time period or it may be obtained from the boot time specification in the "Version message" of the preferred TC.

NOTE A client shall connect only to a single TC and to a single DL at the same time. Connections from one client to multiple TCs or multiple DLs are not permitted in this edition of this part of ISO 11783.

6.8 Data exchange on the network

The TC converts data from the data transfer file set into process data messages to control the devices. These process data messages contain commands and values sent to the participating client control functions. The TC performs calculations to schedule the process data messages to be transmitted to the required addresses on the ISO 11783 network. An example of these calculations are site-specific applications such as when the TC looks up the position of an element of the client in the application grid, combines it with the operation delays of this client, and sends the relevant data to this client. In the opposite direction, the TC processes the process data messages sent by the participating clients and converts these process data variables to task data in order to return these to the data transfer file set.

All TC specific data exchange on the ISO 11783 network is based on process data messages.

It is recommended that control-related process data communication, e.g. section work status data, is triggered on change instead of using a fixed short time interval. The recommendation is to implement sending this type of data with an on change trigger as primary method in combination with a long time interval trigger as a fall-back mechanism. To avoid using an excessive amount of bandwidth on the network for process data communication and to ensure that process data requests and commands are acknowledged, the following process data exchange rules are defined.

a) A maximum of 10 process data messages per process data variable per second can be transmitted for each client to TC connection. This limitation addresses the distribution of process data message bandwidth over the different process data variables. This maximum is a generic requirement. Specific control scenarios that require a short burst such as controlling work states when entering or leaving a headland may exceed this maximum temporarily. These exceptions apply to specific DDIs such as the Setpoint Condensed Work State and the Actual Condensed Work State and are specified in the DDI definitions listed in the ISO 11783-11 data dictionary. Exceptions shall be explicitly listed in this data dictionary.

b) Requests that are transmitted by the TC shall be replied upon by the client before a next request is allowed to be transmitted by the TC to the same client. This synchronization of request-response process data message pairs has been introduced in ISO 11783-10 version 4. Prior to ISO 11783-10 version 4, a block of requests could be transmitted by the TC without waiting for individual responses throughout the block of requests.

c) All measurement commands (Command values 4 to 8) transmitted by the TC shall be acknowledged by the client with a PDACK message. The measurement command shall be replied to by the client before a next measurement command is allowed to be transmitted by the TC to the same client. The measurement command acknowledgement can be a positive acknowledgement when the measurement is supported and executed by the client or a negative acknowledgement when the measurement command is not accepted by the client. In case the measurement command is accepted by the client and the command starts a measurement, the client shall also transmit the value that the process data variable had at the time of starting the measurement for that process data variable to the TC. This ensures that the exact initial value of the process data variable measurement is communicated to the TC. The positive acknowledgement and the transmission of the initial process data value are requirements that have been introduced in ISO 11783-10 version 4. Prior to version 4, only a negative acknowledgement was required to be transmitted in case a measurement command could not be accepted by the client.

d) The PGN response time and time to wait before a request retry as specified in ISO 11783-3 also apply to process data communication. When a TC periodically sends a set of request process data value messages as a fallback method to acquire data when a client does not support a certain measurement, then the client shall respond per these requirements and a TC shall follow the specified retry procedure.

Clients shall initialize their internal process data variables to the correct operational values before activating a device descriptor object pool. For example, in case a client's internal data initialization process takes a certain amount of time to finalize before the correct device configuration values are available, then the device descriptor object pool shall not be activated prior to finalizing this data intialization. Thus, a TC requesting the device geometry upon activation of the device descriptor object pool will receive the correct operational values from the client.

The TC can generate process data messages containing process data variables that are not specified in a task data file. The process data messages control operation and data logging of the participating client control functions. The TC shall only send or request process data variables that are supported by the client control functions. Examples of this type of control are the use of sensor systems and the use of recorded spatial operation of client control functions to control client control functions.

Clients shall only transmit process data value commands to a TC that have been requested by

a) a request value command,

b) the default data logging method, and

c) individual measurement commands.

The interaction between the handling of measurement commands and the state changes of the "Task totals active" bit is as follows:

a) The Task Controller Status message "Task totals active" bit value does not put restrictions on the sending or responding of Process Data commands, except sending a default data logging command.

b) If the TC sends specific Process Data Measurement commands when the "Task totals active" bit value is set to 0, then the TC may also stop measurements by sending specific measurement commands to stop these measurements.

c) In case a TC has started measurements when the "Task totals active" bit is 0, then the TC shall not stop these measurements just before it changes the "Task totals active" bit from 0 to 1.

d) When the "Task totals active" bit goes from 0 to 1 or from 1 to 0, all measurements that were specified by the requesting CF that is transmitting this Task Controller Status message shall be stopped by the client. The TC may retrigger the measurements by sending specific Process Data Measurement commands or by sending the default data logging trigger.

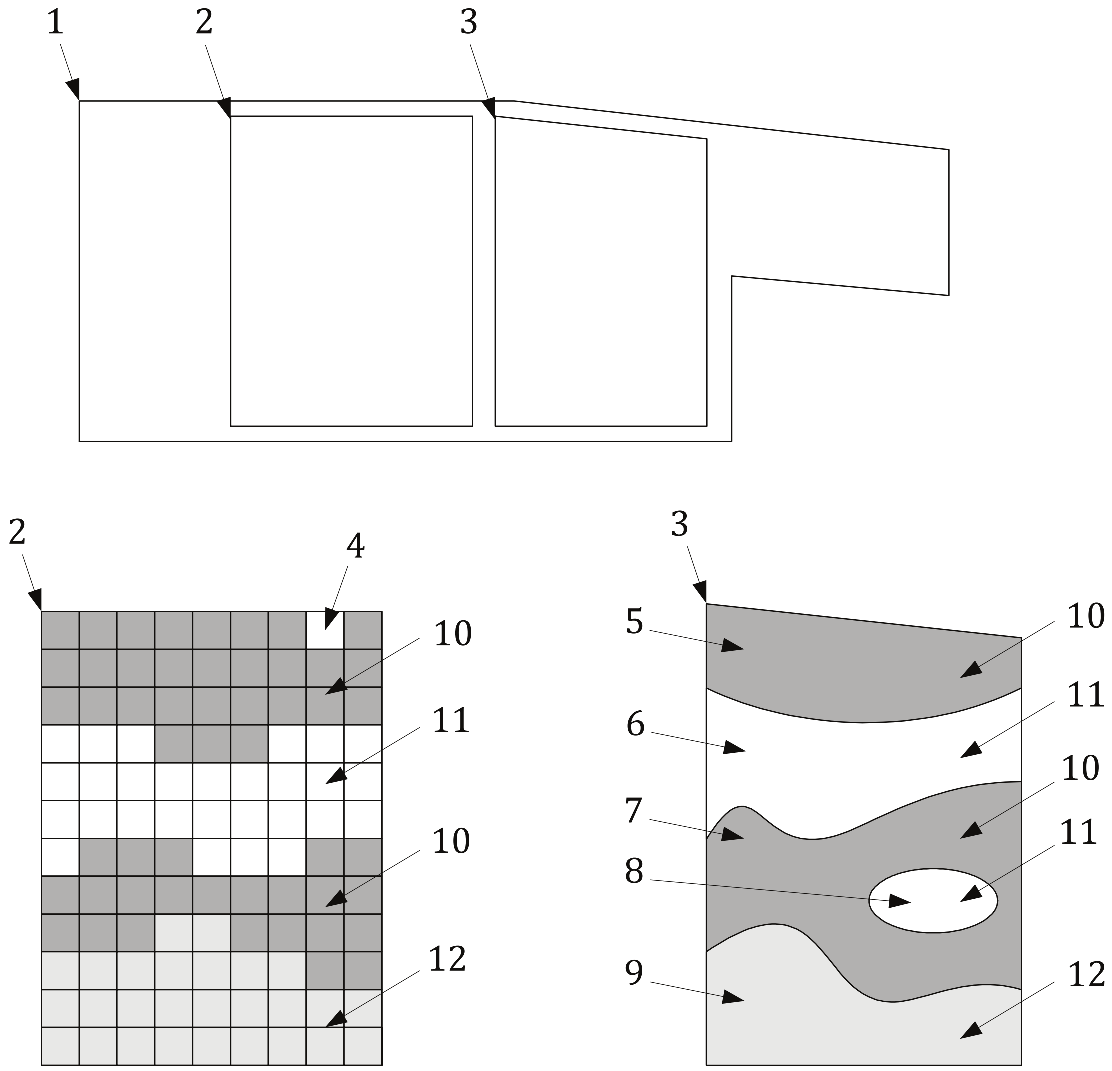
In ISO 11783-10 version 3 and later, it is specified that a TC can start a measurement while a task is not active. In ISO 11783-10 version 2 and earlier, TC clients may or may not support the start of a measurement while a task is not active.

6.8.1 Site-specific application

Site-specific applications require the TC to schedule sending of process data messages according to the actual location. For matching of the actual location with ProcessDataVariable XML element definitions, the geometry for site-specific process data has to be specified in the task data. The geometry definition is either a gridcell or a polygon and shall be labelled with a unique identification. Grid cells and polygons refer to a TreatmentZone to which the site-specific process data variable values are related. When the relevant DeviceElement enters a new TreatmentZone, the new set point values associated with that TreatmentZone are sent over the ISO 11783 network to the appropriate client.

Of the geometry definitions, gridcells have constant length and width dimensions. The gridcell location is relative to the origin of the grid to which a gridcell belongs. The structure and identification of gridcells are specified in the XML element Grid. Polygons can be used to define irregularly shaped TreatmentZones. The XML elements Polygon, LineString, and Point are used to define this type of TreatmentZone. Figure 3 presents a comparison of both types of TreatmentZone definition and Figure 4 the grid definition in more detail. On the right hand side in Figure 3, TreatmentZone 1 is represented by two Polygons that describe two single-bounded areas. The first of these two Polygons (Polygon 1) has a single Linestring bounding its area whereas the second of these two Polygons (Polygon 3) is defined by two Linestrings, an outer boundary and an inner boundary. TreatmentZone 2 is also described by 2 Polygons, Polygon 2 and Polygon 4, both Polygons only have outer boundaries.

When using overlaying polygons, the TC shall always use the exterior polygon definition for a given location. Each interior boundary describes a hole in the exterior polygon.



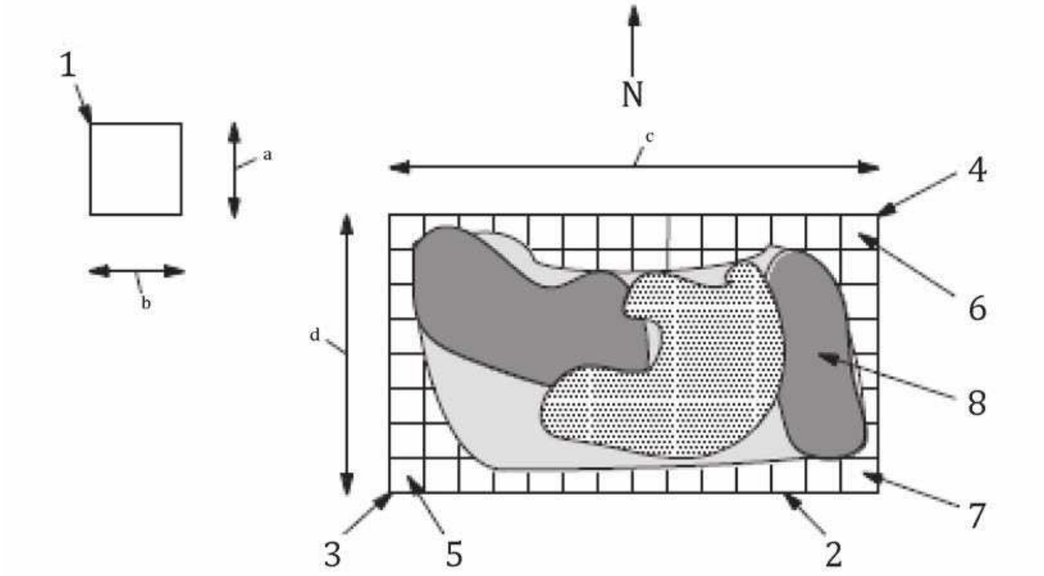
Key

1	frame partfield
2	partfield A
3	partfield B
4	gridcell
5	polygon 1
6	polygon 2
7	polygon 3
8	polygon 4
9	polygon 5
10	treatment zone 1
11	treatment zone 2
12	treatment zone 3

Figure 3 — Frame partfield, partfield, gridcell, polygon, treatment zone relationship

6.8.1.1 Grid based site-specific applications

Inside the XML data definitions, the Grid is defined by the GridMaximumColumns and the GridMaximumRows attributes, the GridMinimumNorthPosition and GridMinimumEastPosition, and the size of each grid cell. The grid cell ordering inside the binary file shall start with the GridMinimumNorthPosition/GridMinimumEastPosition, in ascending order of the columns (moving east), then proceeding in ascending order (moving north) on each row, beginning each time at the GridMinimumEastPosition column for each row. Figure 4 shows an example of a Grid and its grid cells.



Key

1 grid cell
2 frame partfield
3 origin (GridMinimumNorthPosition, GridMinimumEastPosition)
4 FrameNorthMax, FrameEastMax
5 first grid cell (row 0 and column 0)
6 last grid cell (row 7 and column 13)
7 fourteenth grid cell (row 0 and column 13)
8 treatment zone
a GridCellNorthSize.
b GridCellEastSize.
c GridColumns.
d GridRows.

Figure 4 — Definition of frame of grid for partfield

Grid cell definitions data are stored in binary format as separate files. The reference to the separate file is specified in the XML element Grid. There can only be a single Grid XML element per task, referring to both a binary and an XML file of a unique name prefix for all tasks of the data transfer file set. The uniqueness of grid file names shall be guaranteed by the FMIS. Each grid file definition results in two separate files — one to contain the binary data and the other the XML-coded header structure of the binary data set. The header structure defines the maximum data per binary record and enables the correct interpretation of the binary data. The binary file suffix is to be set to ".bin"; the suffix of the XML header structure file is to be set to ".xml".

See 8.6.2 for the encoding details of the XML header and binary data files.

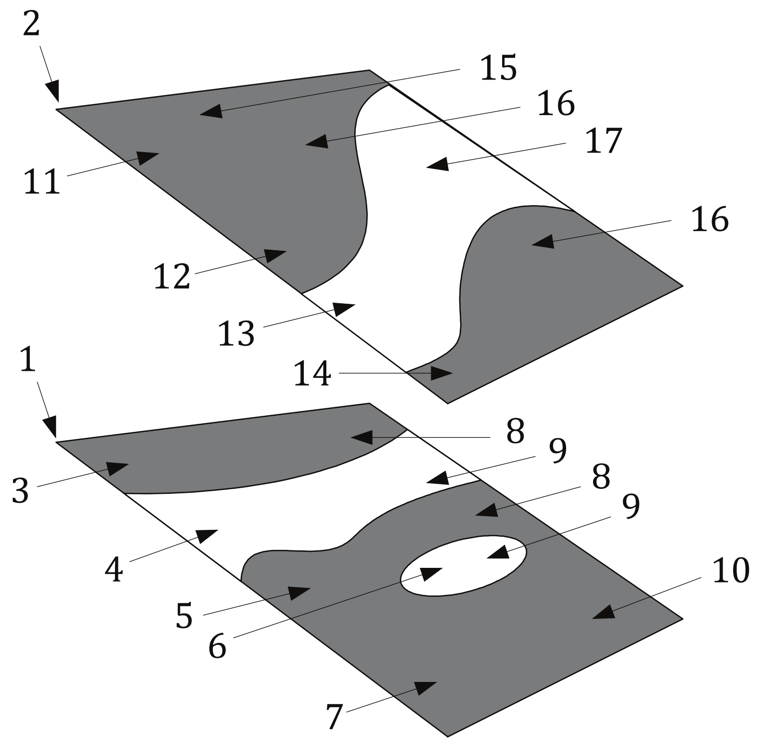
6.8.1.2 Polygon based site-specific applications

The geometry of the different TreatmentZone XML elements can also be specified by polygons. A separate set of polygons shall be defined for each site-specific operation of the task, i.e. each individual product or each individual spatially variable setpoint type has its own set of polygons that define the boundaries between the setpoint values for different treatment zones. For examples of single and multi-layer site-specific tasks with the geometry specified as polygons see D.41. An example of a dual-layer site-specific task is depicted in Figure 5 and the XML encoding for this example is provided in this section.

To transmit a setpoint value to a client, the TC shall determine for each device element that has a setpoint device process data assigned to the ProcessDataVariable XML elements of a variable rate layer whether it is located in one of the TreatmentZone XML elements of that layer.

TreatmentZone XML elements may contain multiple polygons. Each polygon specifies an area bounded by one exterior boundary for which the ProcessDataVariable of that TreatmentZone is valid.

When polygons are used to define the geometry of the TreatmentZone XML element then only a single ProcessDataVariable XML element shall be included in that TreatmentZone. With this constraint, at maximum one variable rate is specified for the group of TreatmentZones and polygons that define one variable rate layer.



Key

1 layer 1
2 layer 2
3 polygon 1
4 polygon 2
5 polygon 3
6 polygon 4
7 polygon 5
8 treatment zone 1
9 treatment zone 2
10 treatment zone 3
11 polygon 6
12 polygon 7
13 polygon 8
14 polygon 9
15 treatment zone 4
16 treatment zone 5
17 treatment zone 6

Figure 5 — Dual layer polygon based site-specific application

See D.50 example 2 for the XML encoding of this set of polygon-based treatment zones.

6.8.2 Data logging

The TC can also be used to log data for transfer back to the FMIS. For this purpose, process data variable values can be requested on the ISO 11783 network. Data log triggers can be specified by the FMIS in the data transfer file set to the TC. The TC is responsible for converting these data log triggers into process data measurement commands and for logging the replied process data variable values. The process data measurement commands for data logging are sent by the TC after starting and resuming a task. The values for the requested process data variables are provided by the connected clients, as long as the task is active. The sending of the requested values is stopped and measurements from the

TC are cancelled when the task is paused. When a client sends more data than requested by the TC for data logging, these additional data shall be ignored by the TC. The description of the XML element DataLogTrigger in Annex D provides more detail on the use of data log triggers.

Note that process data measurement commands can also be used by the TC to request process data when the "Task totals active" bit value is set to 0. This case is outlined in the introduction in 6.8.

6.8.3 Totals

Two different types of totals are distinguished: task totals and lifetime totals. A task total is controllable by the TC, i.e. when a task is started, the TC can set the value of this total to continue counting from. A lifetime total is not controllable by the TC and can only be requested and stored in the data transfer file set. Device process data objects that represent task totals shall have a data log trigger method of type "total" and a process data property bit that specifies that this process data object is settable. Device process data objects that represent lifetime totals shall have a data log trigger method of type "total" and the process data property bit "settable" shall be set to value 0.

A data log trigger with a data log method of type total shall be used to request the TC to store totals. Each total shall be stored once per Time XML element in a task. A task can contain multiple Time XML elements, e.g. in the case when a task is resumed, and each Time XML element can contain a set of totals. In addition to this, total values can also be stored more frequently in the TimeLog related data log files. When a task is resumed, the TC shall send the task totals stored in the most recent Time XML element to the client to continue counting from these task total values. When tasks are imported into the FMIS system, the most recent Time XML element shall contain all totals of the task. The total values may be requested by the TC from the client to obtain the latest total values when the task is stopped. Alternatively, when process data variables that represent total values are transferred as process data messages to the TC cyclically, such a query might not be necessary.

In the time period between the power up of a client and the first transmission of the "Task totals active" bit with value 1 the client shall keep its task total values initialized to a value of 0.

6.8.3.1 Handling of Task Totals

Task totals are required to be handled by TCs and may optionally be used by DLs. If a DL handles task totals, the requirements of handling task totals by a TC apply to that DL. Therefore, in this paragraph, the reference to a TC may be substituted by a DL if that DL supports task totals.

For task totals, it is the TCs responsibility to perform the following:

a) Set up an internal list of all task totals from all connected clients. To do so, the TC must parse the device descriptor object pools of all connected clients.

b) Request all those task totals from all connected clients if the task status changes from running to paused or completed and store them within a Time XML element.

c) Restore all those task totals to all connected clients if a task is resumed.

d) In order to avoid data loss in case of an unexpected shutdown, the TC shall request all task totals on a regular basis, e.g. at a 1 min interval. The TC may use a measurement to request these task totals on a regular basis. In order to support measurements for task totals, task totals are recommended to support the time and distance interval measurement methods.

e) The most recent Time XML element in a Task shall contain all task totals collected for that Task, even if some of the clients were not present during the most recent running period of a Task.

Task totals do not need to be in the default data set of a client because all task totals shall be handled by the TC. For this reason, task totals are recommended not to be added to the default data set of a client.

Definition of task totals behaviour regarding Start/Resume/Pause task events:

a) Task started: the client resets its task totals to zero.

b) Task paused: the TC collects task totals from all clients, and all task totals retain their values.

c) Task resumed: this is to be handled similarly to the Start event for clients. After the task has been restarted, the TC sets all task totals to the previously stored task total values, sends out these values as process data variable values to all appropriate clients, and the clients count from these values onwards. (The TC is responsible for keeping track of previously collected task totals.)

The Task Controller Status message indicating the change of status of the task totals' active bit from inactive to active shall be sent before the task total values are set. The Task Controller Status message indicating the change of status of the task totals' active bit from active to inactive shall be sent before final task total values are requested.

It is the client's responsibility to default its task totals to a value of zero at client startup. The client logic will keep the task totals at zero as long as the task status is set to the counters are not active value (bit value = 0). Totals can be requested before a task is started; the value reported by the client will, in that case, be zero.

6.8.3.2 Handling of Lifetime Totals

Lifetime totals can both be requested and logged by TCs and DLs. When lifetime totals are handled by the TC or DL, it is the TC's or DL's responsibility to perform the following:

a) Set up an internal list of all lifetime totals from all connected clients. To do so, the TC or DL must parse the device descriptor object pools of all connected clients.

b) Request all those lifetime totals from all connected clients if the task status changes from running to paused or completed and store them within a Time XML element.

c) In order to avoid data loss in case of an unexpected shutdown, the TC or DL shall request all lifetime totals on a regular basis. The TC or DL may use a measurement to request these lifetime totals on a regular basis. In order to support measurements for lifetime totals, lifetime totals are recommended to support the time and distance interval measurement methods.

d) The most recent Time XML element in a Task shall contain all lifetime totals collected during that Task, even if some of the clients were not present during the most recent running period of a Task.

Lifetime totals are not recommended to be added to the default data set of a client. This enables TCs or DLs to handle the lifetime totals based upon the data collection requirements of the TC or DL.

Lifetime totals are not affected by the Start/Resume/Pause task events. The updating of lifetime totals is entirely controlled by the clients.

6.8.4 Data log triggers

Each process data variable of a certain device might be asked for, and can be logged, by the TC. The number and types of process data variables to be logged are specified by the XML element DataLogTrigger. The DataLogTrigger XML element can specify precisely which process data variables are required from certain device elements.

Alternatively, for a default data logging mechanism, a process data variable called RequestDefaultProcessData (DDI = $DFFF_{16}$) is specified. Within the default data logging mechanism, the clients are responsible to send the data that the clients want to be logged, at the interval and with the triggers that are specified by the client. Examples of cases for which this data logging mechanism is defined are the following:

a) A client has a large number of DeviceProcessData elements. The TC neither knows an appropriate selection nor a good interval for logging. By using the default data logging mechanism, the client itself decides which data are logged.

b) The FMIS can specify that it requires data logging to be activated by adding the RequestDefaultProcessData (DDI = $DFFF_{16}$) to each task.

In case no datalog triggers for specific DeviceProcessData are specified in a Task, the recommended practice is to include a DataLogTrigger XML element with the RequestDefaultProcessData DDI specified. This will ensure that at least the default data set from each client is logged. Due to that the default data logging can only be stopped by a transition of the "Task totals active" bit value from 1 to 0, the request default data command shall not be used by a TC when the "Task totals active" bit value is 0.

By using the identifier (DDI) of this process data variable in the DataLogTrigger XML element, the TC is commanded to request the specified devices to send all their default process data variable values from their default measurement method. The RequestDefaultProcessData DDI may only be requested from device element 0 of a device. The TC uses a request value command with DDI = $DFFF_{16}$ to request the sending of the default process data variables values. The set of process data variable values that belong to the default set of a device are specified in the device descriptor object pool. When the device descriptor object pool specifies certain DeviceProcessData objects to be part of the default set, a DeviceProcessData object with the RequestDefaultProcessData DDI shall be included in the device descriptor object pool of that device. The ProcessDataTriggerMethod attribute of this DeviceProcessData object shall be set to $1F_{16}$.

EXAMPLE 1 Device descriptor object pool containing the RequestDefaultProcessData DDI and a set of default data:

```
<DVC A="DVC1" B="Tiller" C="1.02*" D="A00484000B2CAF13" F="32A0FE34A56F00"
G="FF000000006E65">
    <DET A="DET1" B="1" C="1" E="0" F="0">
       <DOR A="2"/>
       <DOR A="3"/>
       <DOR A="4"/>
       <DOR A="5"/>
       <DOR A="6"/>
    </DET>
    <DPD A="2" B="0074" C="2" D="16" E="Area" F="9"/>
    <DPD A="3" B="0077" C="2" D="16" E="Time ON" F="7"/>
    <DPD A="4" B="008D" C="1" D="3" E="Work ON/OFF"/>
    <DPD A="5" B="0043" C="1" D="3" E="Width" F="8"/>
    <DPD A="6" B="DFFF" C="0" D="31"/>
    <DVP A="7" B="0" C="2.777778E-04" D="2" E="hr"/>
    <DVP A="8" B="0" C="1.000000E-03" D="1" E="m"/>
    <DVP A="9" B="0" C="1.000000E-04" D="2" E="ha"/>
</DVC>
```

In this example, the process data DDIs $8D_{16}$ and 43_{16} are part of the default set of data. This device supports the request default process data mechanism which is indicated by the DPD with object ID 6.

EXAMPLE 2 RequestDefaultProcessData DataLogTrigger in a task:

```
<TSK A="TSK1" E="PFD1" G="3">
    <DLT A="DFFF" B="31"/>
</TSK>
```

In this example, the task specifies the requesting of default data from connected clients that support the request default process data mechanism.

During data logging, there can be different frequencies of transmission of DDIs and also different time delays of replies to value request commands. The TC logs process data variable values at certain time intervals, and therefore the following rules apply.

a) Time and position data shall be logged at a maximum once per TimeLog instance.

b) Each process data variable value shall be logged at a maximum once per TimeLog instance. If data logging can only be performed at a low frequency because of TC capabilities, the value logged depends on the TC design.

c) If the TC is capable of data logging at the rate given by DDI with the highest update frequency, then a new data log record shall be started each time the next DDI with highest update frequency is received. All other process data values that are received between two values of the highest update frequency DDI are logged in the current record.

d) A received process data value shall be logged a maximum of once. If no new value is available for the next record, then no value shall be logged for that DDI.

Following these rules, logged data values in the same record can be separated by, at a maximum, the time difference that exists between two records. If several DDIs need to be grouped together, the process data variable “Log Count” (DDI = 0093_{16}) shall be used with the highest repetition rate to time tag the data logging of associated process data variable values.

EXAMPLE 3 Process data variable DDI “A” has the highest update frequency; process data variable DDIs “B” and “C” are added to current DataLog:

Client sends value of DDI A_1: TC stores value of A_1 in DataLog 1

Client sends value of DDI B_1: TC adds value of B_1 to DataLog 1

Client sends value of DDI A_2: TC closes DataLog 1; stores value of A_2 in DataLog 2

Client sends value of DDI C_1: TC adds value of C_1 to DataLog 2

Client sends value of DDI A_3: TC closes DataLog 2; stores value of A_3 in DataLog 3

Client sends value of DDI A_4: TC closes DataLog 3; stores value of A_4 in DataLog 4

EXAMPLE 4 Client desires to group process data variable DDIs: “A”, “B”, “C”, and “D”.

Client sends value of DDI (LogCount) LC_1: TC stores value of DDI LC_1 in DataLog 1

Client sends value of DDI A_1: TC adds value of A_1 to DataLog 1

Client sends value of DDI B_1: TC adds value of B_1 to DataLog 1

Client sends value of DDI C_1: TC adds value of C_1 to DataLog 1

Client sends value of DDI D_1: TC adds value of D_1 to DataLog 1

Client sends value of DDI LC_2: TC stores value of DDI LC_2 in DataLog 2

Client 1 sends value of DDI A_2: TC adds value of A_2 to DataLog 2

Client 1 sends value of DDI D_2: TC adds value of D_2 to DataLog 2

Client 1 sends value of DDI B_2: TC adds value of B_2 to DataLog 2

Client 1 sends value of DDI C_2: TC adds value of C_2 to DataLog 2

Client sends value of DDI LC_3: TC stores value of DDI LC_3 in DataLog 3

Client 1 sends value of DDI D_3: TC adds value of D_3 to DataLog 3

Client 1 sends value of DDI B_3: TC adds value of B_3 to DataLog 3

Client 1 sends value of DDI C_3: TC adds value of C_3 to DataLog 3

Client 1 sends value of DDI A_3: TC adds value of A_3 to DataLog 3

The FMIS can determine, based on the LogCounts, that DDIs A, B, C, and D all belong to the same group. There will always be a new DataLog started before the group is transmitted.

6.8.5 Peer Control

Peer Control is the mechanism by which any CF may be a setpoint value source to another CF as a setpoint value user such that the TC controls and records the assignment of sources to users.

Peer Control addresses the requirement for CFs in the ISO 11783 systems to accept setpoint values from sources that cannot be determined at the time that field operations are planned. These setpoint values need to be determined at run time and directed to a rate controller in a standardized manner without the specific knowledge of the setpoint value source or of the controller that is required to process these setpoint values. These sources of control information may include items such as on-the-go sensor systems. These systems may include determination of a rate directly by the setpoint value source or a modification of another rate such as that from a planned application map.

For this mechanism, both the setpoint value source and the setpoint value user shall be CFs that are TC “clients” as defined in this part of ISO 11783 and that upload and activate a device descriptor object pool. Similar to planned site specific applications, the setpoint value user device descriptor object pool shall include Device Process Data (DPD) object(s) that have their property defined as “Settable”. The setpoint value source device descriptor object pool shall include a similar form of Device Process Data object with the property defined by a bit flag set as “Setpoint Source”. The bits for Settable and for Setpoint Source are mutually exclusive. The setpoint value source may also include DPD objects defined as settable. The settable DPD in the setpoint value source can then be used in a “Map Overlay” function where a position dependent received setpoint value from the TC is modified by the setpoint value sources’ specific algorithm and used as the setpoint value to be transferred to the final setpoint value user. The relations between the map source and peer source are depicted in Figure 6.

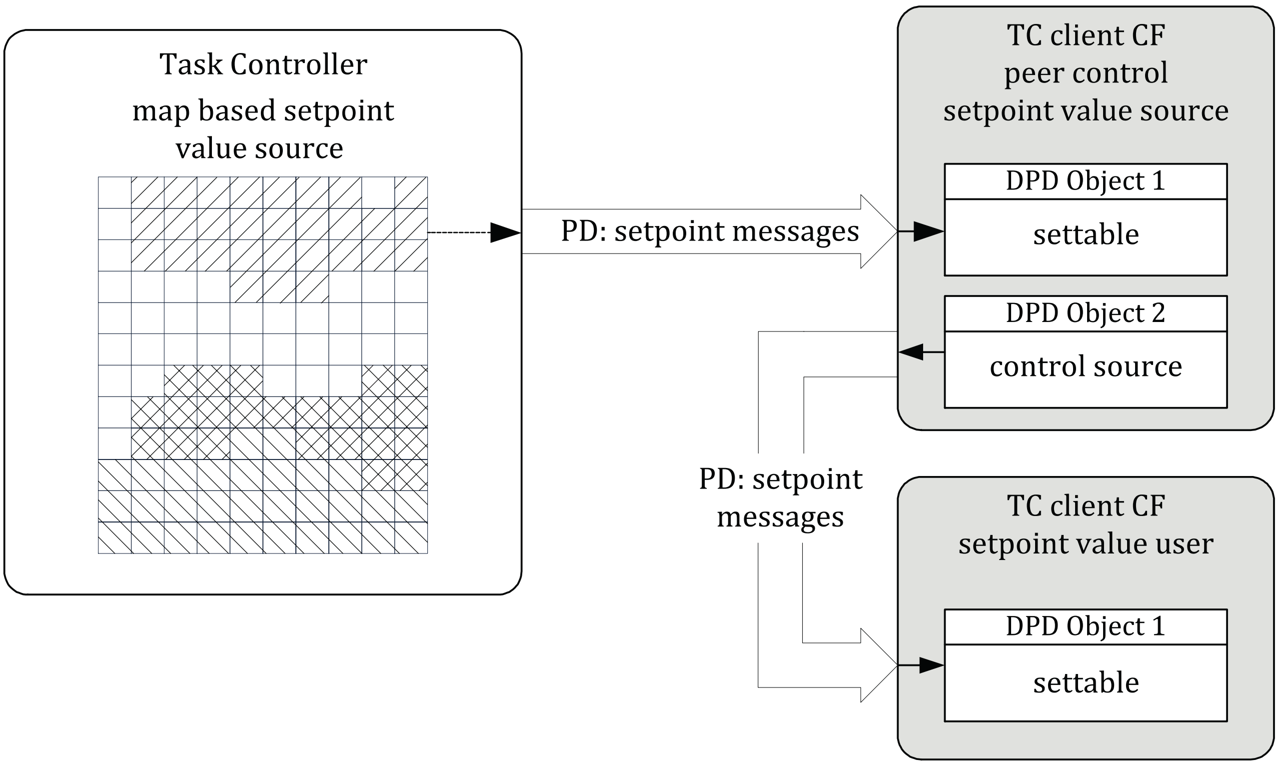


Figure 6 — Relations between a map-based setpoint value source, a peer control setpoint value source, and a setpoint value user

When a task is initiated in the TC, similar mechanisms are used as are available in ISO 11783-10 version 3 and prior TCs to automatically or manually assign controlling settable process data objects to setpoint data sources. In ISO 11783-10 version 3 and prior TCs, setpoint data sources include map layers from variable rate maps defined for a task or (optional) manually programmed control setpoints defined by the operator in the TC user interface. In ISO 11783-10 version 4 and later TCs, the setpoint data sources also include items from any CF that have a device process data element with the control source property set in their device descriptor object pool. All of the same assignment restrictions of DDIs and object properties apply to all setpoint data sources whether originating from an FMIS-planned source (TreatmentZone) or from another external source (CF with a device process data element with the control source property set). When the sources are assigned, the TC shall initiate assignment messages to both the setpoint value source and the setpoint value user. Each client shall respond to acknowledge or reject the assignment within 250 ms. For backwards compatibility, if the target controller does not send a response, the connection is treated as rejected. The assignments enable the setpoint value source to send the setpoint information directly to the setpoint value user without extra message latency and processing overhead of communicating through the TC. The assignment also enables the setpoint value user to qualify received control setpoint data as originating from its assigned setpoint value source CF. This assignment uses a command message from Process Data (PD) messaging. The message specifies the type of the connection (settable or control source property set), the Device Element, the DDI, and the NAME of the other CF. It is the responsibility of the assigned CF to track changes to the CF's SA's when new address claims occur. The assignment procedure is visually represented in Figure 7.

The assignment messages shall be sent following each change of task status from inactive to active. While the task is active, the control setpoint information from the setpoint value source CF shall be sent at a periodic rate of once per second and on-change with a maximum rate of 5Hz. If the control setpoint messages are not received for 3 s, the target controlling CF shall take action in the same manner as if the TC communications were lost. If the communications with the TC is lost (as determined by the periodic Task Controller Status message), both the setpoint value user and the setpoint value source take action based on lost TC communications and act as the task becomes inactive. When a task becomes inactive, the setpoint value user shall not accept setpoint messages, the setpoint value source shall stop sending messages and both CF's shall cancel the current assignments. A new set of assignment messages shall be sent at the event of a task becoming active for each device element assigned to an external source. If no assignment message is received during each active task session or the assignment is invalid, the setpoint value user shall default to only accepting setpoint messages from the TC while the task status is active.

An assignment may be made by the FMIS system and stored as part of the Task (TSK) element of the data transfer file set. If the assignment made by the FMIS is not available at the start of the task, the TC may make the assignment according to the same rules as defined for direct TC to controller usage. If the assignment is made by the TC the assignment shall be stored in the same manner such that the assignment is remembered for future start events of the same task and is available to the FMIS for reference. This assignment is stored by the ControlAssignment (CAT) XML element.

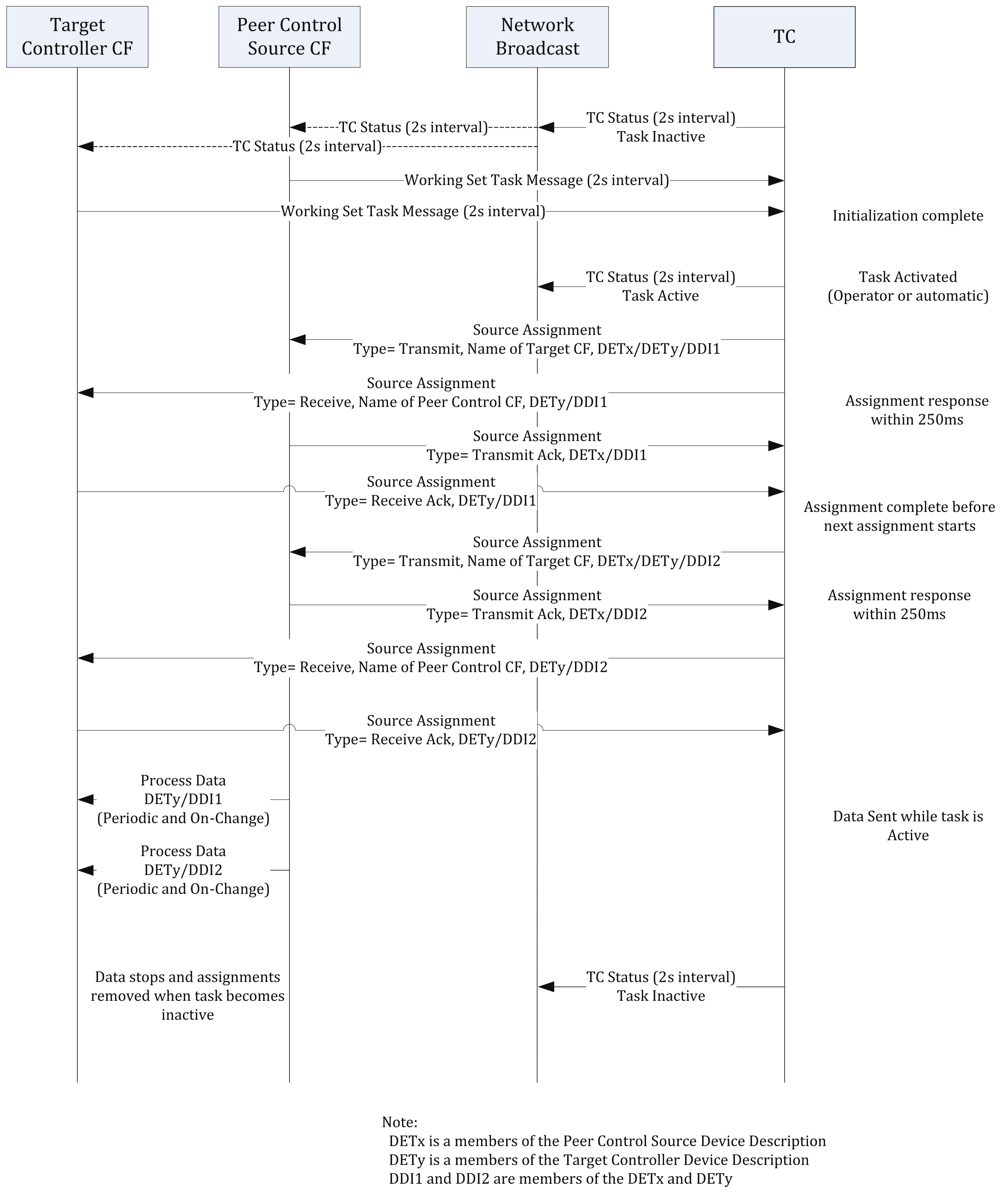


Figure 7 — Peer Control assignment procedure

7 Data logger requirements

7.1 General

A data logger (DL) can be installed on the network to perform non-task related data logging without the need for controlling a TC client. The DL is distinguished from a TC by a separate function definition (see ISO 11783-1).

The DL uses a subset of the TC functionality. The DL has to fulfil the client connection mechanism requirements, the processing of device descriptor object pools, and the logging of data from process data as specified for a TC. In order to keep the CF responsibilities on the network clear, a DL shall not use all the TC functionalities that are described in Clause 6. The network communication rules specified in 6.8 do apply to DL-to-DL client communication.

The DL shall not

— use section control functionality,

— use position based control functionality,

— set up peer control assignments (note that a peer control CF can connect to a DL to supply values for data logging purposes), and

— send process data set command values for DDI's other than task totals to a DL client.

The DL may

— request process data from connected DL clients,

— issue measurement commands to each connected DL client,

— request default data logging from connected DL clients,

— request parameter groups to log parameters, and

— set and Request task totals and control accumulation of task totals via the "Task totals active" bit in its Task Controller Status message (the DL client shall maintain an individual set of task totals for the DL).

7.2 Connection management

The DL client supporting a connection to a DL shall initialize to a DL in the same way as described in 6.6.2. The DL client shall provide a device descriptor object pool to the DL. The device descriptor object pool of a client provided to a TC and the device descriptor object pool of the same client provided to a DL can differ. It is the client designer's responsibility to specify both device descriptor object pools and optimize the content for each purpose.

Multiple DLs can be present on a network. If multiple DLs are present on a network, each DL shall be able to be identified by a unique DL function instance. Configuring the function instance of a DL is, similar to configuring the TC function instance in case multiple TCs are present in a network, done via a proprietary means. The DL with function instance 0 does not have a role that is different from any other DL function instance on the network. In ISO 11783-10 version 4 and prior, a connection from a DL client to, at maximum, one DL is permitted.

The following network configurations are possible:

a) TC and TC client(s)

Every TC client supporting task controller functionality sets up a connection to the TC. This scenario has been available from ISO 11783-10 version 1 and onwards.

b) DL and DL client(s)

Every DL client supporting data logger functionality sets up a connection to the DL. This scenario is, connection wise, similar to scenario 1 and is introduced in ISO 11783-10 version 4.

c) TC and DL and TC/DL client(s)

Every CF supporting both task controller and data logger client functionalities connects to the TC and/or DL. The CF may support connecting to both the TC and DL servers simultaneously. This scenario is introduced in ISO 11783-10 version 4.

7.3 Measurements and totals

If a DL client transmits a device descriptor object pool containing process data values to a DL, it shall maintain a unique set of active measurement commands for the connected DL. In case a client connects to both a TC and a DL simulatenously, there shall be different sets of measurements commands active in parallel in one client.

If a DL client transmits a device descriptor object pool containing task totals to a DL it shall maintain a unique set of task totals for the connected DL. Thus, a client can have a set of task totals for a TC and a set of task totals for a DL, each accumulating in parallel. Each set of task totals may be set, reset, or triggered individually by the corresponding CF. There shall be no interference between these different sets of task totals. Note that even though a DL may not set or reset task totals often, this mechanism is useful for the DL to specify a start point for the accumulation of task total values and to avoid roll-over of task totals when a maximum value is reached.

8 Data transfer

8.1 General

The communication FMIS to MICS to FMIS is based on data transfer files. Each XML data transfer file is formatted according to XML definitions version 1.0. The XML files contain text only and are coded in UTF-8 (see ISO/IEC 10646). Optionally, binary-coded data files for gridcell definitions or logged process data can be part of the data transfer file set. All files shall be contained in the same directory.

Both coding data and task data are stored in the same set of XML files upon transmission from FMIS to the MICS. During processing of tasks by the TC, these files are likely to be modified and, when tasks are finished, can be transferred back to the FMIS.

8.2 Extensible Markup Language

XML is a language for the description of structured documents and data and represents a technological basis for data exchange. XML is a subset of SGML (standardized general markup language).

XML is a hierarchical set of XML elements. XML elements can contain one or more XML attributes. Inside the data transfer XML files, no text is allowed inside XML elements.

An XML element consists at least of an opening label, a number of attributes, and a closing label. For low memory and data transfer bandwidth requirements, abbreviations as specified in the description of the XML elements are used in the xml schema and in the xml files. The following line is an example of an XML element definition called *Worker*:

< WKR A = “WKR2” B = “Miller” >

< /WKR >

Opening and closing labels shall be of the same string, except that the closing label shall be preceded by a “/”. The case is significant in all uses of XML element labels, i.e. < worker > is not the same as < Worker >.

XML attributes carry information contained in XML elements. The syntax of an XML attribute is *attribute-name* = “*description*”.

The XML attribute label follows the same rules as an XML element label. The case is significant. The use of the equals sign is always required. The text string description shall be enclosed in quotes.

XML elements that do not contain child elements, but contain attributes, may be represented in a shortened form:

< WKR A = “WKR1” B = “Smith” / >

If an XML file follows the rules of the XML syntax itself, it is said to be *well formed*. The elements of an XML document can be formally specified using a DTD. If a DTD exists for an XML document, XML parsers with the appropriate extensions can test the XML document against the DTD. A document that passes such a consistency and validation test is said to be *valid*.

XML data transfer files are always text files. Due to the hierarchical structure, XML files can only contain a single XML root element. The elements list of the root element specifies which XML elements might be used as primary elements. Primary elements consist of coding data elements and entities which are not related to a single task. The root element of a data transfer file set is named as ISO11783_TaskData. XML data transfer files shall always be well formed and valid; otherwise, the contents of the file cannot be processed.

8.3 Extensible schema definition

XML provides an application-independent way of sharing and transferring data. With a DTD, it is possible to verify that received data from the outside world is valid. Also, a DTD might be used to verify its own created data.

The purpose of a DTD is to define the legal building blocks of an XML document. It defines the document structure with a list of legal XML elements. A DTD can be declared inline in an XML document or as an external reference. An object-oriented language modelling approach is to use an XSD as DTD.

An XML schema

— defines elements that can appear in a document,

— defines attributes that can appear in a document,

— defines which elements are child elements,

— defines the order of child elements,

— defines the number of child elements,

— defines whether an element is empty or can include text,

— defines data types for elements and attributes, and

— defines default and fixed values for elements and attributes.

XML schemas are formatted in XML and well formed. A well-formed XML document is a document that conforms to the following XML syntax rules:

a) it shall begin with the XML declaration;

b) it shall have one unique root element;

c) all start tags shall match end tags;

d) XML tags are case-sensitive;

e) all elements shall be closed;

f) all elements shall be properly nested;

g) all attribute values shall be quoted.

Schemas support data types and namespaces. With the support for data types, it is possible to describe permissible document content, validate the correctness of data, work with data from a database, define data facets (restrictions on data) and data patterns (data formats), and convert data between different data types.

8.4 XML schema definition

The validity of XML data transfer files is defined by the schema ISO11783_TaskFile. The schema is based on http://www.w3.org/2001/XMLSchema definitions and data types. The ISO11783_TaskFile schema does not have an ISO 11783 specific namespace. The XML schemas are published at the following location: http://dictionary.isobus.net/isobus/file/supportingDocuments. Unique identifiers of XML elements and references to unique identifiers across XML elements are defined as data types ID and IDREF respectively. To create a unique namespace all over the data transfer file set, each XML element identifier shall be defined in its specific XML element namespace. Table 2 defines the valid namespaces of XML element identifiers.

The XML Schema definition has in its filename the version information of the ISO 11783-10 document. The naming convention for the XML Schema definition is: ISO11783_TaskFile_V[VersionMajor]-[VersionMinor].xml. For example, the XML Schema definition filename of the first published international standard is ISO11783_TaskFile_V2-0.xml; the XML Schema definition filename of the first revision to the first published international standard is ISO11783_TaskFile_V3-0.xml. The VersionMinor field in the ISO11783_TaskData XML element is used to enumerate revisions to the XML Schema definition within the same VersionMajor value. The XML Schema definition files are published on the Internet at URL http://dictionary.isobus.net/isobus/file/supportingDocuments. The XML element ISO11783_TaskData VersionMajor and VersionMinor attribute values are set to a fixed value in the corresponding XML Schema definition files.

Implementers of this standard are responsible for keeping their applications backwards compatible with older versions of the standard and their corresponding schemas.

The XML schema definition is to be applied to all XML data transfer files in both data transfer directions. All XML elements are considered as entities. Each entity shall have a unique identifier (if there is an identifier defined as the XML attribute for this element) over the whole data transfer file set. Each identifier starts with the appropriate namespace letters followed by, at a maximum, an 11-digit decimal number without leading zeroes. Unique identifiers are at least 4 bytes long and shall be provided by both the FMIS and the MICS. The values of 0 and −2147483648 for identifiers are reserved and shall not be used inside XML elements. Prior to ISO 11783-10 version 3, the value of 0 was not reserved and was assumed to be interpreted as a positive value identifying FMIS generated XML elements. In ISO 11783-10 version 3, the value of 0 has been introduced as reserved and shall not be used as XML element identifier.

On the MICS, no coding data XML element shall be changed or deleted. Non-coding data XML elements may be edited and changed. If the operator of the TC creates new entities of XML elements, then the TC is responsible for creating unique identifiers for all newly created entities of XML elements. To differentiate between those entities provided by the FMIS and those created on the mobile system, each newly created identifier shall have the appropriate namespace letters followed by a negative decimal number, e.g. “WKR–1”.

Table 2 — XML element types and acronyms

XML element designator	Coding data	XML element name
AllocationStamp		ASP
AttachedFile[a]	X	AFE
BaseStation[a]	X	BSN
CodedComment	X	CCT
CodedCommentGroup	X	CCG
CodedCommentListValue	X	CCL
ColourLegend	X	CLD
ColourRange	X	CRG

[a] This element is introduced in ISO 11783-10 version 4.

Table 2 *(continued)*

XML element designator	Coding data	XML element name
CommentAllocation		CAN
Connection		CNN
ControlAssignment[a]		CAT
CropType	X	CTP
CropVariety	X	CVT
CulturalPractice	X	CPC
Customer	X	CTR
DataLogTrigger		DLT
DataLogValue		DLV
Device	X	DVC
DeviceAllocation		DAN
DeviceElement	X	DET
DeviceObjectReference	X	DOR
DeviceProcessData	X	DPD
DeviceProperty	X	DPT
DeviceValuePresentation	X	DVP
Farm	X	FRM
Grid		GRD
GuidanceAllocation[a]		GAN
GuidanceGroup[a]	X	GGP
GuidancePattern[a]	X	GPN
GuidanceShift[a]		GST
LineString		LSG
OperationTechnique	X	OTQ
OperationTechniqueReference	X	OTR
OperTechPractice		OTP
Partfield	X	PFD
Point		PNT
Polygon		PLN
Position		PTN
ProcessDataVariable		PDV
Product	X	PDT
ProductAllocation		PAN
ProductGroup	X	PGP
ProductRelation[a]	X	PRN
Task		TSK
TaskControllerCapabilities[a]		TCC
Time		TIM
TimeLog		TLG
TreatmentZone		TZN

[a] This element is introduced in ISO 11783-10 version 4.

Table 2 *(continued)*

XML element designator	Coding data	XML element name
ValuePresentation	X	VPN
Worker	X	WKR
WorkerAllocation		WAN
ExternalFileContents		XFC
ExternalFileReference		XFR
[a] This element is introduced in ISO 11783-10 version 4.		

Any XML attribute of type IDREF may only contain a single identifier reference.

8.4.1 Proprietary XML schema extensions

Manufacturer proprietary XML elements and XML attributes in the data transfer file set shall be preceded by a string consisting of a character “P”, the decimal manufacturer code, and an underscore character “_”. The manufacturer proprietary XML element names and tags shall be at maximum 16 characters in length. The data length of each manufacturer proprietary XML attribute shall not exceed 64 characters.

EXAMPLE 1 A proprietary XML element designated “MyElement” in the data transfer file set of a manufacturer with manufacturer code 500 receives the XML element name:

“P500_MyElement”.

EXAMPLE 2 A proprietary XML attribute designated “MyAttribute” from the same manufacturer receives the XML attribute name:

“P500_MyAttribute”.

The contents of the manufacturer proprietary XML elements and XML attributes shall be ignored when the data transfer file set is processed by a MICS or FMIS of another manufacturer. On the MICS, when the data transfer file set originating from the FMIS contains proprietary XML elements or XML attributes, these XML tags and their values may be omitted in the data transfer file set originating from the MICS back to the FMIS.

The XML element AttachedFile (AFE) is introduced in ISO 11783-10 version 4 to enable transfer of manufacturer proprietary data in separate files that are linked to the standard data transfer file set.

It is recommended that the amount of proprietary data are kept as low as possible. Reasons for keeping the amount of proprietary data low are that this data cannot be interpreted in multi-manufacturer systems and may not be retained. Proprietary data are allowed for reasons of being able to extend the standard for manufacturer internal use.

8.5 XML data transfer files

The name of the main XML coded data transfer file containing the root XML element ISO11783_TaskData shall be set to “TASKDATA.XML”. When the data are transferred using a removable storage device, the MICS shall support access to this file in a directory named “TASKDATA” located in the root directory of the transfer medium. Both this directory name and the data transfer file names are case-sensitive and all the characters are in upper case. Any other medium used to transfer the data between FMIS and MICS is proprietary (see Figure 1).

The main data transfer file contains coding data and a number of tasks. Figure 8 illustrates the structure of XML elements in the data transfer file set. There can be references to other XML files containing coding data or tasks inside the main data transfer file. References to XML files are accomplished by use of the XML element XFR. XML file references cannot be nested. This means that only the main XML file can include XML elements XFR. There can be a single XML element in the referenced external XML file. The elements shall be embedded in an XML root element named XFC. A referenced XML file shall only contain top-level XML elements of the same type — for example, it may include definitions of customers but not of customers and of partfields together. Top-level XML elements may be defined both in the main and in the referenced XML files.

Each XML coded data transfer file shall start with an XML identification section and shall be well formed:

```
    < ?xml version = ”1.0” encoding = ”UTF-8”? >
```

In the root data transfer file, all XML elements are always embedded in the global root structure:

```
< ISO11783_TaskData VersionMajor = ”...” VersionMinor = ”...”
TaskControllerManufacturer = ”...”
       TaskControllerVersion = ”...” ManagementSoftwareManufacturer = ”...”
ManagementSoftwareVersion = ”...”
       xmlns:xsi = ”http://www.w3.org/2001/XMLSchema-instance”
       xsi:noNamespaceSchemaLocation = ”… ” DataTransferOrigin = ”...” >
< /ISO11783_TaskData >
```

The number of coding data elements supported by the recipient of a data transfer file set shall, at a minimum, be 2 000 per element type, and the overall number of coding data elements supported by the recipient shall, at a minimum, be 20 000. A data transfer file set containing more than 2 000 coding data elements per element type or more than 20 000 coding data elements is at risk of not being able to be processed by a recipient. This supported data elements limitation does not apply to data elements that are not classified as coding data (see Table 2). An example of data elements not classified as coding data are the geometry elements, e.g. the number of Points in a boundary is not limited.

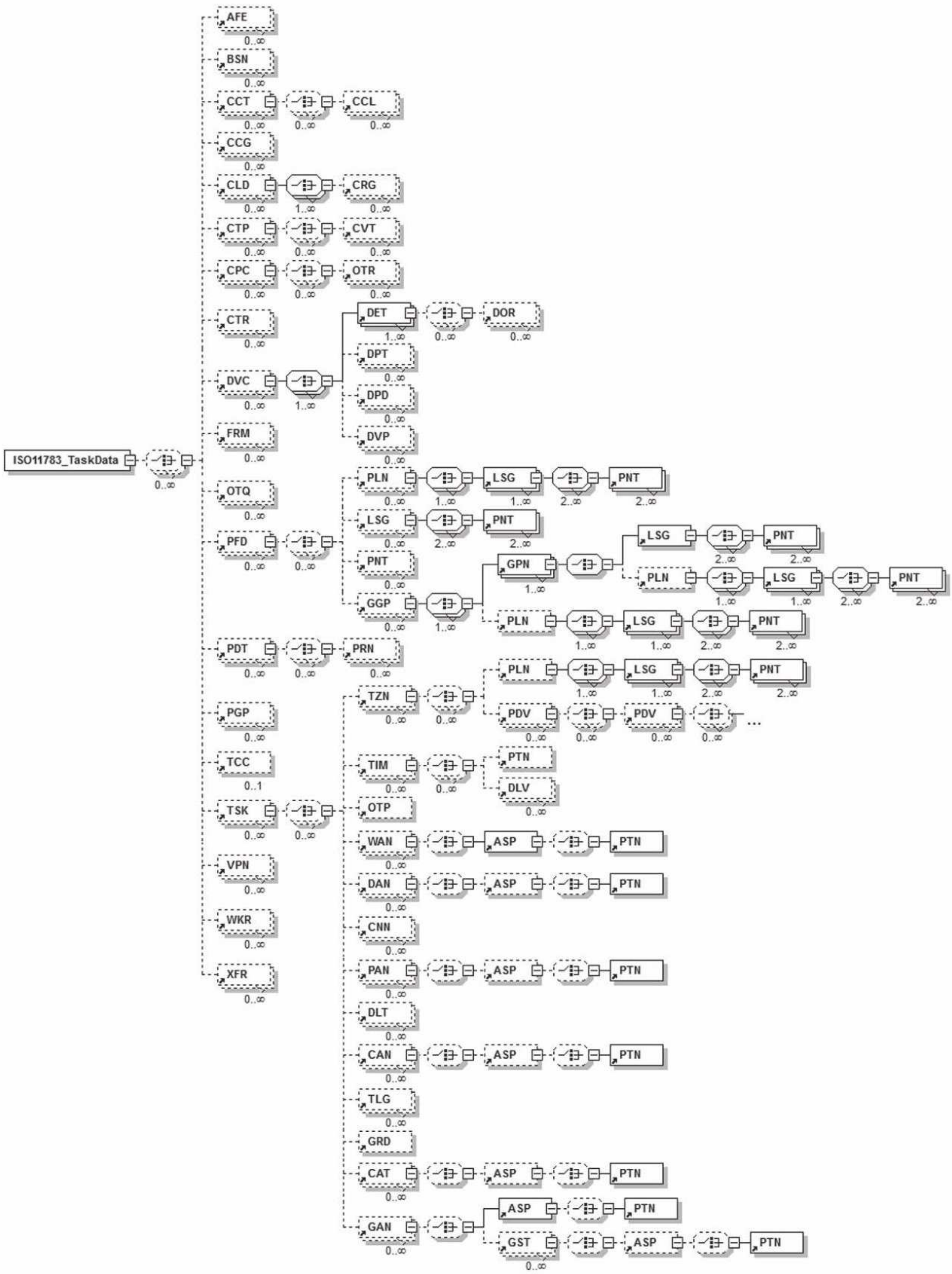


Figure 8 — Schematic structure of the data transfer file set

Figure 9 shows the usage of referenced XML files.

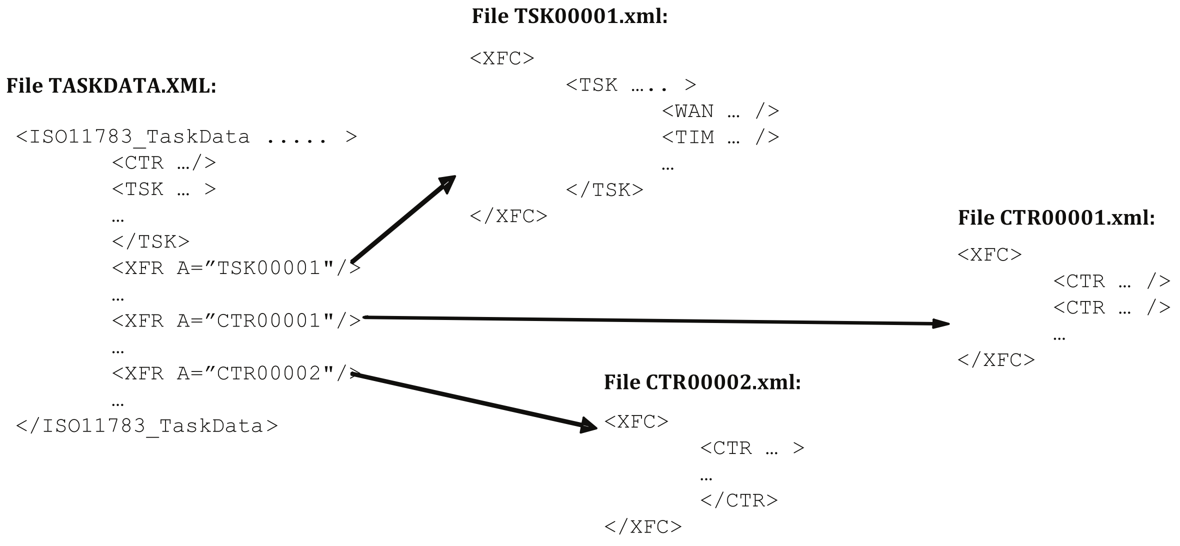


Figure 9 — XFR/XFC XML file references

The direction indication attribute in the XML header section (FMIS→MICS or MICS→FMIS) is to be handled and updated properly by both FMIS and MICS.

For an update of coding data, the MICS is allowed to add new instances of XML elements only. This means that modification of FMIS coding data can only be done by the FMIS.

The TC or TC configuration program shall be prepared to receive either a single XML data transfer file or set of XML data transfer files from the management information system. The management information system shall be prepared to receive either a single data transfer file or a set of XML data transfer files from the MICS, and both can refer to binary files.

8.6 Binary data transfer files

8.6.1 General

There are three types of binary file allowed to be used as part of the data transfer file set. One is to contain gridcell values. The second is to contain binary-coded log data. Both binary files belong to a task. The third binary file, introduced in ISO 11783-10 version 4, is used for the binary encoding of the Point geometry elements.

8.6.2 Grid binary file structure

The gridcells of a Grid can only be defined in a binary file. Two types of grid are supported: the first type of grid contains TreatmentZone codes; the second type of grid contains ProcessDataVariable values.

The first type is used when a limited number of TreatmentZones is defined and the grid functions as a lookup table to the TreatmentZones. In this case, each gridcell in the binary grid file contains a TreatmentZoneCode of the TreatmentZone to which that gridcell belongs. This grid type has at a maximum one value per gridcell, which is an unsigned 8-bit integer coded TreatmentZoneCode.

EXAMPLE 1 Grid that contains TreatmentZoneCodes:

The task data XML file contains a grid definition with an empty TreatmentZoneIdRef attribute:

< GRD A = ”58.096653” B = ”8.54321” C = ”1.5E-3” D = ”1.4E-3” E = ”200” F = ”300” G = ”GRD00001” I = ”1”/ >

The binary file contains: (TreatmentZoneCodes)

(1)(4)(3)(6)…

Each value inside brackets is a record and represents the binary-coded treatment zone code. The brackets are for better reading purposes in this example and are not part of the binary format. The relation between the binary file and the grid type 1 specification is shown in Figure 10.

File TASKDATA.XML:

```
<ISO11783_TaskData ..... >
     <CTR … />
     <TSK … >
          …
          <GRD A="58.096653" B="8.54321" C="1.5E-3" D="1.4E-3" E="200" F="300"
G="GRD00001" I="1"/>
          <TZN A="1" B="MidRate 1" C="2">
               <PDV A="0006" B="10000"/>
          </TZN>
          <TZN A="2" B="MidRate 2" C="3">
               <PDV A="0006" B="12000"/>
          </TZN>
          <TZN A="3" B="HighRate 3" C="4">
               <PDV A="0006" B="14000"/>
          </TZN>
          <TZN A="4" B="HighRate 4" C="5">
               <PDV A="0006" B="15000"/>
          </TZN>
          <TZN A="5" B="MaxRate 5" C="6">
               <PDV A="0006" B="17000"/>
          </TZN>
          <TZN A="6" B="MaxRate 6" C="7">
               <PDV A="0006" B="19000"/>
          </TZN>
          …
     </TSK>
</ISO11783_TaskData>
```

File GRD00001.bin:

Contents of the binary coded GridCell file GRD00001.bin:

(1)(4)(5)(1)……

(Each record represents a reference to the treatment zone for the grid cells in order of the first grid cell in the grid. So, grid cell 1 would reference treatment zone 1, grid cell 2 would reference 4, grid cell 3 would reference 5 and so on.)

Figure 10 — Grid type 1 binary file example

The second type of grid is used when the ProcessDataValue attribute of the ProcessDataVariable is not classified into a limited number of TreatmentZones. In this case, one TreatmentZone, possibly with multiple ProcessDataVariables, is defined as a template for the binary grid file. There are no values of the XML attributes “ProcessDataValue” in this template definition in the XML file specified and, instead, these values are coded in the binary grid file as signed 32-bit integers. For byte ordering of the 32-bit integer data type, the same rules apply as defined in ISO 11783-6.

EXAMPLE 2 Grid that contains ProcessDataVariable ProcessDataValues

The task data XML file contains a Grid definition with a TreatmentZoneCode attribute:

< GRD A = ”58.096653” B = ”8.54321” C = ”1.5E-3” D = ”1.4E-3” E = ”200” F = ”300” G = ”GRD00001” I = ”2” J = ”6”/ >

and a TreatmentZone prototype without ProcessDataValue attributes:

< TZN A = ”6” B = ”Precision Farming” C = ”2” >

< PDV A = ”0001” B = ”0”/ >

< PDV A = ”0006” B = ”0”/ >

< PDV A = ”001A” B = ”0”/ >

< /TZN >

The binary file contains: (ProcessDataValue records [])

[(11100)(15000)(190)]

[(14000)(20000)(200)]

[(19000)(25000)(200)]……

where the first record represents the first gridcell with the following ProcessDataValues:

< PDV A = ”0001” B = ”11100”/ >

< PDV A = ”0006” B = ”15000”/ >

< PDV A = ”001A” B = ”190”/ >

Each value inside the square brackets in the binary file example is a record and represents the ProcessDataValues for one gridcell. The rounded brackets delimit the individual ProcessDataValue values when more than one ProcessDataVariable is present in the referenced TreatmentZone. The brackets are for better reading purposes in this example and are not part of the binary format. Figure 11 shows the binary file related to the grid type 2 specification.

File TASKDATA.XML:

```
<ISO11783_TaskData ..... >
     <CTR … />
     <TSK … >
          …
          <GRD A="58.096653" B="8.54321" C="1.5E-3" D="1.4E-3" E="200" F="300"
G="GRD00001" I="2" J=”1"/>
          <TZN A=”1" B=”Precision Application”>
               <PDV A=”0001" B=”0"/>
               <PDV A=”0006" B=”0"/>
               <PDV A=”001A" B=”0"/>
          </TZN>
          …
     </TSK>
</ISO11783_TaskData>
```

File GRD00001.bin:

Contents of the binary coded GridCell file GRD00001.bin:

[(11100)(15000)(190)]
[(14000)(20000)(200)]
[(19000)(25000)(200)]

(Each record represents the values of the ProcessDataVariables in the grid referenced treatment zone. The first record contains the first grid cell in the
grid. So, the grid references treatment zone 1, grid cell 1 contains 3 values for the 3 processdatavariables, grid cell 2 contains 3 values for the 3 ProcessDataVariables and so on.)

Figure 11 — Grid type 2 binary file example

8.6.3 Log data binary file structure

The TimeLog XML element is used in the task data file to refer to a binary log file. In the task data file, a TimeLog XML element without referenced Time, Position, or DataLogValues shall be used. This TimeLog XML element refers to two files: a TimeLog XML header file and a binary log file. Inside the TimeLog XML header file, the XML elements shall include attributes without any values to define the record structure of the binary log file. All attributes with non-empty value definitions in the TimeLog XML header file contain fixed values which are valid for all binary-coded records in the binary log file. Only the values of the empty value attributes of the TimeLog XML header file are written in the binary log file. The empty value attributes of the Time, Position, and DataLogValue XML elements may be written in any order. The order of the corresponding values in the binary log file shall be according to the order listed in Table 3. An example of an XML-coded TimeLog file that specifies the structure of the binary TimeLog file is given by:

< TIM A = ”” D = ”4” >

< !– PositionNorth, PositionEast and PositionStatus are the required attributes of element Position. – >

< !– In this sample only these three position attributes are written to the binary TimeLog file. – >

< PTN A = ”” B = ”” D = ””/ >

< !– DLV 0 – >

< DLV A = ”0815” B = ”” C = ”DET1”/ >

< !– DLV 1 – >

< DLV A = ”4711” B = ”” C = ”DET2”/ >

< !– DLV 2 – >

< DLV A = ”4522” B = ”” C = ”DET3”/ >

< /TIM >

TimeLog binary data formats are shown in Table 3. Only the attributes that are listed in Table 3 can be used in the binary log file. The attributes from the Time, Position, and DataLogValue XML elements that are not listed in Table 3 shall not be used in a binary log file. The attributes shall be written in the order as listed in Table 3. Note that the local time zone information introduced in ISO 11783-10 version 4 in the Time XML element does not change the definition of the use of the TimeStart attribute in the binary logfile. The TimeStart attribute in the binary logfile shall continue to be expressed in local time.

Table 3 — Binary file record value definitions

Value	XML reference	Binary data type	Not Available value	Definition
TimeStart: time of day	TIM, A	Unsigned 32-bit integer	$FFFFFFFF_{16}$	Local time zone milliseconds since midnight
TimeStart: date	TIM, A	Unsigned 16-bit integer	$FFFF_{16}$	Local time zone days since 1980–01–01
PositionNorth	PTN, A	32-bit integer	$FFFFFFFF_{16}$	10^{-7} degrees WGS-84
PositionEast	PTN, B	32-bit integer	$FFFFFFFF_{16}$	10^{-7} degrees WGS-84
PositionUp	PTN, C	32-bit integer	$FFFFFFFF_{16}$	Millimetres relative to WGS-84 ellipsoid

Table 3 *(continued)*

Value	XML reference	Binary data type	Not Available value	Definition
PositionStatus	PTN, D	Byte	FF_{16}	Position status. Definition references NMEA2000 MethodGNSS parameter. 0 = no GPS fix 1 = GNSS fix 2 = DGNSS fix 3 = Precise GNSS, no deliberate degradation (such as SA), and higher resolution code (P-code) and 2 frequencies are used to correct atmospheric delays 4 = RTK Fixed Integer 5 = RTK Float 6 = Est(DR)mode 7 = Manual input 8 = Simulate mode 9–13 = Reserved 14 = Error 15 = PositionStatus value not available
PDOP	PTN, E	Unsigned 16-bit integer	$FFFF_{16}$	10^{-1} PDOP quality information
HDOP	PTN, F	Unsigned 16-bit integer	$FFFF_{16}$	10^{-1} HDOP quality information
NumberOfSatellites	PTN, G	Byte	FF_{16}	Number of used satellites
GpsUtcTime	PTN, H	Unsigned 32-bit integer	$FFFFFFFF_{16}$	UTC milliseconds since midnight
GpsUtcDate	PTN, I	Unsigned 16-bit integer	$FFFF_{16}$	UTC days since 1980–01–01
#DLV		Byte	n.a.	Number of PDV to follow
DLVn		Byte	n.a.	Ordering number of PDV to follow, starting with 0 for first DataLogValue definition
ProcessDataValue	DLV, B	32-bit integer	n.a.	According to DDI

All attributes in the header file that contain values are specified as having these constant values for all records of the binary file. All DLV elements are indexed and referred to in the binary records according to their definitions order. The number of DataLogValues actually stored is defined as #DLV. An example of the binary log data file is shown in Figure 12, in which there are a total of three DLVs. To allow a dynamic set of DLVs in the binary record, each DLV is identified by its ordering number, DLVn, of definition in the XML file. The example of an XML-coded TimeLog file given above, just before Table 3, specifies that all following records consist of the following set of values:

(TimeStart,PositionNorth,PositionEast,PositionStatus,#DLV,DLV0,PDV0,DLV1,PDV1,DLV2,PDV2)

This means that a time entry can have a maximum of 255 PDVs.

File TASKDATA.XML:

```
<ISO11783_TaskData ..... >
     <CTR … />
     <TSK … >
          <TLG A=”TLG00001"/>
          …
     </TSK>
     …
</ISO11783_TaskData>
```

File TLG00001.xml:

```
<TIM A=”” D=”4">
     <PTN A=”” B=”” D=””/>
     <DLV A=”0815" B=””
C=”DET1"/>
     <DLV A=”4711" B=””
C=”DET2"/>
</TIM>
```

File TLG00001.bin:

The binary file will contain the log values in the following record format in binary:

(TimeStart, PositionNorth, PositionEast, PositionStatus, #DLV, DLV0, PDV0, DLV1, PDV1)

for example:

(2005-05-02T16:32:00, 51.00678, 6.03489, 1,2,0,10,1,15)

Figure 12 — Log data binary file example

8.6.4 Point data binary file structure

A series of Point XML elements can be transferred in a binary file as an alternative encoding to its XML definition. This data transfer method provides the possibility to use a higher range of accuracy. Within the binary format, the Position North and East are specified as 64-bit-integers. In the task data file, the Point XML element may refer to a binary file. In case of a reference to a binary file, the Point XML elements shall include attributes without any values to define the record structure of the binary file. All attributes with non-empty value definitions in the Point XML element definition contain fixed values which are valid for all binary-coded records in its referenced binary file. Only the values of the empty value attributes of the Point XML element are written in the binary file. The empty value attributes may be defined in the Point XML element in any order. The order in which the values for these attributes are written to or read from the binary file shall be according to the order of their listing in Table 4. An example of an XML-coded Point element that specifies the structure of the binary Point file is given by

```
< LSG A = ”6” E = ”1” B = ”Line1” D = ”2000” C = ”20” >
   < PNT A = ”2” C = ”” D = ”” E = ”” F = ”1” J = ”PNT00001”/ >
< /LSG >
```

The attributes of the Point XML element that can be transferred via a binary file are specified in Table 4:

Table 4 — Binary Point file record value definitions

Value	XML reference	Binary Data Type	Not Available value	Definition
PointType	PNT, A	Unsigned 8-bit integer	FF_{16}	Enumeration of PointTypes
PositionNorth	PNT, C	64-bit integer	$FFFFFFFFFFFFFFFF_{16}$	10^{-16} degrees WGS-84
PositionEast	PNT, D	64-bit integer	$FFFFFFFFFFFFFFFF_{16}$	10^{-16} degrees WGS-84

Table 4 *(continued)*

Value	XML reference	Binary Data Type	Not Available value	Definition
PositionUp	PNT, E	32-bit integer	$FFFFFFFF_{16}$	Millimetres relative to WGS-84 ellipsoid
PointColour	PNT, F	Unsigned 8-bit integer	FF_{16}	Colour of the point Format: palette like ISO 11783-6
PointHorizontalAccuracy	PNT, H	Unsigned 16-bit integer	$FFFF_{16}$	RMS error in mm
PointVerticalAccuracy	PNT, I	Unsigned 16-bit integer	$FFFF_{16}$	RMS error in mm

NOTE The binary representation for transferring point data are introduced in ISO 11783-10 version 4.

8.7 Device descriptor object pool

The purpose of the device descriptor object pool is to specify device-related properties on both the MICS and the FMIS. The device descriptor object pool can be supplied by the device manufacturer as a set of XML definitions and can be integrated into the FMIS through a data import. Once the device descriptor object pool is known at the FMIS, the device descriptor object pool can be used for task planning and included in the data transfer file set from FMIS to MICS.

If in a separate file, the device descriptor object pool shall begin with an XML identification section. Its contents shall follow the definitions of the data transfer file set schema for devices. The used identifiers of XML elements shall be unique and will be transferred into the unique identifier namespace of the farm-management system by the FMIS. The device descriptor data file has to contain all information about a device that is necessary when using it for a task. For example, a device descriptor for a sprayer contains the geometry of the sections or nozzles, the number of tanks and their volumes, and supported process data variables. This information shall be specified by DeviceElement, DeviceProperty, and DeviceProcessData XML elements in the file.

The first element of DeviceElements represents the device itself and gets ElementNumber = 0. For this DeviceElement, the ParentIdRef refers to the XML element device. See the XSD for the description of allowed values. For all other (sub-)device elements on a device, the ParentIdRef can refer to a DeviceElement to describe a hierarchical structure or to a device for device elements that do not have a sub-position in a hierarchy but belong directly to the device.

The device descriptor data can originate from devices on the MICS. Device descriptors are transmitted to the TC on the ISO 11783 network using the appropriate process data commands specified in Annex B. Thus, the TC shall be able to receive the device descriptor to determine whether a task can be executed as specified. The TC shall also store the device descriptor for transfer back to the FMIS. For this transfer of the device descriptor data over the ISO 11783 network, a set of objects is defined. The objects containing the device descriptor data are described in Annex A.

The procedure for transfer of the device descriptor data over the network is the following.

a) The client shall determine if the TC has available memory by transmitting a request object-pool transfer message. The TC acknowledges this message with a request object-pool transfer response message. The client shall check the status code returned. If enough memory is available, the client can proceed.

b) The client uses the transport protocol (see ISO 11783-3) or extended transport protocol (see ISO 11783-3) to transmit the updated or new object(s) to the TC. Normal handshaking, error checking, and retransmission, in accordance with ISO 11783-3, shall be implemented.

c) Upon completion, the client shall transmit an object-pool activate message to the TC to indicate that the update is now complete and ready for use.

d) The TC shall respond with the object-pool activate response message.

A client may update its device descriptor object pool at runtime. The runtime updates to the device descriptor shall be limited to changes related to language and/or units of measure settings. These changes enable a TC user interface to update the device related information to appear in a language and units of measure format that the operator selected. No restrictions are made on when during runtime, device descriptor object pool updates may take place; however, client designers shall avoid, as much as possible, performing device descriptor updates during those times when a task is active. When the client updates its device descriptor object pool at runtime, the update shall include the device object, where the original structure label shall remain unchanged and the localization label shall be updated to the new localization settings. Next to the device object, only objects of type DeviceValuePresentation may be transmitted. Transfer of the updated objects is done using the same messages and procedures used to transfer the device descriptor object pool at initialization. The designators may be updated with the change designator command.

If the client wants to change its pool with changes other than these, it shall do so by transmitting a complete device descriptor object pool, where the device object shall include an updated structure label. The structure label uniquely identifies the set of device elements, their property values, and their process data variables. The procedure for transmitting an updated device descriptor object pool shall be as follows:

a) The client transmits a de-activate object pool message.

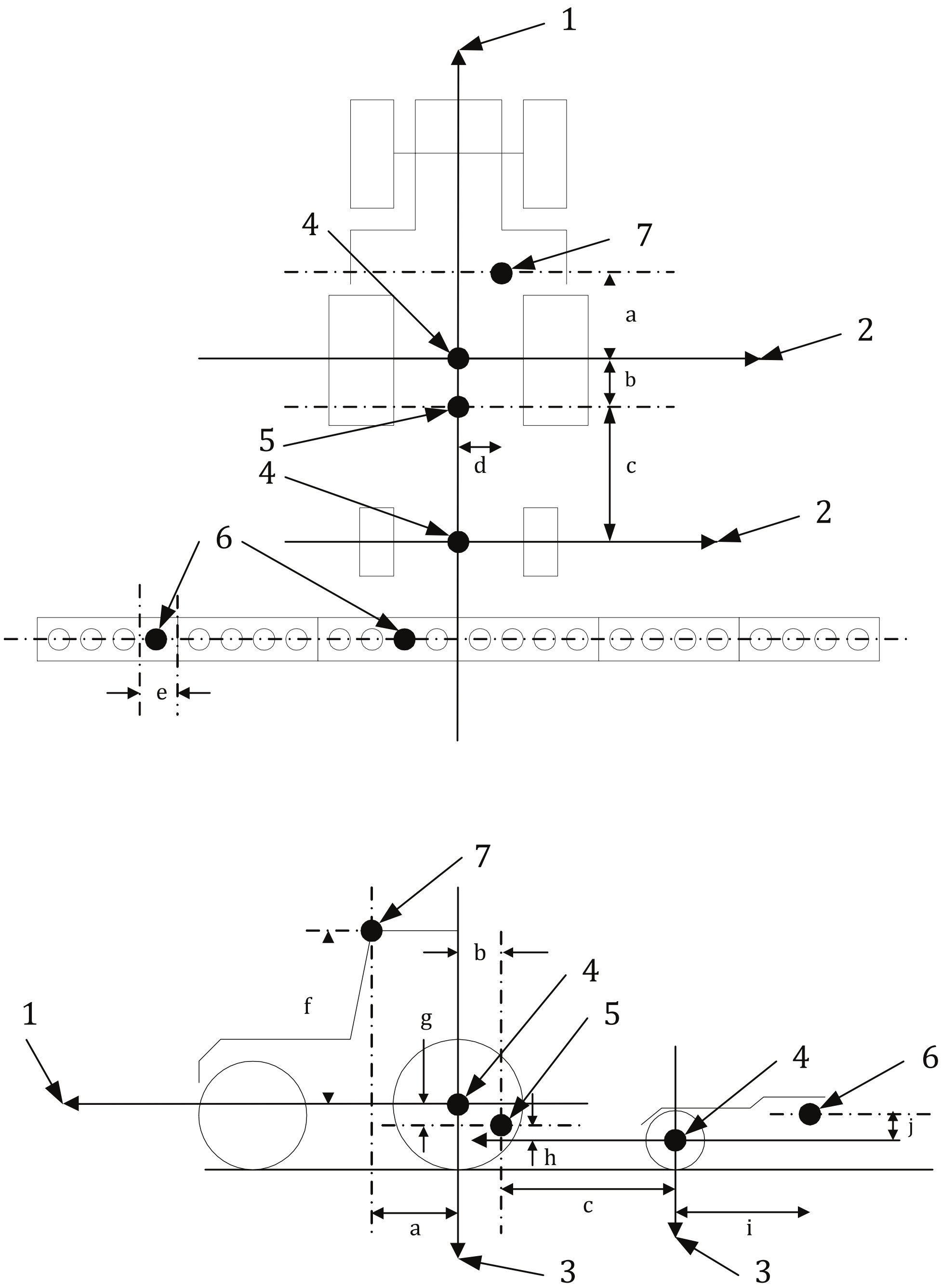
b) The client transfers the device descriptor object pool by using the same messages and procedures used to transfer the pool at initialization.

In the case of errors in the update to the device descriptor object pool, the TC shall indicate the errors in an object-pool activate response message. The TC shall delete the entire device descriptor object pool from volatile memory (including the device descriptor object pool as it existed prior to the update) and may inform the operator, by an alarm-type method, of the suspension of the client and the reason for the deletion of the device descriptor object pool from volatile memory.

For the geometry description of a device, the following information is needed: numbering and specification of DeviceElements, supported ProcessDataVariables, specification of DeviceProperties.

NOTE In order to obtain a general description, no difference is made between an implement, a sensor system, or a tractor. The device descriptor is usable both for implements and sensor systems with a GPS receiver and for a tractor with fixed DeviceElements.

Figure 13 shows the reference points in a connected system. All reference points are described by DeviceElements and the relative distances can be specified by process data variable values, defined in the data dictionary specified in ISO 11783-11.



Key

1	device x-axis (positive direction)	7	NavigationReferencePoint (NRP)	f	NRP_Z.
2	device y-axis (positive direction)	a	NRP_X.	g	CRP_Z (Tractor).
3	device z-axis (positive direction)	b	CRP_X (Tractor).	h	CRP_Z (Implement).
4	DeviceReferencePoint (DRP)	c	CRP_X (Implement).	i	ERP_X.
5	ConnectorReferencePoint (CRP)	d	NRP_Y.	j	ERP_Z.
6	DeviceElementReferencePoint (ERP)	e	ElementWorkingWidth.		

Figure 13 — Device geometry definitions

Each device has a coordinate system. The centre of the device coordinate system is defined as a DeviceReferencePoint with coordinates (0,0,0). The location of a DeviceElement inside a device is relative to the DRP and the coordinate system is a right-hand rule coordinate system with the following axis definitions.

— The x-axis is specified as positive in the normal driving direction.

— The y-axis is specified as positive to the right side of the device relative to the normal driving direction.

— The z-axis is specified as positive downward toward the ground plane.

For a tractor, the DRP is the centre of the rear axle. For a wheeled implement, the DRP is the centre of the front axle. In other cases, the DRP can be chosen freely, e.g. DRP = CRP or DRP = ERP. If there are angles on the device geometry, the device control function is responsible for recalculating the new DRP/CRP and sending this as dynamic data to the TC.

The NavigationReferencePoint and ConnectorReferencePoints refer to the location of the DeviceElements that are of DeviceElementType “Navigation” and “Connection”. In case a NavigationReferencePoint is not reported by a tractor but is required by a TC, the TC can provide alternative means to specify the offsets of the navigation system’s reference point.

The ConnectorReferencePoint specifies the position of one or more mounting facilities, e.g. rear and front three-point hitch or drawbar. For the three-point hitch, the CRP is the centre of the lower link points. If there are dynamic changes (e.g. by a steering axle of the device), the device has to calculate the new CRP and send it to the TC.

The ERP is the centre of one (or of a group of) device element(s), i.e. the centre of a spraying boom.

DeviceProperty or DeviceProcessData objects shall be defined for a device and referenced by DeviceElementObjects to provide the values for DeviceElement offsets, time delays, capacities, etc. In case these values are expressed in DeviceProperties, changes shall only be applied through a new Device descriptor upload. In case these values are communicated through DeviceProcessData, changes can be made during a connection with a TC and while a task is active. Each of the parameters for the DeviceElement offsets, time delays, capacities, etc. shall be included in the Device Element as either a DeviceProperty or as a DeviceProcessData object. Including the same parameter, i.e. same DDI in a Device Element via both a DeviceProperty and a DeviceProcessData object is not allowed.

To avoid conflicts with Device descriptors and logged data previously stored within the TC, DL, and FMIS, it is the responsibility of the client to ensure that the structure label uniquely describes the relationships between device elements including element and object numbering in perpetuity.

Annex A
(normative)

Device descriptor objects

A.1 General

Each device, whether machine or sensor system, is defined by the XML element “Device”, at least one XML element “DeviceElement” and the optional device-element-related XML elements “DeviceProcessData”, “DeviceProperty”, and “ValuePresentation”. By defining these device descriptors of XML elements as objects, similar to the way in which objects are used inside an object-pool definition of ISO 11783-6, all device-related data can be transferred to the TC by means of the ISO 11783 transport protocol and/or extended transport protocol. This annex describes the representation of the device descriptor XML elements as binary objects. Note that several attributes in this representation are coded as UTF-8 strings. These strings do not have a preceding byte-order mark (BOM). The maximum number of bytes per character of a UTF-8 coded string is 4 bytes. Therefore, the maximum length of the byte arrays is four times the length of the UTF-8 coded strings as specified in the corresponding XML definitions.

All object IDs shall be unique inside the whole device descriptor object pool.

When no object is referenced, the NULL Object ID shall be used. The NULL Object ID has value 65535 ($FFFF_{16}$).

Data transfer of device descriptor objects enables the TC to get the actual description of a connected device. The received object definitions of a certain device are converted into XML elements and stored in the data transfer file set. If the object definitions are not part of the current task data file (as received from FMIS), then the new device shall be added to the task data file. If the same device exists in the current task data file, then the software and description structure label between both data sets shall be checked. It is up to the mobile system either to update the definitions inside the task data file by the newly received device descriptor data or store the received description objects as new device XML elements.

A.2 Definition of DeviceObject

DeviceObject is the object definition of the XML element device. Each device shall have one single DeviceObject in its device descriptor object pool. Table A.1 lists the attributes of the DeviceObject. The structure label and the extended structure label attributes are defined to retrieve a specific device descriptor object pool of a device. In case a device has multiple configurations, each configuration can be identified by a unique structure label and extended structure label attribute combination. The contents of the Device structure label and the Device extended structure label are manufacturer specific.

The two attributes for the extended structure label shall only be used when the versions of the TC and of the client are both reported as version 4 or higher. When either the TC or the client in a connection are version 3 or earlier, then the communication shall fall back to the lowest common version and the 2 attributes for the extended structure label shall both not be used. A version 4 or later client that does not use the extended structure label shall report the extended structure label to be of 0 bytes length by setting the value of the Number of Extended Structure Label bytes to 0 when transferring the DeviceObject to a version 4 or later TC.

The ClientNAME as provided in the DeviceObject shall be the same as the CF’s address claim.

Table A.1 — DeviceObject definition

Attribute Name	Type	Size bytes	Range or value	Record byte	Description
Table ID	String	3	DVC	1–3	XML element namespace for device.
Object ID	Integer	2	0	4–5	Unique object identifier inside this device descriptor object pool. DeviceObject ID = 0.
Number of designator bytes (N)[a]	Byte	1	0 to 128	6	The number of bytes of the following designator UTF-8 string.
Device designator	UTF-8 string, no BOM	0 to 128		7	Descriptive text to identify this device. This text can be displayed to the operator. The maximum number of characters of the designator is 32.
Number of software version bytes (M)[a]	Byte	1	0 to 128	7+N	The number of bytes of the following software version UTF-8 string.
Device software version	UTF-8 string, no BOM	0 to 128		8+N	Software version indicating text. The maximum number of characters of the software version is 32.
ClientNAME	Double integer	8		8+N+M	NAME of client device. The NAME structure is defined in ISO 11783-5.
Number of DeviceSerial-Numberbytes (O)[a]	Byte	1	0 to 128	16+N+M	The number of bytes of the following DeviceSerialNumber UTF-8 string.
DeviceSerial Number	UTF-8 string, no BOM	0 to 128		17+N+M	Device and manufacturer-specific serial number of the Device or Machine represented by the client, e.g. a unique Vehicle or Product identification code[c]. The maximum number of characters of the serial number is 32.
Device structure label	Byte array	7	0..254 for each byte	17+N+M+O	Label given by device to identify the device descriptor structure. This label allows the device to identify the current version of the device descriptor object pool present in a TC and to determine whether an update is needed. The device structure label byte 1 is transmitted closest to the DeviceSerialNumber parameter.
Device localization label	Byte array	7	0..254 for each of bytes 1 to 6, 255 for byte 7	24+N+M+O	Label given by device to identify the device descriptor localization. Bytes 1 to 6 are defined by the language command PGN (see ISO 11783-7). Byte 7 is reserved and set to FF_{16}. Byte 1 is transmitted closest to the device structure label parameter and is the least significant byte of the DeviceLocalizationLabel in the XML element device.
Number of Extended Structure Label bytes[b]	Byte	1	0 to 32	31+N+M+O	Length of following Extended Structure Label byte array.

[a] The value range for this attribute was extended from 32 to 128 in ISO 11783-10 version 4.

[b] This attribute is introduced in ISO 11783-10 version 4.

[c] Defined in ISO 11783-12, 2nd edition Diagnostics services.

Table A.1 *(continued)*

Attribute Name	Type	Size bytes	Range or value	Record byte	Description
Extended Structure Label[b]	Byte Array	0 to 32	0 to 255 for each byte	32+N+M+O	Continuation of the Label given by Device to identify the Device descriptor Structure. This label allows the device to identify the current version of the device descriptor object pool present in a TC and to determine whether an update is necessary. The Extended Structure Label byte 1 is transmitted closest to the NumberofExtendedStructureLabellbytes parameter.

[a] The value range for this attribute was extended from 32 to 128 in ISO 11783-10 version 4.

[b] This attribute is introduced in ISO 11783-10 version 4.

[c] Defined in ISO 11783-12, 2nd edition Diagnostics services.

A.3 Definition of DeviceElementObject

DeviceElementObject is the object definition of the XML element DeviceElement. The attribute DeviceElementType specifies the type of this particular element definition. The type "device" represents the complete device itself and therefore can exist only once per device descriptor object pool. See Table A.2.

Referable child objects:

DeviceProcessDataObject

DevicePropertyObject

Table A.2 — DeviceElementObject definition

Attribute Name	Type	Size bytes	Range or value	Record byte	Description
Table ID	String	3	DET	1–3	XML element namespace for DeviceElement.
Object ID	Integer	2	1 to 65534	4–5	Unique object identifier.
DeviceElement-Type	Byte	1	1 to 7	6	1 = device 2 = function 3 = bin 4 = section 5 = unit 6 = connector 7 = navigation reference See detailed description below this table.
Number of designator bytes (N)[a]	Byte	1	0 to 128	7	The number of bytes of the following designator UTF-8 string.
DeviceElement Designator	UTF-8 string, no BOM	0 to 128		8	Descriptive text to identify this device element. The maximum number of characters of the designator is 32.

[a] The value range for this attribute was extended from 32 to 128 in ISO 11783-10 version 4.

Table A.2 *(continued)*

Attribute Name	Type	Size bytes	Range or value	Record byte	Description
DeviceElement Number	Integer	2	0 to 4095	8+N	Element number for process data variable addressing according to the definitions in process data variable definitions in Annex B.
Parent ObjectId	Integer	2	0 to 65534	10+N	Object ID of parent DeviceElementObject or DeviceObject in order to establish a hierarchical order of DeviceElements.
Number of objects to follow	Integer	2		12+N	Number of following object references.
REPEAT: ObjectId	Integer	2		14+N+object	List of references to DeviceProcessDataObjects or DevicePropertyObjects.

[a] The value range for this attribute was extended from 32 to 128 in ISO 11783-10 version 4.

A.3.1 Device element type — Device

The device descriptor object pool shall have one device element of type device with device element number = 0, which represents the complete device and makes it addressable for the TC.

A.3.2 Device element type — Function

This device element type can be used as a generic device element to define individually accessible components of a device like valves or sensors. The detailed function of this device element type is defined by its list of device process data and device properties attributes. Examples of how the function device element types are used to describe device functions are provided in Annex F.

A.3.3 Device element type — Bin

This is, for instance, the tank of a sprayer or the bin of a seeder.

A.3.4 Device element type — Section

This is, for instance, the section of a spray boom, seed toolbar, or planter toolbar. A section may provide device geometry definitions (x, y, z) and a working width next to supported process data elements as device process variable values or device property values.

A.3.5 Device element type — Unit

This device element type is, for example, used for spray boom nozzles, seeder openers, or planter row units. It is intended as a layer below the device element type section in the hierarchical device element structure. A device element of type unit is typically the most detailed level that can be addressed in a device. An example is a number of row units that belong to the same section.

A.3.6 Device element type — Connector

This device element type specifies the mounting/connection position of the device. More than one connector can be defined for one device (e.g. a tractor may provide front-end mounting and rear-end mounting connection locations). A connector element shall provide its device geometry definitions (x, y, z) relative to the device reference point as device process data values or as device property values, even when the device reference point is the same as the location of the connector (x = y = z = 0).

A.3.7 Device element type — Navigation reference

This device element type defines the navigation reference position for navigation devices such as GPS receivers. Such elements have to reference their position in the x-, y-, and z-direction as device process data values or device property values.

A.4 Definition of DeviceProcessDataObject

The DeviceProcessDataObject is the object definition of the XML element DeviceProcessData. Each object contains a single process data variable definition. See Table A.3.

Referable child object: DeviceValuePresentationObject.

Table A.3 — DeviceProcessDataObject definition

Attribute Name	Type	Size bytes	Range or value	Record byte	Description
Table ID	String	3	DPD	1–3	XML element namespace for DeviceProcess-Data.
Object ID	Integer	2	1 to 65534	4–5	Unique object identifier.
Process data DDI	Integer	2	0 to 65535	6–7	Identifier of process data variable (DDI) according to definitions in Annex B and ISO 11783-11.
Process data properties	Byte	1	0 to 7	8	A bitset combined of: Bit 1 = 1 = member of default set. Bit 2 = 1 = settable. Bit 3 = 1 = control source.[a] Note: Bits 2 and 3 are mutually exclusive.
Process data available trigger methods	Byte	1	0 to 31	9	A bitset combined of: Bit 1 = 1 = time interval. Bit 2 = 1 = distance interval. Bit 3 = 1 = threshold limits. Bit 4 = 1 = on change. Bit 5 = 1 = total. See detailed description in A.4.1 to A.4.5.
Number of designator bytes (N)[b]	Byte	1	0 to 128	10	The number of bytes of the following designator UTF-8 string.
Process data designator	UTF-8 string, no BOM	0 to 128		11	Descriptive text for this device process data. The maximum number of characters of the designator is 32.
Device value presentation object ID	Integer	2	1 to 65535	11+N	Object identifier of DeviceValuePresentation-Object. Use NULL object ID when no Device-ValuePresentationObject is referenced.

a Bit 3 of this attribute is introduced in ISO 11783-10 version 4.

b The value range for this attribute was extended from 32 to 128 in ISO 11783-10 version 4.

A.4.1 Device process data trigger method — Time interval

The device can provide these device process data based on a time interval.

A.4.2 Device process data trigger method — Distance interval

The device can provide these device process data based on a distance interval.

A.4.3 Device process data trigger method — Threshold limits

The device can provide these device process data based on a surpassing of the value threshold.

A.4.4 Device process data trigger method — On change

The device can provide these device process data when its value changes.

A.4.5 Device process data trigger method — Total

These device process data are a total. See 6.6.2 for a description of total functionality.

A.5 Definition of DevicePropertyObject

DevicePropertyObject is the object definition of the XML element DeviceProperty. Each object contains a single DeviceElementProperty definition. See Table A.4.

Referable child object: DeviceValuePresentationObject.

Table A.4 — DevicePropertyObject definition

Attribute name	Type	Size bytes	Range or value	Record byte	Description
Table ID	String	3	DPT	1–3	XML element namespace for DeviceProperty.
Object ID	Integer	2	1 to 6 553 4	4–5	Unique object identifier.
Property DDI	Integer	2	0 to 6 553 5	6–7	Identifier of property (DDI) according to definitions in Annex B and ISO 11783-11.
Property value	Signed integer	4	-2^{31} to $(2^{31}-1)$	8–11	Value of property.
Number of designator bytes (N)[a]	Byte	1	0 to 128	12	The number of bytes of the following designator UTF-8 string.
Property designator	UTF-8 string, no BOM	0 to 128		13	Descriptive text for this device property. The maximum number of characters of the designator is 32.
Device value presentation object ID	Integer	2	1 to 6 553 5	13+N	Object identifier of DeviceValuePresentationObject. Use NULL object ID when no DeviceValuePresentationObject is referenced.

[a] The value range for this attribute was extended from 32 to 128 in ISO 11783-10 version 4.

A.6 Definition of DeviceValuePresentationObject

DeviceValuePresentationObject is the object definition of the XML element DeviceValuePresentation. This object contains the presentation information to display the value of a DeviceProcessData or DeviceProperty object. The device can update these objects when the language and/or units of measure are changed by the operator. See Table A.5.

Referable child objects: none.

Table A.5 — DeviceValuePresentationObject definition

Attribute name	Type	Size bytes	Range or value	Record byte	Description
Table ID	String	3	DVP	1–3	XML element namespace for DeviceValuePresentation.
Object ID	Integer	2	1 to 65534	4–5	Unique object identifier.
Offset	Signed integer	4	-2^{31} to $(2^{31}-1)$	6–9	Offset to be applied to the value for presentation.

[a] The value range for this attribute was extended from 32 to 128 in ISO 11783-10 version 4.

Table A.5 *(continued)*

Attribute name	Type	Size bytes	Range or value	Record byte	Description
Scale	Float	4	0,000000001 ... 100000000,0	10–13	Scale to be applied to the value for presentation.
Number of decimals	Byte	1	0 to 7	14	Specify number of decimals to display after the decimal point.
Number of Unit-Designator bytes (N)[a]	Byte	1	0 to 128	15	The number of bytes of the following unit designator UTF-8 string.
UnitDesignator	UTF-8 string, no BOM	0 to 128		16	Unit designator for this value presentation. The maximum number of characters of the designator is 32.
[a] The value range for this attribute was extended from 32 to 128 in ISO 11783-10 version 4.					

A.7 Object hierarchy

Figure A.1 illustrates the hierarchy of device-related objects. A device descriptor object pool shall contain only a single device object and can have multiple DeviceElement, DeviceProcessData, DeviceProperty, and DeviceValuePresentation objects. The DeviceProcessData and DeviceProperty objects can be referenced from multiple DeviceElement objects. The DeviceValuePresentation object can be referenced from multiple DeviceProcessData or DeviceProperty objects.

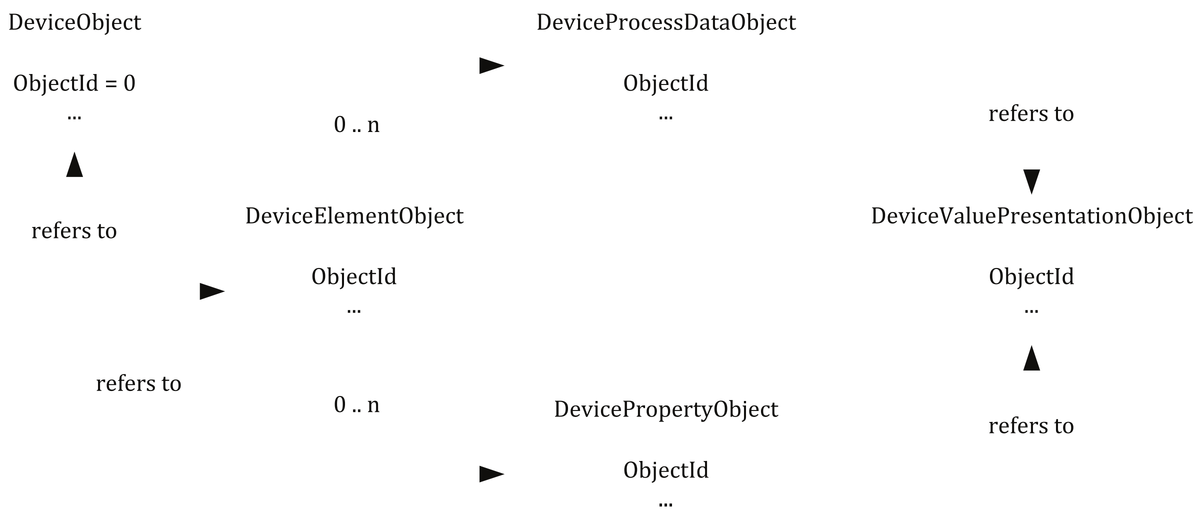


Figure A.1 — Device descriptor object relations

Annex B
(normative)

Message definitions

B.1 General

B.1.1 Process Data message

The Process Data message is used for the transmission of device descriptor data, measured data, and/or setpoint commands to one or more controllers. The first nibble of the first byte of the message identifies the command or the action that the controller(s) are required to perform. The remainder of the process data message for commands 0_{16} and 1_{16} is different from the remainder of the process data message of the commands 2_{16} until A_{16} and D_{16} until F_{16}. Table B.1 lists these commands and specifies the definition or location in this annex.

Process Data messages can be transmitted on the network regardless of whether a TC or DL sets the "Task totals active" bit to 1 in the Task Controller Status message. Process Data messages may also occur on the network outside the scope of the communication between TC or DL and their connected clients.

When the Process Data message is used with proprietary data contents, i.e. a DDI value out of the proprietary DDI value range is referenced then the format of the Process Data message shall be according to the Process Data message definition in this annex.

Process Data requests and measurement commands shall be replied to before a next request or measurement command is allowed to be transmitted. This synchronization of request-response process data message pairs has been introduced in ISO 11783-10 version 4. See 6.8 for additional detail regarding bandwidth management of process data communication.

B.1.2 Command parameter

Data length:	4 bits
Data range:	0 to 15
SPN:	5 199

B.2 Process Data message parameter group

The Process Data message parameter group is defined as:

Data page:	0
Extended data page:	0
PDU format:	203
PDU specific:	Destination address
Default priority:	3 for messages with Command values 3_{16}, A_{16}, E_{16}, or F_{16}.
	4 for messages with Command value D_{16}

	5 for messages with Command values 0_{16}, 1_{16}, 2_{16}, or 4_{16} through 9_{16}
Parameter group number:	51968 ($00CB00_{16}$)
Data length:	Variable, minimum of 8 bytes.

The message may be sent to the global destination address, requiring that all controllers parse and determine the disposition of the message. The message should only be processed when the controller has determined that it has been addressed to its location.

The differentiation of the default priority of the Process Data message into 3 different levels depending on the Command value has been introduced in ISO 11783-10 version 4. Prior to version 4, the default priority was 3 for all Process Data messages. Using different message priorities has been introduced in version 4 to align this part of ISO 11783 with the priority recommendations specified in ISO 11783-3, giving higher priority to control and connection maintenance messages versus request and acknowledgement messages.

A single TC shall be represented by a single control function on the network. All TC communication on the network shall occur between the client identified as a working set master and the TC server. This requirement has been introduced in ISO 11783-10 version 4. The introduction in ISO 11783-10 version 4 of the peer control communication method has eliminated the need for a TC consisting of multiple control functions represented by a multi-member working set on the network. Note that a multi-member working set can still be announced on the network for the connection of such a multi-member working set to a VT. If from this multi-member working set the same working set master establishes a connection as client to a TC, then only the working set master from that working set shall communicate with the TC.

Table B.1 — Process Data commands

Command value	Meaning
0_{16}	Subcommand for determining the technical capabilities of a TC, DL, or client. These technical data messages are defined in B.5.
1_{16}	Subcommand for the transfer and management of device descriptors. These device descriptor messages are defined in B.6.
2_{16}	Request value command. The value of the data entity specified by the data dictionary identifier is requested. The layout of this message is defined in B.3.
3_{16}	Value command: the process data value is the value of the data entity specified by the data dictionary identifier. This command is used both to answer a request value command and to set the value of a process data entity. The layout of this message is defined in B.3.
4_{16}	Measurement command: the process data value is the time interval for sending the data element specified by the data dictionary identifier. The client has to send the value of this data element to the TC or DL cyclic with this time interval. The unit of the time interval is milliseconds. The layout of this message is defined in B.3.
5_{16}	Measurement command: the process data value is the distance interval for sending the data element specified by the data dictionary identifier. The client has to send the value of this data element to the TC or DL cyclic with this distance interval. The unit of the distance interval is millimetres. The layout of this message is defined in B.3.
6_{16}	Measurement command: the process data value is the minimum within threshold for sending the data element specified by the data dictionary identifier. The client has to send the value of this data element to the TC or DL when the value is higher than the threshold value. The unit of the threshold is specified by the data dictionary identifier definition. The layout of this message is defined in B.3.

[a] This command value definition is introduced in ISO 11783-10 version 4.

Table B.1 *(continued)*

Command value	Meaning
7_{16}	Measurement command: the process data value is the maximum within threshold for sending the data element specified by the data dictionary identifier. The client has to send the value of this data element to the TC or DL when the value is lower than the threshold value. The unit of the threshold is specified by the data dictionary identifier definition. The layout of this message is defined in B.3.
8_{16}	Measurement command: the process data value is the change threshold for sending the data entity specified by the data dictionary identifier. The client has to send the value of this data element to the TC or DL when the value change is higher than or equal to the change threshold since last transmission. Prior to ISO 11783-10 version 3, the client had to send the value when the value change was higher than the change threshold; ISO 11783-10 version 3 and later clarified this to be higher than or equal to the change threshold in order to make this consistent with the DataLogTrigger definition in D.17. The unit of the threshold is specified by the data dictionary identifier definition. The layout of this message is defined in B.3.
9_{16}[a]	Peer control assignment command. This message is used to establish a connection between a setpoint value source and a setpoint value user. A successful assignment enables the setpoint value source to send setpoint value commands directly to the setpoint value user. The layout of this message is defined in B.4.
A_{16}[a]	Set Value and Acknowledge command. This command is used to set the value of a process data entity and request a reception acknowledgement from the recipient. The set value command process data value is the value of the data entity specified by the data dictionary identifier. The layout of this message is defined in B.3. The reply shall be a Process Data Acknowledge message (D_{16}). This message is defined in B.7.
B_{16} and C_{16}	Reserved.
D_{16}	Message is a Process Data Acknowledge (PDACK). This message is specified in B.7.
E_{16}	Message is a Task Controller Status message. This message is specified in B.8.1
F_{16}	Client Task message — sent by the client — is defined in B.8.2.

a This command value definition is introduced in ISO 11783-10 version 4.

B.3 Value and measurement commands

B.3.1 Parameter fields

The second nibble of the first byte and the second byte identify the controllable element that must perform the commanded action. The third and fourth bytes contain the data dictionary identifier that identifies the data entity of which the value is specified in the remainder of the message. The last four bytes in the Process Data message contain the value of the data to be used to perform the specified action.

The byte ordering of the process variable parameter fields is:

Byte 1:	Bits 4 to 1	Command, possible values 2_{16} until A_{16} (see Table B.1)
Byte 1:	Bits 8 to 5	Element number (LSNibble) (see B.3.2)
Byte 2:		Element number (MSB) (see B.3.2)
Byte 3:		DD identifier (LSB) (see B.3.3)
Byte 4:		DD identifier (MSB) (see B.3.3)
Bytes 5 to 8:		Process variable value (see B.3.4)

B.3.2 Element number

The element number is a 12-bit field that comprises Byte 1, bits 5 to 8 and Byte 2. It indicates the specific controllable element that must act on the command. The set of numbers is defined in the device descriptor. The syntax of {element number, parent number} is used to describe the configuration.

Data length: 12 bits

Data range: 0 to 4095

SPN: 5200

EXAMPLE A sprayer of three boom sections with six nozzles on the first boom section, eight nozzles on the second boom section, and six nozzles on the third boom section {element number, parent number}:

{0,Null}

{1,0}, {2,0}, {3,0}

{4,1}, {5,1}, {6,1}, {7,1}, {8,1}, {9,1}

{10,2}, {11,2}, {12,2}, {13,2}, {14,2}, {15,2}, {16,2}, {17,2}

{18,3}, {19,3}, {20,3}, {21,3}, {22,3}, {23,3}

The element number would be 0 to address the implement sprayer. The element number would be 2 to address the second boom section. To address the second nozzle on the third boom, the element number would be 19.

The element numbers for elements of the same type that are located at the same level in the hierarchy shall have ascending values from left to right, from front to rear or from top to bottom within the device. When same type and hierarchy level elements are located in a matrix layout, counting shall start in the left to right direction, continues with front to rear and ends with the top to bottom direction. The device elements along a left to right, a front to rear, or a top to bottom arrangement are not required to be geometrically connected to each other. Overlaps or gaps may exist but the recommendation is that device elements that represent the sections of a device are adjacent to each other. When controlling devices based upon device location, device element offsets, and device element dimensions, the device element offset and dimension values take precedence over the element numbering.

B.3.3 Data dictionary identifier

This two-byte parameter is the identifier of the data dictionary entity that defines the data contained in the process variable value parameter. The data dictionary entities are listed in ISO 11783-11.

Data length: 2 bytes

Data range: 0 to 65255

SPN: 5201

B.3.4 Process variable value

This four-byte parameter contains the data value for the Process Data message. This value is defined as signed long integer data type.

Data length: 4 bytes

Resolution: per data dictionary entity definition

Data range: per data dictionary entity definition

SPN: 5202

B.4 Peer control assignment messages

The peer control assignment message is used to establish a connection between a setpoint value source and a setpoint value user. The length of this command depends on the direction of this message. When sent from TC to another CF, the message length is 14 bytes and uses transport protocol. When sent to the TC in response to an assignment command, the message length is 8 bytes. This message has been introduced in ISO 11783-10 version 4.

Transmission repetition rate:			On request
Data length:			14 bytes (assignment command), 8 bytes (assignment response)
Parameter group number:			Process data, destination-specific
Byte 1:	Bits 4 to 1	1001	Control assignment
Byte 1:	Bits 8 to 5		Element number (LSNibble) (see B.3.2)
Byte 2:			Element number (MSB) (see B.3.2)
Byte 3:			DD identifier (LSB) (see B.3.3)
Byte 4:			DD identifier (MSB) (see B.3.3)
Byte 5:	Bits 4 to 1		Control assignment mode
	Value = 0: Assign Receiver		The ElementNumber/DDI combination specified in this command starts to receive data from the CF function specified by the NAME in this command. This assignment mode value is sent from the TC to the setpoint value user CF.
	Value = 1: Unassign Receiver		The ElementNumber/DDI combination specified in this command no longer receives data from the CF function specified by the NAME in this command. This assignment mode value is sent from the TC to the setpoint value user CF.
	Value = 2: Receiver Ack		The assignment or unassignment for receiving control data by the Element / DDI combination specified in this command has been accepted. This assignment mode value is sent from the setpoint value user CF to the TC.
	Value = 3: Assign Transmitter		The ElementNumber/DDI combination specified in this command is enabled to send setpoint value commands to the setpoint value user CF specified by the NAME in this command. This assignment mode value is sent from the TC to setpoint value source CF.
	Value = 4: Unassign Transmitter		The ElementNumber/DDI combination specified in this command no longer sends setpoint value commands to the setpoint value user CF specified by the NAME in this command. This assignment mode value is sent from the TC to the setpoint value source CF.

Value = 5: Transmitter Ack			The assignment or unassignment for sending setpoint value data by the Element/DDI combination specificied in this command has been accepted. This assignment mode value is sent from the setpoint value source CF to the TC.
Value = 6 to 15			Reserved for future use
Byte 5:	Bits 8 to 5		Destination Element number (LSNibble) of the setpoint value user that this assignment is made with (format of Element number: see B.3.2)
Byte 6:			Destination Element number (MSB) of the setpoint value use that this assignment is made with (format of element number: see B.3.2)
Bytes 7 to 14:			NAME of the CF that this assignment is made with.

NOTE Destination Element number in bytes 5 and 6 is only used if the Type of Assignment is Transmit. The Element Number is that of the setpoint value user device descriptor. This value is to be used by the setpoint value source in the “Value Command” Process Data message sent to the setpoint value user. For the Receiver Assignment messages, set the Destination Element number as FFF_{16}. There exists only one DDI because it must be the same between the setpoint value source CF device descriptor and the target controller CF device descriptor in order for the assignment by the TC to be able to be made.

The assignment response is made by sending the same message back to the TC with the Type of Assignment (Byte 5, bits 4 to 1) changed to the correct value. Byte 5, bits 8 to 5 are set to F_{16} and bytes 6 to 8 are set of FF_{16} and the message is sent as a single packet. Only one assignment or unassignment message shall be sent to a CF before a response or Timeout has occurred.

If the assignment is rejected, the PDACK shall be used as the response. The defined error bits shall be used to indicate the reason for rejection.

The connection is disconnected on the event of task totals status change from active to inactive. If there is a need to disconnect the assignment before the task is inactive or if the other device does not accept the assignment, the assignment message will be sent with the all fields set the same to define the unique data element, except the control assignment mode shall be set for the corresponding unassignment.

B.5 Technical data messages

B.5.1 General

The technical data messages are used to request the characteristics of the TC and participating clients.

B.5.2 Request Version message

The Request Version message allows the TC, DL, and the client to determine the ISO 11783-10 version of the implementation.

Transmission repetition rate:			On request
Data length:			8 bytes
Parameter group number:			Process data, destination-specific
Byte 1:	Bits 4 to 1	0000	Command technical data
	Bits 8 to 5	0000	Parameter Request Version

Bytes 2 to 8:			Reserved, transmit as FF_{16}

B.5.3 Version message

The Version message is sent in response to the request version message and contains the ISO 11783-10 version information of the TC, DL, or client implementation.

Transmission repetition rate:			In response to Request Version message
Data length:			8 bytes
Parameter group number:			Process data, destination-specific
Byte 1:	Bits 4 to 1	0000	Command technical data
	Bits 8 to 5	0001	Parameter Version
Byte 2:	Version number		The version of this part of ISO 11783 that this control function meets.
	0		The version of the DIS (draft International Standard).
	1		The version of the FDIS.1 (final draft International Standard, first edition).
	2		The version of the FDIS.2 and the first edition published as an International Standard.
	3		The version of the second edition published as a draft International Standard (E2.DIS).
	4		The version of the second edition published as the final draft International Standard (E2.FDIS) and as the International Standard (E2.IS).
Byte 3:	Boot time		Maximum number of seconds from a power cycle to transmission of first “Task Controller Status message” (See 6.7.1). Transmit as FF_{16} when this information is not available. Providing boot time information shall only be used by the version information message sent by the TC or DL, the client shall set this value to FF_{16}. The definition of this byte is introduced in ISO 11783-10 version 3, the value for ISO 11783-10 version 2 and prior is FF_{16}.
Byte 4:	Provided options		The options specified in ISO 11783-10 that this TC, DL, or client meets. The definition of this byte is introduced in ISO 11783-10 version 3, the value for ISO 11783-10 version 2 and prior is FF_{16}.
		Bit 1 = 1 =	Supports documentation. NOTE: task totals need to be handled at any level. TC-BAS functionality. The details of the TC functionalities are listed in Annex F.
		Bit 2 = 1 =	Supports TC-GEO without position-based control. This definition is defined for position based data logging.

[a] A TC client is required to adjust the number of booms, sections, and position-based control channels down to the requirements specified in the TC-SC and TC-GEO functionalities (see Annex F). The values reported by the TC client in the version message shall not be modified to the adjusted number of booms, sections, and position-based control channels when connected to a TC server but describe the capabilities of the TC client.

	Bit 3 = 1 =	Supports TC-GEO with position based control.
	Bit 4 = 1 =	Supports peer control assignment.
	Bit 5 = 1 =	Supports implement section control, TC-SC functionality.
	Bits 6 to 8 = 0 =	Reserved for ISO assignment

Byte 5: Provided options byte 2

Bits 1 to 8 = 0 = Reserved for ISO assignment. The definition of this byte is introduced in ISO 11783-10 version 3, the value for ISO 11783-10 version 2 and prior is FF_{16}.

Byte 6: Number of booms for section control. When reported by the TC, this is the maximum number of section control booms that is supported.[a] When reported by the client, this is the number of section control booms that the client is capable to control.[a] A section control boom is defined as a Device Element that references the Section Control State DDI in one of its Device Process Data Objects (see F.3.5.3). The definition of this byte is introduced in ISO 11783-10 version 3, the value for ISO 11783-10 version 2 and prior is FF_{16}.

Byte 7: Number of sections for section control. This is the total number of sections across all booms. When reported by the TC, this is the maximum number of sections that is supported.[a] When reported by the client, this is the number of sections that the client is capable to control.[a] The definition of this byte is introduced in ISO 11783-10 version 3, the value for ISO 11783-10 version 2 and prior is FF_{16}.

Byte 8: Number of control channels for position based control. When reported by the TC, this is the maximum number of individual control channels that is supported.[a] When reported by the client, this is the number of individual control channels that the client is capable to control.[a] A position-based control channel is defined as a Device Element that references the Prescription Control State DDI in one of its Device Process Data Objects (see F.3.4.4). The definition of this byte is introduced in ISO 11783-10 version 3, the value for ISO 11783-10 version 2 and prior is FF_{16}

[a] A TC client is required to adjust the number of booms, sections, and position-based control channels down to the requirements specified in the TC-SC and TC-GEO functionalities (see Annex F). The values reported by the TC client in the version message shall not be modified to the adjusted number of booms, sections, and position-based control channels when connected to a TC server but describe the capabilities of the TC client.

B.5.4 Request Identify Task Controller message

The Request Identify Task Controller message may be sent by either clients or TCs. Upon receipt of this message, the TC shall display, for a period of 3 s, the TC Number (See Clause 3). This message is intended to be sent Destination-Global; however, it may be sent Destination-Specific.

The presentation of the TC Number is considered TC proprietary and the TC designer may choose to present other information indicating the purpose of the TC Number (see 3.18).

The TC Number shall be in the range of 1 to 32, corresponding to Function Instances 0 to 31. The TC may then be referenced as TC Number 1, TC Number 2, etc.

TC Number = TC Function Instance + 1

NOTE 1 The offset of 1 is in support of operators, which may not be familiar with a zero-based numbering system.

NOTE 2 This message is valid for a DL CF.

NOTE 3 This message is available in ISO 11783-10 version 4 and later.

Transmission repetition rate:			On request
Data length:			8 bytes
Parameter group number:			Process data, destination-specific
Byte 1:	Bits 4 to 1	0000	Command technical data
	Bits 8 to 5	0010	Identify Task Controller
Bytes 2 to 8:			Reserved, transmit as FF_{16}

B.5.5 Identify Task Controller response message

The TC uses this message to respond to the Identify Task Controller message if it was received Destination-Specific.

NOTE 1 A TC without a GUI may exist and, in that case, the instance number might not be able to be displayed.

NOTE 2 This message is valid for a DL CF.

NOTE 3 This message is available in ISO 11783-10 version 4 and later.

Transmission repetition rate:			On request
Data length:			8 bytes
Parameter group number:			Process data, destination-specific
Byte 1:	Bits 4 to 1	0000	Command technical data
	Bits 8 to 5	0010	Identify Task Controller response
Bytes 2 to 8:			Reserved, transmit as FF_{16}

B.6 Device descriptor messages

B.6.1 General

The device descriptor messages are used to transfer the device descriptor from client to TC or DL and to maintain the device descriptor object pool.

B.6.2 Request Structure Label message

The Request Structure Label message allows the client to determine the availability of the requested device descriptor structure at the TC or DL. If the requested structure label is present, a structure label message with the requested structure label shall be transmitted by the TC or DL to the sender of the Request Structure Label message. Otherwise, a structure label message with 7 structure label bytes set to value = FF_{16} shall be transmitted by the TC or DL to the sender of the Request Structure Label message.

If bytes 2 to 8 are transmitted as FF_{16}, the Request Structure Label message allows the client to determine the version of the latest device descriptor structure present at the TC or DL. If a structure label is not present, a structure label message with 7 structure label bytes set to value = FF_{16} shall be transmitted by the TC or DL to the sender of the Request Structure Label message. This method is backwards compatible with ISO 11783-10 version 3 and prior.

Transmission repetition rate:	On request

Data length:			Variable (8 to 40 bytes)
Parameter group number:			Process data, destination-specific
Byte 1:	Bits 4 to 1	0001	Device descriptor
	Bits 8 to 5	0000	Request Structure Label
Bytes 2 to n:			Device Descriptor Structure Label and if needed the Extended Structure Label. Byte 2 is data array byte 1 of the Device Descriptor Structure Label and Byte 8 is data array byte 7 of the Device Descriptor Structure Label. If the existence of a specific Extended Structure Label is requested, then byte 9 is byte 1 of the Extended Structure Label and Byte n is the last data array byte of the Extended Structure Label. This definition is introduced in ISO 11783-10 version 4, prior versions use exactly 7 bytes with the values set to FF_{16}. If a specific Extended Structure Label is requested, transport protocol shall be used.

B.6.3 Structure Label message

The Structure Label message is sent by the TC or DL to inform the client about the availability of the requested version of the device descriptor structure at the TC or DL.

Transmission repetition rate:			In response to Request Structure Label message
Data length:			Variable (8 to 40 bytes)
Parameter group number:			Process data, destination-specific
Byte 1:	Bits 4 to 1	0001	Device descriptor
	Bits 8 to 5	0001	Structure label
Bytes 2 to n:			Device Descriptor Structure Label and if needed the Extended Structure Label. Byte 2 is the Device Descriptor Structure Label data array byte 1 and Byte 8 is the Device Descriptor Structure Label data array byte 7. If an Extended Structure Label is available, then byte 9 is byte 1 of the Extended Structure Label and Byte n is the last Extended Structure Label data array byte. Transmit a single CAN frame 8 byte message with 7 bytes with value = FF_{16} when the requested structure label is not present. If the Extended Structure Label is used, transport protocol shall be used.

B.6.4 Request Localization Label message

The Request Localization Label message allows the client to determine the availability of the requested device descriptor localization at the TC or DL. If the requested localization label is present, a localization label message with the requested localization label shall be transmitted by the TC or DL to the sender of the Request Localization Label message. Otherwise, a localization label message with all localization label bytes set to value = FF_{16} shall be transmitted by the TC or DL to the sender of the Request Localization Label message.

If bytes 2-8 are transmitted as FF_{16}, the Request Localization Label message allows the client to determine the localization version of the latest device descriptor present at the TC or DL. If a localization label is not present, a localization label message with all localization label bytes set to value = FF_{16} shall be transmitted by the TC or DL to the sender of the Request Localization Label message.

Transmission repetition rate:			On request
Data length:			8 bytes
Parameter group number:			Process data, destination-specific
Byte 1:	Bits 4 to 1	0001	Device descriptor
	Bits 8 to 5	0010	Request Localization Label
Bytes 2 to 8:			7-byte Device Descriptor Localization Label. Byte 2 is data array byte 1 and Byte 8 is data array byte 7. This definition is introduced in ISO 11783-10 version 4, Prior versions use FF_{16}.

B.6.5 Localization Label message

The Localization Label message is sent by the TC or DL to inform the client about the availability of the requested localization version of the device descriptor at the TC or DL.

Transmission repetition rate:			In response to Request Localization Label message
Data length:			8 bytes
Parameter group number:			Process data, destination-specific
Byte 1:	Bits 4 to 1	0001	Device descriptor
	Bits 8 to 5	0011	Localization Label
Bytes 2 to 8:			7 byte device descriptor localization label. Bytes 1 to 6 are defined by the language command PGN (see ISO 11783-7). Byte 7 is reserved and set to FF_{16}. Transmit all bytes with value = FF_{16} when the requested localization label is not present.

B.6.6 Request Object-pool Transfer message

The Request Object-pool Transfer message allows the client to determine whether it is allowed to transfer (part of) the device descriptor object pool to the TC or DL.

Transmission repetition rate:			On request
Data length:			8 bytes
Parameter group number:			Process data, destination-specific
Byte 1:	Bits 4 to 1	0001	Device descriptor
	Bits 8 to 5	0100	Request Object-pool Transfer
Bytes 2 to 5:			Requested transfer data size in bytes
Bytes 6 to 8:			Reserved, transmit as FF_{16}

B.6.7 Request Object-pool Transfer Response message

Transmission repetition rate:			In response to Request Object-pool Transfer message
Data length:			8 bytes
Parameter group number:			Process data, destination-specific
Byte 1:	Bits 4 to 1	0001	Device descriptor
	Bits 8 to 5	0101	Request Object-pool Transfer Response
Byte 2:	Status	0	There may be enough memory available. However, because there is overhead associated with object storage, it is impossible to predict whether there is enough memory available.
Bytes 6 to 8:		1	There is not enough memory available. Do not transmit device descriptor object pool.
Bytes 3 to 8:			Reserved, transmit as FF_{16}

B.6.8 Object-pool Transfer message

The Object-pool Transfer message enables the client to transfer (part of) the device descriptor object pool to the TC or DL. The transfer of the device descriptor object pool can be split up over multiple object-pool transfer messages. When the object-pool transfer is split up over multiple Object-pool Transfer messages, each single object-pool transfer shall contain complete object descriptions.

Transmission repetition rate:			On request
Data length:			Variable
Parameter group number:			Process data, destination-specific
Byte 1:	Bits 4 to 1	0001	Device descriptor
	Bits 8 to 5	0110	Object-pool Transfer
Bytes 2 to n:			Object-pool records

B.6.9 Object-pool Transfer Response message

Transmission repetition rate:			In response to Object-pool Transfer message
Data length:			8 bytes
Parameter group number:			Process data, destination-specific
Byte 1:	Bits 4 to 1	0001	Device descriptor
	Bits 8 to 5	0111	Object-pool Transfer Response
Byte 2:	Error code	0	No error, transfer OK
		1	TC or DL ran out of memory during transfer
		2	Any other error
Bytes 3 to 6:			Received data size in bytes

Bytes 7 to 8:			Reserved, transmit as FF_{16}

B.6.10 Object-pool Activate / Deactivate message

This message is sent by a client to complete its connection procedure to a TC or DL or to disconnect from a TC or DL.

To complete the connection procedure, this message is sent by a client with the value of the command attribute byte set to “Activate”. This indicates that the device descriptor object pool is complete and ready for use. This command is sent after the transfer of the device descriptor object pool, after any object is redefined or added to its device descriptor object pool during operation or when comparison of the requested labels indicates that the version of the device descriptor object pool in the device is the same as the version of the device descriptor object pool available in the TC or DL. In case that no device descriptor object pool was uploaded, the TC or DL shall activate the device descriptor object pool from the latest Structure Label request which was successfully responded to.

To disconnect from a TC or DL, this message is sent with the value of the command attribute byte set to “Deactivate”. This indicates that the client intentionally disconnects from the TC or DL.

The definition of the command attribute byte value for deactivation of the connection to a TC or DL has been introduced in ISO 11783-10 version 4.

Transmission repetition rate:			On request
Data length:			8 bytes
Parameter group number:			Process data, destination-specific
Byte 1:	Bits 4 to 1	0001	Device descriptor
	Bits 8 to 5	1000	Object-pool Activate/Deactivate
Byte 2:			Command Attribute, FF_{16} = Object-pool activation command, 0 = Object-pool deactivation command value.
Bytes 3 to 8:			Reserved, transmit as FF_{16}

B.6.11 Object-pool Activate / Deactivate Response message

This message is sent by the TC or DL to a client to acknowledge the device descriptor Object-pool Activate/Deactivate message.

Upon activation of a device descriptor object pool, when the TC or DL replies with an error of any type, the TC or DL shall delete the device descriptor object pool from the volatile memory. The TC or DL can optionally inform the operator of the reason for deletion. Communication with the client shall not continue in the case of a device descriptor already being present in the data transfer file set from the FMIS in case this device descriptor is not able to be activated by the client.

Device descriptors that contain errors shall not be written to the data transfer file set for transfer from the MICS to the FMIS.

Upon successful deactivation of a device descriptor object pool, the TC or DL shall reply with the Byte 2 set to indicate no errors. The maximum response time is 2 s.

Transmission repetition rate:	In response to Object-pool Activate message
Data length:	8 bytes

[a] The definition of this bit is introduced in ISO 11783-10 version 4.

Parameter group number:			Process data, destination-specific
Byte 1:	Bits 4 to 1	0001	Device descriptor
	Bits 8 to 5	1001	Object-pool Activate / Deactivate Response
Byte 2:			Error codes (0 = no errors)
		Bit 1 = 1 =	There are errors in the device descriptor object pool, refer to Bytes 3 to 7 for additional error information
		Bit 2 = 1 =	TC ran out of memory during activation
		Bit 3 = 1 =	Any other error
		Bit 4 = 1 =	Object Pool not activated. Different DDOP with same structure label exists already in TC[a]
		Bits 5 to 8 =	Reserved, transmitted as 0 (zero)
Bytes 3, 4			Parent object ID of faulty object, transmit as NULL object ID if there are no object-pool errors
Bytes 5, 6			Object ID of faulty object, transmit as NULL object ID if there are no object-pool errors
Byte 7			Object-pool error codes (0 = no errors)
		Bit 1 = 1 =	Method or attribute not supported by the TC or DL
		Bit 2 = 1 =	Unknown object reference (missing object)
		Bit 3 = 1 =	Any other error
		Bit 4 = 1 =	Device descriptor object pool was deleted from volatile memory
Bits 5 to 8 =			Reserved, transmitted as 0 (zero)
Byte 8			Reserved, transmit as FF_{16}

a The definition of this bit is introduced in ISO 11783-10 version 4.

B.6.12 Object-pool Delete message

This is a message to delete the device descriptor object pool for the client that sends this message. The Object-pool Delete message enables a client to delete the entire device descriptor object pool before sending an updated or changed device descriptor object pool with the object-pool transfer message.

Only device descriptor object pools that have been defined by the client may be deleted. Device descriptor object pools received from the FMIS are not affected.

For TC or DL version 3 or prior, the Object-pool Delete message was defined for a client to delete the entire device descriptor object pool before sending an updated or changed device descriptor object pool with the Object-pool Transfer message. Due to possible references of logged data to a device descriptor or the device already being allocated to a task, there are situations that a TC server cannot execute this message succesfully. In these situations, for TC version 3 or prior, it was not clear whether the Object-pool Delete Response message had to be used to respond the object pool delete error to the client or whether the Object-pool Delete message could be rejected with a process data acknowledge message.

In TC or DL version 4 and later, each Object-pool Delete message shall be replied to by transmitting an Object-pool Delete Response message. An error detail attribute has been added to the Object-pool Delete Response message to indicate the reason in case an object-pool cannot be deleted.

Transmission repetition rate:			On request
Data length:			8 bytes
Parameter group number:			Process data, destination-specific
Byte 1:	Bits 4 to 1	0001	Device descriptor
	Bits 8 to 5	1010	Object-pool Delete
Bytes 2 to 8:			Reserved, transmit as FF_{16}

B.6.13 Object-pool Delete Response message

Transmission repetition rate:			In response to Object-pool Delete message
Data length:			8 bytes
Parameter group number:			Process data, destination-specific
Byte 1:	Bits 4 to 1	0001	Device descriptor
	Bits 8 to 5	1011	Object-pool Delete Response
Bytes 2 to 8:			Error code: 0 = no error; 1 = error
Byte 2:			Error detail: [a]
Byte 3:			0 = object-pool is referenced by task data
			1 = server cannot check object-pool references
			FF_{16} = error details are not available
Bytes 4 to 8:			Reserved, transmit as FF_{16}

[a] The definition of this byte is introduced in ISO 11783-10 version 4, in ISO 11783-10 version 3 and earlier; this byte was transmitted as FF_{16}.

B.6.14 Change Designator message

This message is to update the designator of an object.

Transmission repetition rate:			On request
Data length:			Variable
Parameter group number:			Process data, destination-specific
Byte 1:	Bits 4 to 1	0001	Device descriptor
	Bits 8 to 5	1100	Change Designator
Bytes 2, 3:			Object ID

Bytes 4:			Designator length, number of bytes, value range is 0 to 128 to transmit maximum 32 UTF-8 characters.
Bytes 5 to n:			UTF-8 encoded designator, no BOM

B.6.15 Change Designator Response message

Transmission repetition rate:			In response to Change Designator message
Data length:			8 bytes
Parameter group number:			Process data, destination-specific
Byte 1:	Bits 4 to 1	0001	Device descriptor
	Bits 8 to 5	1101	Change Designator Response
Bytes 2, 3:			Object ID
Bytes 4:			Error code: 0 = no error; 1 = error
Bytes 5 to 8:			Reserved, transmit as FF_{16}

B.7 Process Data Acknowledge (PDACK) message

This message is sent by the client Master or by the TC or DL to acknowledge or reject commands and process data. The reasons are given in the least significant byte of the process data value. When the process data error code reported is not associated with a specific element number or a specific DDI, the values of the element number or DDI shall be set to “not available”. The “not available” value of the element number is FFF_{16} and the “not available” value of the DDI is $FFFF_{16}$.

To avoid unnecessary busload, a Process Data Acknowledge message with the Process Data Error Code = 0 shall not be transmitted in response to the Process Data commands: 0_{16}, 1_{16}, 2_{16}, 3_{16}, 9_{16}, D_{16}, E_{16}, and F_{16},

Transmission repetition rate:			On request
Data length:			8 bytes
Parameter group number:			Process data, destination-specific
Byte 1:	Bits 4 to 1	1101	Process Data Acknowledge (PDACK)
	Bits 8 to 5		Element number (LSNibble), send as F_{16} when not applicable to the error code
Byte 2:			Element number (MSB), send as FF_{16} when not applicable to the error code
Bytes 3, 4:			DDI, send as $FFFF_{16}$ when not applicable to the error code
Byte 5:			Process Data Error Codes (0 = no errors)
		Bit 1 = 1 =	Process Data Command not supported

[a] The definition of this bit is introduced in ISO 11783-10 version 4.

[b] The definition of these 4 bits is introduced in ISO 11783-10 version 4.

	Bit 2 = 1 =		Invalid element number
	Bit 3 = 1 =		DDI not supported by element
	Bit 4 = 1 =		Trigger method not supported
	Bit 5 = 1 =		Process data not settable
	Bit 6 = 1 =		Invalid or unsupported interval or threshold
	Bit 7 = 1 =		Process data value does not conform to DDI definition[a]
	Bit 8 = 1 =		Process data value is outside the operational range of this device[a]
Byte 6:	Bits 4 to 1		Process Data Command that this PDACK message is transmitted for, send as F_{16} when not applicable to the error code[b]. Prior to version 4, these bits are transmitted as F_{16}.
	Bits 8 to 5		Reserved, transmit as F_{16}
Bytes 7, 8:			Reserved, transmit as FF_{16}

a The definition of this bit is introduced in ISO 11783-10 version 4.

b The definition of these 4 bits is introduced in ISO 11783-10 version 4.

B.8 Status messages

The status messages allow the client to determine the health of the TC or DL and to monitor the progress of tasks in the TC or DL. They also allow the TC or DL to monitor the health of clients and supported commands, and processing of data by the clients.

B.8.1 Task Controller Status message

This message shall be sent by the TC to indicate the current task status.

Transmission repetition rate:			2 s and on change of any byte in this message. At least 200 ms shall pass between status messages.
Data length:			8 bytes
Parameter group number:			Process data, global destination
Byte 1:	Bits 4 to 1	1110	Task Controller Status
	Bits 8 to 5	1111	Element number, set to not available[a]
Byte 2:		FF_{16}	Element number, set to not available[a]
Bytes 3, 4:		$FFFF_{16}$	DDI, set to not available[a]
Byte 5:			Actual TC / DL status
	Bit 1 = 1 =		Task totals active, a task is started or resumed, task totals may accumulate.

a The not available value for this attribute is introduced in ISO 11783-10 version 2. Clients shall ignore this attribute when processing the Task Controller Status messages.

	On 0 to 1 transition:		Internal Control Function task totals are reset to zero and counting starts
	On 1 to 0 transition:		Internal Control Function task totals are stopped and may be queried by TC or DL through a series of Request Value commands
	Bit 2 = 1 =		TC or DL is busy saving data to non-volatile memory
	Bit 3 = 1 =		TC or DL is busy reading data from non-volatile memory
	Bit 4 = 1 =		TC or DL is busy executing a B.6 command (device descriptor messages)
	Bits 5 to 7 = 0 =		Reserved
	Bit 8 = 1 =		TC or DL is out of memory
Byte 6:			Source address of client for which B.6 command is being executed (valid only if Byte 5, Bit 4 is set, or else transmit value = 0)
Byte 7:			B.6 command (Byte 1) that is being executed (valid only if Byte 5, Bit 4 is set, or else transmit value = 0)
Byte 8:			Reserved
Notes:			

a The not available value for this attribute is introduced in ISO 11783-10 version 2. Clients shall ignore this attribute when processing the Task Controller Status messages.

B.8.2 Client Task message

This message shall be sent by all clients to the TC or DL to indicate the current Client Task status.

Transmission repetition rate:			2 s and on each task status change. At least 200 ms shall pass between Client Task messages.
Data length:			8 bytes
Parameter group number:			Process data, destination-specific
Byte 1:	Bits 4 to 1	1111	Client Task
	Bits 8 to 5	1111	Element number, set to not available[a]
Byte 2:		FF_{16}	Element number, set to not available[a]
Bytes 3, 4:		$FFFF_{16}$	DDI, set to not available[a]
Bytes 5 to 8:	Bit 1		Actual TC or DL status: task totals active (as received in Task Controller Status message, Byte 5, Bit 1)

a The not available value for this attribute is introduced in ISO 11783-10 version 2. TCs or DLs shall ignore this attribute when processing the Task Controller Status messages.

Bits 2 to 32	0	Reserved

Notes:

[a] The not available value for this attribute is introduced in ISO 11783-10 version 2. TCs or DLs shall ignore this attribute when processing the Task Controller Status messages.

Annex C
(normative)

XML elements relationship diagram

C.1 XML element relationship

Figures C.1 and C.2 show the entity relationship diagram for all entities. The contents of an XML file shall be according to the relationships as specified in the diagram. The entities shown in Figure C.1 and Figure C.2 do not always contain foreign key identifiers, since this can be determined by the sequence of definition in the XML file.

The ERD specifies, for example, that the *Task* entity has a one-to-zero-or-more relation to *WorkerAllocation*; the task can have no workers assigned, but also one or more. When one or more workers are assigned, this means that the *WorkerAllocation* definition shall follow the *TaskHeaderData* definition, in order to get its relation to the task. *Worker* has also a direct one-to-zero-or-more relation to *Task*. This means that a *Worker* can be the “Responsible Worker” for zero or more *Tasks*.

The line ends at each relation shall be read as “opposite line end” and determine multiplicity from the originating entity towards the other entity. For example, the relation between *Worker* and *WorkerAllocation* is read as: *Worker* can occur in zero or more *WorkerAllocations* (0 + crow foot) and a *WorkerAllocation* entity is bound to exactly 1 worker (single bar).

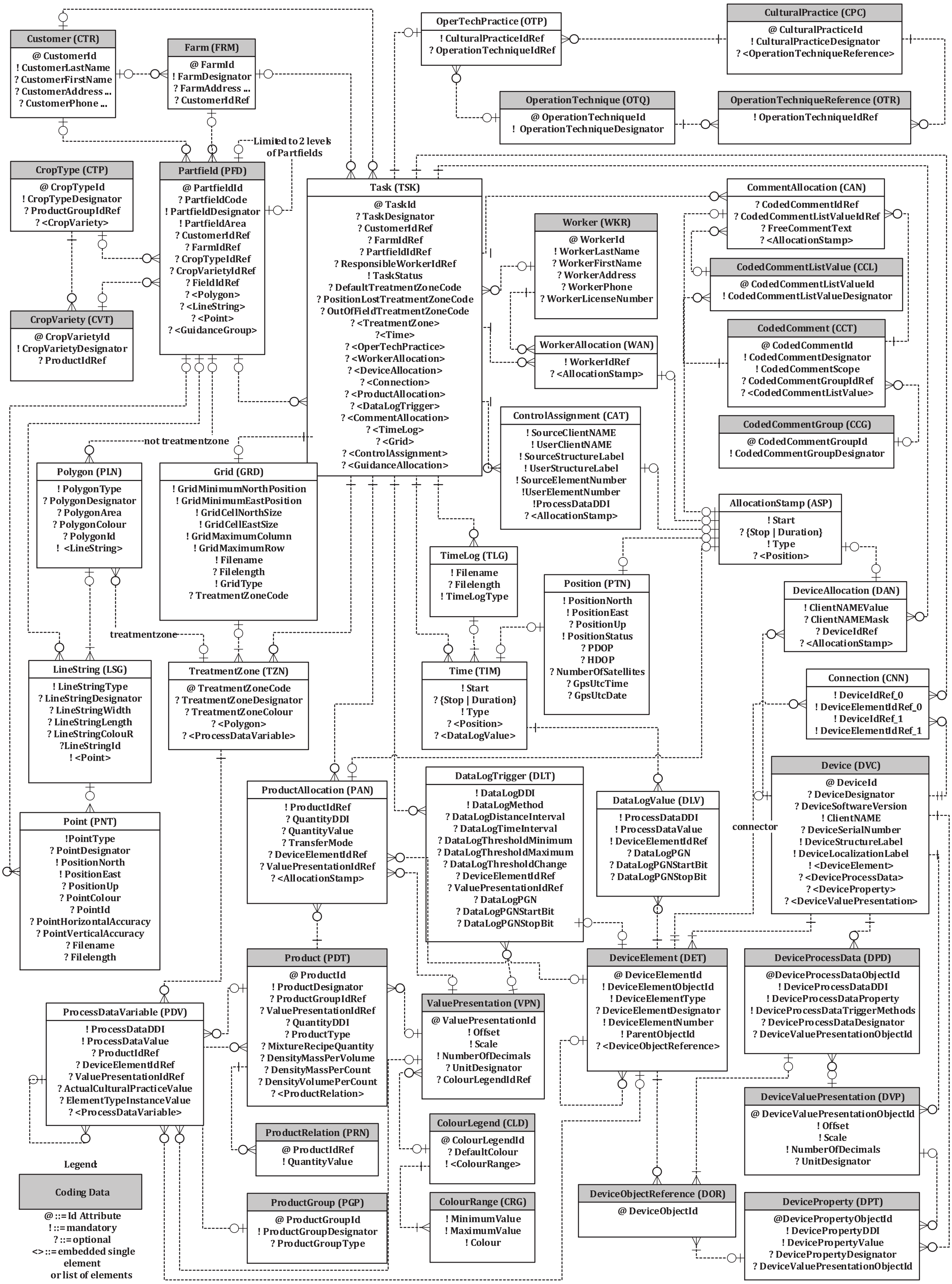


Figure C.1 — XML elements relationship diagram, part 1

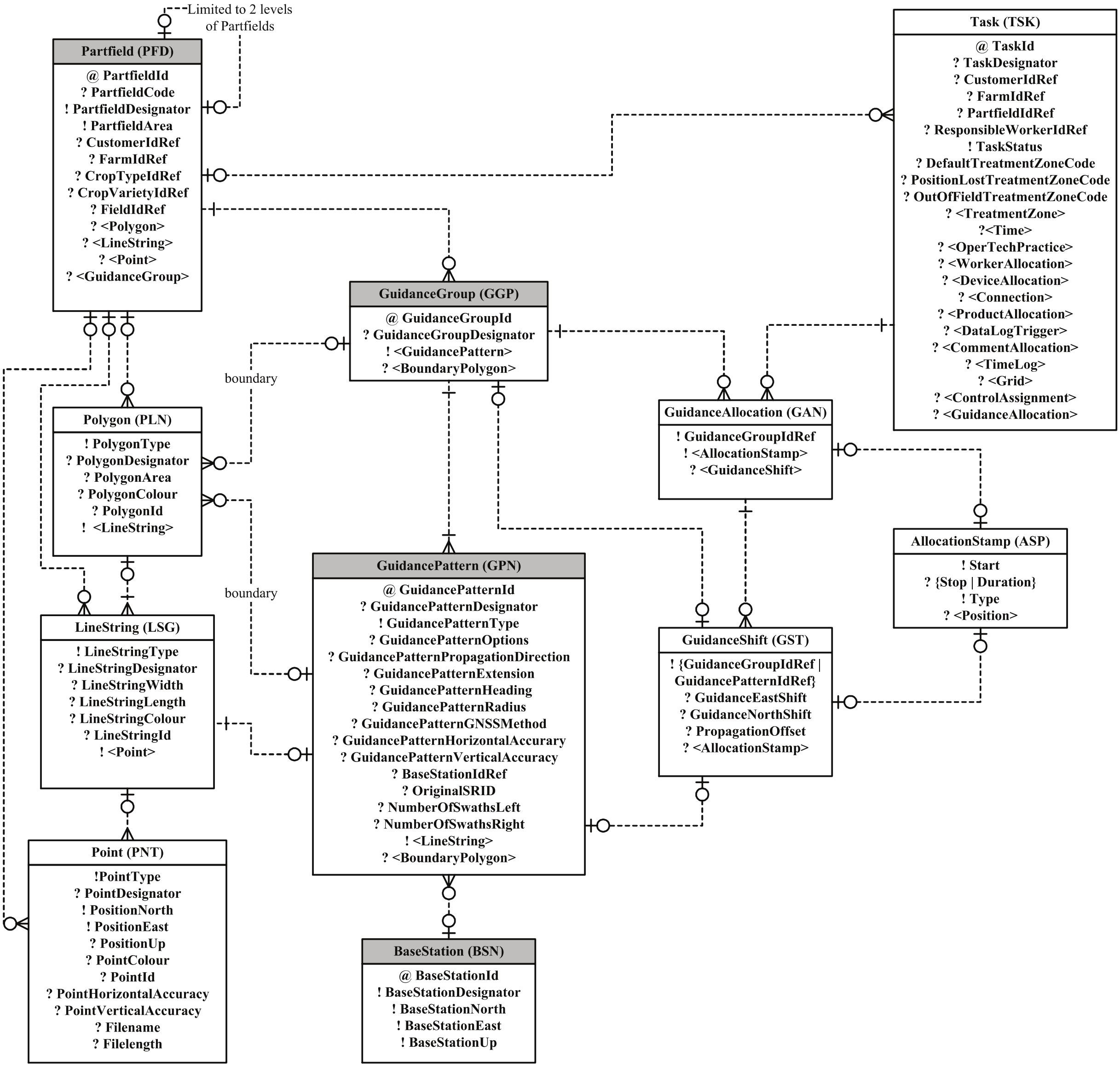


Figure C.2 — XML elements relationship diagram, part 2

Annex D
(normative)

XML elements and attributes

D.1 XML elements

The XML attributes of an XML element are described in this annex in tables with six columns:

— **Attribute**, name of the XML attribute.

— **XML**, XML attribute tag.

— **Use**, “r” denotes required, “o” denotes optional.

— **Type**, XML type specification of this attribute.

— **Length/range**, number of characters or value range of this attribute.

— **Comment**, additional explanation or restrictions on format of this attribute.

NOTE Child XML elements are also listed in the XML attributes.

Table D.1 — Example XML element attributes

Attribute	XML	Use	Type	Length/ range	Comment
Attribute_1	A	r	xs:datetime	Max. 29	example of a required attribute
Attribute_2	B	o	xs:datetime	Max. 29	example of an optional attribute
Element_1		o	xs:element		example of optionally including a single XML element Element_1

The type xsd:dateTime represents a specific date and time in the format CCYY-MM-DDThh:mm:ss.sss, which is a concatenation of the xsd:date and xsd:time forms, separated by a literal letter “T”. All of the same rules that apply to the xsd:date and xsd:time types are applicable to xsd:dateTime.

An optional time zone expression may be added at the end of the value. The letter Z is used to indicate Coordinated Universal Time (UTC). All other time zones are represented by their difference from Coordinated Universal Time in the format +hh:mm or -hh:mm. These values may range from −14:00 to 14:00. For example, US Eastern Standard Time, which is five hours behind UTC, is represented as −05:00. If no time zone value is present, it is considered to be local time; it is not assumed to be UTC.

D.2 AllocationStamp — ASP

Type:

Task data

Description:

The AllocationStamp XML element specifies a recording of an allocation event. Optionally, a position can be recorded at an AllocationStamp specification. All AllocationStamp XML elements that were provided by the FMIS shall be of the type *planned* and all AllocationStamp XML elements that were provided by the MICS shall be of the type *effective*.

All AllocationStamp XML elements shall have a Start attribute value defined. Duration shall always contain a positive value and is equal to the time elapsed between Start and Stop. Thus, a finite AllocationStamp XML element specification can consist of either a Start and a Stop (Duration can be calculated) or a Start and a Duration (Stop time can be calculated). The specification of an infinite AllocationStamp XML element, e.g. only recording the Start is allowed.

In ISO 11783-10 version 4, the datetime data type has been extended to include time zone information or to be explicitly set to UTC. When the time zone is known, it is required to either specify times in UTC by adding the “Z” indicator or include time zone offsets in the start and stop attributes. Using either UTC explicitly or using the time zone information when local time is used by the FMIS to normalize data coming from multiple time zones. In case no time zone information is known, then all times are expressed in local time without time zone information.

Prior to version 4, the time zone can only be calculated from the logged records in the TimeLog binary data file if both the UTC time values and the local time values are recorded by subtracting the local time from the UTC time. In this case, the calculated time zone can be applied to all timestamps in the same data transfer file set.

Included by XML elements:

— *CommentAllocation*

— *DeviceAllocation*

— *ProductAllocation*

— *WorkerAllocation*

Includes XML element:

— *Position*

Table D.2 — AllocationStamp attributes

Attribute	XML	Use	Type	Length/ range	Comment
Start	A	r	xs:datetime	Max. 29	Time of start. Format: yyyy-mm-ddThh:mm:ss.sss plus an optional time zone indication.[a]
Stop	B	o[b]	xs:datetime	Max. 29	Time of end. Format: yyyy-mm-ddThh:mm:ss.sss plus an optional time zone indication.[a]
Duration	C	o[b]	xs:unsignedLong	0 to (2^{32}–2)	Time between start and stop in number of seconds.
Type	D	r	xs:NMTOKEN	1, 4	Type of the AllocationStamp, possible values: 1 = Planned 4 = Effective (Realized)
Position		o	xs:element		Include up to 2 Position XML elements which represent the position at the start and stop attribute values of the AllocationStamp.

[a] The time zone information is introduced in ISO 11783-10 version 4.

[b] Either Stop or Duration need to be specified in addition to Start to create a finite AllocationStamp XML element.

EXAMPLE 1 No time zone information is available, time is recorded in local time.

```
<ASP A=”2003-08-20T08:10:00” D=”4”>
<PTN A=”54.588945” B=”9.989209” D=”3”/>
```

```
</ASP>
<!- timestamp with second precision - >
<ASP A="2003-08-20T08:10:00" C="3512" D="4"/>
    <!- timestamp that includes a sub-second precision - >
<ASP A="2003-08-20T08:10:00.245" C="3512" D="4"/>
```

EXAMPLE 2 Time zone information is available in the system, but time is recorded explicitly in UTC only.

```
<ASP A="2003-08-20T07:10:00Z" C="3512" D="4"/>
```

EXAMPLE 3 Time zone information is available, time is recorded in local time with time zone information, e.g. for Central European Time a start time for the same moment of time as example 2 would be

```
<ASP A="2003-08-20T08:10:00+01:00" C="3512" D="4"/>
```

or for a location in time zone 0, this same moment of time is

```
<ASP A="2003-08-20T07:10:00+00:00" C="3512" D="4"/>
```

D.3 AttachedFile — AFE

Type:

External data

Description:

The AttachedFile XML element specifies a manufacturer proprietary file to be part of the data transfer file set.

The file must reside in the same directory as the TASKDATA.XML. The filename of the attached files should follow the 8.3 naming convention and be all upper case.

If the attached file originates at another manufacturer (for example, a FMIS manufacturer), contents can be ignored by the TC if the attached file is of a non-predefined type. If the Preserve XML attribute has a value of 1, then the TC can omit the AFE XML element from the received data transfer file set and delete the attached file immediately. A value of 2 for the Preserve XML attribute signals that the TC must preserve this XML element and the file and include these in the data transfer file set back to the FMIS.

To prevent unnecessary data transfers, attribute B (Preserve) shall be set to 1 if there is no need for the Task Controller to transfer the attached file back to the FMIS. An updated Task Controller language file could be an example of this use case.

Included by XML elements:

— *ISO11783_TaskData*

Table D.3 — AttachedFile attributes

Attribute	XML	Use	Type	Length/ range	Comment
FilenameWithExtension	A	r	xs:ID	12	Unique name of the attached file. For predefined file types, the file name may be fixed. Format is a 8.3 filename consisting of uppercase characters in the range A-Z and 0–9 (e.g. "PROPDATA.BIN").
Preserve	B	r	xs:NMTOKEN	1 to 2	1 = Task Controlller does not need to preserve attached file. 2 = Preserve on TC and send back to FMIS.

Table D.3 *(continued)*

Attribute	XML	Use	Type	Length/ range	Comment
ManufacturerGLN	C	r	xs:anyURI	max. 32	Empty for attachments predefined by ISO 11783. For other attachments, this shall contain the manufacturer's GS1 GLN (Global Location Number).
FileType	D	r	xs:unsignedByte	1 to 254	Predefined types (ManufacturerGLN is empty): 1 = LINKLIST (fixed filename 'LINKLIST.XML', all uppercase). Note: a maximum of one LINKLIST.XML file can be referenced in the TASKDATA.XML. The range of 1 to 127 is reserved for ISO predefined file types. The range of 128 to 254 is for manufacturer proprietary file types. For a nonempty ManufacturerGLN, it is up to the manufacturer to set an appropriate value.
FileVersion	E	o	xs:string	max. 32	File version number to create a unique relation between the TASKDATA.XML and the attached file.
FileLength	F	o	xs:unsignedLong	0 to (2^{32}-2)	Length of attached file in number of bytes.

EXAMPLE

```
<AFE A="LINKLIST.XML" B="1" C="" D="1" E="3.2" F="6400"/>
<AFE A="CLAASEXT.DAT" B="1" C="urn:epc:id:sgln:4250285.50000.9" D="1"/>
<AFE A="PRESERVE.DAT" B="2" C="urn:epc:id:sgln:0614141.33254.1" D="7"/>
```

D.4 BaseStation — BSN

Type:

Coding data

Description:

The BaseStation XML element describes a positioning system base station.

Note: this XML element is introduced in ISO 11783-10 version 4.

Referenced by XML element:

— *GuidancePattern*

Included by XML element:

— *ISO11783_TaskData*

Table D.4 — BaseStation attributes

Attribute	XML	Use	Type	Length/range	Comment
BaseStationId	A	r	xs:ID	min. 4 to max. 14	Unique identifier of BaseStation Format: (BSN|BSN-)([0–9])+ Records generated on MICS have negative IDs

Table D.4 *(continued)*

Attribute	XML	Use	Type	Length/range	Comment
BaseStationDesignator	B	r	xs:string	max. 32	Designator of BaseStation
BaseStationNorth	C	r	xs:decimal	−90.0 to 90.0	BaseStation position north WGS84
BaseStationEast	D	r	xs:decimal	−180.0 to 180.0	BaseStation position east WGS84
BaseStationUp	E	r	xs:long	$(-2^{31}+1)$ to $(2^{31}-1)$	BaseStation position altitude WGS84 ellipsoid in mm

EXAMPLE

```
<BSN A="BSN4" B="RTK Reference 1" C="54.588945" D="9.989209" E="523867"/>
```

D.5 CodedComment — CCT

Type:

Coding data

Description:

The CodedComment XML element describes predefined comments that can be used to annotate tasks. Comments can be grouped in CodedCommentGroups by establishing a reference to that group in CodedCommentGroupIdRef. A coded comment can include a list of possible values (e.g. low, medium, high). These values are described in CodedCommentValues. One of these values can be referenced when a comment is assigned to a task.

Referenced by XML element:

— *CommentAllocation*

Includes XML element:

— *CodedCommentListValue*

Table D.5 — CodedComment attributes

Attribute	XML	Use	Type	Length/range	Comment
CodedCommentId	A	r	xs:ID	min. 4 to max. 14	Unique identifier of CodedComment Format: (CCT|CCT-)([0–9])+ Records generated on MICS have negative IDs
CodedCommentDesignator	B	r	xs:string	max. 32	Designator of CodedComment
CodedCommentScope	C	r	xs:NMTOKEN	1 to 3	Selection of one attribute: 1 = point 2 = global 3 = continuous
CodedCommentGroupIdRef	D	o	xs:IDREF	min. 4 to max. 14	Reference to ID of CodedCommentGroup Format: (CCG|CCG-)([0–9])+
CodedCommentListValue		o	xs:element		Includes a list of XML element CodedCommentListValue

EXAMPLE 1 Point Scope CodedComments

```
<CCT A="CCT1" B="Refuel" C="1"/>
<CCT A="CCT2" B="Field operation interrupted" C="1"/>
```

EXAMPLE 2 Global Scope CodedComments with CodedCommentListValues

```
<CCT A="CCT3" B="Weeds" C="2">
   <CCL A="CCL12" B="no weeds"/>
   <CCL A="CCL13" B="some small weeds"/>
   <CCL A="CCL14" B="some large weeds"/>
   <CCL A="CCL15" B="numerous small weeds"/>
   <CCL A="CCL16" B="numerous large weeds"/>
</CCT>
<CCT A="CCT4" B="Weather" C="2">
   <CCL A="CCL17" B="Drizzle"/>
   <CCL A="CCL18" B="Humid"/>
   <CCL A="CCL19" B="Frost, Firm Ground"/>
   <CCL A="CCL20" B="Rain"/>
   <CCL A="CCL21" B="No Wind"/>
   <CCL A="CCL22" B="Slight Frost"/>
   <CCL A="CCL23" B="Light Wind"/>
   <CCL A="CCL24" B="Shower"/>
   <CCL A="CCL25" B="Very Hot"/>
   <CCL A="CCL26" B="Very Windy"/>
   <CCL A="CCL27" B="Dry"/>
</CCT>
<CCT A="CCT5" B="Soil Condition" C="2">
   <CCL A="CCL28" B="damp"/>
   <CCL A="CCL29" B="dry on surface, wet below"/>
   <CCL A="CCL30" B="wet on surface, dry below"/>
   <CCL A="CCL31" B="seed bed coarse"/>
   <CCL A="CCL32" B="seed bed good"/>
   <CCL A="CCL33" B="seed bed finely granulated"/>
   <CCL A="CCL34" B="very porous"/>
   <CCL A="CCL35" B="very wet"/>
   <CCL A="CCL36" B="dry"/>
</CCT>
```

EXAMPLE 3 Continuous Scope CodedComments with CodedCommentListValues

```
<CCT A="CCT7" B="Thistles" C="3">
   <CCL A="CCL1" B="10 P./sqm"/>
   <CCL A="CCL2" B="20 P./sqm"/>
   <CCL A="CCL3" B="30 P./sqm"/>
</CCT>
```

D.6 CodedCommentGroup — CCG

Type:

Coding data

Description:

The CodedCommentGroup XML element can be used to combine predefined CodedComments into groups. The purpose is to have a better navigation and selection of CodedComments on the mobile system. Each CodedComment can belong to only one CodedCommentGroup. A CodedCommentGroup can, for instance, be "weeds", which contains the CodedComments "camomile", "couch grass", and "thistle". To set up a list of CodedComments that belong to a particular CodedCommentGroup on the MICS, all CodedComment XML elements shall be examined for a match between XML attributes CodedCommentGroupIdRef and the identifier of that CodedCommentGroup.

Referenced by XML element:

— *CodedComment*

Table D.6 — CodedCommentGroup attributes

Attribute	XML	Use	Type	Length/range	Comment
CodedCommentGroupId	A	r	xs:ID	min. 4 to max. 14	Unique identifier of CodedCommentGroup Format: (CCG|CCG-)([0–9])+ Records generated on MICS have negative IDs
CodedCommentGroupDesignator	B	r	xs:string	max. 32	Designator of CodedCommentGroup

EXAMPLE

```
<CCG A="CCG1" B="Weeds" />
   <CCT A="CCT6" B="Thistles" C="3" D="CCG1" >
   <CCL A="CCL37" B="10 P./sqm"/>
   <CCL A="CCL38" B="20 P./sqm"/>
   <CCL A="CCL39" B="30 P./sqm"/>
</CCT>
```

D.7 CodedCommentListValue — CCL

Type:

Coding data

Description:

The CodedCommentListValue XML element provides a value to qualify a coded comment. Each CodedCommentListValue always belongs to a single CodedComment only. One of the CodedCommentListValues that belongs to a CodedComment can be referenced in CommentAllocation when that CodedComment is assigned to a task. Examples of CodedCommentListValues are sets like "low", "medium", and "high" or definitions such as "crop growth stage codes".

Referenced by XML element:

— *CommentAllocation*

Included by XML element:

— *CodedComment*

Table D.7 — CodedCommentListValue attributes

Attribute	XML	Use	Type	Length/range	Comment
CodedCommentListValueId	A	r	xs:ID	min. 4 to max. 14	Unique identifier of CodedCommentListValue Format: (CCL|CCL-)([0–9])+ Records generated on MICS have negative IDs
CodedCommentListValueDesignator	B	r	xs:string	max. 32	Designator of CodedCommentListValue

EXAMPLE

```
<CCL A="CCL37" B="10 P./sqm"/>
<CCL A="CCL38" B="20 P./sqm"/>
<CCL A="CCL39" B="30 P./sqm"/>
```

D.8 ColourLegend — CLD

Type:

Coding data

Description:

The ColourLegend XML element describes a colour legend that can be used to display values in different colours on a grid map. A ColourLegend XML element includes ColourRange XML elements which specify the colour to be used when presenting the value. The attribute DefaultColour is used to present the value when no ColourRange within which the value falls is specified.

Referenced by XML element:

— *ValuePresentation*

Includes XML element:

— *ColourRange*

Table D.8 — ColourLegend attributes

Attribute	XML	Use	Type	Length/range	Comment
ColourLegendId	A	r	xs:ID	min. 4 to max. 14	Unique identifier of ColourLegend Format: (CLD|CLD-)([0–9])+ Records generated on MICS have negative IDs
DefaultColour	B	o	xs:unsignedByte	0 to 254	Default colour of the ColourLegend Format: palette like ISO 11783-6
ColourRange		r	xs:element		Includes a list of XML element ColourRange

EXAMPLE

```
<CLD A="CLD1" B="0">
   <CRG A="0" B="9999" C="1"/>
   <CRG A="10000" B="14999" C="2"/>
   <CRG A="15000" B="19999" C="3"/>
   <CRG A="20000" B="29999" C="4"/>
   <CRG A="30000" B="99999" C="5"/>
</CLD>
```

D.9 ColourRange — CRG

Type:

Coding data

Description:

The ColourRange XML element specifies the colour to use to display values within a certain value range on a grid map. A ColourRange XML element is included in a ColourLegend XML element to specify a range of colours to be used to present different ranges of values.

Included by XML element:

— *ColourLegend*

Table D.9 — ColourRange attributes

Attribute	XML	Use	Type	Length/range	Comment
MinimumValue	A	r	xs:long	$(-2^{31}+1)$ to $(2^{31}-1)$	ColourRange minimum value. The value is included in the range.
MaximumValue	B	r	xs:long	$(-2^{31}+1)$ to $(2^{31}-1)$	ColourRange maximum value. The value is included in the range.
Colour	C	r	xs:unsigned-Byte	0 to 254	ColourRange colour Format: palette according to ISO 11783-6 colour definition

EXAMPLE

```
<CLD A="CLD1">
   <CRG A="0" B="9999" C="1"/>
   <CRG A="10000" B="14999" C="2"/>
   <CRG A="15000" B="19999" C="3"/>
   <CRG A="20000" B="29999" C="4"/>
   <CRG A="30000" B="99999" C="5"/>
</CLD>
```

D.10 CommentAllocation — CAN

Type:

Task data

Description:

The CommentAllocation XML element allocates a CodedComment or a free comment text to a task. The allocation of comments is specified at a certain position and time by inclusion of XML element AllocationStamp. Free comment can be added in XML attribute FreeCommentText. In case of assigning a CodedComment and a CodedCommentListValue to a task, the CodedCommentListValue shall be out of the list of values of the assigned CodedComment. A single CommentAllocation can assign either a CodedComment or contain a FreeCommentText exclusively.

Only a single CommentAllocation shall be recorded for each allocation event of a CodedComment in a Task. For the CodedComment type “continuous”, this means that the included Position XML element shall contain the position at the start time of the recording of the continuous comment. When the continuous CodedComment finishes, the CommentAllocation is updated to contain both the start and stop time attributes in its AllocationTimeStamp.

When a CodedComment of type “continuous” is activated, the TC shall check whether datalogging of time and position in binary log files is active. If data logging of time and position in binary log files is not active, then the TC shall activate it with a 1 Hz update rate for storing time and position in the binary log files. If no CodedComments of type “continous” are active and no other request for data logging is present, the TC shall stop data logging of time and position in binary log files. This ensures that the positions are logged as long as a CodedComment of type “continuous” is active.

Includes XML element:

— *AllocationStamp*

Included by XML element:

— *Task*

References XML elements:

— *CodedComment*

— *CodedCommentListValue*

Table D.10 — CommentAllocation attributes

Attribute	XML	Use	Type	Length/range	Comment
CodedCommentIdRef	A	o	xs:IDREF	min. 4 to max. 14	Reference to XML element Coded-Comment Format: (CCT|CCT-)([0–9])+
CodedCommentListValue-IdRef	B	o	xs:IDREF	min. 4 to max. 14	Reference to XML element Coded-CommentListValue Format: (CCL|CCL-)([0–9])+
FreeCommentText	C	o	xs:string	max. 32	Operator-definable free comment text
AllocationStamp		o	xs:element		Includes a single XML element Allo-cationStamp to specify the time and the position of the comment.

EXAMPLE 1 Operator defined free comment text CommentAllocation

```
<CAN C=”bad driving conditions”>
   <ASP A=”2003-08-20T08:00:20” D=”4”>
      <PTN A=”51.23456” B=”13.23456” D=”2”/>
   </ASP>
</CAN>
```

EXAMPLE 2 Referenced global type CodedComment and CodedCommentListValue

```
<CAN A=”CCT3” B=”CCL13”>
   <ASP A=”2003-08-20T08:00:20” D=”4”>
      <PTN A=”51.23456” B=”13.23456” D=”2”/>
   </ASP>
</CAN>
```

EXAMPLE 3 Referenced continuous type CodedComment and CodedCommentListValue

A continuous CodedComment is started:

```
<CAN A=”CCT6” B=”CCL38”>
   <ASP A=”2003-08-20T08:00:20” D=”4”>
   <!–- A = start of continuous allocation, comment is active ->
      <PTN A=”51.23456” B=”13.23456” D=”2”/>
      <!—- position at the start of the continuous comment allocation ->
   </ASP>
</CAN>
```

and about 35 min later, the same continuous CodedComment is stopped:

```
<CAN A=”CCT6” B=”CCL38”>
   <ASP A=”2003-08-20T08:00:20” B=”2003-08-20T08:35:45” D=”4”>
   <!—- A = start of continuous comment allocation ->
   <!—- B = end of continuous comment allocation, comment is complete ->
      <PTN A=”51.23456” B=”13.23456” D=”2”/>
      <!—- position at the start of the continuous comment allocation ->
   </ASP>
</CAN>
```

D.11 Connection — CNN

Type:

Task data

Description:

The purpose of the Connection XML element is to specify how two devices are connected to each other within a single task. A connection specification consists of references to the two DeviceElements of type “connector” of the two devices which are connected. The connection specification enables the TC to determine the position of DeviceElements of one device relative to the position of, for instance, the NavigationReferencePoint of another device. The referenced DeviceElement in DeviceElementIdRef_0 shall be part of the device referenced in DeviceIdRef_0 and shall be of the type “connector”. The referenced DeviceElement in DeviceElementIdRef_1 shall be part of the device referenced in DeviceIdRef_1 and shall also be of the type “connector”.

Included by XML element:

— *Task*

References XML elements:

— *Device*

— *DeviceElement*

Table D.11 — Connection attributes

Attribute	XML	Use	Type	Length/range	Comment
DeviceIdRef_0	A	r	xs:IDREF	min. 4 to max. 14	Reference to XML element Device Format: (DVC|DVC-)([0–9])+
DeviceElementIdRef_0	B	r	xs:IDREF	min. 4 to max. 14	Reference to XML element DeviceElement Format: (DET|DET-)([0–9])+
DeviceIdRef_1	C	r	xs:IDREF	min. 4 to max. 14	Reference to XML element Device Format: (DVC|DVC-)([0–9])+
DeviceElementIdRef_1	D	r	xs:IDREF	min. 4 to max. 14	Reference to XML element DeviceElement Format: (DET|DET-)([0–9])+

EXAMPLE

```
<CNN A=”DVC2” B=”DET2” C=”DVC1” D=”DET1”/>
```

D.12 ControlAssignment — CAT

Type:

Task data

Description:

The ControlAssignment XML element describes an assignment of a setpoint value source CF to a setpoint value user CF. An example would be a process data value of an element of a sensor system and a process data value of an element of an application controller device. This element is used to record the assignment that is made during execution of task and enables making the same assignment without further operator interaction when a task is restarted. It may be used by the FMIS as a planned assignment which the TC uses after starting of a task. Multiple ControlAssignment XML elements with the same control assignment CFs and process data DDI may exist in a task, e.g. to record control assignment time periods. In this case, the control assignment with the latest timestamp shall be used for re-establishing the assignmennt in a later task start.

The ControlAssignment XML element is added in ISO 11783-10 version 4.

Included by XML element:

— *Task*

Includes XML element:

— *AllocationStamp*

Table D.12 — ControlAssignment attributes

Attribute	XML	Use	Type	Length/ Range	Comment
SourceClientNAME	A	r	xs:hexBinary	0000000000000000$_{16}$ to FFFFFFFFFFFFFFFF$_{16}$	NAME of the setpoint value source CF (see ISO 11783-5)
UserClientNAME	B	r	xs:hexBinary	0000000000000000$_{16}$ to FFFFFFFFFFFFFFFF$_{16}$	NAME of the setpoint value user CF (see ISO 11783-5)
SourceDeviceStructureLabel	C	r	xs:hexBinary	00$_{16}$ to FE$_{16}$ per byte for the most significant 7 bytes, 00$_{16}$ to FF$_{16}$ per byte for the extended structure label bytes. The maximum number of bytes of the structure label extension is 32.	Structure Label of device descriptor of the setpoint value source Array Byte 1 is most significant and Array Byte n is least significant.
UserDeviceStructureLabel	D	r	xs:hexBinary	00$_{16}$ to FE$_{16}$ per byte for the most significant 7 bytes, 00$_{16}$ to FF$_{16}$ per byte for the extended structure label bytes. The maximum number of bytes of the structure label extension is 32.	Structure Label of device descriptor of the setpoint value user Array Byte 1 is most significant and Array Byte n is least significant.

Table D.12 *(continued)*

Attribute	XML	Use	Type	Length/ Range	Comment
SourceDeviceElementNumber	E	r	xs:unsignedShort	0 to 4095	Unique number of the device element of the setpoint value source, refer to Annex B: ProcessDataVariable element numbering
UserDeviceElementNumber	F	r	xs:unsignedShort	0 to 4095	Unique number of the device element of the setpoint value user, refer to Annex B: ProcessDataVariable element numbering
ProcessDataDDI	G	r	xs:hexBinary	0000_{16} to $FFFF_{16}$	Unique number which defines the process data DDI (specified in ISO 11783-11) for both the setpoint value source and the setpoint value user
AllocationStamp		o	xs:element		Includes a single XML element AllocationStamp

EXAMPLE 1 ControlAssignment definition

```
<CAT A="A02282000CE03039" B="A00A84000B8131A3" C="01020304050607" D="07060504030201"
E="0" F="0" G="0006">
    <ASP A="2003-08-20T08:00:00" B="2003-08-20T17:00:00" D="4"/>
</CAT>
```

EXAMPLE 2 Device descriptor for a Setpoint Value Source

The following example shows only the basic elements for a setpoint value source. Other elements are omitted for example purposes. The primary item shown is the DPD for the setpoint application rate, attribute C has value 5 which is a combination of 4 for "control source" and 1 for "belongs to default set".

```
<DVC A="DVC1" B="RT200ISO" C="0.5.10" D="A02282000CE03039" E="2" F="01020304050607"
G="FF565A01506E65">
    <DET A="DET1" B="65523" C="1" D="NDVISensors" E="0" F="0">
        <DOR A="65524" />
    </DET>
    <DPD A="65524" B="0006" C="5" D="1" E="SetPointRate"/>
</DVC>
```

D.13 CropType — CTP

Type:

Coding data

Description:

The CropType XML element describes a crop that can be cultivated on a partfield. A CropType XML element can include several CropVariety XML elements.

The optional attribute ProductGroupIdRef has been introduced in version 4 to include a reference to a ProductGroup of type "CropType". This reference enables cross referencing the use of a commodity with the CropType attribute of a Partfield.

Referenced by XML elements:

— *Partfield*

Includes XML elements:

— *CropVariety*

Table D.13 — CropType attributes

Attribute	XML	Use	Type	Length/range	Comment
CropTypeId	A	r	xs:ID	min. 4 to max. 14	Unique identifier of CropType Format: (CTP|CTP-)([0–9])+ Records generated on MICS have negative IDs
CropTypeDesignator	B	r	xs:string	max. 32	Name of the crop
ProductGroupIdRef [a]	C	o	xs:IDREF	min. 4 to max. 14	Reference to XML element ProductGroup Format: (PGP|PGP-)([0–9])+
CropVariety		o	xs:element		Includes a list of XML element CropVariety

[a] This attribute is introduced in ISO 11783-10 version 4.

EXAMPLE 1 CropType and CropVariety definition

```
<CTP A="CTP1" B="wheat">
   <CVT A="CVT1" B="Ritmo B"/>
   <CVT A="CVT2" B="Dekan B"/>
   <CVT A="CVT3" B="Ares C"/>
</CTP>
<CTP A="CTP2" B="barley"/>
<CTP A="CTP3" B="oats"/>
```

EXAMPLE 2 CropType definition linked to a ProductGroup definition

```
<PGP A="PGP1" B="potato" C="2"/>
<CTP A="CTP4" B="potato" C="PGP1"/>
```

D.14 CropVariety — CVT

Type:

Coding data

Description:

The CropVariety XML element describes varieties of a crop specified by CropType that can be cultivated on a partfield. Each CropVariety definition belongs to a single CropType definition.

The optional attribute ProductIdRef has been introduced in version 4 to include a reference to a Product that is grouped within a ProductGroup of type "CropType". This reference enables cross referencing the use of a commodity with the CropVariety attribute of a Partfield.

Included by XML elements:

— *CropType*

Referenced by XML elements:

— *Partfield*

Table D.14 — CropVariety attributes

Attribute	XML	Use	Type	Length	Comment
CropVarietyId	A	r	xs:ID	min. 4 to max. 14	Unique identifier of CropVariety Format: (CVT|CVT-)([0–9])+ Records generated on MICS have negative IDs
CropVarietyDesignator	B	r	xs:string	max. 32	Name of the CropVariety
ProductIdRef [a]	C	o	xs:IDREF	min. 4 to max. 14	Reference to XML element Product Format: (PDT|PDT-)([0–9])+

[a] This attribute is introduced in ISO 11783-10 version 4.

EXAMPLE 1 CropVariety definition

```
<CVT A=”CVT1” B=”Ritmo B”/>
<CVT A=”CVT2” B=”Dekan B”/>
<CVT A=”CVT3” B=”Ares C”/>
```

EXAMPLE 2 CropVariety linked to a Product definition

```
<PGP A=”PGP1” B=”Wheat” C=”2”/>
<PDT A=”PDT1” B=”Ritmo B” C=”PGP1”/>
<CTP A=”CTP1” B=”Wheat” C=”PGP1” />
   <CVT A=”CVT1” B=”Ritmo B” C=”PDT1”/>
</CTP>
```

D.15 CulturalPractice — CPC

Type:

Coding data

Description:

A cultural practice describes one or a series of activities to realize an objective in crop production. The CulturalPractice XML element describes cultural practices that through the OperTechPractice XML element can be allocated to a task. Examples of CulturalPractice definitions are “primary soil tillage” or “seeding”. A CulturalPractice can refer to a list of references to several OperationTechniques (e.g. cultural practice: “fertilization” - > operation techniques: “liquid fertilization”, “organic fertilization”, “gaseous fertilization”).

Referenced by XML elements:

— *OperTechPractice*

Includes XML elements:

— *OperationTechniqueReference*

Table D.15 — CulturalPractice attributes

Attribute	XML	Use	Type	Length	Comment
CulturalPracticeId	A	r	xs:ID	min. 4 to max. 14	Unique identifier of Cultural-Practice Format: (CPC|CPC-)([0–9])+ Records generated on MICS have negative IDs
CulturalPracticeDesignator	B	r	xs:string	max. 32	Designator of the CulturalPractice
OperationTechniqueReference		o	xs:element		Includes a list of XML element OperationTechniqueReference

EXAMPLE

```
<CPC A="CPC1" B="fertilization">
   <OTR A="OTQ1"/>
</CPC>
<CPC A="CPC2" B="seeding"/>
<CPC A="CPC3" B="harvest"/>
```

D.16 Customer — CTR

Type:

Coding data

Description:

The Customer XML element describes a customer. A customer can be referenced in a task, farm, and partfield. The relationship between a customer and farms and partfields can be multiple. To determine which farms or partfields belong to a specific customer, the CustomerIdRef values of all farms or partfields, respectively, shall be examined for a match with a particular CustomerId value.

Referenced by XML elements:

— *Task*

— *Farm*

— *Partfield*

Table D.16 — Customer attributes

Attribute	XML	Use	Type	Length/range	Comment
CustomerId	A	r	xs:ID	min. 4 to max. 14	Unique identifier of customer Format: (CTR|CTR-)([0–9])+ Records generated on MICS have negative IDs
CustomerLastName	B	r	xs:string	max. 32	Customer's last name
CustomerFirstName	C	o	xs:string	max. 32	Customer's first name
CustomerStreet	D	o	xs:string	max. 32	Street
CustomerPOBox	E	o	xs:string	max. 32	PO box
CustomerPostalCode	F	o	xs:string	max. 10	Postal code
CustomerCity	G	o	xs:string	max. 32	City
CustomerState	H	o	xs:string	max. 32	State or county

Table D.16 *(continued)*

Attribute	XML	Use	Type	Length/range	Comment
CustomerCountry	I	o	xs:string	max. 32	Country
CustomerPhone	J	o	xs:string	max. 20	Telephone number
CustomerMobile	K	o	xs:string	max. 20	Mobile phone number
CustomerFax	L	o	xs:string	max. 20	Fax number
CustomerEMail	M	o	xs:string	max. 64	E-mail

EXAMPLE

```
<CTR A=”CTR1” B=”Smith” C=”John” G=”Munich”/>
```

D.17 DataLogTrigger — DLT

Type:

Task data

Description:

The DataLogTrigger XML element is included in the task and contains the information about which ProcessDataVariables values shall be logged as DataLogValues during task processing. The reference to the DeviceElement can be added on the mobile system, as soon as a device is allocated to the task. When the reference to the DeviceElement is already given at FMIS then a specific device was planned for the task. When a reference to a DeviceElement is not specified in a DataLogTrigger, then the TC shall log the requested DDI from all DeviceElements that can supply this DDI. The attributes of the DataLogTrigger define the behaviour of the TC, regarding how to collect and store the ProcessDataVariable values.

The data log methods “time interval”, “distance interval”, and “on change” can be used in any combination. The logging is triggered by the event that occurs first, and all active methods of these three data log methods are then restarted. Additionally, the “threshold limits” method can be added. With this addition, the logging is triggered when the logging value enters the value range specified by the threshold limit definitions and is enabled as long as the value is within the value range specified by the threshold limit definitions.

If the DataLogThresholdMinimum is smaller than the DataLogThresholdMaximum, data logging is enabled when the value is between the DataLogThresholdMinimum and the DataLogThresholdMaximum. If the DataLogThresholdMinimum is larger than the DataLogThresholdMaximum, data logging is enabled when the value is larger than the DataLogThresholdMinimum or smaller than the DataLogThresholdMaximum.

The data log method “total” is independent of the other data log methods and can be used with any combination described above. DataLogValues of the type “total” shall be stored once per Time XML Element in a task in the data transfer file set.

Values from parameter groups can be logged by specification of the attributes DataLogPGN, DataLogPGNStartBit and DataLogPGNStopBit. When these attributes are specified, the DataLogDDI attribute shall be set to $DFFE_{16}$ (PGN log value).

Included by XML elements:

— *Task*

References XML elements:

— *DeviceElement*

Table D.17 — DataLogTrigger attributes

Attribute	XML	Use	Type	Length/range	Comment
DataLogDDI	A	r	xs:hexBinary	0000_{16} to $FFFF_{16}$	Unique number, which identifies the ProcessDataVariable (as specified in ISO 11783-11)
DataLogMethod	B	r	xs:unsignedByte	1 to 31	Selection of the log method: 1 = time interval 2 = distance interval 4 = threshold limits 8 = on change 16 = total
DataLogDistanceInterval	C	o	xs:long	0 to 1000000	Distance interval for data log in mm, 0 stops measurement.
DataLogTimeInterval	D	o	xs:long	0 to 60000	Time interval for data log in ms, 0 stops measurement, 100 ms is the minimum time interval.
DataLogThresholdMinimum	E	o	xs:long	$(-2^{31}+1)$ to $(2^{31}-1)$	Minimum threshold to activate the data log. Threshold limit value is included in the value range to log. $(2^{31}-1)$ stops measurement.
DataLogThresholdMaximum	F	o	xs:long	$(-2^{31}+1)$ to $(2^{31}-1)$	Maximum threshold to activate the data log. Threshold limit value is included in the value range to log. $(-2^{31}+1)$ stops measurement.
DataLogThresholdChange	G	o	xs:long	$(-2^{31}+1)$ to $(2^{31}-1)$	Change threshold to activate the data log, 0 stops measurement, 1 logs each change.
DeviceElementIdRef	H	o	xs:IDREF	min. 4 to max. 14	Reference to XML element DeviceElement Format (DET|DET-)([0–9])+
ValuePresentationIdRef	I	o	xs:IDREF	min. 4 to max. 14	Reference to XML element ValuePresentation Format (VPN|VPN-)([0–9])+
DataLogPGN	J	o	xs:unsignedLong	0 to $(2^{18}-1)$	Parameter Group to log a value from.
DataLogPGNStartBit	K	o	xs:unsignedByte	0 to 63	First bit of the value to log from a parameter group. 0 specifies the least significant bit of Byte 1 in the data field of a data frame (zero based bit index). The start bit is included in the value and becomes the least significant bit.
DataLogPGNStopBit	L	o	xs:unsignedByte	0 to 63	Stop bit of the value to log from a parameter group. The stop bit is included in the value and becomes the most significant bit.

EXAMPLE

```
<DLT A="1122" B="1" D="1000" H="DET2"/>
```

D.18 DataLogValue — DLV

Type:

Task data

Description:

The DataLogValue specifies a single value of a single ProcessDataVariable, specified by its DDI and supplied by a single DeviceElement. The XML attribute DeviceElementIdRef references the appropriate DeviceElement. The position and time of a DataLogValue are specified through the XML element Time. Time is included in the task and by this relation all DataLogValues belong to a task. DataLogValues are part of the data logging functionality of the TC.

When the values are logged from a parameter group, the attributes DataLogPGN, DataLogPGNStartBit and DataLogPGNStopBit are used and the DataLogDDI attribute shall be set to $DFFE_{16}$ (PGN log value).

Included by XML elements:

— *Time*

References XML elements:

— *DeviceElement*

Table D.18 — DataLogValue attributes

Attribute	XML	Use	Type	Length/range	Comment
ProcessDataDDI	A	r	xs:hexBinary	0000_{16} to $FFFF_{16}$	Unique number, which defines the ProcessDataVariable (as specified in Annex B and ISO 11783-11)
ProcessDataValue	B	r	xs:long	-2^{31} to $(2^{31}-1)$	Value
DeviceElementIdRef	C	r	xs:IDREF	min. 4 to max. 14	Reference to DeviceElement Format: (DET|DET-)([0–9])+
DataLogPGN	D	o	xs:unsigned-Long	0 to $(2^{18}-1)$	Parameter Group to log a value from.
DataLogPGNStartBit	E	o	xs:unsigned-Byte	0 to 63	First bit of the value to log from a parameter group. 0 is the least significant bit of Byte 1 in the data field of a data frame (zero based bit index). The start bit is included in the value and becomes the least significant bit.
DataLogPGNStopBit	F	o	xs:unsigned-Byte	0 to 63	Stop bit of the value to log from a parameter group. The stop bit is included in the value and becomes the most significant bit.

EXAMPLE

```
<DLV A="0815" B="10" C="DET1"/>
<DLV A="4711" B="15" C="DET2"/>
<DLV A="4522" B="20" C="DET3"/>
```


D.19 Device — DVC

Type:

Coding data

Description:

The Device XML element describes a complete device, like a machine or sensor system. Each device shall have at least one DeviceElement.

Includes XML elements:

— *DeviceElement*

— *DeviceProcessData*

— *DeviceProperty*

— *DeviceValuePresentation*

Referenced by XML elements:

— *Connection*

— *DeviceAllocation*

Table D.19 — Device attributes

Attribute	XML	Use	Type	Length/range	Comment
DeviceId	A	r	xs:ID	min. 4 to max. 14	Unique identifier of device Format: (DVC|DVC-)([0–9])+
DeviceDesignator	B	o	xs:string	max. 32	Device designator
DeviceSoftwareVersion	C	o	xs:string	max. 32	Software version of device
ClientNAME	D	r	xs:hexBinary	0000000000000000_{16} to $FFFFFFFFFFFFFFFF_{16}$	NAME of client device (see ISO 11783-5)
DeviceSerialNumber	E	o	xs:string	max. 32	Device and manufacturer-specific serial number of the Device or Machine represented by the client. E.g. a unique Vehicle or Product identification code [a].
DeviceStructureLabel	F	r	xs:hexBinary	00_{16} to FE_{16} per byte for the least significant 7 bytes, 00_{16} to FF_{16} per byte for the extended structure label bytes. The maximum number of bytes of the structure label extension is 32.	Label of device descriptor structure. Structure label array byte 1 is the least significant byte in the xs:hexBinary value and structure label array Byte n is the most significant byte in the xs:hexBinary value. See example 1 in this section.

[a] Defined in ISO 11783-12, 2nd edition Diagnostics services.

Table D.19 *(continued)*

Attribute	XML	Use	Type	Length/range	Comment
DeviceLocalizationLabel	G	r	xs:hexBinary	Bytes 1 to 6: 00_{16} to FE_{16} per byte, Byte 7 = FF_{16}	Label of device descriptor localization. Bytes 1 to 6 are defined by the language command PGN (see ISO 11783-7). Byte 7 is reserved and set to FF_{16}. Language command PGN Byte 1 is least significant and language command Byte 7 is most significant.
DeviceElement		r	xs:element		Includes a list of XML element DeviceElement
DeviceProcessData		o	xs:element		Includes a list of XML element DeviceProcessData
DeviceProperty		o	xs:element		Includes a list of XML element DeviceProperty
DeviceValuePresentation		o	xs:element		Includes a list of XML element DeviceValuePresentation

[a] Defined in ISO 11783-12, 2nd edition Diagnostics services.

EXAMPLE 1 Structure label and localization label

The language command bytes 1 and 2 contain the language code. For example, a language code ‘en’ is represented by bytes 65_{16} and $6E_{16}$, the least significant byte in the localization label in this case has the value 65_{16}.

Byte order in the DeviceObject in the XML file:

```
<DVC A=”DVC1” B=”sprayer 4711” C=”1.0” D=”A00C8400073FFFC7” F=”F9FAFBFCFDFE39”
G=”FF000000006E65”>
```

Structure label array byte [7][6][5][4][3][2][1]: F9 FA FB FC FD FE 39

Localization label byte [7][6][5][4][3][2][1]: FF 00 00 00 00 6E 65

Byte order in the DeviceObject transferred over the CAN bus:

Structure label array byte order on the CAN bus [1][2][3][4][5][6][7]: 39 FE FD FC FB FA F9

localization label byte order on the CAN bus [1][2][3][4][5][6][7]: 65 6E 00 00 00 00 FF

EXAMPLE 2 Extended structure label of 8 bytes added to the 7 byte structure label

Byte order in the DeviceObject in the XML file:

```
<DVC A=”DVC1” B=”sprayer 4711” C=”1.0” D=”A00C8400073FFFC7”
F=”E8E7E6E5E4E3E2E1F9FAFBFCFDFE39” G=”FF000000006E65”>
```

Structure label array byte [15][14][13][12][11][10][9][8][7][6][5][4][3][2][1]: E8 E7 E6 E5 E4 E3 E2 E1 F9 FA FB FC FD FE 39

Byte order in the DeviceObject transferred over the CAN bus:

Structure label array byte order on the CAN bus[1][2][3][4][5][6][7][8][9][10][11][12][13][14][15]: 39 FE FD FC FB FA F9 .. .. 08 E1 E2 E3 E4 E5 E6 E7 E8

EXAMPLE 3 Device descriptor

```
<DVC A=”DVC1” B=”sprayer 4711” C=”1.0” D=”A00C84000B20408B” F=”30353035344042”
G=”FF000000006C6E”>
```



```
    <DET A="DET1" B="1" C="1" D="all elements" E="0" F="0">
      <DOR A="3"/>
      <DOR A="4"/>
      <DOR A="8"/>
      <DOR A="9"/>
      <DOR A="10"/>
      <DOR A="11"/>
    </DET>
    <DET A="DET2" B="2" C="3" D="main tank" E="1" F="1">
      <DOR A="5"/>
      <DOR A="6"/>
      <DOR A="7"/>
      <DOR A="8"/>
      <DOR A="9"/>
      <DOR A="10"/>
      <DOR A="12"/>
    </DET>
    <DPD A="3" B="1234" C="3" D="1"/>
    <DPD A="4" B="8765" C="1" D="1"/>
    <DPD A="5" B="1111" C="3" D="1"/>
    <DPD A="6" B="1112" C="3" D="1"/>
    <DPD A="7" B="1133" C="1" D="2"/>
    <DPT A="8" B="4301" C="0"/>
    <DPT A="9" B="4302" C="0"/>
    <DPT A="10" B="4303" C="0"/>
    <DPT A="11" B="4305" C="2700"/>
    <DPT A="12" B="4304" C="4500"/>
  </DVC>
```

D.20 DeviceAllocation — DAN

Type:

Task data

Description:

The DeviceAllocation XML element includes information on which device(s) the planned task was created for and which devices were actually used during task processing. For a planned task, the DeviceAllocation describes the ClientNAMEValue and optionally a NAME mask to enable a range of NAME values which can specify devices that are also allowed for the task processing. During task processing, the TC adds a new DeviceAllocation to the task with the client information that was actually used to perform the task.

Includes XML elements:

— *AllocationStamp*

Included by XML elements:

— *Task*

References XML elements:

— *Device*

Table D.20 — DeviceAllocation attributes

Attribute	XML	Use	Type	Length/range	Comment
ClientNAMEValue	A	r	xs:hexBinary	000000_{16} to FFFFFFFF FFFFFFF$_{16}$	NAME of client device (see ISO 11783-5) the task is planned for or was processed with.

Table D.20 *(continued)*

Attribute	XML	Use	Type	Length/range	Comment
ClientNAMEMask	B	o	xs:hexBinary	000000_{16} to FFFFFFFFFFFFFFF$_{16}$	Bit-Mask, which is to be used for a logical AND operation to ClientNAMEValue, to allow more then one specific device for the task. bit value = 1 = > relevant bit of the ClientNAMEValue bit value = 0 = > bit of the ClientNAMEValue is not relevant.
DeviceIdRef	C	o	xs:IDREF	min. 4 to max. 14	Reference to XML element Device Format: (DVC|DVC-)([0–9])+
AllocationStamp		o	xs:element		Includes a single XML element AllocationStamp

EXAMPLE 1 Planned DeviceAllocation, the mask specifies to assign a sprayer machine controller, first function instance found on the network. This planned DeviceAllocation does not have a reference to a specific Device.

```
<DAN B="7FFFFFFF00000000" A="200C840000000000">
   <ASP A="2003-08-20T08:00:00" B="2003-08-20T17:00:00" D="1"/>
</DAN>
```

EXAMPLE 2 Planned DeviceAllocation to assign one exact instance of a sprayer machine controller. This planned DeviceAllocation does have a reference to a specific Device.

```
<DAN B="FFFFFFFFFFFFFFFF" A="A00C84000B20408B" C="DVC1">
   <ASP A="2003-08-20T08:00:00" B="2003-08-20T17:00:00" D="1"/>
</DAN>
```

EXAMPLE 3 Recorded DeviceAllocation of a sprayer machine controller. This DeviceAllocation is added to a task for the actual Device that is used in the task.

```
<DAN A="A00C84000B20408B" C="DVC1">
   <ASP A="2003-08-21T07:34:09" B="2003-08-21T12:40:23" D="4"/>
</DAN>
```

D.21 DeviceElement — DET

Type:

Coding data

Description:

The DeviceElement XML element describes the functional or physical elements of a device. To establish a hierarchical structure of element groups, a DeviceElement shall refer to another DeviceElement or to the device itself. The DeviceElementType attribute is defined in A.3. The ParentObjectId attribute is used to refer either to the DeviceObject (object ID = 0) or to a parent DeviceElementObject to establish a hierarchical ordering of DeviceElements.

Included by XML elements:

— *Device*

Includes XML elements:

— *DeviceObjectReference*

References XML elements:

— *DeviceElement*

— *Device*

Referenced by XML elements:

— *Connection*

— *DataLogTrigger*

— *DataLogValue*

— *ProcessDataVariable*

— *ProductAllocation*

Table D.21 — DeviceElement attributes

Attribute	XML	Use	Type	Length/range	Comment
DeviceElementId	A	r	xs:ID	min. 4 to max. 14	Unique identifier of DeviceElement Format: (DET|DET-)([0–9])+ Records generated on MICS have negative IDs
DeviceElementObjectId	B	r	xs:unsigned-Short	1 to 65534	Object ID unique inside a device descriptor
DeviceElementType	C	r	xs:NMTOKEN	1 to 7	Selection of type: 1 = device 2 = function 3 = bin 4 = section 5 = unit 6 = connector 7 = navigation
DeviceElementDesigna-tor	D	o	xs:string	max. 32	Designator of element
DeviceElementNumber	E	r	xs:unsigned-Short	0 to 4095	Unique number of the element, refer to Annex B: ProcessDataVariable element numbering
ParentObjectId	F	r	xs:unsigned-Short	0 to 65534	Object ID of parent DeviceElement or Device
DeviceObjectReference		o	xs:element		Includes a list of XML element Devi-ceObjectReference

EXAMPLE

```
<DET A="DET1" B="1" C="1" D="all elements" E="0" F="0">
   <DOR A="3"/>
   <DOR A="4"/>
   <DOR A="8"/>
   <DOR A="9""/>
   <DOR A="10"/>
   <DOR A="11"/>
</DET>
```

D.22 DeviceObjectReference — DOR

Type:

Coding data

Description:

The DeviceObjectReference XML element describes a reference to a DeviceProcessData object or a DeviceProperty object.

This XML element is part of the device descriptor.

References XML elements:

— *DeviceProcessData*

— *DeviceProperty*

Included by XML elements:

— *DeviceElement*

Table D.22 — DeviceObjectReference attributes

Attribute	XML	Use	Type	Length/range	Comment
DeviceObjectId	A	r	xs:unsigned-Short	1 to 65534	Object ID of DeviceProcessData or DeviceProperty

EXAMPLE

```
<DOR A="3" />
```

D.23 DeviceProcessData — DPD

Type:

Coding data

Description:

The DeviceProcessData XML element describes ProcessDataVariable DDIs, supported by the DeviceElement that references this XML element. The DeviceProcessDataProperty attribute options "settable" and "control source" are mutually exclusive, only one of these options may be set in the DeviceProcessDataProperty attribute value. The DeviceProcessDataTriggerMethods attribute specifies the available trigger methods for a ProcessDataVariable DDI.

This XML element is part of the device descriptor.

References XML element:

— *DeviceValuePresentation*

Referenced by XML element:

— *DeviceObjectReference*

Included by XML elements:

— *Device*

Table D.23 — DeviceProcessData attributes

Attribute	XML	Use	Type	Length/range	Comment
DeviceProcessDataObjectId	A	r	xs:unsigned-Short	1 to 65534	Unique number inside a single device
DeviceProcessDataDDI	B	r	xs:hexBinary	0000_{16} to $FFFF_{16}$	Unique number which specifies the process data variable (defined in ISO 11783-11)
DeviceProcessDataProperty	C	r	xs:NMTOKEN	0 to 7	Bit combination to specify the ProcessDataVariable property: 1 = belongs to default set 2 = settable 4 = control source[a]
DeviceProcessDataTrigger-Methods	D	r	xs:integer	0 to 31	Bit combination to specify supported trigger methods: 1 = time interval 2 = distance interval 4 = threshold limits 8 = on change 16 = total
DeviceProcessDataDesignator	E	o	xs:string	max. 32	Designator of DeviceProcess-Data
DeviceValuePresentationObjectId	F	o	xs:unsigned-Short	1 to 65534	Object ID of DeviceValueP-resentation

a This bit value is introduced in ISO 11783-10 version 4.

EXAMPLE

```
<DPD A="1" B="1234" C="3" D="1"/>
```

D.24 DeviceProperty — DPT

Type:

Coding data

Description:

The DeviceProperty XML element describes a property of a DeviceElement by means of a reference and a value for a DDI. This XML element is part of the device descriptor.

References XML element:

— *DeviceValuePresentation*

Referenced by XML element:

— *DeviceObjectReference*

Included by XML elements:

— *Device*

Table D.24 — DeviceProperty attributes

Attribute	XML	Use	Type	Length/range	Comment
DevicePropertyObjectId	A	r	xs:unsignedShort	1 to 65534	Unique number inside a single device
DevicePropertyDDI	B	r	xs:hexBinary	0000_{16} to $FFFF_{16}$	Unique number which defines the property (specified in ISO 11783-11)
DevicePropertyValue	C	r	xs:long	$–2^{31}$ to $(2^{31}–1)$	Property value
DevicePropertyDesignator	D	o	xs:string	max. 32	Optional designator for property
DeviceValuePresentationObjectId	E	o	xs:unsignedShort	1 to 65534	Object ID of DeviceValuePresentation

EXAMPLE

```
<DPT A="8" B="1235" C="-65233"/>
```

D.25 DeviceValuePresentation — DVP

Type:

Coding data

Description:

The DeviceValuePresentation XML element is used to specify the presentation of the data dictionary entity-defined integer values that are used within a single device. The presentation shall be according to the following formula:

Presented value = (integer value + Offset) * Scale

Presented values are always rounded to the number of decimals specified in the NumberOfDecimals attribute.

Referenced by XML elements:

— *DeviceProcessData*

— *DeviceProperty*

Included by XML elements:

— *Device*

Table D.25 — DeviceValuePresentation attributes

Attribute	XML	Use	Type	Length/range	Comment
DeviceValuePresentationObjectId	A	r	xs:unsignedShort	1 to 65534	Unique number inside a single device
Offset	B	r	xs:long	$–2^{31}$ to $(2^{31}–1)$	Offset to be applied to the value for presentation
Scale	C	r	xs:decimal	0.000000001 to 100000000.0	Scale to be applied to the value for presentation
NumberOfDecimals	D	r	xs:unsignedByte	0 to 7	Number of decimals to be presented after the decimal separator

Table D.25 *(continued)*

Attribute	XML	Use	Type	Length/ range	Comment
UnitDesignator	E	o	xs:string	max. 32	Optional unit designator string

EXAMPLE

```
<DVP A="1" B="0" C="1.0" D="0" E="kg"/>
<DVP A="2" B="18" C="1.8" D="1" E="°F"/>
```

D.26 Farm — FRM

Type:

Coding data

Description:

The Farm XML element contains all required information to describe a farm. In the data transfer file set, a farm can hold a collection of fields that is managed independently from other farms. The relationships between a customer and farms and partfields can be multiple. To determine which farms or partfields belong to a specific customer, the CustomerIdRef values of all farms or partfields, respectively, shall be examined for a match with a particular CustomerId value.

References XML element:

— *Customer*

Referenced by XML elements:

— *Partfield*

— *Task*

Table D.26 — Farm attributes

Attribute	XML	Use	Type	Length/range	Comment
FarmId	A	r	xs:ID	min. 4 to max. 14	Unique identifier of Farm Format: (FRM|FRM-)([0–9])+ Records generated on MICS have negative IDs
FarmDesignator	B	r	xs:string	max. 32	Farm designator/name
FarmStreet	C	o	xs:string	max. 32	Street
FarmPOBox	D	o	xs:string	max. 32	PO Box
FarmPostalCode	E	o	xs:string	max. 10	Postal code
FarmCity	F	o	xs:string	max. 32	City
FarmState	G	o	xs:string	max. 32	State or county
FarmCountry	H	o	xs:string	max. 32	Country
CustomerIdRef	I	o	xs:IDREF	min. 4 to max. 14	Reference to XML element customer Format: (CTR|CTR-)([0–9])+

EXAMPLE

```
<FRM A="FRM1" B="bonanza ranch" />
```

D.27 Grid — GRD

Type:

Task data

Description:

The Grid XML element describes the dimension and position of a set of gridcells. There shall be a definition of the minimum north/east position, the size of each gridcell, and the number of gridcells in the north/east direction. There can only be a single grid specified per task. The grid is always related to a partfield but the definition of grid is always task-specific. The grid cells of a grid contain a reference to a TreatmentZone or a process data variable value. The grid shall be specified as a complete array of gridcells in ascending order, as gridcells do not contain any ordering information.

The gridcells shall be defined in a binary format in a separate file. There can only be a single binary file per grid and per task. The gridcell files shall exist in the same directory as the other files of the data transfer file set. The name of the gridcell files shall be unique over all grids referred to by tasks of a data transfer file set.

A Task can have at a maximum one Grid. TaskControllers (with limited processing capacity) can handle only one Grid per Task. This means that in case of more OperTechPractises, the FMIS shall formulate a common Grid which specifies the TreatmentZones which are valid for all OperTechPractises. Note that a single TreatmentZone may contain more than one ProcessDataVariable to be controlled by a position-based controller. Multi-variable position-based control is possible with both grid type 1 and grid type 2.

References XML element:

— *TreatmentZone*

Included by XML elements:

— *Task*

Table D.27 — Grid attributes

Attribute	XML	Use	Type	Length/ range	Comment
GridMinimumNorthPosition	A	r	xs:decimal	−90.0 to 90.0	Minimum north position of the grid Format: WGS84
GridMinimumEastPosition	B	r	xs:decimal	−180.0 to 180.0	Minimum east position of the grid Format: WGS84
GridCellNorthSize	C	r	xs:double	0.0 to 1.0	North direction gridcell size Format: WGS84
GridCellEastSize	D	r	xs:double	0.0 to 1.0	East direction gridcell size Format: WGS84
GridMaximumColumn	E	r	xs:unsignedLong	0 to $(2^{32}–1)$	Number of the gridcells in east direction
GridMaximumRow	F	r	xs:unsignedLong	0 to $(2^{32}–1)$	Number of the gridcells in north direction
Filename	G	r	xs:ID	8	Unique name of gridcell file Format: GRD[0–9][0–9][0–9][0–9][0–9]
Filelength	H	o	xs:unsignedLong	0 to $(2^{32}–2)$	Length of gridcell file in number of bytes

Table D.27 *(continued)*

Attribute	XML	Use	Type	Length/ range	Comment
GridType	I	r	xs:NMTOKEN	1 to 2	Grid type specification: 1 = grid type 1 2 = grid type 2
TreatmentZoneCode	J	o	xs:unsigned-Byte	0 to 254	Grid type 2 TreatmentZoneCode

EXAMPLE

Grid type 1 XML element specification:

```
<GRD A="58.096653" B="8.54321" C="0.012" D="0.012" E="200" F="300" G="GRD00001" I="1"/>
```

See 8.6.2 for examples of both grid type 1 and grid type 2 specifications.

D.28 GuidanceAllocation — GAN

Type:

Task data

Description:

The GuidanceAllocation XML element describes the allocation of a GuidanceGroup to a Task. The AllocationStamp entry describes the start/stop time of allocations and enables tracking of the changes of the guidance allocations to a task.

The included GuidanceShift XML element specifies geospatioal shifts applied to the GuidancePatterns in the allocated GuidanceGroup. Each GuidanceShift XML element contains its won AllocationStamp XML element. If multiple shift operations occur, then each shift operation is recorded in a new GuidanceShift XML element in the GuidanceAllocation XML element.

The GuidanceAllocation XML element is added in ISO 11783-10 version 4.

References XML element:

— *GuidanceGroup*

Included by XML element:

— *Task*

Includes XML elements:

— *AllocationStamp*

— *GuidanceShift*

Table D.28 — GuidanceAllocation attributes

Attribute	XML	Use	Type	Length/ range	Comment
GuidanceGroupIdRef	A	r	xs:IDREF	min. 4 to max. 14	Reference to XML element Guidance-Group Format: (GGP|GGP-)([0–9])+ Records generated on MICS have negative IDs
AllocationStamp		r	xs:element		Includes a single XML element Alloca-tionStamp
GuidanceShift		o	xs:element		Includes a list of XML element Guid-anceShift

EXAMPLE

```
<PFD A="PFD3" C="hill" D="32000" G="CTP3">
   <PLN A="1" B="Field Boundary" C="32000" D="1">
      <LSG A="1" B="Line1" C="20" D="280000" E="1" >
         <PNT A="2" B="start" C="40.84982" D="-96.596045" E="150234" F="1" />
         <PNT A="2" B="mid" C="40.84982" D="-96.592655" E="148987" F="1" />
         <PNT A="2" B="end" C="40.846573" D="-96.592526" E="148284" F="1" />
      </LSG>
   </PLN>

   <GGP A="GGP1">
      <GPN A="GPN1" C="1" D="1">
         <LSG A="5" B="Guidance1" C="6000" D="1500000" E="1">
            <PNT A="6" B="start" C="40.84934" D="-96.593445" E="148987" F="1" H="0.05"/>
            <PNT A="7" B="end" C="40.84683" D="-96.592225" E="150234" F="1" H="0.04"/>
         </LSG>
      </GPN>

      <GPN A="GPN2" C="1" D="1">
         <LSG A="5" E="1" B="Guidance2" D="900000" C="18000" F="LSG2">
            <PNT A="2" B="start" C="40.84934" D="-96.122322" E="123506" F="1" I="2.0"/>
            <PNT A="2" B="end" C="40.84911" D="-96.591122" E="122544" F="1" I="2.0"/>
         </LSG>
      </GPN>
   </GGP>
</PFD>

<TSK A="TSK1" F="WKR1" G="1">
   <GAN A="GGP1">
      <ASP A="2003-08-20T08:00:00" B="2003-08-20T17:00:00" D="1"/>
   </GAN>
</TSK>
```

D.29 GuidanceGroup — GGP

Type:

Coding data

Description:

The GuidanceGroup XML element groups together one or more Guidance Patterns (GPN). Guidance Patterns in the group are intended to be used simultaneously. Typically a guidance group will contain two headland guidance patterns and a single mainfield guidance pattern, but it could contain any combination of main and headland guidance patterns. Situations where a partfield is bounded with multiple exterior polygons can require the use of multiple headland/mainland guidance patterns. A guidance group can contain only a single guidance pattern, in which case that guidance pattern shall be a mainland guidance pattern.

The GuidanceGroup XML element is added in ISO 11783-10 version 4.

Includes XML element:

— *GuidancePattern*

— *Polygon*

Included by XML element:

— *Partfield*

Referenced by XML element:

— *GuidanceAllocation*

Table D.29 — GuidanceGroup attributes

Attribute	XML	Use	Type	Length/ range	Comment
GuidanceGroupId	A	r	xs:ID	min. 4 to max. 14	Unique identifier of GuidanceGroup Format: (GGP|GGP-)([0–9])+ Records generated on MICS have negative IDs
GuidanceGroupDesignator	B	o	xs:string	max. 32	Designator of GuidanceGroup
GuidancePattern		r	xs:element		Includes a list of XML element GuidancePattern
BoundaryPolygon		o	xs:element		Includes a list of XML element Polygon to define the boundary for this GuidanceGroup. The boundary polygons are used to limit the geometry of the GuidancePatterns in this GuidanceGroup to a certain area inside the field boundary. Note that each boundary polygon can have interior boundaries in addition to the exterior boundary to designate exclusion zones.

EXAMPLE

```
<GGP A=”GGP1”>
   <GPN A=”GPN1” B=”Adaptive Curve” C=”2” D=”1”>
      <LSG A=”5”>
       <PNT A=”6” C=”58.8754321” D=”8.945632”/>
       <PNT A=”9” C=”58.8757777” D=”8.94777”/>
       <PNT A=”7” C=”58.8789999” D=”8.99889099”/>
      </LSG>
   </GPN>
</GGP>
```

D.30 GuidancePattern — GPN

Type:

Coding data

Description:

The GuidancePattern XML element describes the properties necessary to transfer the data for executing a guidance activity. The GuidancePattern XML element contains the properties that classify a guidance pattern while the LineString XML child element describes the geospatial information.

Every GuidancePattern XML element will contain one and only one LineString XML element. The points in the LineString will be determined by the GuidancePattern classification. The swathe width that separates the adjacent paths in the guidance pattern is represented by the LineStringWidth attribute in the LineString XML element. The following table defines the points that shall be used for each GuidancePattern type:

Table D.30 A — GuidancePattern types

Line Type	Required Points	Optional Points	Notes
AB	The start point shall be A and the end point shall be B.	No optional points	
A+	A single A point	No optional points	Shall have a heading defined. A+ lines with no heading are invalid
Curve	Start point shall be A, followed by any number of guidance points, followed by the end point B	Any number of guidance points (type 9) may exist between the A and B points	A curve is a 1-dimensional geometric primitive, representing the continuous image of a line. The curve type is considered an identical curve unless otherwise specified in the options attribute. Crossing its own line is allowed for curve type.
Pivot	A single Center point	A single centre point, followed by A, followed by B.	The A and B points are used to determine the angle of the start and end points for non-full circles. Pivots are considered full circles unless otherwise specified in the options attribute.
Spiral	A single A point followed by any number of guidance points followed by a single B point	Any number of guidance points (type 9) may exist between the A and B points	

Examples of these 5 pattern types are given in Figure D.1.

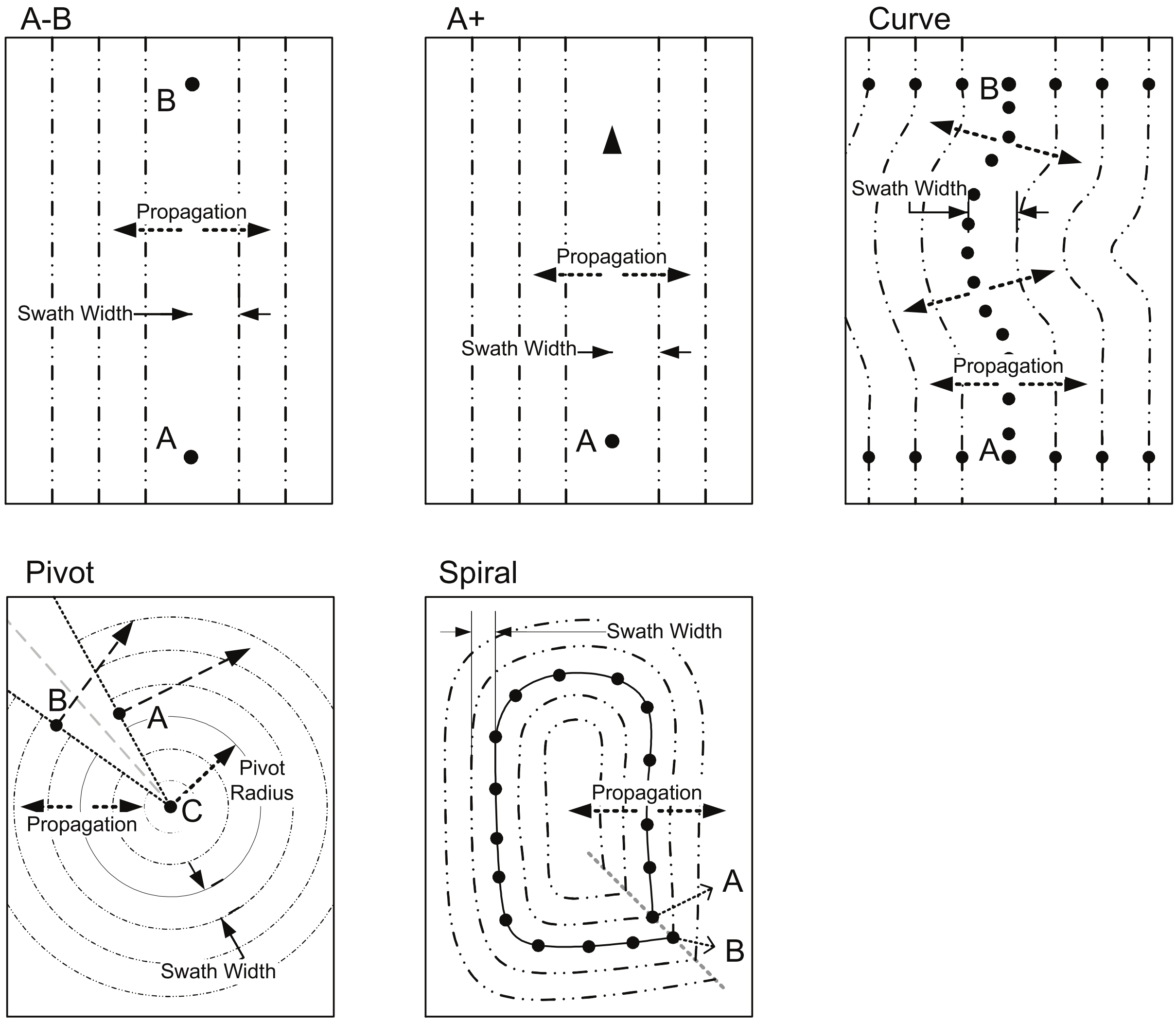


Figure D.1 — Guidance pattern types

AB, A+, and Curve lines will be generated parallel to the previous line and offset a distance determined by the swathe width.

Spiral lines will be generated end-to-end of the previous line and offset a distance determined by the swathe width. These will commonly be headlands.

Guidance patterns can reference the base stations they were recorded with. Monitors are expected to operate on guidance patterns without base station references.

Propagation direction indicates how the parallel offset lines should grow from the original. The direction is defined while standing on the first point and looking to the second point. The propagation definition supplied in attributes GuidancePatternPropagationDirection, NumberOfSwathsLeft and BoundaryPolygon shall not contradict each other.

The horizontal and vertical accuracies can be defined once for the entire GuidancePattern or can be defined at each individual Point. If these accuracies are present in both XML elements, then the accuracies defined in the individual points take precedence.

The boundary polygon defines the region in which a guidance pattern line is propagated. If a boundary polygon is defined, then this polygon must fit inside the defined field boundary polygon in which this guidance pattern is contained.

The GuidancePattern XML element is added in ISO 11783-10 version 4.

Included in XML element:

— *GuidanceGroup*

Includes XML element:

— *LineString*

— *Polygon*

References XML element:

— *BaseStation*

Table D.30 B — GuidancePattern attributes

Attribute	XML	Use	Type	Length/range	Comment
GuidancePatternId	A	r	xs:ID	min. 4 to max. 14	Unique identifier of GuidancePattern Format: (GPN|GPN-)([0–9])+ Records generated on MICS have negative IDs
GuidancePatternDesignator	B	o	xs:string	max. 32	Designator of GuidancePattern
GuidancePatternType	C	r	xs:NMTOKEN	1…5	Selection of type: 1 = AB 2 = A+ 3 = Curve 4 = Pivot 5 = Spiral
GuidancePatternOptions	D	o	xs:NMTOKEN	1…3	1 = Clockwise - for pivot 2 = Counter-clockwise - for pivot 3 = Full Circle - for pivot
GuidancePatternPropagationDirection	E	o	xs:NMTOKEN	1…4	1 = Both directions 2 = Left direction only 3 = Right direction only 4 = No propagation The direction is defined while standing on the first point (A) and looking to the next point. When this attribute is not defined in the GuidancePattern then propagation shall be applied in both directions.

Table D.30 *(continued)*

Attribute	XML	Use	Type	Length/range	Comment
GuidancePatternExtension	F	o	xs:NMTOKEN	1..4	1 = From both first and last point 2 = From first point (A) only 3 = From last point (B) only 4 = No extensions When this attribute is not defined in the GuidancePattern then extensions shall be applied from both first and last point. Extension means that guidance shall be continued when the specified point of the Guidance-Pattern is passed. The Guidan-cePattern is extended by the guidance system.
GuidancePatternHeading	G	o	xs:decimal	0.0…360.0	Decimal degrees counting clock-wise relative to the true north
GuidancePatternRadius	H	o	xs:unsignedLong	0 to (2^{32}-2)	Pivot radius information in mm
GuidancePatternGNSS-Method	I	o	xs:NMTOKEN	0…17	Reference to NMEA 2000 GNSS Method parameter: 1 = GNSS fix 2 = DGNSS fix 3 = Precise GNSS 4 = RTK Fixed Integer 5 = RTK Float 6 = Estimated (DR) mode 7 = Manual Input 8 = Simulate mode (NMEA 2000 defines at maxi-mum 16 tokens for GNSS Method) Additional non- NMEA 2000 methods: 16 = Desktop generated data 17 = Other
GuidancePatternHorizonta-lAccuracy	J	o	xs:decimal	0.0 to 65.0	Horizontal estimated accuracy (RMS error) in m
GuidancePatternVerticalAc-curacy	K	o	xs:decimal	0.0 to 65.0	Vertical estimated accuracy (RMS error) in m
BaseStationIdRef	L	o	xs:IDREF	min. 4 to max. 14	Reference to XML element BaseS-tation Format: (BSN|BSN-)([0–9])+

Table D.30 *(continued)*

Attribute	XML	Use	Type	Length/range	Comment
OriginalSRID	M	o	xs:string	32	Coordinate system and projection method used by the guidance system at the time of creating the GuidancePattern. This information can assist with the representation of the GuidancePattern when it is presented on a GUI after a GuidancePattern reload. The OriginalSRID attribute consists of an organization identifier and a well known id (WKID) for the coordinate system and/or projection method. An example OriginalSRID value is “EPSG:4326”. Note that the units of the GuidancePattern do not change when the OriginalSRID is not a WGS-84 type coordinate system and projection. The coordinates of the GuidancePattern are communicated in the data transfer file set as WGS-84 latitude, longitude, and altitude values.
NumberOfSwathsLeft	N	o	xs:unsignedLong	0 to (2^{32}-2)	Number of swathes to propagate GuidancePattern to the left. The direction is defined while standing on the first point (A) and looking to the next point. When this attribute is not defined, the propagation to the left is not bound by a specific number of swathes and follows attribute GuidancePatternPropagationDirection.
NumberOfSwathsRight	O	o	xs:unsignedLong	0 to (2^{32}-2)	Number of swathes to propagate GuidancePattern to the right. The direction is defined while standing on the first point (A) and looking to the next point. When this attribute is not defined, the propagation to the right is not bound by a specific number of swathes and follows attribute GuidancePatternPropagationDirection.
LineString		r	xs:element		Includes a single XML element LineString

Table D.30 *(continued)*

Attribute	XML	Use	Type	Length/range	Comment
BoundaryPolygon		o	xs:element		Includes a list of XML element Polygon An optional boundary for this line. The boundary can be used to limit the validity of a GuidancePattern to a certain area. This boundary shall fit inside the GuidanceGroup and Partfield boundaries. Note that each boundary polygon can have interior boundaries in addition to the exterior boundary to designate exclusion zones.

EXAMPLE 1 AB guidance pattern

```
<GPN A="GPN1" B="AB" C="1" E="3" F="3">
   <LSG A="5">
      <PNT A="6" C="58.8754321" D="8.945632"/>
      <PNT A="7" C="58.8789999" D="8.99889099"/>
   </LSG>
</GPN>
```

EXAMPLE 2 Curve guidance pattern

```
<GPN A="GPN1" B="Adaptive Curve" C="3">
   <LSG A="5">
      <PNT A="6" C="58.8754321" D="8.945632"/>
      <PNT A="9" C="58.8757777" D="8.94777"/>
      <PNT A="7" C="58.8789999" D="8.99889099"/>
   </LSG>
</GPN>
```

EXAMPLE 3 Pivot guidance pattern

Full circle, centre specified as a single point:

```
<GPN A="GPN1" B="Pivot (full circle)" C="4" D="3" E="1" H="700000">
   <LSG A="5">
      <PNT A="8" C="58.8789999" D="8.99889099"/>
   </LSG>
</GPN>
```

Partial circle, clockwise direction, centre, start and end point defined in C A B point order:

```
<GPN A="GPN1" B="Pivot (clockwise)" C="4" D="1" E="1" H="700000">
   <LSG A="5">
      <PNT A="8" C="58.8789999" D="8.99889099"/>
      <PNT A="6" C="58.8754321" D="8.945632"/>
      <PNT A="7" C="58.8757777" D="8.94777"/>
   </LSG>
</GPN>
```

D.31 GuidanceShift — GST

Type:

Task data

Description:

The GuidanceShift XML element offers the possibility to add shift information data to a certain guidance pattern as it is used during field operations. The shift information shall be applied to all propagated patterns.

Figure D.2 illustrates how the EastShift and NorthShift attributes are applied to a straight line pattern type, a curve pattern type and a pivot pattern type. When the EastShift and NorthShift attributes are applied to a spiral pattern type, a similar shift of both the specified pattern and the propagated patterns results. The typical purpose of the EastShift and NorthShift attributes is to adjust the guidance pattern for positioning system drift.

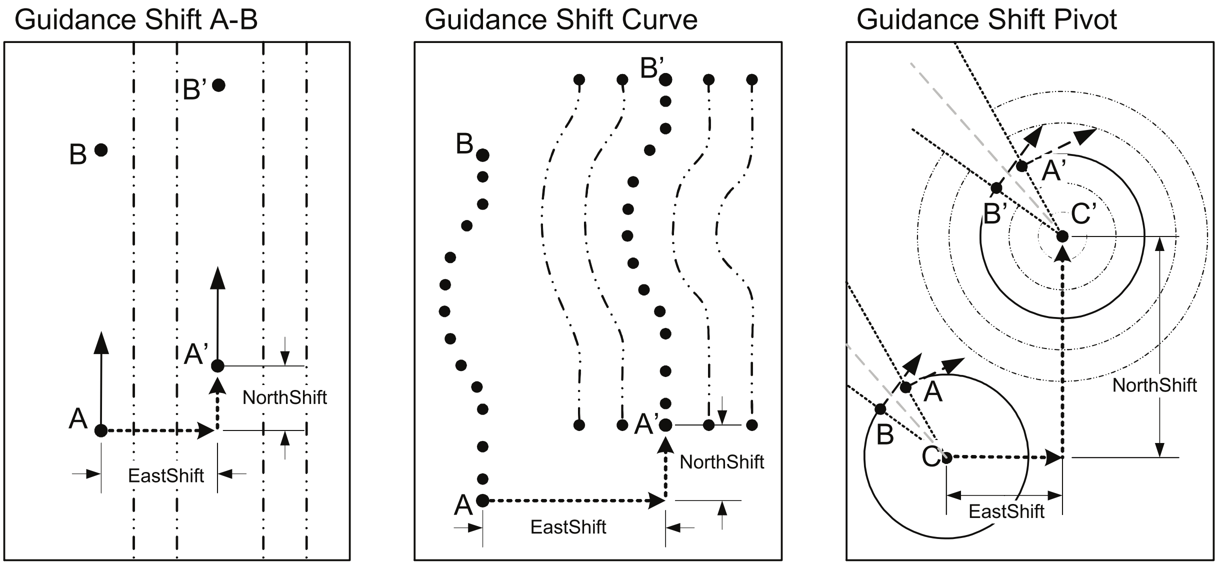


Figure D.2 — Guidance Pattern Shifts

The PropagationOffset attribute offsets the propagated guidance patterns perpendicular to the original guidance pattern. As can be seen in the Figure D.3, the effect of an offset on the straight guidance pattern is similar to the combination of a NorthShift and EastShift. For the curve, pivot and spiral pattern types however, the PropagationOffset differs from a NorthShift and EastShift combination in that the propagated guidance patterns are not shifted but regenerated.

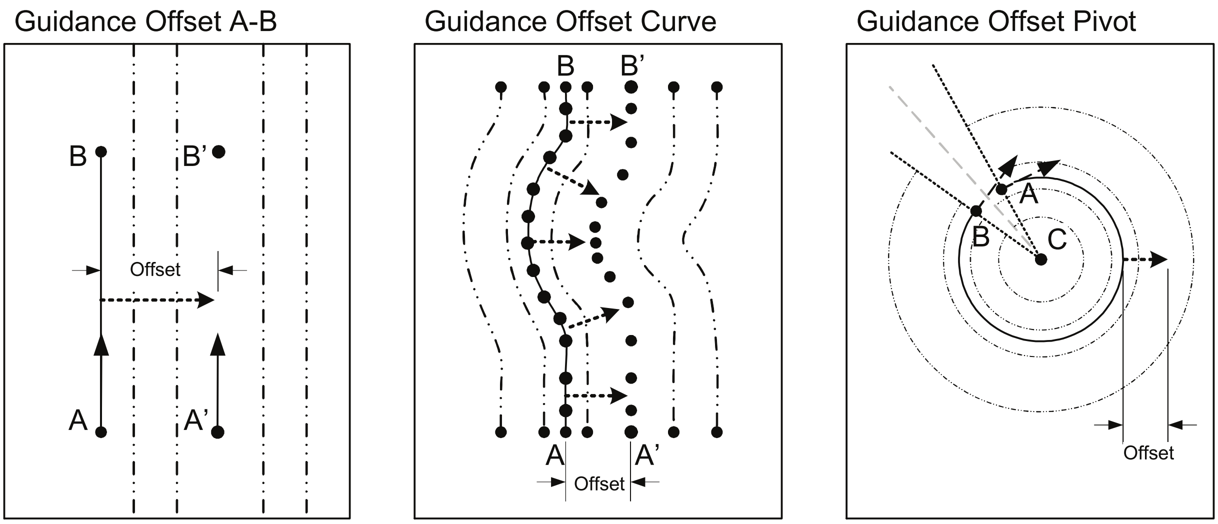


Figure D.3 — Guidance Pattern Offsets

The GuidanceShift XML element is added in ISO 11783-10 version 4.

References XML element:

— *GuidancePattern*

Included by XML element:

— *GuidanceAllocation*

Includes XML element:

— *AllocationStamp*

Table D.31 — GuidanceShift attributes

Attribute	XML	Use	Type	Length/ range	Comment
GuidanceGroupIdRef	A	r	xs:IDREF	min.4 to max.14	Reference to XML element Guidance-Group Format: (GGP|GGP-)([0–9])+ Records generated on MICS have negative IDs Note: either a GuidanceGroup or a GuidancePattern reference is required
GuidancePatternIdRef	B	r	xs:IDREF	min.4 to max.14	Reference to XML element Guidance-Pattern Format: (GPN|GPN-)([0–9])+ Records generated on MICS have negative IDs Note: either a GuidanceGroup or a GuidancePattern reference is required
GuidanceEastShift	C	o	xs:signedLong	-2^{31} to $(2^{31}-1)$	Guidance shift information in mm on the projected tangent plane with point A as reference.
GuidanceNorthShift	D	o	xs:signedLong	-2^{31} to $(2^{31}-1)$	Guidance shift information in mm on the projected tangent plane with point A as reference.
PropagationOffset	E	o	xs:signedLong	-2^{31} to $(2^{31}-1)$	The propagation offset in mm. The propagation offset is perpendicular to the direction of the guidance pattern. A positive value offsets to the right side of the guidance pattern, a negative value offsets to the left side of the guidance pattern.
AllocationStamp		o	xs:element		Includes a single XML element AllocationStamp

EXAMPLE

```
<TSK A=”TSK1” D=”FRM1” G=”1” E=”PFD1”>
   <GAN A=”GGP1”>
      <ASP A=”2003-08-20T08:00:00” D=”1”/>
      <GST  B=”GPN2” C=”100” D=”150””>
         <ASP A=”2003-08-20T08:10:00” D=”1”/>
      </GST>
```

```
    </GAN>
</TSK>
```

D.32 ISO11783_TaskData

Type:

Root element

Description:

The XML element ISO11783_TaskData is the main XML element called a root element and contains definitions about the construct of the XML File (Version number…) and the use of primary XML elements.

Includes XML elements:

— *Task*

— *CodedComment*

— *CodedCommentGroup*

— *ColourLegend*

— *CropType*

— *CulturalPractice*

— *Customer*

— *Farm*

— *Device*

— *OperationTechnique*

— *Partfield*

— *Product*

— *ProductGroup*

— *ValuePresentation*

— *Worker*

— *ExternalFileReference*

— *TaskControllerCapabilities*

Table D.32 — ISO11783_TaskData attributes

Attribute	XML	Use	Type	Length/ range	Comment
VersionMajor	VersionMajor	r	xs:NMTOKEN	0 to 4	List element release number (major), used to specify the version of ISO 11783-10 that this task data file meets: 0 = The version of the DIS (draft International Standard) 1 = The version of the FDIS.1 (final draft International Standard, first edition) 2 = The version of the FDIS.2 and the first edition published as an International Standard 3 = The version of second edition published as a draft International Standard (E2.DIS). 4 = The version of the second edition published as a final draft International Standard (E2.FDIS).
VersionMinor	VersionMinor	r	xs:NMTOKEN	0 to 99	List element version number (minor), used to refer to revisions of the XML schema that this element complies to. See 8.4 for the naming convention of the XML schema.
ManagementSoftwareManufacturer	ManagementSoftwareManufacturer	o/r[a]	xs:string	32	Name of management-software manufacturer
ManagementSoftwareVersion	ManagementSoftwareVersion	o/r[a]	xs:string	32	Version of management software
TaskControllerManufacturer	TaskControllerManufacturer	o/r[a]	xs:string	32	Name of TC manufacturer
TaskControllerVersion	TaskControllerVersion	o/r[a]	xs:string	32	Version of TC software
DataTransferOrigin	DataTransferOrigin	r	xs:NMTOKEN	1	Describes the origin of the XML file: 1 = FMIS 2 = MICS This attribute specifies the system that most recently generated this data transfer file set
DataTransferLanguage[b]	lang	o	xs:language	32	Specifies the language used by the system that generated the data transfer file set

[a] The optional or required status of this attribute depends on the data transfer origin, this attribute is not used when the data transfer origin is FMIS. TC information is required when the data transfer origin is MICS.

[b] This attribute is introduced in ISO 11783-10 version 4.

[c] This element is introduced in ISO 11783-10 version 4.

Table D.32 *(continued)*

Attribute	XML	Use	Type	Length/ range	Comment
AttachedFile	AFE	o	xs:element		Includes a list of XML element AttachedFile
BaseStation	BSN	o	xs:element		Includes a list of XML element BaseStation
CodedComment	CCT	o	xs:element		Includes a list of XML element CodedComment
CodedCommentGroup	CCG	o	xs:element		Includes a list of XML element CodedCommentGroup
ColourLegend	CLD	o	xs:element		Includes a list of XML element ColourLegend
CropType	CTP	o	xs:element		Includes a list of XML element CropType
CulturalPractice	CPC	o	xs:element		Includes a list of XML element CulturalPractice
Customer	CTR	o	xs:element		Includes a list of XML element Customer
Device	DVC	o	xs:element		Includes a list of XML element Device
Farm	FRM	o	xs:element		Includes a list of XML element Farm
OperationTechnique	OTQ	o	xs:element		Includes a list of XML element OperationTechnique
Partfield	PFD	o	xs:element		Includes a list of XML element Partfield
Product	PDT	o	xs:element		Includes a list of XML element Product
ProductGroup	PGP	o	xs:element		Includes a list of XML element ProductGroup
Task	TSK	o	xs:element		Includes a list of XML element Task
TaskControllerCapabilities[c]	TCC	o/r[a]	xs:element		Includes a single XML element TaskControllerCapabilities
ValuePresentation	VPN	o	xs:element		Includes a list of XML element ValuePresentation
Worker	WKR	o	xs:element		Includes a list of XML element Worker
ExternalFileReference	XFR	o	xs:element		Includes a list of XML element ExternalFileReference

[a] The optional or required status of this attribute depends on the data transfer origin, this attribute is not used when the data transfer origin is FMIS. TC information is required when the data transfer origin is MICS.

[b] This attribute is introduced in ISO 11783-10 version 4.

[c] This element is introduced in ISO 11783-10 version 4.

EXAMPLE

```
<ISO11783_TaskData VersionMajor="4" VersionMinor="0"
TaskControllerManufacturer="FarmCtrl" TaskControllerVersion="1.0"
ManagementSoftwareManufacturer="FarmSystem"
ManagementSoftwareVersion="1.0" DataTransferOrigin="1" lang="en-GB">
   <TSK A="TSK1" F="WKR1" G="1">
```

```
        ......
        </TSK>
        ......
    </ISO11783_TaskData>
```

D.33 LineString — LSG

Type:

Task data

Description:

The LineString XML element describes the position, length, and appearance of a line. LineStrings of the type flag can be used to assign a comment to all positions of the lineString. This is to enable the TC to display farm-side-generated comments stored in the XML attribute LineStringDesignator at certain positions as informational messages to the operator.

LineStrings shall be drawn as a series of lines connecting the Points in the order in which they are listed in the LineString XML element. A LineString shall be closed if it is a polygon exterior or interior boundary (types 1 and 2). In this case, the first Point in the LineString shall be repeated as the last Point, in accordance with how a LineString is closed in the corresponding GML object. Note that prior to version 4, for a LineString of type 1 or 2, the first point did not need to be included as last point to draw a closed boundary, the requirements in versions 2 and 3 where that a line was also required to be drawn between the last point and first point in the list of Point XML elements for LineStrings of type 1 and 2.

The LineString ID attribute shall only be used when the LineString XML element is not a child element of a Polygon XML element or a GuidancePattern XML element.

Includes XML element:

— *Point*

Included by XML elements:

— *Partfield*

— *Polygon*

Table D.33 — LineString attributes

Attribute	XML	Use	Type	Length/range	Comment
LineStringType	A	r	xs:NMTOKEN	1 to 9	Type of the LineString, possible values: 1 = PolygonExterior 2 = PolygonInterior 3 = TramLine 4 = SamplingRoute 5 = GuidancePattern 6 = Drainage 7 = Fence 8 = Flag 9 = Obstacle, any kind of obstacle, e.g. used to generate a warning when approaching an obstacle.[a]

[a] This definition is introduced in ISO 11783-10 version 4.

Table D.33 *(continued)*

Attribute	XML	Use	Type	Length/range	Comment
LineStringDesignator	B	o	xs:string	max. 32	LineString name or comment
LineStringWidth	C	o	xs:unsigned-Long	0 to (2^{32}–2)	Width of the LineString in mm. This is the width in real world units, i.e. as measured on a Partfield. For LineStringType 5 (GuidancePattern), the LineStringWidth represents the swathe width that separates the adjacent paths of the guidance pattern.
LineStringLength	D	o	xs:unsigned-Long	0 to (2^{32}–2)	Length of the LineString in mm. This is the length in real world units, i.e. as measured on a Partfield.
LineStringColour	E	o	xs:unsigned-Byte	0 to 254	Colour of the LineString Format: palette like ISO 11783-6
LineStringId [a]	F	o	xs:ID	min.4 to max.14	Unique identifier of LineString Format: (LSG|LSG-)([0–9])+ Records generated on MICS have negative IDs
Point		r	xs:element		Includes a list of XML element Point

[a] This definition is introduced in ISO 11783-10 version 4.

EXAMPLE

```
<LSG A="6" E="1" B="Line1" D="2000" C="20">
   <PNT A="2" C="58.8754321" D="8.945632" F="1" B="start" E="50"/>
   <PNT A="2" C="58.8789999" D="8.99889099" F="1" B="end" E="50"/>
</LSG>
```

D.34 OperationTechnique — OTQ

Type:

Coding data

Description:

The OperationTechnique XML element describes operation techniques like “drilling”, “spreading”, “gaseous”.

Referenced by XML elements:

— *OperTechPractice*

— *OperationTechniqueReference*

Table D.34 — OperationTechnique attributes

Attribute	XML	Use	Type	Length/range	Comment
OperationTechniqueId	A	r	xs:ID	min. 4 to max. 14	Unique identifier of Operation-Technique Format: (OTQ|OTQ-)([0–9])+ Records generated on MICS have negative IDs
OperationTechniqueDesignator	B	r	xs:string	max. 32	Designator of OperationTech-nique

EXAMPLE

```
<OTQ A="OTQ1" B="drilling"/>
<OTQ A="OTQ2" B="spreading"/>
<OTQ A="OTQ3" B="gaseous"/>
```

D.35 OperationTechniqueReference — OTR

Type:

Coding data

Description:

The OperationTechniqueReference XML element contains a reference to a single OperationTechnique.

Included by XML element:

— *CulturalPractice*

References XML element:

— *OperationTechnique*

Table D.35 — OperationTechniqueReference attributes

Attribute	XML	Use	Type	Length/range	Comment
OperationTechnique-IdRef	A	r	xs:IDREF	min. 4 to max. 14	Reference to XML element OperationTech-nique Format: (OTQ|OTQ-)([0–9])+

EXAMPLE

```
<OTR A="OTQ1" />
```

D.36 OperTechPractice — OTP

Type:

Task data

Description:

A task-related OperTechPractice XML element provides an assignment of the combination of a specific operation technique with a single cultural practice.

Included by XML element:

— *Task*

References XML elements:

— *OperationTechnique*

— *CulturalPractice*

Table D.36 — OperTechPractice attributes

Attribute	XML	Use	Type	Length	Comment
CulturalPracticeIdRef	A	r	xs:IDREF	min. 4 to max. 14	Reference to XML element CulturalPractice Format: (CPC|CPC-)([0–9])+
OperationTechnique-IdRef	B	o	xs:IDREF	min. 4 to max. 14	Reference to XML element OperationTech-nique Format: (OTQ|OTQ-)([0–9])+

EXAMPLE

```
<OTP A=”CPC1” B=”OTQ1”/>
```

D.37 Partfield — PFD

Type:

Coding data

Description:

The Partfield XML element describes the pieces of land that a Task can be allocated to. A Partfield is a dynamic object that is created at the moment that it is known that it will be treated as one unit. The Partfield is bare or grown with one CropType. Only the primary CropType is listed in the case of and undersown crop. A Partfield ends when a new crop production unit is started or when it is combined with a neighbouring Partfield. A Partfield can be a field as a whole or be a part of a field. When a Partfield consists of multiple pieces of land, all pieces shall be located near each other, e.g. only separated by relatively narrow strips of land. Each piece of land that belongs to one Partfield is bounded by a single Polygon. Prior to version 4, only a single Polygon representing a boundary could be included in a Partfield. In version 4 and later, multiple Polygons, each representing a bounded area of the Partfield, can be included in a single Partfield.

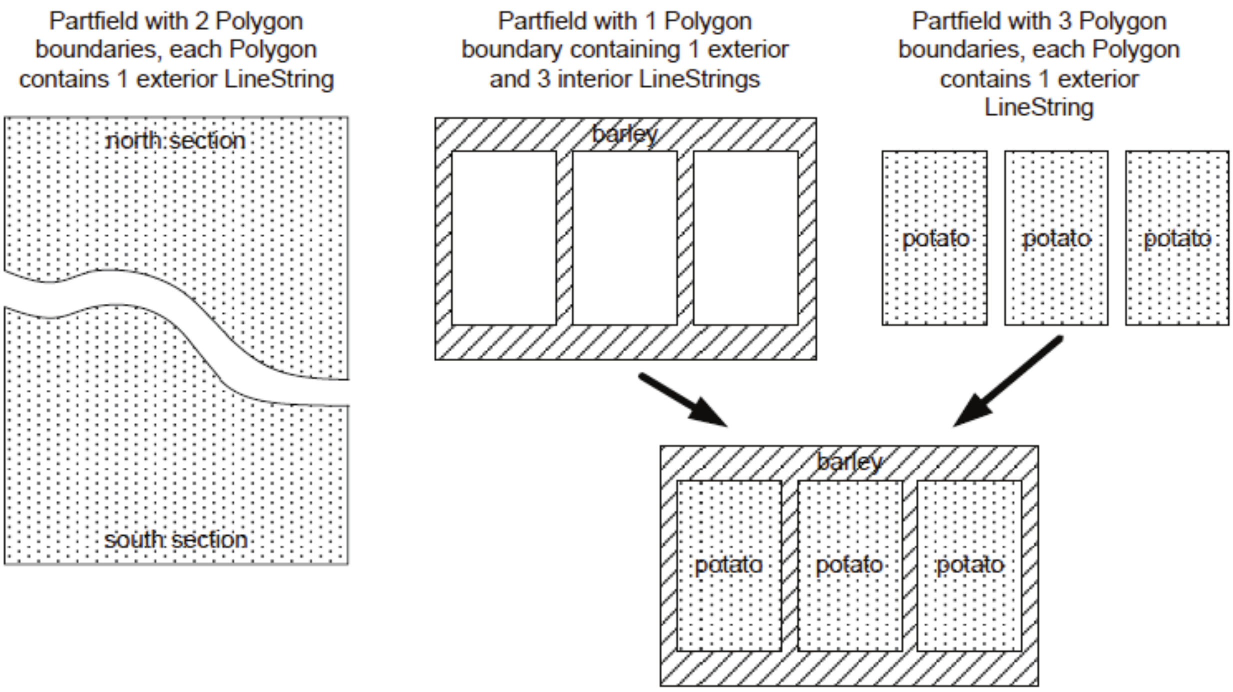


Figure D.4 — Partfield boundary Polygon and LineString examples

Partfield elements can only refer to Polygons, LineStrings, and Points that are not related to TreatmentZones.

Partfields can be child XML elements of other Partfields. The recursion depth of this relation shall be limited to 2 levels. A Partfield can have at maximum one parent Partfield and that parent Partfield cannot be included as a child XML element in another Partfield XML element. An example of the definition of two Partfields that are children of another Partfield is to keep track of the relation between two crop varieties grown in different areas of one field. The parent Partfield lists the crop species while both child Partfields can list the individual varieties.

Referenced by XML element:

— Task

References XML elements:

— CropType

— CropVariety

— Customer

— Farm

Includes XML elements:

— Polygon

— LineString

— Point

— GuidanceGroup

Table D.37 — Partfield attributes

Attribute	XML	Use	Type	Length/range	Comment
PartfieldId	A	r	xs:ID	min. 4 to max. 14	Unique identifier of partfield Format: (PFD|PFD-)([0–9])+ Records generated on MICS have negative IDs
PartfieldCode	B	o	xs:string	max. 32	Partfield number from the FMIS
PartfieldDesignator	C	r	xs:string	max. 32	Partfield designator/name
PartfieldArea	D	r	xs:unsignedLong	0 to (2^{32}–2)	Size of the partfield in m^2
CustomerIdRef	E	o	xs:IDREF	min. 4 to max. 14	Reference to XML element Customer Format: (CTR|CTR-)([0–9])+
FarmIdRef	F	o	xs:IDREF	min. 4 to max. 14	Reference to XML element Farm Format: (FRM|FRM-)([0–9])+
CropTypeIdRef	G	o	xs:IDREF	min. 4 to max. 14	Reference to XML element CropType Format: (CTP|CTP-)([0–9])+
CropVarietyIdRef	H	o	xs:IDREF	min. 4 to max. 14	Reference to XML element CropVariety Format: (CVT|CVT-)([0–9])+
FieldIdRef	I	o	xs:IDREF	min. 4 to max. 14	Unique identifier of Partfield, used as parent Partfield Format: (PFD|PFD-)([0–9])+ Records generated on MICS have negative IDs
Polygon		o	xs:element		Includes a list of XML element Polygon
LineString		o	xs:element		Includes a list of XML element LineString
Point		o	xs:element		Includes a list of XML element Point
GuidanceGroup		o	xs:element		Includes a list of XML element GuidanceGroup

EXAMPLE

```
<PFD A="PFD3" C="hill" D="32000" G="CTP3"/>
```

D.38 Point — PNT

Type:

Task data

Description:

The Point XML element describes the position and appearance of a point location. Points of type flag can be used to assign a comment to the position of the point. This is to enable the TC to display farm-side-generated comments stored in XML attribute PointDesignator at a certain position as informational messages to the operator.

The Point ID attribute shall only be used when the Point XML element is not a child element of a LineString XML element.

Included by XML elements:

— *LineString*

— *Partfield*

Table D.38 — Point attributes

Attribute	XML	Use	Type	Length/range	Comment
PointType	A	r	xs:NMTOKEN	1 to 11	Defines the meaning of this point: 1 = Flag 2 = other 3 = Field Access, entry point of a field[a] 4 = Storage, location where product is stored on a field[a] 5 = Obstacle, any kind of obstacle, e.g. used to generate warnings when approaching a obstacle[a] 6 = Guidance Reference A, start point of a guidance line, e.g the A point in an A-B line[a] 7 = Guidance Reference B, end point of a guidance line, e.g. the B point in an A-B line[a] 8 = Guidance Reference Center, centre point of a guidance pivot pattern[a] 9 = Guidance Point, guidance related point that is neither an A, B or Center reference point[a] 10 = Partfield Reference Point, location in e.g. a Partfield that identifies the field for administrative purposes[a] 11 = Homebase[a]
PointDesignator	B	o	xs:string	max. 32	Name of the point, description or comment
PointNorth	C	r	xs:decimal	−90.0 to 90.0	GPS position north Format: WGS84
PointEast	D	r	xs:decimal	−180.0 to 180.0	GPS position east Format: WGS84
PointUp	E	o	xs:long	$(-2^{31}+1)$ to $(2^{31}-1)$	GPS position altitude in mm Format: WGS84 ellipsoid
PointColour	F	o	xs:unsignedByte	0 to 254	Colour of the point Format: palette like ISO 11783-6

a This definition is introduced in ISO 11783-10 version 4.

Table D.38 *(continued)*

Attribute	XML	Use	Type	Length/range	Comment
PointId[a]	G	o	xs:ID	min.4 to max.14	Unique identifier of point Format: (PNT|PNT-)([0–9])+ Records generated on MICS have negative IDs
PointHorizontalAccuracy[a]	H	o	xs:decimal	0.0 to 65.0	Point horizontal estimated accuracy (RMS error) in m
PointVerticalAccuracy[a]	I	o	xs:decimal	0.0 to 65.0	Point vertical estimated accuracy (RMS error) in m
Filename[a]	J	o	xs:ID	8	Unique name of point binary file Format: PNT[0–9][0–9][0–9][0–9][0–9]
Filelength[a]	K	o	xs:unsignedLong	0 to (2^{32}–2)	Length of point binary file in number of bytes

[a] This definition is introduced in ISO 11783-10 version 4.

EXAMPLE 1 LineString with Points in XML format

```
<LSG A="6" E="1" B="Line1" D="2000" C="20">
   <PNT A="2" C="58.8754321" D="8.945632" F="1" B="start" E="50"/>
   <PNT A="2" C="58.8789999" D="8.99889099" F="1" B="end" E="50"/>
</LSG>
```

EXAMPLE 2 LineString with Points in binary format

```
<LSG A="6" E="1" B="Line1" D="2000" C="20">
    <PNT A="2" C="" D="" E="" F="1" J="PNT00001" K="40"/>
</LSG>
```

D.39 Polygon — PLN

Type:

Task data

Description:

The Polygon XML element describes an area by inclusion of LineStrings. A Polygon can be used to specify the boundary of a partfield or the area of a TreatmentZone.

Prior to version 4, multiple outer boundary LineStrings could be included in one Polygon. In version 4 and later, a Polygon shall only describe a single surface, which corresponds to the definition of a polygon in GML.

Polygons of type flag can be used to assign a comment to all positions inside the Polygon. This is to enable the TC to display farm-side-generated comments stored in XML attribute PolygonDesignator at certain positions as informational messages to the operator.

Included by XML elements:

— *Partfield*

— *TreatmentZone*

Includes XML element:

— *LineString*

Table D.39 — Polygon attributes

Attribute	XML	Use	Type	Length/range	Comment
PolygonType	A	r	xs:NMTOKEN	1 to 12	Type of the polygon, possible values: 1 = Partfield Boundary 2 = TreatmentZone 3 = WaterSurface 4 = Building 5 = Road 6 = Obstacle 7 = Flag 8 = Other 9 = Mainfield, main area of the field used for cultivations[a] 10 = Headland, area of the field used for turning, usually planted in another direction than the main field area. Definition and example of main field area is Partfield boundary minus obstacles, water surface, headlands[a] 11 = BufferZone, area of the field that has to be treated differently due to environmental regulations[a] 12 = Windbreak, surrounding of a field with the purpose to reduce wind on the field[a]
PolygonDesignator	B	o	xs:string	max. 32	Polygon designator/name
PolygonArea	C	o	xs:unsigned-Long	0 to (2^{32}–2)	Polygon area in m^2
PolygonColour	D	o	xs:unsigned-Byte	0 to 254	Colour of the polygon Format: palette like ISO 11783-6
PolygonId [a]	E	o	xs:ID	min.4 to max.14	Unique identifier of polygon Format: (PLN|PLN-)([0–9])+ Records generated on MICS have negative IDs

[a] This definition is introduced in ISO 11783-10 version 4.

Table D.39 *(continued)*

Attribute	XML	Use	Type	Length/range	Comment
LineString		r	xs:element		Includes a list of XML element LineString. At a maximum one LineString of type outer boundary may be included. Multiple LineStrings of type inner boundary may be included where each inner boundary shall not cross the outer boundary or another inner boundary. Inner boundaries may touch an outer boundary or inner boundary.

[a] This definition is introduced in ISO 11783-10 version 4.

EXAMPLE

```
<PLN A="1" B="Field Boundary" C="32000" D="1">
   <LSG A="1" B="Line1" C="20" D="280000" E="1" >
      <PNT A="2" B="start" C="40.84982" D="-96.596045" E="150234" F="1" />
      <PNT A="2" B="mid" C="40.84982" D="-96.592655" E="148987" F="1" />
      <PNT A="2" B="end" C="40.846573" D="-96.592526" E="148284" F="1" />
      <PNT A="2" B="start" C="40.84982" D="-96.596045" E="150234" F="1" />
   </LSG>
</PLN>
```

D.40 Position — PTN

Type:

Task data

Description:

The Position XML element describes a measured position. The position is part of an AllocationStamp or Time specification. With the latter, it can be used to, for example, log DataLogValues together with a position.

Included by XML elements:

— *AllocationStamp*

— *Time*

Table D.40 — Position attributes

Attribute	XML	Use	Type	Length/ range	Comment
PositionNorth	A	r	xs:decimal	−90.0 to 90.0	GPS position north WGS84
PositionEast	B	r	xs:decimal	−180.0 to 180.0	GPS position east WGS84
PositionUp	C	o	xs:long	-2^{31} to $(2^{31}-1)$	GPS position altitude WGS84 ellipsoid in mm

Table D.40 *(continued)*

Attribute	XML	Use	Type	Length/ range	Comment
PositionStatus	D	r	xs:NMTOKEN	0 to 15	Position status. Definition references NMEA2000 MethodGNSS parameter. 0 = no GPS fix 1 = GNSS fix 2 = DGNSS fix 3 = Precise GNSS, no deliberate degradation (such as SA), and higher resolution code (P-code) and 2 frequencies are used to correct atmospheric delays 4 = RTK fixed integer 5 = RTK float 6 = Est(DR)mode 7 = Manual input 8 = Simulate mode 9–13 = Reserved 14 = Error 15 = PositionStatus value is not available
PDOP	E	o	xs:decimal	0.0 to 99.9	PDOP quality information
HDOP	F	o	xs:decimal	0.0 to 99.9	HDOP quality information
NumberOfSatellites	G	o	xs:unsigned-Byte	0 to 254	Number of used satellites
GpsUtcTime	H	o	xs:unsigned-Long	0 to (2^{32}–2)	UTC time in milliseconds since midnight
GpsUtcDate	I	o	xs:unsigned-Short	0 to (2^{16}–2)	UTC date in days relative to 1980–01–01

EXAMPLE

```
<PTN A=”54.588945” B=”9.989209” D=”3”/>
```

D.41 ProcessDataVariable — PDV

Type:

Task data

Description:

The ProcessDataVariable XML element is included in a TreatmentZone. It contains the ProcessDataDDI accompanied by the value for that ProcessDataDDI and optional product and device element information to be used in this TreatmentZone. A ProcessDataVariable can contain another ProcessDataVariable to describe the planned allocation of multiple products to a single DeviceElement. In that case, the “parent” ProcessDataVariable specifies the ProcessDataDDI to be sent by the TaskController to a DeviceElement while the “child” ProcessDataVariables contain the product specifications. The “child” ProcessDataVariable(s) shall not contain other ProcessDataVariables and shall not specify a DeviceElementIdRef attribute.

The optional attributes ActualCulturalPracticeValue and ElementTypeInstanceValue have been introduced to enable planning and tracking of the assignment of process data values to specific operations or specific element type instances. This enables distinguishing sets of ProcessDataVariables when multiple operations or multiple instances of the same device element type are present in a single Task. An example of this is simultaneously controlling the application rates of a seeding operation and a fertilizing operation within one Task. In this case, to separate the seeding operation application rate process data from the fertilizing operation application rate process data, each set of application rate process data setpoints contains its own ActualCulturalPractice value. An example of using the ElementTypeInstanceValue attribute is to separate sets of process data variables intended to control application rates with the same DDIs but going to separate rate control device elements. The use of the ElementTypeInstanceValue attributes enables grouping these process data by rate controller instance without requiring these to be pre-assigned to an actual device element. Prior to ISO 11783-10 version 4, this grouping of process data could only be achieved by a planned allocation to an actual device element. The summary for the grouping of process data in different ISO 11783-10 versions is:

ISO 11783-10 version 3 and prior:	1) Use the ProductIdRef attribute to group prescription rates by product. This requires the actual product to be present in the data transfer file set. 2) Use the DeviceElementIdRef attribute to group prescription values by an element of a machine. This requires the actual machine device descriptor to be present in the data transfer file set.
ISO 11783-10 version 4 and later:	1) Use the ProductID reference to group prescription rates by product in case the product is known and present in the data transfer file set. 2) Use the ActualCulturalPracticeValue attribute to group prescription values by the cultural practice(s) that the task is specified for. Device elements may contain an ActualCulturalPractice DDI value that enables matching the operation to the correct device element in case tasks or devices are capable to execute multiple operations. 3) Use the ElementTypeInstanceValue attribute to group prescription values with the same DDI and ActualCulturalPracticeValue by their planned control device element instance on a device. Device elements may contain an ElementTypeInstance DDI value that enables automatic matching of each group of process data variables to the correct device element on a device. 4) During the execution of a Task, once an actual machine is selected, the process data variables DeviceElementIdRef attribute may be updated to store the allocation of process data to device elements.

The DeviceElementIdRef attribute in the ProcessDataVariable XML element is an optional attribute. Task controller implementations are required to be able to assign the ProcessDataVariable to a connected client that has a device descriptor with a matching ProcessDataDDI that is able to accept this ProcessDataVariable as a setpoint. The grouping of ProcessDataVariables by ProductIdRef, ActualCulturalPracticeValue, and ElementTypeInstanceId assist this ProcessDataVariable to controllable DeviceElement assignment process.

Included by XML element:

— *TreatmentZone*

References XML elements:

— *Product*

— *DeviceElement*

— *ValuePresentation*

Includes XML element:

— *ProcessDataVariable*

Table D.41 — ProcessDataVariable attributes

Attribute	XML	Use	Type	Length/range	Comment
ProcessDataDDI	A	r	xs:hexBinary	0000_{16} to $FFFF_{16}$	Unique number which defines the process data DDI (specified in ISO 11783-11)
ProcessDataValue	B	r	xs:long	-2^{31} to $(2^{31}-1)$	Contains the value of the process data DDI
ProductIdRef	C	o	xs:IDREF	min. 4 to max. 14	Reference to XML element Product Format: (PDT|PDT-)([0–9])
DeviceElementIdRef	D	o	xs:IDREF	min. 4 to max. 14	Reference to XML element DeviceElement Format: (DET|DET-)([0–9])
ValuePresentationIdRef	E	o	xs:IDREF	min. 4 to max. 14	Reference to XML element ValuePresentation Format: (VPN|VPN-)([0–9])
ActualCulturalPracticeValue[a]	F	o	xs:long	-2^{31} to $(2^{31}-1)$	Contains the value of the ActualCulturalPractice DDI that this process data are planned for or assigned to.
ElementTypeInstanceValue[a]	G	o	xs:long	-2^{31} to $(2^{31}-1)$	Contains the value of the ElementTypeInstance DDI that this process data are planned for or assigned to.
ProcessDataVariable		o	xs:element		Includes a list of XML element ProcessDataVariable

[a] This attribute is introduced in ISO 11783-10 version 4.

EXAMPLE 1 ProcessDataVariable referencing a Product and a DeviceElement.

A Setpoint Mass Per Area Application Rate (DDI 6) with a value of 15000 mg/m^2 (150 kg/ha) of the product identified by PDT1 is specified for a device element identified by DET2:

```
<PDT A="PDT1" B="Granular Fertilizer"/>
<VPN A="VPN1" B="0" C="0.01" D="0" E="kg/ha"/>
<TSK A="TSK1" G="1" H="0">
   <TZN A="0" B="Default Treatment Zone">
      <PDV A="0006" B="15000" C="PDT1" D="DET2" E="VPN1"/>
   </TZN>
</TSK>
```

EXAMPLE 2 ProcessDataVariables with ActualCulturalPracticeValue attributes specified to distinguish between two operations, position-based control specified as polygon geometries.

Treatment zone, multiple process data variables, enable grouping by ActualCulturalPractice value:

```
<PDT A="PDT1" B="Granular Fertilizer"/>
<PDT A="PDT2" B="Sunflower"/>
```



```
<VPN A="VPN1" B="0" C="0.01" D="0" E="kg/ha"/>
<TSK A="TSK1" G="1" H="0">

   <TZN A="1" B="Layer 1, Rate 1" C="2">
      <!- fertilizer operation low rate ->
      <PDV A="0006" B="5000" C="PDT1" E="VPN1" F="1"/>
      … list of polygons to specify the areas for this treatmentzone …
   </TZN>
   <TZN A="2" B="Layer 1, Rate 2" C="3">
      <!- fertilizer operation high rate ->
      <PDV A="0006" B="6500" C="PDT1" E="VPN1" F="1"/>
      … list of polygons to specify the areas for this treatmentzone …
   </TZN>
   <TZN A="3" B="Layer 2, Rate 1" C="9">
      <!- planting operation low rate ->
      <PDV A="0006" B="800" C="PDT2" E="VPN1" F="2"/>
      … list of polygons to specify the areas for this treatmentzone …
   </TZN>
   <TZN A="4" B="Layer 2, Rate 2" C="10">
      <!- planting operation high rate ->
      <PDV A="0006" B="1000" C="PDT2" E="VPN1" F="2"/>
      … list of polygons to specify the areas for this treatmentzone …
   </TZN>
</TSK>
```

EXAMPLE 3 ProcessDataVariables for a planter device equipped with two fertilizer bins and one seed bin, distinguish setpoint process data variables for 3 mass/area application rates. The ProcessDataVariables have the ActualCulturalPracticeValue and ElementTypeInstanceValue attributes specified to distinguish between two operations and between two product bins for one of the operations. Position-based control is specified as polygon geometries.

Treatment zones with polygons, multiple process data variables, enable grouping by ActualCulturalPractice and by ElementTypeInstance values:

```
<PDT A="PDT1" B="Granular N Fertilizer"/>
<PDT A="PDT2" B="Granular P Fertilizer"/>
<PDT A="PDT3" B="Sunflower"/>
<VPN A="VPN1" B="0" C="0.01" D="0" E="kg/ha"/>
<TSK A="TSK1" G="1" H="0">
   <TZN A="1" B="Layer 1, Rate 1" C="2">
      <!- N fertilizer operation low rate ->
      <PDV A="0006" B="5000" C="PDT1" E="VPN1" F="1" G="0"/>
      … list of polygons to specify the areas for this treatmentzone …
   </TZN>
   <TZN A="2" B="Layer 1, Rate 2" C="3">
      <!- N fertilizer operation high rate ->
      <PDV A="0006" B="6500" C="PDT1" E="VPN1" F="1" G="0"/>
      … list of polygons to specify the areas for this treatmentzone …
   </TZN>
   <TZN A="3" B="Layer 2, Rate 1" C="12">
      <!- P fertilizer operation low rate ->
      <PDV A="0006" B="450" C="PDT2" E="VPN1" F="1" G="1"/>
      … list of polygons to specify the areas for this treatmentzone …
   </TZN>
   <TZN A="4" B="Layer 2, Rate 2" C="13">
      <!- P fertilizer operation high rate ->
      <PDV A="0006" B="550" C="PDT2" E="VPN1" F="1" G="1"/>
      … list of polygons to specify the areas for this treatmentzone …
   </TZN>
   <TZN A="5" B="Layer 3, Rate 1" C="9">
      <!- planting operation low rate ->
      <PDV A="0006" B="800" C="PDT3" E="VPN1" F="2"/>
      … list of polygons to specify the areas for this treatmentzone …
   </TZN>
   <TZN A="6" B="Layer 3, Rate 2" C="10">
      <!- planting operation high rate ->
      <PDV A="0006" B="1000" C="PDT3" E="VPN1" F="2"/>
      … list of polygons to specify the areas for this treatmentzone …
```



```
      </TZN>
</TSK>
```

EXAMPLE 4 ProcessDataVariables for a planter device equipped with two fertilizer bins and one seed bin, distinguish setpoint process data variables for 3 mass/area application rates. The ProcessDataVariables have the ActualCulturalPracticeValue and ElementTypeInstanceValue attributes specified to distinguish between two operations and between two product bins for one of the operations. Position-based control is specified on a grid type 2 geometry.

Grid type 2 application with a treatment zone template for the grid cell values, multiple process data variables, enable grouping by ActualCulturalPractice and by ElementTypeInstance values:

```
   <PDT A=”PDT1” B=”Granular N Fertilizer”/>
   <PDT A=”PDT2” B=”Granular P Fertilizer”/>
   <PDT A=”PDT3” B=”Sunflower”/>
   <VPN A=”VPN1” B=”0” C=”0.01” D=”0” E=”kg/ha”/>
   <TSK A=”TSK1” G=”1” H=”0”>
      <TZN A=”1” B=”Layer 1, Rate 1” C=”2”>
         <!- N fertilizer operation low rate ->
         <PDV A=”0006” B=”” C=”PDT1” E=”VPN1” F=”1” G=”0”/>
         <!- P fertilizer operation low rate ->
         <PDV A=”0006” B=”” C=”PDT2” E=”VPN1” F=”1” G=”1”/>
         <!- planting operation low rate ->
         <PDV A=”0006” B=”” C=”PDT3” E=”VPN1” F=”2”/>
      </TZN>
      <GRD A=”51.812025” B=”4.284920” C=”1.5E-3” D=”1.4E-3” E=”200” F=”300” G=”GRD00001”
I=”2” J=”1”/>
   </TSK>
```

EXAMPLE 5 ProcessDataVariables for a planter device equipped with two bins, capable to apply two different varieties of planting material at the same time. Distinguish the setpoint process data for two planting populations of two varieties. The ProcessDataVariables have the ElementTypeInstanceValue attribute specified to distinguish between two product bins for two varieties.

Treatment zone with polygons, multiple process data variables, enable grouping by ElementTypeInstance values:

```
   <PGP A=”PGP1” B=”Soy Bean” C=”2”/>
   <PDT A=”PDT1” B=”Glenn”/>
   <PDT A=”PDT2” B=”Stoddard”/>
   <VPN A=”VPN1” B=”0” C=”0.01” D=”0” E=”kseeds/ha”/>
   <TSK A=”TSK1” G=”1” H=”0”>
      <TZN A=”1” B=”Layer 1, Rate 1” C=”2”>
         <!- variety 1 low rate ->
         <PDV A=”000B” B=”25000” C=”PDT1” E=”VPN1” G=”0”/>
         … list of polygons to specify the areas for this treatmentzone …
      </TZN>
      <TZN A=”2” B=”Layer 1, Rate 2” C=”3”>
         <!- variety 1 high rate ->
         <PDV A=”000B” B=”29000” C=”PDT1” E=”VPN1” G=”0”/>
         … list of polygons to specify the areas for this treatmentzone …
      </TZN>
      <TZN A=”3” B=”Layer 2, Rate 1” C=”12”>
         <!- variety 2 low rate ->
         <PDV A=”000B” B=”23000” C=”PDT2” E=”VPN1” G=”1”/>
         … list of polygons to specify the areas for this treatmentzone …
      </TZN>
      <TZN A=”4” B=”Layer 2, Rate 2” C=”13”>
         <!- variety 2 high rate ->
         <PDV A=”000B” B=”27000” C=”PDT2” E=”VPN1” G=”1”/>
         … list of polygons to specify the areas for this treatmentzone …
      </TZN>
   </TSK>
```

D.42 Product — PDT

Type:

Coding data

Description:

The Product XML element describes a single product, a product mixture, produce of a crop variety, or planting material of a crop variety. A product can be related to a ProductGroup. Product XML elements that define produce or planting material of crop varieties are grouped into ProductGroup XML elements of type "CropType".

To determine which products belong to a specific ProductGroup, the ProductGroupIdRef values of all products shall be examined for a match with a particular ProductGroupId value. Products refer both to products being used in a task (i.e. crop protection chemicals) and to products being produced by a task (agricultural products).

The ValuePresentationIdRef and QuantityDDI can be used to specify the presentation and definition of the quantity of the product. The following DDIs shall be used for QuantityDDI:

72 Actual Volume Content (ml)

75 Actual Mass Content (g)

78 Actual Count Content (count)

These QuantityDDI units can also be used to select the setpoint application rate DDI unit that is required to control the application of a product.

The ProductType is used to set the type of the product definition, single or mixture. In case of a product mixture, the MixtureRecipeQuantity is used to define the entire quantity of the recipe of the mixture. If the ProductType is not defined, the default type single shall be used.

The MixtureRecipeQuantity does not specify the quantity for a specific task but is the quantity the mixture recipe totals to. It shall be equal to the resultant quantity of all ingredients. For example, a typical MixtureRecipeQuantity would be 1000 in case the ingredients quantities are specified in 1/1000 value increments.

The embedded ProductRelations are used to define the ingredients of a product mixture. For a product of type "single", ProductRelations shall not be used. ProductRelations(s) shall not contain references to products of type "mixture".

If a Product of type "single" is referenced in an XML element ProductRelation, then the QuantityDDI value shall be specified in that Product. If the referenced Product is of type "mixture", then that Product shall contain values for QuantityDDI and MixtureRecipeQuantity.

A product mixture that is intended to be reusable and selectable in a list of products either in the FMIS or in the MICS will have a ProductType of "Mixture". In some cases, a product mixture is created in the field with no intention of reusing the mixture later. This type of mixture needs to be documented but shall not be able to be selected from a list of products in the FMIS or MICS. In this case, the mixture shall get the ProductType "TemporaryMixture" value.

References XML element:

— *ProductGroup*

— *ValuePresentation*

Referenced by XML element:

— *ProductAllocation*

— *ProcessDataVariable*

— *ProductRelation*

Includes XML element:

— *ProductRelation*

Table D.42 — Product attributes

Attribute	XML	Use	Type	Length/range	Comment
ProductId	A	r	xs:ID	min. 4 to max. 14	Unique identifier of product Format: (PDT|PDT-)([0–9])+ Records generated on MICS have negative IDs
ProductDesignator	B	r	xs:string	max. 32	Product designator/name
ProductGroupIdRef	C	o	xs:IDREF	min. 4 to max. 14	Reference to XML element Product-Group Format: (PGP|PGP-)([0–9])+
ValuePresentationIdRef	D	o	xs:IDREF	min. 4 to max. 14	Reference to XML element ValueP-resentation Format: (VPN|VPN-)([0–9])+
QuantityDDI	E	o	xs:hexBinary	0000_{16} to $FFFF_{16}$	Unique number, which defines the DDI (specified in ISO 11783-11) of the quantity
ProductType[a]	F	o	xs:NMTOKEN	1 to 3	Type of product definition 1 = Single (default) 2 = Mixture 3 = TemporaryMixture
MixtureRecipeQuantity[a]	G	o	xs:long	0 to $(2^{31}–1)$	Contains the resultant quantity of the mixtures' recipe components
DensityMassPerVolume[a]	H	o	xs:long	0 to $(2^{31}–1)$	Product density in mg/l, corre-sponds to DDI 121
DensityMassPerCount[a]	I	o	xs:long	0 to $(2^{31}–1)$	Product density in mg/1000, corre-sponds to DDI 122
DensityVolumePer-Count[a]	J	o	xs:long	0 to $(2^{31}–1)$	Product density in ml/1000, corre-sponds to DDI 123
ProductRelation[a]		o	xs:element		Includes a list of XML element Pro-ductRelation in case the Product-Type is set to 2 or 3.

[a] This attribute is introduced in ISO 11783-10 version 4.

EXAMPLE 1 Product definition without product type or quantity information

```
<PDT A="PDT1" B="agent 1"/>
<PDT A="PDT2" B="agent 2"/>
```

EXAMPLE 2 Product definition providing product type and quantity tracking information

```
<PDT A="PDT3" B="agent 3" D="VPN100" E="0048" F="1"/>
```

EXAMPLE 3 Product mix definition with three constituents, two liquids (quantity is measured as a volume), and one solid (quantity is measured as mass), with the assumption that the solid constituent dissolves completely into the liquids

```
<!- Liquid mixture product definition ->
<VPN A="VPN1" B="0" C="0.001" D="3" E="l"/>
```



```
<VPN A="VPN2" B="0" C="0.001" D="3" E="kg"/>
<!- Define liquid product 10, QuantityDDI 0048₁₆ (72) = Actual   VolumeContent ->
<PDT A="PDT10" B="product 10" D="VPN1" E="0048"/>
<!- Define liquid product 20, QuantityDDI 0048₁₆ (72) = Actual VolumeContent ->
<PDT A="PDT20" B="product 20" D="VPN1" E="0048" F="1"/>
<!- Define solid product 24, QuantityDDI 004B₁₆ (75) = Actual Mass Content ->
<PDT A="PDT24" B="product 24" D="VPN2" E="004B" F="1"/>
<!- Define liquid mixture 12, QuantityDDI 0048₁₆ (72) = Actual Volume Content ->
<PDT A="PDT250" B="mixture 12" D="VPN1" E="0048" F="2" G="100000">
    <!- To get 100 l of mixture 12, one needs: ->
    <!- 80 l of product 10 ->
    <PRN A="PDT10" B="80000"/>
    <!- 20 l of product 20 ->
    <PRN A="PDT20" B="20000"/>
    <!- 0.5 kg of product 24, which dissolves completely in the 100 l mixture ->
    <PRN A="PDT24" B="500"/>
</PDT>
```

When during a task a total of 7600 l of mixture 12 was applied and the FMIS needs the quantities of the mixture ingredient products 10, 20 and 24, the FMIS shall calculate these as follows:

f = 7600 l / 100 l = 76   // The total quantity applied corresponds to 76 times the recipe.

Total volume applied of product 10 is f * PRN.B = 76 * 80,000 ml = 6,080,000 ml = 6,080 l

Total volume applied of product 20 is f * PRN.B = 76 * 20,000 ml = 1,520,000 ml = 1,520 l

Total mass applied of product 24 is f * PRN.B = 76 * 500 mg = 38,000 mg = 38 kg

EXAMPLE 4 Product mix definition with two constituents, one liquid (quantity is measured as a volume), and one solid (quantity is measured as mass). In this case, the dry chemical slightly adds to the volume of the resulting liquid.

```
<!- Liquid mixture product definition ->
<VPN A="VPN1" B="0" C="0.001" D="3" E="l"/>
<VPN A="VPN2" B="0" C="0.001" D="3" E="kg"/>
<!- Define water (liquid), QuantityDDI 0048₁₆ (72) = Actual VolumeContent ->
<PDT A="PDT60" B="water" D="VPN1" E="0048" F="1"/>
<!- Define solid product 70, QuantityDDI 004B₁₆ (75) = Actual Mass Content ->
<PDT A="PDT70" B="product 70" D="VPN2" E="004B" F="1"/>
<!- Define liquid mixture, QuantityDDI 004816 (72) = Actual Volume Content ->
<PDT A="PDT251" B="mixture 13" D="VPN1" E="0048" F="2" G="500000">
    <!- To get 500 l of mixture 13, one needs: ->
    <!- 498 l of water ->
    <PRN A="PDT60" B="498000"/>
    <!- 22.5 kg of product 70, which increases the volume by 2 l ->
    <PRN A="PDT70" B="22500"/>
</PDT>
```

When during a task a total of 18200 l of mixture 13 was applied and the FMIS needs the quantities of the mixture ingredients water (PDT60) and the dry chemical product 70 (PDT70), the FMIS shall calculate these as follows:

f = 18200 l / 500 l = 36,4   // The total quantity applied corresponds to 36.4 times the recipe.

Total volume applied of PDT60 (water) is f * PRN.B = 36,4 * 498,000 ml = 18,127,200 ml = 18,127 l

Total mass applied of PDT70 (product 70) is f * PRN.B = 36,4 * 22 500 mg = 819,000 mg = 819 kg

EXAMPLE 5 Product mix definition with three constituents, one liquid (quantity is measured as a volume), and two solids (quantity is measured as mass), a resultant density of the product mix is specified as an optional attribute.

```
<!- Solid mixture product definition ->
<VPN A="VPN1" B="0" C="0.001" D="0" E="l"/>
<VPN A="VPN2" B="0" C="0.001" D="0" E="kg"/>
```

```
<!- Define liquid product 10, QuantityDDI 0048₁₆ (72) = Actual VolumeContent ->
<PDT A="PDT10" B="product 10" D="VPN1" E="0048" F="1" H="910000"/>
<!- Define solid product 23, QuantityDDI 004B₁₆ (75) = Actual Mass Content ->
<PDT A="PDT23" B="product 23" D="VPN2" E="004B" F="1"/>
<!- Define solid product 24, QuantityDDI 004B₁₆ (75) = Actual Mass Content ->
<PDT A="PDT24" B="product 24" D="VPN2" E="004B" F="1"/>
<!- Define solid mixture 7, QuantityDDI 004B₁₆ (75) = Actual Mass Content ->
<PDT A="PDT275" B="mixture 7" D="VPN2" E="004B" F="2" G="375000" H="990000">
    <!- To get 375 kg of mixture 7, one needs: ->
    <!- 250 kg of product 23 ->
    <PRN A="PDT23" B="250000"/>
    <!- 120 kg of product 24 ->
    <PRN A="PDT24" B="120000"/>
    <!- 5.5 l of product 10 ->
    <PRN A="PDT10" B="5500"/>
    <!- The 5.5 l of product 10 add 5 kg of mass to the mixture. ->
    <!- The density of the liquid product 10 is 0.91 kg/l. Since information ->
    <!- on density is not required, it's up to the operator or FMIS to ->
    <!- calculate the correct MixtureQuantity and MixtureDensity ->
</PDT>
```

When during a task a total of 12,562.5 kg of mixture 7 was applied and the FMIS needs the quantities of the mixture ingredient products 23, 24, and 10, the FMIS shall calculate these as follows:

f = 12562.5 kg / 375 kg = 33.5 // The total quantity applied corresponds to 33.5 times the recipe.

Total mass applied of product 23 is f * PRN.B = 33.5 * 250,000 mg = 8,375,000 mg = 8,375 kg

Total mass applied of product 24 is f * PRN.B = 33.5 *120,000 mg = 4,020,000 mg = 4,020 kg

Total volume applied of product 10 is f * PRN.B = 33.5 * 5,500 ml = 184,250 ml = 184.25 l

EXAMPLE 6 Product density attribute is used to convert between a task setpoint unit and a device metering unit (e.g. mass/area setpoint to a volumetric metering unit or volume/area to a pressure based metering unit)

```
  <VPN A="VPN1" B="0" C="0.001" D="0" E="l"/>
  <!- Define liquid product 10, QuantityDDI 0048₁₆ (72) = Actual VolumeContent ->
  < !- Product density is 0.91 kg/l - >
  <PDT A="PDT10" B="product 10" D="VPN1" E="0048" F="1" H="910000"/>
  <!- Device accepts a volume based setpoint rate and a density set command ->
  <DVC A="DVC1" B="Sprayer" C="1.0" D="A00C84000B20408B" F="30353035344042"
G="FF000000006C6E">
     <DET A="DET1" B="1" C="1" D="all elements" E="0" F="0">
        <DOR A="3"/>
     </DET>
     <DET A="DET2" B="2" C="3" D="main tank" E="1" F="1">
        <DOR A="3"/>
        <DOR A="4"/>
        <DOR A="5"/>
        <DOR A="6"/>
        <DOR A="7"/>
        <DOR A="8"/>
     </DET>
     <DPD A="3" B="008D" C="1" D="9" E="WorkSwitch"/>
     <DPD A="4" B="009E" C="2" D="8" E="Prescription Control State"/>
     <DPD A="5" B="0043" C="1" D="9" E="Actual Working Width"/>
     <DPD A="6" B="0001" C="2" D="8" E="Setpoint Rate"/>
     <DPD A="7" B="0002" C="1" D="1" E="Actual Rate"/>
     <DPD A="8" B="0079" C="2" D="8" E="Product Density"/>
  </DVC>
  <!- Task with a Product with product density definition ->
  < TSK A = "TSK1" G = "1" H = "1" >
     <TZN A="1" B="Default Rate" C="2">
        <PDV A="0001" B="15000" C="PDT10" D="DET2"/>
     </TZN>
  < /TSK >
```


D.43 ProductAllocation — PAN

Type:

Task data

Description:

The ProductAllocation XML element specifies the allocation of a single product to a task. A ProductAllocation is optionally related to a DeviceElement. This is to be able to track the assignment of a single product or of a number of different products to a bin or to another type of DeviceElement. The transferred quantity can optionally be specified by a combination of the attributes QuantityDDI and QuantityValue. The following DDIs shall be used for QuantityDDI:

72 Actual Volume Content (ml)

75 Actual Mass Content (g)

78 Actual Count Content (count)

If totals are used to define a product allocation then the TC shall calculate the correct values for the actual filling, emptying and remainder QuantityValue(s). The “filling” and “emptying” ProductAllocation types use transferred quantities (quantity going in or out of a bin), the “remainder” ProductAllocation type uses absolute quantity (quantity in bin).

Includes XML element:

— *AllocationStamp*

Included by XML element:

— *Task*

References XML elements:

— *DeviceElement*

— *ValuePresentation*

— *Product*

Table D.43 — ProductAllocation attributes

Attribute	XML	Use	Type	Length/range	Comment
ProductIdRef	A	r	xs:IDREF	min. 4 to max. 14	Reference to XML element Product Format: (PDT|PDT-)([0–9])+
QuantityDDI	B	o	xs:hexBinary	0000_{16} to $FFFF_{16}$	Unique number which defines the quantity (specified in ISO 11783-11)
QuantityValue	C	o	xs:long	0 to (2^{31}–1)	Value for the quantity of the product transferred (for filling or emptying) or the quantity of the product in a bin (for remainder)
TransferMode	D	o	xs:NMTOKEN	1 to 3	Type of transfer: 1 = Filling 2 = Emptying 3 = Remainder[a]

[a] This token is introduced in ISO 11783-10 version 4.

Table D.43 *(continued)*

Attribute	XML	Use	Type	Length/range	Comment
DeviceElementIdRef	E	o	xs:IDREF	min. 4 to max. 14	Reference to XML element DeviceElement Format: (DET|DET-)([0–9])+
ValuePresentationIdRef	F	o	xs:IDREF	min. 4 to max. 14	Reference to ValuePresentation Format: (VPN|VPN-)([0–9])+
AllocationStamp		o	xs:element		Includes a single XML element AllocationStamp
[a] This token is introduced in ISO 11783-10 version 4.					

EXAMPLE 1 Mass-based product allocation

```
<PAN A="PDT2" B="004B" C="20000" D="1" E="DET3" F="VPN1">
   <ASP A="2003-11-12T08:00:00" D="4"/>
</PAN>
<PDT A="PDT2" B="Agent"/>
<VPN A="VPN1" B="0.001" C="1.0" D="0" E="kg"/>
```

EXAMPLE 2 Liquid-based product mixture

```
<PDT A="PDT250" B="liquid_mix_12" D="VPN1" E="0048" F="2" G="505">
   <PRN A="PDT1" B="400"/>
   <PRN A="PDT2" B="100"/>
   <PRN A="PDT24" B="5"/>
</PDT>
```

EXAMPLE 3 Complete task cycle

```
<TSK A="TSK11" B="filling example" D="FRM1" E="PFD1" G="1" H="1" F="WKR1">

   <!- Planning ->
   <PAN A="PDT250" B="0048" C="50000" D="1" E="DET3" F="VPN1">
     <ASP A="2010-02-05T13:00:00" D="1"/>
   </PAN>

   <!— On start of the task the tank isn't empty ->
   <PAN A="PDT250" B="0048" C="2000" D="3" E="DET3" F="VPN1">
     <ASP A="2010-02-05T13:38:00" D="4"/>
   </PAN>

   <!- Filling ->
   <PAN A="PDT250" B="0048" C="50000" D="1" E="DET3" F="VPN1">
     <ASP A="2010-02-05T13:40:00" D="4"/>
   </PAN>

   <!- Tank remainder ->
   <PAN A="PDT250" B="0048" C="10000" D="3" E="DET3" F="VPN1">
     <ASP A="2010-02-05T18:50:00" D="4"/>
   </PAN>

   <!- Refilling ->
   <PAN A="PDT250" B="0048" C="50000" D="1" E="DET3" F="VPN1">
     <ASP A="2010-02-05T19:00:00" D="4"/>
   </PAN>

   <!- Product change  tank remainder ->
   <PAN A="PDT250" B="0048" C="0" D="3" E="DET3" F="VPN1">
     <ASP B="2010-02-05T20:40:00" D="1"/>
   </PAN>

   <!- Filling new product ->
   <PAN A="PDT258" B="0048" C="30000" D="1" E="DET3" F="VPN1">
```



```
      <ASP A="2010-02-05T21:00:00" D="4"/>
    </PAN>
  </TSK>
```

In case the product (mixture) of a device is changed during task execution, the FMIS must rely on the PAN-elements to calculate the different product quantities because the total of the device in the TIM-Element of the task is not product specific.

If there is not PAN of type 3 (remainder) on startup of a task, the FMIS assumes a remainder of 0. This also applies to the end of the task. Without a finishing PAN-element of type 3 (remainder), the tank is assumed empty.

Product quantity calculation in the sample above:

13:00	The task is planned with a total of 50 l of PDT250 to be filled in DET3.
13:38	When starting the task, there is a remainder of 2 l of PDT250 in tank.
13:40	Tank is filled. There now are 50 l of PDT250 in the tank (→ the driver filled in 48 l).
18:50	A remainder of 10 l is in the tank (→ 50 l – 10 l = 40 l were applied so far).
19:00	Refill up to a total of 50 l of PDT250 (→ the driver filled in 40 l).
20:40	Tank is empty (remainder of 0). In total 40 l + 50 l = 90 l of PDT250 were applied so far.
21:00	Tank is filled with 30 l of PDT258.
??.??	Task is finished (time is to be taken out of the task time relation). Since there is no final PAN element of type 3 (remainder), the FMIS assumes the tank to be empty. Therefore 30 l of PDT258 were applied in the task.

So in the task DET3 applied 90 l of PDT250 and 30 l of PDT258. The total in the TIM element would state 120 l: < TIM … > < DLV A = "0080" B = "120" C = "DET3"/ > < /TIM >

D.44 ProductGroup — PGP

Type:

Coding data

Description:

The ProductGroup XML element is used to group products or produce and planting material of crop varieties. Products can be assigned to a Task during the Task execution by means of adding a ProductAllocation element to the Task. With this method, all changes in the allocation of products or in the specification of crop varieties that are being processed can be recorded while a Task is active.

To determine which products or produce or planting material of crop varieties belong to a specific product group or crop type, the ProductGroupIdRef values of all products shall be examined for a match with a particular ProductGroupId value. Similar, to determine which crop varieties belong to a specific crop type, the ProductGroupIdRef values of all products that match a ProductGroup of type CropType define the set of crop varieties belonging to a particular crop type. Within the XML elements CropType and CropVariety, references can be included to the ProductGroup and Product to cross reference the CropType and CropVariety specified for a Partfield with the commodity specified in Product and ProductGroup.

The ProductGroupType attribute shall be used to distinguish between products (i.e. crop protection chemicals) and crops (i.e. crop varieties cultivated on a partfield). If the ProductGroupType attribute is not set, then the default type shall be ProductGroup.

Referenced by XML element:

— *Product*

Table D.44 — ProductGroup attributes

Attribute	XML	Use	Type	Length/range	Comment
ProductGroupId	A	r	xs:ID	min. 4 to max. 14	Unique identifier of ProductGroup Format: (PGP|PGP-)([0–9])+ Records generated on MICS have negative IDs
ProductGroupDesignator	B	r	xs:string	max. 32	ProductGroup designator/ name
ProductGroupType[a]	C	o	xs:NMTOKEN	1 to 2	Type of ProductGroup 1 = ProductGroup (default), contains products 2 = CropType, contains crop varieties

[a] This attribute is introduced in ISO 11783-10 version 4.

EXAMPLE

```
<PGP A="PGP1" B="Herbicides"/>
<PDT A="PDT1" B="agent 1" C="PGP1"/>
<PGP A="PGP2" B="Potato" C="2" />
<PDT A="PDT1" B="Asterix" C="PGP2"/>
```

D.45 ProductRelation – PRN

Type:

Coding Data

Description:

The ProductRelation XML element specifies the allocation of a single product to a mixture product definition. The QuantityValue attribute defines the portion that the Product referenced in attribute ProductIdRef contributes to the referencing product mixture that it is included in.

Products referenced by a PRN element shall contain values for the attributes QuantityDDI and MixtureQuantity to enable calculation and handling of the product mixture portions of all ingredients.

The Product referenced in ProductRelation shall not be a mixture product definition (multi-level product mixtures shall not be defined).

The Product referenced in ProductRelation shall not be the same as the Product in which ProductRelation is included (circular references shall not occur).

Compound fertilizers, which combine N-P-K-S and any other minerals in the pellets are seen as a single product. When assembled on the farm from single-component fertilizers, the product is seen as a mixture with relation to the single products.

The ProductRelation XML element is added in ISO 11783-10 version 4.

References XML element:

— *Product*

Included by XML element:

— *Product*

Table D.45 — ProductRelation attributes

Attribute	XML	Use	Type	Length/Range	Comment
ProductIdRef	A	r	xs:IDREF	min.4 to max.14	Reference to XML element product Format: (PDT|PDT-)([0–9])+
QuantityValue	B	r	xs:long	0 to (2^{31}–1)	Contains the portion of the mixture

EXAMPLE

```
<PDT A="PDT250" B="liquid_mix_12" D="VPN1" E="0048" F="2" G="505">
   <PRN A="PDT1" B="400"/>
   <PRN A="PDT2" B="100"/>
   <PRN A="PDT24" B="5"/>
</PDT>
```

D.46 Task — TSK

Type:

Task data

Description:

The Task XML element describes an ISO 11783 task. The task is the central XML element in the data transfer file set. It contains references to various other XML elements to specify and record the allocation of resources and operations.

By specifying the XML attributes CustomerIdRef, FarmIdRef and PartfieldIdRef in a task, redundant relations between task, customer, farm, and partfield can be created. This redundancy is allowed and shall not result in conflicts in the relations between these XML elements.

The XML attribute ResponsibleWorkerIdRef is a reference to a single worker responsible for this task. This can, for example, be the worker that specified this task or a worker that can be contacted to provide additional detail about this task. The recording of, for example, start time and duration of one or more workers contributing to execution of a task shall be done by the use of XML element WorkerAllocation in a task. Multiple WorkerAllocation XML elements can be specified in a single task.

The XML attribute TaskStatus has values 1 to 4 and 6 for specifying the lifecycle of a Task. In addition to these 5 states, the TaskStatus can be specified as “template” (value 5). When a Task with status “template” is started, then a new Task, which is a copy of the template Task, shall be created and started.

At least one Time element shall be added to a Task after it has been started by the MICS and time information is present on the ISO 11783 network.

References XML elements:

— *Customer*

— *Farm*

— *Partfield*

— *(Responsible) Worker*

— *(Default and PositionLost) TreatmentZones*

Includes XML elements:

- *TreatmentZone*
- *Time*
- *OperTechPractice*
- *WorkerAllocation*
- *DeviceAllocation*
- *Connection*
- *ProductAllocation*
- *DataLogTrigger*
- *CommentAllocation*
- *Grid*
- *TimeLog*
- *ControlAssignment*
- *GuidanceAllocation*

Table D.46 — Task attributes

Attribute	XML	Use	Type	Length/range	Comment
TaskId	A	r	xs:ID	min. 4 to max. 14	Unique identifier of task Format: (TSK|TSK-)([0–9])+ Records generated on MICS have negative IDs
TaskDesignator	B	o	xs:string	max. 32	Task designator/name
CustomerIdRef	C	o	xs:IDREF	min. 4 to max. 14	Reference to XML element Customer Format: (CTR|CTR-)([0–9])+
FarmIdRef	D	o	xs:IDREF	min. 4 to max. 14	Reference to XML element Farm Format: (FRM|FRM-)([0–9])+
PartfieldIdRef	E	o	xs:IDREF	min. 4 to max. 14	Reference to XML element Partfield Format: (PFD|PFD-)([0–9])+
ResponsibleWorkerIdRef	F	o	xs:IDREF	min. 4 to max. 14	Reference to XML element Worker Format: (WKR|WKR-)([0–9])+
TaskStatus	G	r	xs:NMTOKEN	1 to 6	Status of the task, possible values: 1 = planned 2 = running 3 = paused 4 = completed 5 = template[a] 6 = canceled[a]

[a] This definition is introduced in ISO 11783-10 version 4.

Table D.46 *(continued)*

Attribute	XML	Use	Type	Length/range	Comment
DefaultTreatmentZoneCode	H	o	xs:unsigned-Byte	0 to 254	Reference to XML element Treat-mentZone
PositionLostTreatmentZone-Code	I	o	xs:unsigned-Byte	0 to 254	Reference to XML element Treat-mentZone
OutOfFieldTreatmentZone-Code	J	o	xs:unsigned-Byte	0 to 254	Reference to XML element Treat-mentZone
TreatmentZone		o	xs:element		Includes a list of XML element TreatmentZone
Time		o	xs:element		Includes a list of XML element Time
OperTechPractice		o	xs:element		Includes a single XML element OperTechPractice
WorkerAllocation		o	xs:element		Includes a list of XML element WorkerAllocation
DeviceAllocation		o	xs:element		Includes a list of XML element DeviceAllocation
Connection		o	xs:element		Includes a list of XML element Connection
ProductAllocation		o	xs:element		Includes a list of XML element ProductAllocation
DataLogTrigger		o	xs:element		Includes a list of XML element DataLogTrigger
CommentAllocation		o	xs:element		Includes a list of XML element CommentAllocation
TimeLog		o	xs:element		Includes a list of XML element TimeLog
Grid		o	xs:element		Includes a single XML element Grid
ControlAssignment[a]		o	xs:element		Includes a list of XML element ControlAssignment
GuidanceAllocation[a]		o	xs:element		Includes a list of XML element GuidanceAllocation
[a] This definition is introduced in ISO 11783-10 version 4.					

EXAMPLE

```
<TSK A=”TSK1” F=”WKR1” G=”1”>
   <OTP A=”CPC1” B=”OTQ1”/>
   <WAN A=”WKR1”>
      <ASP A=”2003-08-20T08:00:00” B=”2003-08-20T17:00:00” D=”1”/>
   </WAN>
   <DAN B=”7FFFFFFF00000000” A=”200C840000000000”>
      <ASP A=”2003-08-20T08:00:00” B=”2003-08-20T17:00:00” D=”1”/>
   </DAN>
   <CNN A=”DVC2” B=”DET2” C=”DVC1” D=”DET2”/>
   <DLT A=”1122” B=”1” H=”DET2”/>
   <CAN A=”CCT5”>
      <ASP A=”2003-08-20T08:00:20” D=”4”/>
   </CAN>
   <TZN A=”1” B=”200” C=”140”/>
   <TIM A=”2003-08-20T08:00:00” B=”2003-08-20T17:00:00” D=”1”/>
</TSK>
```

D.47 TaskControllerCapabilities — TCC

The TaskControllerCapabilities XML element contains the implementation version and the capabilities of the task controller which generated the data transfer file set. This XML element shall only be included in the data transfer file set when the data transfer origin is MICS and, at a maximum, only one TaskControllerCapabilities XML element shall, in that case, be included in the data transfer file set.

The TaskControllerCapabilities XML element is added in ISO 11783-10 version 4.

Included by XML element:

— *ISO11783_TaskData*

Table D.47 — TaskControllerCapabilities attributes

Attribute	XML	Use	Type	Length/ range	Comment
TaskControllerControlFunction NAME	A	r	xs:hexBinary	000000000 0000000_{16} to FFFFFFFFF $FFFFFFF_{16}$	NAME of the task controller (see ISO 11783-5)
TaskControllerDesignator	B	r	xs:string	max 153 characters	Product Name as provided by the manufacturer. Contains the product name specified in the Product Identification Information parameter group and defined in ISO 11783-12.
VersionNumber	C	r	xs:NMTOKEN	0 to 4	Version of ISO 11783-10 that this task controller meets: 0 = The version of the DIS (draft International Standard) 1 = The version of the FDIS.1 (final draft International Standard, first edition) 2 = The version of the FDIS.2 and the first edition published as an International Standard 3 = The version of second edition published as a draft International Standard (E2. DIS). 4 = The version of the second edition published as a final draft International Standard (E2.FDIS).

Table D.47 *(continued)*

Attribute	XML	Use	Type	Length/ range	Comment
ProvidedCapabilities	D	r	xs:unsignedByte	0 to 63	Supported capabilities: 1 = Supports documentation 2 = Supports TC-GEO without position based control 4 = Supports TC-GEO with position based control 8 = Supports peer control assignment 16 = Supports implement section control 32 = Supports polygon prescription maps
NumberOfBoomsSectionControl	E	r	xs:unsignedByte	0 to 254	States the maximum number of booms the task controller supports for section control or 0 when no section control is supported
NumberOfSectionsSectionControl	F	r	xs:unsignedByte	0 to 254	States the maximum number of section the task controller supports for section control or 0 when no section control is supported
NumberOfControlChannels	G	r	xs:unsignedByte	0 to 254	States the maximum number of control channels the task controller supports or 0 when no control channels are supported

D.48 Time — TIM

Type:

Task data

Description:

The Time XML element specifies a recording of a time event. Optionally, a position can be recorded at a time specification. The type attribute is used to specify what type of Time is specified or recorded. All Time XML elements that are provided by FMIS shall be of the type planned. When the MICS does not provide a detailed time recording distinction by using all of the types 2 (Preliminary) to types 8 (Powered Down), then all Time XML elements that are provided by the MICS shall be of type 4 (Effective). In this case, the sum of these recorded effective times is the total working time and cannot be broken down to more detailed time types. See Annex G for a description of the more detailed time types.

A Time XML element with Type 1 (Planned) is used to specify the planned start and stop or duration of a Task in which this Time XML element is included. The use of this Type value is for Task planning and scheduling purposes only. At maximum, one Time XML element of type planned may be included in each task.

A Time XML element with Type 8 (Powered Down) is used to record the time of powering down the machine of which the TC is a part of.

All Time XML elements shall have a Start attribute value defined. Duration shall always contain a positive value and is equal to the time elapsed between Start and Stop. Thus, a finite Time XML element specification can consist of either a Start and Stop (Duration can be calculated) or a Start and Duration (Stop time can be calculated). The specification of an infinite Time XML element, e.g. only recording the Start, is allowed.

In ISO 11783-10 version 4, the datetime data type has been extended to include time zone information or to be explicitly set to UTC. When the time zone is known, it is required to either specify times in UTC by adding the "Z" indicator or include time zone offsets in the start and stop attributes. Using either UTC explicitly or using the time zone information when local time is used by the FMIS to normalize data coming from multiple time zones. In case no time zone information is known, then all times are expressed in local time without time zone information.

Prior to version 4, the time zone can only be calculated from the logged records in the TimeLog binary data file if both the UTC time values and the local time values are recorded by subtracting the local time from the UTC time. In this case, the calculated time zone can be applied to all timestamps in the same task data transfer file set.

Included by XML elements:

— *Task*

— *TimeLog*

Includes XML elements:

— *Position*

— *DataLogValue*

Table D.48 — Time attributes

Attribute	XML	Use	Type	Length/ range	Comment
Start	A	r	xs:datetime	Max. 29	Time of start. Format: yyyy-mm-ddThh:mm:ss.sss plus an optional time zone indication.[a]
Stop	B	o[b]	xs:datetime	Max. 29	Time of end. Format: yyyy-mm-ddThh:mm:ss.sss plus an optional time zone indication.[a]
Duration	C	o[b]	xs:unsigned-Long	0 to (2^{32}–2)	Time between start and stop in number of seconds.

[a] The time zone information is introduced in ISO 11783-10 version 4.

[b] Either Stop or Duration need to be specified in addition to Start to create a finite Time XML element.

[c] The value 3 (Preparation) has been deprecated in ISO 11783-10 version 4, this value did not provide additional detail over value 2 (Preliminary) time.

[d] This token is introduced in ISO 11783-10 version 4.

Table D.48 *(continued)*

Attribute	XML	Use	Type	Length/range	Comment
Type	D	r	xs:NMTOKEN	1 to 2, 4 to 8	Type of the specified or recorded time, possible values[c]: 1 = Planned 2 = Preliminary 4 = Effective 5 = Ineffective 6 = Repair 7 = Clearing 8 = Powered Down[d]
Position		o	xs:element		Include up to 2 Position XML elements which represent the position at the start and stop attribute values of the Time.
DataLogValue		o	xs:element		Includes a list of XML element DataLog-Value

[a] The time zone information is introduced in ISO 11783-10 version 4.

[b] Either Stop or Duration need to be specified in addition to Start to create a finite Time XML element.

[c] The value 3 (Preparation) has been deprecated in ISO 11783-10 version 4, this value did not provide additional detail over value 2 (Preliminary) time.

[d] This token is introduced in ISO 11783-10 version 4.

EXAMPLE 1 No time zone information is available, time is recorded in local time

```
<TIM A="2003-08-20T08:10:00" B="2003-08-20T12:04:23" D="4">
   <PTN A="54.588945" B="9.989209" D="3"/>
   <DLV A="0815" B="10" C="DET1"/>
   <DLV A="4711" B="15" C="DET2"/>
</TIM>
<TIM A="2003-08-20T08:10:00" C="3612" D="6"/>
```

EXAMPLE 2 Time zone information is available in the system, but time is recorded explicitly in UTC only

```
<TIM A="2003-08-20T07:10:00Z" C="3612" D="6"/>
```

EXAMPLE 3 Time zone information is available, time is recorded in local time with time zone information, e.g. for Central European Time a start time for the same moment of time as example 2 would be

```
<TIM A="2003-08-20T08:10:00+01:00" C="3612" D="6"/>
```

or for a location in time zone 0 this same moment of time is

```
<TIM A="2003-08-20T07:10:00+00:00" C="3612" D="6"/>
```

D.49 TimeLog — TLG

Type:

Task data

Description:

Time XML elements are used as embedded lists inside the XML data transfer file set or as a time template specification in the TimeLog XML element. TimeLog enables the collection of all DataLogValues in a binary-coded data log file. Within the TimeLog XML element, the TimeType attribute of the Time XML element shall be set to 4 (effective). The TimeLogType attribute is defined to enable future expansion of the data-logging method. The value of the TimeLogType attribute shall be set a value of “1” for the current data-logging method which is specified in 6.8.2 and 6.8.4.

TimeLog is always related to a task and refers to a set of two files of a unique name. Both files shall exist in the same directory as the other files in the data transfer file set. The name of the files shall be unique over all TimeLogs referred to by all tasks of a data transfer file set.

Included by XML elements:

— *Task*

Table D.49 — TimeLog attributes

Attribute	XML	Use	Type	Length/ range	Comment
Filename	A	r	xs:ID	8	Unique name of TimeLog file Format: TLG[0–9][0–9][0–9][0–9][0–9]
Filelength	B	o	xs:unsigned-Long	0 to (2^{32}–2)	Length of TimeLog file in number of bytes
TimeLogType	C	r	xs:NMTOKEN	1	1 = binary timelog file type 1

EXAMPLE TimeLog in the data transfer file set

```
<TLG A="TLG00001" C="1"/>
```

The TimeLog XML file TLG00001.xml is as follows:

```
<TIM A="" D="4">
   <PTN A="" B="" D=""/>
   <DLV A="0002" B="" C="DET1"/>
   <DLV A="0007" B="" C="DET2"/>
</TIM>
```

All attributes that refer to any values are considered to be fixed values throughout all records of the binary file. The XML file above specifies that all the following records consist of the following set of values:

(TimeStart, PositionNorth, PositionEast, PositionStatus, #DLV, DLV0, PDV0, DLV1, PDV1)

This shall represent the following values in binary format:

(2005–05–02T16:32:00, 51.00678, 6.03489, 1, 2, 0, 10, 1, 15)

D.50 TreatmentZone — TZN

Type:

Task data

Description:

The TreatmentZone describes an area to be treated with the same process data DDIs and the same values.

Inside a task, treatment zones can be referenced. The DefaultTreatmentZone has a special meaning concerning the included ProcessDataVariables. The values of those ProcessDataVariables are to be applied globally to the whole task. All ProcessDataVariable values inside the DefaultTreatmentZone are sent by the TC to all appropriate clients on starting and resuming a task. A DefaultTreatmentZone can exist for non-site-specific tasks. Non-site-specific tasks are tasks without any polygons or grid defined. The PositionLostTreatmentZone contains the ProcessDataVariables that shall be sent to the clients when the positioning becomes unavailable. The OutOfFieldTreatmentZone contains the ProcessDataVariables that shall be sent to the appropriate device elements of clients when (part of) the clients leave the area bounded by a field boundary polygon. The specification of the PositionLostTreatmentZone and of the OutOfFieldTreatmentZone is intended for site-specific tasks.

Prior to ISO 11783-10 version 4, only a single Polygon could be included in a TreatmentZone to specify its geometry. In version 4 and later, a list of Polygons can be included in a TreatmentZone and each Polygon has been limited to describe one single surface.

When the geometry of the TreatmentZone is defined by one or more polygons, then only a single XML element ProcessDataVariable shall be included. This change for polygon bounded treatment zones from a list to single ProcessDataVariable XML element inclusion has been introduced in ISO 11783-10 version 4. TreatmentZones without a geometry definition, e.g. as referenced in a Grid or in Task attributes, may include a list of XML element ProcessDataVariable.

Referenced by XML elements:

— *Grid*

Included by XML elements:

— *Task*

Includes XML elements:

— *ProcessDataVariable*

— *Polygon (of type TreatmentZone only)*

Table D.50 — TreatmentZone attributes

Attribute	XML	Use	Type	Length/ range	Comment
TreatmentZoneCode	A	r	xs:unsigned-Byte	0 to 254	Unique TreatmentZoneCode inside a task
TreatmentZoneDesig-nator	B	o	xs:string	max. 32	Name of TreatmentZone
TreatmentZoneColour	C	o	xs:unsigned-Byte	0 to 254	Colour of TreatmentZone Format: palette like ISO 11783-6
Polygon		o	xs:element		Includes a list [a] of XML element Polygon
ProcessDataVariable		o	xs:element		Includes a single or a list [a] of XML element ProcessDataVariable
[a] The multiplicity of the inclusion of this attribute has been changed in ISO 11783-10 version 4.					

EXAMPLE 1 TreatmentZone with multiple ProcessDataVariables

```
<TZN A="6" B="MidRate 1" C="2">
   <PDV A="0001" B="100"/>
   <PDV A="0006" B="1200"/>
</TZN>
```

EXAMPLE 2 XML encoding of a polygon based site-specific application (see Figure 5)

```
<PDT A="PDT1" B="Granular Fertilizer"/>
<PDT A="PDT2" B="Sunflower"/>
<VPN A="VPN1" B="0" C="0.01" D="0" E="kg/ha"/>
<TSK A="TSK1" G="1" H="0">
   <TZN A="1" B="Layer 1, TreatmentZone 1" C="102">
      <PDV A="0006" B="5800" C="PDT1" E="VPN1" F="1"/>
      <PLN A="2" B="Polygon 1">
```

```
        <LSG A="1" B="outer boundary">
          <PNT A="2" C="38.13336" D="-97.41707"/>
          <PNT A="2" C="38.13322" D="-97.41675"/>
          <PNT A="2" C="38.13319" D="-97.41675"/>
          <PNT A="2" C="38.13318" D="-97.41683"/>
          <PNT A="2" C="38.13318" D="-97.41690"/>
          <PNT A="2" C="38.1332" D="-97.417"/>
          <PNT A="2" C="38.13323" D="-97.41707"/>
          <PNT A="2" C="38.13336" D="-97.41707"/>
        </LSG>
      </PLN>
      <PLN A="2" B="Polygon 3">
        <LSG A="1" B="outer boundary">
          <PNT A="2" C="38.1331" D="-97.41707"/>
          <PNT A="2" C="38.13312" D="-97.41704"/>
          <PNT A="2" C="38.13312" D="-97.41702"/>
          <PNT A="2" C="38.1331" D="-97.41699"/>
          <PNT A="2" C="38.13309" D="-97.41696"/>
          <PNT A="2" C="38.13309" D="-97.41694"/>
          <PNT A="2" C="38.133105" D="-97.41689"/>
          <PNT A="2" C="38.13311" D="-97.41683"/>
          <PNT A="2" C="38.1331" D="-97.41675"/>
          <PNT A="2" C="38.132955" D="-97.41675"/>
          <PNT A="2" C="38.13296" D="-97.41679"/>
          <PNT A="2" C="38.132955" D="-97.41683"/>
          <PNT A="2" C="38.13294" D="-97.41689"/>
          <PNT A="2" C="38.132945" D="-97.41694"/>
          <PNT A="2" C="38.13295" D="-97.41696"/>
          <PNT A="2" C="38.13297" D="-97.417"/>
          <PNT A="2" C="38.13297" D="-97.41703"/>
          <PNT A="2" C="38.13295" D="-97.41707"/>
          <PNT A="2" C="38.1331" D="-97.41707"/>
        </LSG>
        <LSG A="2" B="inner boundary">
          <PNT A="2" C="38.133055" D="-97.41690"/>
          <PNT A="2" C="38.13306" D="-97.41688"/>
          <PNT A="2" C="38.13306" D="-97.41684"/>
          <PNT A="2" C="38.13305" D="-97.41681"/>
          <PNT A="2" C="38.13304" D="-97.41680"/>
          <PNT A="2" C="38.13302" D="-97.41680"/>
          <PNT A="2" C="38.13301" D="-97.41681"/>
          <PNT A="2" C="38.133" D="-97.41684"/>
          <PNT A="2" C="38.132995" D="-97.41689"/>
          <PNT A="2" C="38.133" D="-97.41691"/>
          <PNT A="2" C="38.13301" D="-97.41692"/>
          <PNT A="2" C="38.13303" D="-97.41692"/>
          <PNT A="2" C="38.13305" D="-97.41691"/>
          <PNT A="2" C="38.133055" D="-97.41690"/>
        </LSG>
      </PLN>
    </TZN>
    <TZN A="2" B="Layer 1, TreatmentZone 2" C="1">
      <PDV A="0006" B="6500" C="PDT1" E="VPN1" F="1"/>
      <PLN A="2" B="Polygon 2">
        <LSG A="1" B="outer boundary">
          <PNT A="2" C="38.13323" D="-97.41707"/>
          <PNT A="2" C="38.1332" D="-97.417"/>
          <PNT A="2" C="38.13318" D="-97.4169"/>
          <PNT A="2" C="38.13318" D="-97.41683"/>
          <PNT A="2" C="38.13319" D="-97.41675"/>
          <PNT A="2" C="38.1331" D="-97.41675"/>
          <PNT A="2" C="38.13311" D="-97.41683"/>
          <PNT A="2" C="38.133105" D="-97.41689"/>
          <PNT A="2" C="38.13309" D="-97.41694"/>
          <PNT A="2" C="38.13309" D="-97.41696"/>
          <PNT A="2" C="38.1331" D="-97.41699"/>
          <PNT A="2" C="38.13312" D="-97.41702"/>
          <PNT A="2" C="38.13312" D="-97.41704"/>
          <PNT A="2" C="38.1331" D="-97.41707"/>
          <PNT A="2" C="38.13323" D="-97.41707"/>
        </LSG>
```



```
    </PLN>
    <PLN A="2" B="Polygon 4">
      <LSG A="1" B="outer boundary">
        <PNT A="2" C="38.133055" D="-97.41690"/>
        <PNT A="2" C="38.13306" D="-97.41688"/>
        <PNT A="2" C="38.13306" D="-97.41684"/>
        <PNT A="2" C="38.13305" D="-97.41681"/>
        <PNT A="2" C="38.13304" D="-97.41680"/>
        <PNT A="2" C="38.13302" D="-97.41680"/>
        <PNT A="2" C="38.13301" D="-97.41681"/>
        <PNT A="2" C="38.133" D="-97.41684"/>
        <PNT A="2" C="38.132995" D="-97.41689"/>
        <PNT A="2" C="38.133" D="-97.41691"/>
        <PNT A="2" C="38.13301" D="-97.41692"/>
        <PNT A="2" C="38.13303" D="-97.41692"/>
        <PNT A="2" C="38.13305" D="-97.41691"/>
        <PNT A="2" C="38.133055" D="-97.41690"/>
      </LSG>
    </PLN>
  </TZN>
  <TZN A="3" B="Layer 1, TreatmentZone 3" C="7">
    <PDV A="0006" B="5000" C="PDT1" E="VPN1" F="1"/>
    <PLN A="2" B="Polygon 5">
      <LSG A="1" B="outer boundary">
        <PNT A="2" C="38.13295" D="-97.41707"/>
        <PNT A="2" C="38.13297" D="-97.41703"/>
        <PNT A="2" C="38.13297" D="-97.417"/>
        <PNT A="2" C="38.13295" D="-97.41696"/>
        <PNT A="2" C="38.132945" D="-97.41694"/>
        <PNT A="2" C="38.13294" D="-97.41689"/>
        <PNT A="2" C="38.132955" D="-97.41683"/>
        <PNT A="2" C="38.13296" D="-97.41679"/>
        <PNT A="2" C="38.132955" D="-97.41675"/>
        <PNT A="2" C="38.13286" D="-97.41675"/>
        <PNT A="2" C="38.13286" D="-97.41707"/>
        <PNT A="2" C="38.13295" D="-97.41707"/>
      </LSG>
    </PLN>
  </TZN>
  <TZN A="4" B="Layer 2, TreatmentZone 4" C="7">
    <PDV A="0006" B="800" C="PDT2" E="VPN1" F="2"/>
    <PLN A="2" B="Polygon 6">
      <LSG A="1" B="outer boundary">
        <PNT A="2" C="38.13336" D="-97.41707"/>
        <PNT A="2" C="38.13325" D="-97.41682"/>
        <PNT A="2" C="38.13324" D="-97.4169"/>
        <PNT A="2" C="38.13323" D="-97.41694"/>
        <PNT A="2" C="38.1332" D="-97.41697"/>
        <PNT A="2" C="38.13317" D="-97.416982"/>
        <PNT A="2" C="38.13312" D="-97.41698"/>
        <PNT A="2" C="38.133108" D="-97.416985"/>
        <PNT A="2" C="38.1331" D="-97.417"/>
        <PNT A="2" C="38.13309" D="-97.41707"/>
        <PNT A="2" C="38.13336" D="-97.41707"/>
      </LSG>
    </PLN>
  </TZN>
  <TZN A="5" B="Layer 2, TreatmentZone 5" C="102">
    <PDV A="0006" B="900" C="PDT2" E="VPN1" F="2"/>
    <PLN A="2" B="Polygon 7">
      <LSG A="1" B="outer boundary">
        <PNT A="2" C="38.13309" D="-97.41707"/>
        <PNT A="2" C="38.1331" D="-97.417"/>
        <PNT A="2" C="38.133108" D="-97.416985"/>
        <PNT A="2" C="38.13312" D="-97.41698"/>
        <PNT A="2" C="38.13317" D="-97.416982"/>
        <PNT A="2" C="38.1332" D="-97.41697"/>
        <PNT A="2" C="38.13323" D="-97.41694"/>
        <PNT A="2" C="38.13324" D="-97.4169"/>
        <PNT A="2" C="38.13325" D="-97.41682"/>
        <PNT A="2" C="38.13322" D="-97.41675"/>
```

```
            <PNT A="2" C="38.13319" D="-97.41675"/>
            <PNT A="2" C="38.13318" D="-97.41679"/>
            <PNT A="2" C="38.13315" D="-97.41685"/>
            <PNT A="2" C="38.133095" D="-97.4169"/>
            <PNT A="2" C="38.133055" D="-97.41695"/>
            <PNT A="2" C="38.133025" D="-97.417"/>
            <PNT A="2" C="38.133005" D="-97.41707"/>
            <PNT A="2" C="38.13309" D="-97.41707"/>
         </LSG>
      </PLN>
      <PLN A="2" B="Polygon 9">
         <LSG A="1" B="outer boundary">
            <PNT A="2" C="38.1329" D="-97.41707"/>
            <PNT A="2" C="38.132905" D="-97.41703"/>
            <PNT A="2" C="38.13291" D="-97.417"/>
            <PNT A="2" C="38.132925" D="-97.41698"/>
            <PNT A="2" C="38.13296" D="-97.41695"/>
            <PNT A="2" C="38.132995" D="-97.41689"/>
            <PNT A="2" C="38.133005" D="-97.41683"/>
            <PNT A="2" C="38.132995" D="-97.41679"/>
            <PNT A="2" C="38.13297" D="-97.41675"/>
            <PNT A="2" C="38.13286" D="-97.41675"/>
            <PNT A="2" C="38.13286" D="-97.41707"/>
            <PNT A="2" C="38.1329" D="-97.41707"/>
         </LSG>
      </PLN>
   </TZN>
   <TZN A="6" B="Layer 2, TreatmentZone 6" C="1">
      <PDV A="0006" B="1000" C="PDT2" E="VPN1" F="2"/>
      <PLN A="2" B="Polygon 8">
         <LSG A="1" B="outer boundary">
            <PNT A="2" C="38.133005" D="-97.41707"/>
            <PNT A="2" C="38.133025" D="-97.417"/>
            <PNT A="2" C="38.133055" D="-97.41695"/>
            <PNT A="2" C="38.133095" D="-97.4169"/>
            <PNT A="2" C="38.13315" D="-97.41685"/>
            <PNT A="2" C="38.13318" D="-97.41679"/>
            <PNT A="2" C="38.13319" D="-97.41675"/>
            <PNT A="2" C="38.13297" D="-97.41675"/>
            <PNT A="2" C="38.132995" D="-97.41679"/>
            <PNT A="2" C="38.133005" D="-97.41683"/>
            <PNT A="2" C="38.132995" D="-97.41689"/>
            <PNT A="2" C="38.13296" D="-97.41695"/>
            <PNT A="2" C="38.132925" D="-97.41698"/>
            <PNT A="2" C="38.13291" D="-97.417"/>
            <PNT A="2" C="38.132905" D="-97.41703"/>
            <PNT A="2" C="38.1329" D="-97.41707"/>
            <PNT A="2" C="38.133005" D="-97.41707"/>
         </LSG>
      </PLN>
   </TZN>
</TSK>
```


D.51 ValuePresentation — VPN

Type:

Coding data

Description:

The ValuePresentation XML element is used to specify the presentation of data dictionary entity-defined integer values. The presentation shall be according to the following formula:

Presented value = (integer value + Offset) * Scale

Presented values are always rounded to the number of decimals specified in the NumberOfDecimals attribute.

References XML elements:

— *ColourLegend*

Referenced by XML elements:

— *DataLogTrigger*

— *ProcessDataVariable*

— *Product*

— *ProductAllocation*

Table D.51 — ValuePresentation attributes

Attribute	XML	Use	Type	Length/range	Comment
ValuePresentationId	A	r	xs:ID	min. 4 to max. 14	Unique identifier of ValuePresentation Format: (VPN|VPN-)([0–9])+
Offset	B	r	xs:long	-2^{31} to $(2^{31}-1)$	Offset to be applied to the value for presentation
Scale	C	r	xs:decimal	0.000000001 to 100000000.0	Scale to be applied to the value for presentation
NumberOfDecimals	D	r	xs:unsignedByte	0 to 7	Number of decimals to be presented after the decimal separator
UnitDesignator	E	o	xs:string	max. 32	Optional Unit Designator string
ColourLegendIdRef	F	o	xs:IDREF	min. 4 to max. 14	Reference to XML element ColourLegend Format: (CLD|CLD-)([0–9])+

EXAMPLE

```
<VPN A=”VPN1” B=”0” C=”1.0” D=”0” E=”kg”/>
<VPN A=”VPN2” B=”18” C=”1.8” D=”1” E=”°F”/>
```

D.52 Worker — WKR

Type:

Coding data

Description:

The Worker XML element describes a worker that can be referenced by a task. All worker allocations are logged with time information inside the data transfer file set. The task attribute ResponsibleWorkerIdRef has a special meaning for a task. This worker is directly referenced by the task without any additional logged data.

Referenced by XML elements:

— *Task*

— *WorkerAllocation*

Table D.52 — Worker attributes

Attribute	XML	Use	Type	Length/range	Comment
WorkerId	A	r	xs:ID	min. 4 to max. 14	Unique identifier of worker Format: (WKR|WKR-)([0–9])+ Records generated on MICS have negative IDs
WorkerLastName	B	r	xs:string	max. 32	Last name
WorkerFirstName	C	o	xs:string	max. 32	First name
WorkerStreet	D	o	xs:string	max. 32	Street
WorkerPOBox	E	o	xs:string	max. 32	Postbox
WorkerPostalCode	F	o	xs:string	max. 10	Postal code
WorkerCity	G	o	xs:string	max. 32	City
WorkerState	H	o	xs:string	max. 32	State, province
WorkerCountry	I	o	xs:string	max. 32	Country
WorkerPhone	J	o	xs:string	max. 20	Telephone number
WorkerMobile	K	o	xs:string	max. 20	Mobile phone number
WorkerLicenseNumber	L	o	xs:string	max. 32	Worker-specific licence number
WorkerEmail	M	o	xs:string	max. 64	E-mail

EXAMPLE

```
<WKR A="WKR1" B="Smith" C="John"/>
<WKR A="WKR2" B="Miller" C="Karl"/>
```

D.53 WorkerAllocation — WAN

Type:

Task data

Description:

The WorkerAllocation XML element describes the allocation of workers to a task. The AllocationStamp entry describes the start/stop time of worker allocations and the changes of the worker allocations inside a task.

References XML elements:

— *Worker*

Included by XML elements:

— *Task*

Includes XML elements:

— *AllocationStamp*

Table D.53 — WorkerAllocation attributes

Attribute	XML	Use	Type	Length/range	Comment
WorkerIdRef	A	r	xs:IDREF	min. 4 to max. 14	Reference to XML element Worker Format: (WKR|WKR-)([0–9])+

Table D.53 *(continued)*

Attribute	XML	Use	Type	Length/range	Comment
AllocationStamp		o	xs:element		Includes a single XML element AllocationStamp

EXAMPLE

```
<WAN A="WKR1">
   <ASP A="2003-08-20T08:00:00" D="4" />
</WAN>
<WAN A="WKR2">
   <ASP A="2003-08-20T08:05:00" D="4" />
</WAN>
```

D.54 ExternalFileContents — XFC

Type:

Task data

Description:

The XML element ExternalFileContents is used to group all XML elements of an XML file external to the main XML data transfer file in order to keep the external file well formed.

Includes XML elements:

— *CodedCommentGroup*

— *CodedComment*

— *ColourLegend*

— *CulturalPractice*

— *CropType*

— *Customer*

— *Device*

— *Farm*

— *OperationTechnique*

— *Product*

— *Partfield*

— *ProductGroup*

— *Task*

— *ValuePresentation*

— *Worker*

D.55 ExternalFileReference — XFR

Type:

Task data

Description:

The XML element ExternalFileReference is used to refer to an XML file external to the main XML file. The external file can only include top-level XML elements. Top-level XML elements are the elements that can be included in XML element ISO11783_TaskData. Inside an external XML file, only a single type of XML element can be specified per file. There shall be no recursive use of XFR elements and no recursive use of XFC elements.

Included by XML elements:

— *ISO11783_TaskData*

Includes XML elements:

— *None*

Table D.54 — ExternalFileReference attributes

Attribute	XML	Use	Type	Length/ range	Comment
Filename	A	r	xs:ID	8	Specification of filename Format: (CCG|CCT|CLD|CPC|CTP|CTR|DVC| FRM|OTQ|PDT|PFD|PGP|TSK| VPN|WKR)[0–9][0–9][0–9][0–9][0–9]
Filetype	B	r	xs:NMTOKEN	1	1 = XML

EXAMPLE

```
<XFR A="FRM00000" B="1"/>
<XFR A="CTP00000" B="1"/>
<XFR A="PFD00000" B="1"/>
<XFR A="WKR00000" B="1"/>
<XFR A="CCT00000" B="1"/>
<XFR A="PGP00000" B="1"/>
<XFR A="PDT00000" B="1"/>
<XFR A="DVC00000" B="1"/>
<XFR A="TSK00000" B="1"/>
<XFR A="TSK00001" B="1"/>
<XFR A="TSK00002" B="1"/>
<XFR A="VPN00000" B="1"/>
<XFR A="CTR00000" B="1"/>
```

Annex E
(normative)

Predefined ISO-11783 attachments

E.1 Link List

There are situations where an FMIS and a MICS want to exchange additional key information about objects of the XML data set. The predefined Link List attachment provides a standardised way to relate a XML element of the data transfer file set to one or more additional key values. The existence of an object ID is a precondition for this relationship. The associations (henceforth also called "mappings") between XML element object IDs and additional key values are stored in a separate file, the link list file, with the name 'LINKLIST.XML' (all uppercase).

The link list file shall start with an XML identification section and shall be well formed. The following xml version specification shall be at the start of the link list file:

```
<?xml version="1.0" encoding="UTF-8"?>
```

The link list file shall contain one root XML element containing all the link definitions.

The link list file and the XML elements contained in this file are introduced in ISO 11783-10 version 4.

E.2 ISO 11783LinkList

Type:

Root element

Description:

The XML element ISO 11783LinkList is the root XML element of the link list file and contains definitions about the construction of the XML file and the inclusion of the primary XML elements.

The FileVersion attribute shall be used to define a unique relation between the LINKLIST.XML and the TASKDATA.XML files.

The TASKDATA.XML file shall contain an AttachedFile (AFE) element, < AFE A = "LINKLIST.XML" B = "1" C = "" D = "1" *F* = "12988"/ > , to specify the transfer of the attached LINKLIST.XML file,

A maximum of one LINKLIST.XML file can be referenced in the TASKDATA.XML file.

Includes XML elements:

— *LinkGroup*

Table E.1 — ISO 11783LinkList attributes

Attribute	XML	Use	Type	Length/range	Comment
VersionMajor	VersionMajor	r	xs:NMTOKEN	4 to 99	List element release number (major), used to specify the version of ISO 11783-10 that this Link List file meets: 4 = The version of the second edition published as a final draft International Standard (E2.FDIS). Note: The ISO 11783LinkList is introduced in version 4.
VersionMinor	VersionMinor	r	xs:NMTOKEN	0 to 99	List element version number (minor), used to refer to revisions of the XML schema that this element complies to. See 8.4 for the naming convention of the XML schema.
ManagementSoftwareManufacturer	Management Software Manufacturer	r	xs:string	32	Name of management-software manufacturer
ManagementSoftwareVersion	Management Software Version	r	xs:string	32	Version of management software
TaskControllerManufacturer	TaskControllerManufacturer	o	xs:string	32	Name of TC manufacturer
TaskControllerVersion	TaskControllerVersion	o	xs:string	32	Version of TC software
FileVersion	FileVersion	o	xs:string	32	Version of the linklist file
DataTransferOrigin	DataTransfer Origin	r	xs:NMTOKEN	1 to 2	Describes the origin of the XML file: 1 = FMIS 2 = MICS
LinkGroup	LGP	r	xs:element		Includes a list of XML element LinkGroup

EXAMPLE

```
<ISO11783LinkList VersionMajor="4" VersionMinor="0"
TaskControllerManufacturer="FarmCtrl" TaskControllerVersion="1.0"
ManagementSoftwareManufacturer="FarmSystem"
ManagementSoftwareVersion="1.0" FileVersion="3.2" DataTransferOrigin="2">
   <LGP A="LGP10" B="1" E="Sample Of UUID Links">
      <LNK A="FRM1" B="{1059B14E-929F-4C4C-BCD4-C4F52A6A076A}"/>
      <LNK A="PFD3" B="{92043685-072D-4CAA-B3A8-C1E9D23BDB31}" C="Headland"/>
      <LNK A="PDT19" B="{C89AA8C5-A9B3-4CA4-BE47-4449B9CD336E}"/>
      <LNK A="TSK-21" B="{EB689DCE-A287-400C-9F0A-E8777716DDE6 }"/>
   </LGP>
   <LGP A="LGP31" B="3" D="urn:epc:id:sgtin:" E="Sample Of GS1 GTIN Links">
      <LNK A="PDT19" B="0764011.9854564"/>
      <LNK A="PDT20" B="0764012.9852311"/>
   </LGP>
   ……
</ISO11783LinkList>
```

E.3 Link — LNK

Type:

Coding data

Description:

The Link XML element links / maps / associates an XML element (object) in the ISO 11783 data transfer file set and a key value corresponding to an entity outside the scope of ISO 11783. The format and meaning of the key value is dependent on the LinkGroupType defined in the parent LinkGroup. An optional designator for the link can also be specified.

Included by XML element:

— *LinkGroup*

Table E.2 — Link attributes

Attribute	XML	Use	Type	Length/Range	Comment
ObjectIdRef	A	r	xs:ID	min.4 to max.14	Identifier of object, the scope of uniqueness of this identifier is the ISO 11783 data transfer file set.
LinkValue	B	r	xs:token	max. 255	Link value for the object.
LinkDesignator	C	o	xs:string	max. 32	Designator of Link

EXAMPLE

```
<ISO11783LinkList VersionMajor=”4” VersionMinor=”0”
TaskControllerManufacturer=”FarmCtrl” TaskControllerVersion=”1.0”
ManagementSoftwareManufacturer=”FarmSystem”
ManagementSoftwareVersion=”1.0” FileVersion=”3.2” DataTransferOrigin=”2”>
   <LGP A=”LGP10” B=”1” E=”UUIDs”>
      <LNK A=”FRM1” B=”{1059B14E-929F-4C4C-BCD4-C4F52A6A076A}”/>
      <LNK A=”PFD3” B=”{92043685-072D-4CAA-B3A8-C1E9D23BDB31}” C=”Headland”/>
      <LNK A=”PDT19” B=”{C89AA8C5-A9B3-4CA4-BE47-4449B9CD336E}”/>
      <LNK A=”TSK-21” B=”{EB689DCE-A287-400C-9F0A-E8777716DDE6 }”/>
      <LNK A=”PFD-3” B=”{8D96B98A-62C2-44D0-9709-29467A4169F3}”/>
   </LGP>
   ......
   <LGP A=”LGP20” B=”3” D=”urn:epc:id:sgtin:” E=”GS1 GTIN Links”>
      <LNK A=”PDT19” B=”0764011.9854564”/>
      <LNK A=”PDT20” B=”0764012.9852311”/>
   </LGP>
</ISO11783LinkList>
```

E.4 LinkGroup — LGP

Type:

Coding data

Description:

Link (LNK) XML elements may be created for different purposes and may involve different types of external entities (keys). The LinkGroup XML element groups Link elements of a single common key type and also holds data common to all the Link elements of the group. The LinkGroup’s type attribute (LGP.B) describes the key type contained in the LinkGroup.

Includes XML element:

— *Link*

Table E.3 — LinkGroup attributes

Attribute	XML	Use	Type	Length/Range	Comment
LinkGroupId	A	r	xs:ID	min.4 to max.14	Unique identifier (within scope of link list file) of LinkGroup Format: (LGP|LGP-)([0–9])+ Records generated on MICS have negative IDs.
LinkGroupType	B	r	xs:NMTOKEN	1 to 4	Selection of one attribute that represents the type of external entity (key) participating in the link: **1** - UUIDs Standardized UUIDs. All the group's Link child elements contain a valid UUID in the LinkValue (LNK.B) attribute. For newly-created elements the MICS and FMIS should create UUIDs using ISO/IEC 9834-8 (technically compatible to RFC 4122). Within the standard, the version 4 algorithm (Random) shall be used. **2** - Manufacturer Proprietary When attribute B is set to 2, a valid GLN is mandatory in the ManufacturerGLN attribute. **3** - Unique Resolvable URIs This type can be used to link data transfer elements with GS1 codes, for example. The complete link value is a concatenation of LinkGroupNamespace and LinkValue (LGP.D and LNK.B). **4** - Informational Resolvable URIs Type for a (loose) collection of informational links. The complete link value is a concatenation of LinkGroupNamespace and LinkValue (LGP.D and LNK.B).
ManufacturerGLN	C	o	xs:anyURI	max. 64	GS1 Global Location Number (GLN) of the manufacturer. For existing GLNs, or to apply for a new one, see http://www.gs1.org.
LinkGroupNamespace	D	o	xs:token	max. 255	For the resolvable URI types (3 and 4) this attribute holds a prefix value which is applied to all LinkValues of the contained Link elements In a LinkGroup of type 3 it shall hold the URI up to and including the last colon.
LinkGroupDesignator	E	o	xs:string	max. 32	Designator of LinkGroup
Link		o	xs:element		Includes a list of element Link

EXAMPLE

```
  <LGP A="LGP10" B="1" E="UUIDs">
     <LNK A="FRM1" B="{1059B14E-929F-4C4C-BCD4-C4F52A6A076A}"/>
     <LNK A="PFD3" B="{92043685-072D-4CAA-B3A8-C1E9D23BDB31}" C="Headland"/>
     <LNK A="PDT19" B="{C89AA8C5-A9B3-4CA4-BE47-4449B9CD336E}"/>
     <LNK A="TSK-21" B="{EB689DCE-A287-400C-9F0A-E8777716DDE6 }"/>
     <LNK A="PFD-3" B="{8D96B98A-62C2-44D0-9709-29467A4169F3}"/>
  </LGP>
  <LGP A="LGP31" B="3" D="urn:epc:id:sgtin:" E="GS1 GTIN/EAN">
     <LNK A="PDT19" B="0764011.9854564"/>
     <LNK A="PDT20" B="0764012.9852311"/>
  </LGP>
  <LGP A="LGP32" B="4" D="https://portal.bvl.bund.de/psm/jsp/DatenBlatt.jsp?kennr=" E="BVL
Links">
     <LNK A="PDT103" B="024658-00"/>
     <LNK A="PDT104" B="024145-00"/>
     <LNK A="PDT105" B="004960-00"/>
     <LNK A="PDT106" B="024309-00"/>
  </LGP>
```

E.4.1 LinkGroupType 1: Universally Unique Identifiers

The uniqueness of the object identifiers in the ISO 11783 data transfer file set is limited to the scope of the transfer file set itself. To enable FMIS and MICS to synchronize objects over a wider system scope, a universally unique identifier (UUID) can be associated to every XML element that has an object ID attribute.

If the MICS has the capability to create UUIDs, then the TC may add the new entries to the LINKLIST. XML file. Each newly created object identifier shall have the appropriate namespace letters followed by a negative decimal number.

The UUIDs shall be created according to ISO/IEC 9834-8 (technically compatible to RFC 4122). Within this standard, the random algorithm (version 4) shall be used.

The UUIDs of elements newly created on the MICS shall always be added to a type 1 LinkGroup with empty ManufacturerGLN. From within LinkGroups with empty ManufacturerGLN, any XML element of the data transfer file set shall have a maximum of one Link element.

EXAMPLE

```
  <LGP A="LGP10" B="1" E="UUIDs Sample, Group 1">
     <LNK A="FRM1" B="{1059B14E-929F-4C4C-BCD4-C4F52A6A076A}"/>
     <LNK A="PFD3" B="{92043685-072D-4CAA-B3A8-C1E9D23BDB31}" C="Headland"/>
     <LNK A="PDT19" B="{C89AA8C5-A9B3-4CA4-BE47-4449B9CD336E}"/>
     <LNK A="TSK-21" B="{EB689DCE-A287-400C-9F0A-E8777716DDE6}"/>
     <LNK A="PFD-3" B="{8D96B98A-62C2-44D0-9709-29467A4169F3}"/>
  </LGP>
  <LGP A="LGP20" B="3" D="urn:epc:id:sgtin:" E="GS1 GTIN Links">
     <LNK A="PDT19" B="0764011.9854564"/>
     <!- The same object (PDT19) can be referenced in multiple LinkGroups. The ->
     <!- exception to that rule are LinkGroups of type 1 with empty ManufacturerGLN. ->
     <LNK A="PDT20" B="0764012.9852311"/>
  </LGP>
  <LGP A="LGP11" B="1" E="UUIDs Sample, Group 2">
     <LNK A="FRM2" B="{84A46623-2D31-4427-BB4E-288FAC4A48AB}"/>
     <LNK A="PFD5" B="{F9B87F36-7F4A-445C-9083-C23D0F2492A8}"/>
  </LGP>
  <LGP A="LGP12" B="1">
     <!- <LNK A="FRM2" B="{446278DF-4035-4101-92FD-47525D1AF077}"/> ->
     <!- Not allowed! There shall be a maximum of one reference to the same XML ->
     <!- element from within LinkGroups of type 1 with empty ManufacturerGLN! ->
  </LGP>
```

E.4.2 LinkGroupType 2: Manufacturer-Proprietary Link Values

Link Groups of type 2 are suitable for exchanging manufacturer-proprietary keys. The samples below show mapping of Customers and Products to proprietary ID values. Note that multiple links to the same element are allowed. Within a proprietary LinkGroup, it is up to the manufacturer to know (and communicate to partners) how to use the Link elements, provided that they pass the validation against the schema. The key value of the LNK element can be a UUID value.

EXAMPLE

```
  <LGP A="LGP29" B="2" C="urn:epc:id:sgln:0614141.33254.1" D="" E="Proprietary Customer
Codes">
     <LNK A="CTR103" B="1909-19"/>
     <LNK A="CTR104" B="1909-20"/>
     <LNK A="CTR105" B="1909-21"/>
     <!- CTR105 is customer 'Waldo & Beer' ->
     <!- Since this was a merger (of customer Waldo and customer Beer) there ->
     <!- exists also the former id of customer Beer (1909-34). Therefore we have ->
     <!- a second LNK element for CTR105. ->
     <LNK A="CTR105" B="1909-34"/>
     <LNK A="CTR106" B="1909-22"/>
  </LGP>
  <LGP A="LGP29" B="2" C="urn:epc:id:sgln:0614141.33254.1" D="" E="Proprietary Product
Codes">
     <LNK A="PDT943" B="{3950035A-FE41-4A5B-89BB-D9AAF1F5FB99}" C="Product X"/>
     <LNK A="PDT944" B="{DC759896-9026-4800-A548-657001A1FCF9}" C="Product Y"/>
     <LNK A="PDT945" B="{D8159354-BCC6-48A6-B566-23193AB95841}" C="Product Z"/>
  </LGP>
  <LGP A="LGP30" B="2" C="urn:epc:id:sgln:0123456.78901.2" D="" E="Product Codes from an
FMIS">
     <LNK A="PDT943" B="{2AF969DA-E47A-4F18-B34F-99BD00C70A2D}" C="Advise System
(agbusiness)"/>
     <LNK A="PDT943" B="{F95829FD-7AFB-46A2-B213-9A7F00D8EDB1}" C="Advise System
(wizard)"/>
     <!- In this case, the acquisition of two chemical manufacturers by a single parent
company ->
     <!- has resulted in the same product (i.e. even having the same registration number)
->
     <!- being marketed under two different manufacturers, but with the same name. ->
     <!- A participating FMIS company's controlled vocabulary of approximately 20,000
products ->
     <!- contained 92 sets of 2 or more products: 198 total records, approximately 1% of
the total. ->
  </LGP>
  ...
  <!- The following Type 1 (UUID) Link Group has nothing to do with the example, but
illustrates that there may be ->
  <!- several Link Groups in one LINKLIST.XML file. ->
  <LGP A="LGP30" B="1">
     <LNK A="PDT103" B="{84A46623-2D31-4427-BB4E-288FAC4A48AB}"/>
     <LNK A="PDT104" B="{F9B87F36-7F4A-445C-9083-C23D0F2492A8}"/>
  </LGP>
```

E.4.3 LinkGroupType 3: Unique Resolvable URIs

Link Group type 3 relates an XML element of the data transfer file set to a universal resource identifier, or URI, which must uniquely identify the element and be consistent over time. Examples of URIs include:

de.wikipedia.org
user@example.com:8080
192.0.2.16:80
[2001:db8::7]

The concatenation of LGP.D and LNK.B shall be a resolvable URI. This is to omit redundant data and also to make the content of a LinkGroup programmatically recognizable.

In the special case when the URI in question is a Universal Resource Name, or URN, the URN must be split at the last occurrence of a colon used to specify the namespace. The first part (the namespace, including the trailing colon) goes into attribute D of the LinkGroup (LinkGroupNamespace). The second part is the LinkValue (LNK.B). Note that multiple links to the same element are allowed.

EXAMPLE


```
  <LGP A="LGP31" B="3" D="urn:epc:id:sgtin:" E="GS1 GTIN/EAN">
     <LNK A="PDT19" B="0764011.9854564"/>
     <!- Completes to urn:epc:id:sgtin:0764011.9854564 ->
     <LNK A="PDT20" B="0764012.9852311"/>
  </LGP>
  <LGP A="LGP32" B="3" D="urn:epc:id:sgln:" E="GS1 GLN">
     <LNK A="CTR9" B="4260159.94000.8"/>
     <LNK A="CTR10" B="4014689.00000.4"/>
     <LNK A="CTR11" B="4399901.95917.0"/>
     <LNK A="CTR11" B=" 4399902.09257.9"/>
     <!- Note how it is possible to have more than one reference to "CTR11" within the
same LinkGroup of type 3 ->
  </LGP>
  <LGP A="LGP33" B="3" D="" E="Base Station IP Addresses">
     <LNK A="BSN1" B="192.0.2.16"/>
     <LNK A="BSN2" B="192.0.2.22"/>
  </LGP>
```

E.4.4 LinkGroupType 4: Informational Resolvable URIs

The purpose of a Link Group of type 4 is to provide a mechanism for storing links. These could be the homepage of a manufacturer, farm, a product information sheet, or similar.

Type 4 is suitable for volatile internet pages. Type 4 Links may not be used for identification or mapping. The concatenated value of LGP.D and LNK.B shall be a resolvable URI.

Within Link Groups of type 4, there may be more than one Link element referring to the same XML element of the data transfer file set.

A type 4 Link Group can make use of the LinkGroupNamespace. If the LinkGroupNamespace is nonempty, the URI value is a concatenation of LGP.D and LNK.B. There is no regulation regarding where to split the URI.

EXAMPLE

```
  <LGP A="LGP29" B="4">
     <LNK A="PDT9" B="http://www.bayer.de/products?234234"/>
     <LNK A="PDT19" B="http://www.monsanto.com/7832637abc"/>
     <LNK A="PDT27" B="http://www.kemira.com/service/products/info#6733a93"/>
     <LNK A="PDT27" B=" https://portal.bvl.bund.de/psm/jsp/DatenBlatt.jsp?kennr=024658-00
"/>

     <LNK A="CTR22" B="http://www.gut-schrockwede.de/start.jsp"/>

     <LNK A="WKR7" B="http://www.facebook.com/max.muster?fref=ts" C="facebook"/>
     <LNK A="WKR7" B="http://www.max-muster.de" C="personal homepage"/>
  </LGP>
  <LGP A="LPG30" B="4" D=" http://en.wikipedia.org/wiki/">
     <LNK A="CTP209" B="Soybean"/>
     <LNK A="CTP210" B="Maize"/ >
     <LNK A="CTP211" B="Wheat"/ >
     <LNK A="CTP212" B="Oat"/ >
  </LGP>
  <LGP A="LGP31" B="4" D="https://portal.bvl.bund.de/psm/jsp/DatenBlatt.jsp?kennr=" E="BVL
Links">
     <LNK A="PDT103" B="024658-00"/>
     <!- Complete Link: https://portal.bvl.bund.de/psm/jsp/DatenBlatt.jsp?kennr=024658-00
->
     <LNK A="PDT104" B="024145-00"/>
     <LNK A="PDT105" B="004960-00"/>
     <LNK A="PDT106" B="024309-00"/>
  </LGP>
  <!- LGP31 is LinkGroupType 4 because LGP.D referrs to an existing web site. -->
  <!-So it's a URL depending on the existance of https://portal.bvl.bund.de-->
```


Annex F
(normative)

TC Functionalities and Device Descriptor Object Pool definitions

F.1 General

With the set of objects provided in Annex A, many different device descriptor object pools can be constructed for one and the same device. Next to variation in the construction of device descriptors, the extent to which products implement the communication methods and system features that are described in this part of ISO 11783 may vary. For a system in which products from different manufacturers are required to function at a common feature level, a further classification and definition of these product features is required.

This annex provides guidelines and examples for product designers regarding which sets of features to design for and how to structure a device descriptor such that it best reflects the controllable aspects of the device and that it provides data in way that is easiest to handle in data processing applications.

These guidelines and examples have been drafted in collaboration with AEF,[2] facilitating the use of the ISO 11783 standards in products by means of conformance testing and implementation support activities.

F.2 TC functionalities

The definition of distinct sets of ISO 11783-10-based features is provided by TC functionalities. TC functionalities identify the features that need to be implemented in products to interface and operate correctly in a system. To support future developments and accommodate product's technical limitations, TC functionalities are versioned by a generation number and have in some cases optional features identified. The use of the functionality generation and the functionaly options enable product designers to specify which parts of this part of ISO 11783 are implemented and what the operational limitations are for compatibility with a certain set of system requirements.

The following TC functionalities are defined:

a) TC-BAS (basic) generation 1

b) TC-GEO (geo-based) generation 1

c) TC-SC (section control) generation 1

d) LOG (data logger) generation 1

These TC functionalities, their requirements, and the functionality options are explained in the following clauses.

F.2.1 TC-BAS generation 1 functionality

The functionality TC-BAS is defined as task-based collection of totals, i.e. the reading and writing of total values for a task for one or more of the TC client's device elements. Totals are accumulated values for input or output resources, distance, time, or rates of those values (e.g. total area, total harvest mass, etc.); totals are not necessarily site-specific and therefore do not require time or position data.

All DDIs defined as totals in ISO 11783-11 and listed in the ISOBUS Data Dictionary shall be supported by a TC service type in order for that TC server to conform to the TC-BAS functionality.

A TC-BAS conform TC client can provide a subset of the totals. At least DDI 119, total time, is required to be provided by the TC client to be TC-BAS conform. In case a rate DDI is defined within the DDOP also, the corresponding total value shall be provided. A list of additional recommended total values per type of device is provided in Table F.1

Table F.1 — Recommended Total DDIs per Device Class

Device Class		DDIs (defined in ISO 11783-11)
1	Tractor	117, 118, 119, 120, 148
2	Tillage	116, 117, 119
3	Secondary Tillage	116, 117, 119
4	Seeders/Planters	81 or 82, 116, 117, 119
5	Fertilizers	81, 116, 117, 119
6	Sprayers	80, 116, 117, 119
7	Harvesters	89 or 90 or 91, 116, 117, 119
8	Root Harvesters	116, 117, 119
9	Forage	116, 117, 119
10	Irrigation	80, 119
11	Transport/Trailers	117, 119
12	Farmstead operations	119, 120
13	Powered auxiliary devices	116, 117, 119
14	Special crop	116, 117, 119
15	Earthworks	119
16	Skidders	119, 148
17	Sensor Systems	119
25	Slurry Applicators	80, 116, 117, 119

The FMIS, TC, and TC client implementations according to the TC-BAS generation 1 shall publish the Task Controller version number 3 in the data transfer file set (e.g. ISO11783_TaskData) and the Version message.

FMIS products shall consider the limitations of embedded implementations of TC and vice versa, TC implementations shall avoid generation of large task data transfer file set. At a minimum, a task data transfer file set with 20 000 XML elements in total and 2 000 XML elements per element type shall be supported by FMIS and TC implementations.

FMIS or TC server implementations according to TC-BAS generation 1 are not required to process the following XML elements:

a) OperationTechnique

b) OperationTechniqueReference

c) CodedCommentGroup

d) Connection

e) All geometry or geo referencing elements

Not being required to process these XML elements means that when any of these elements are present in the data transfer file set, they are allowed to be skipped by the data processing. The rest of the content of the data transfer file set that contains these XML elements shall however be read and the not supported elements shall not cause the data transfer file set to be rejected by the data processing.

A series of device descriptor object pools that meet the TC-BAS functionality generation 1 are provided in F.3.

F.2.2 TC-GEO generation 1 functionality

The functionality TC-GEO defines the processing of site specific values for a task for the TC client's device elements based on their geographic positions. TC-GEO requires a position source such as a GNSS receiver connected to the TC. Although TC-GEO functionality refers to varying the rate of applications, the TC-GEO functionality is not limited to devices with controllable process data elements. TC-GEO also applies, for example, to TC clients without controllable elements that provide "as-applied" process data for mapping purposes (e.g. combine harvester yield sensor data, or planter population data).

All DDIs defined in ISO 11783-11 and listed in the ISOBUS Data Dictionary shall be supported by a TC service type in order for that TC server to conform to the TC-GEO functionality.

A TC-GEO conform TC client can provide a subset of the DDIs that are relevant for the processes it controls or provides data to be logged for. A list of recommended DDIs per type of device is provided in Table F.2. In case the device has rate control processes, at least one setpoint rate and actual rate DDI pair is recommended to be supported. In addition to controlling application rates, other setpoint type DDIs can be controlled as well, e.g. working depth for tillage operations.

Table F.2 — Recommended Rate DDIs per Device Class

Device Class		DDIs (defined in ISO 11783-11)
4	Seeders/Planters	6, 7, 11, 12, 16, 17
5	Fertilizers	1, 2, 6, 7
6	Sprayers	1, 2, 6, 7
7	Harvesters	7, 84, 181
8	Root Harvesters	7, 84
9	Forage	7, 84, 181
10	Irrigation	1, 2
25	Slurry Applicators	1, 2, 6, 7

When the TC-GEO functionality is supported, then the functionality TC-BAS shall be supported as well.

The FMIS, TC, and TC client implementations according to the TC-GEO generation 1 shall publish the Task Controller version number 3 in the data transfer file set (e.g. ISO11783_TaskData) and the Version message.

Processing of the Grid types 1 and 2 is required. The minimum supported number of rows and columns in a Grid is 350 rows and 350 columns.

The processing of Polygon-based GEO presciptions is optional. When a TC supports Polygon-based prescriptions, then this shall be specified in the Control function functionalities message. The Control function functionalities message is defined in ISO 11783-12.

The number of position based control channels supported by the TC server shall be at a minimum 1 channel. The number of position based control channels that a TC server can handle in parallel or that a TC client can control shall be specified in the Control function functionalities message and the Version message. The reported number of control channels in the Version message shall be used by the TC client to limit the number of control channels reported in the device descriptor transmitted to the TC server.

A series of device descriptor object pools that meet the TC-GEO functionality generation 1 are provided in F.3.

F.2.3 TC-SC generation 1 functionality

The functionality TC-SC defines the on/off control of TC client's device elements based on their geographic position. The TC-SC functionality includes the capability to determine whether these device elements encountered previously covered or other specific (e.g. headland or out-of-field) areas to minimize application overlaps and skips. The TC-SC functionality requires a position source such as a GNSS receiver connected to the TC.

The TC-SC server shall support at least 1 boom and 3 sections while the TC client conforms starting with 1 boom and 1 section. If the TC-SC server can control less booms or sections than the TC client can report and control physically, then the TC client has to adjust the number of booms and sections in its device descriptor down to match the number of booms and sections that the TC server can control. In this case, the TC client shall combine its physically controllable device elements such that a lower number of booms or sections is communicated the TC server. The number of booms and sections that is either supported by a TC server or that can be controlled by a TC client shall be specified in the Control function functionalities message and the Version message. The TC client shall use the information supplied in the Version message to adjust its device descriptor object pool to the TC server in case the number of booms or sections that TC client can control exceeds the TC server TC-SC capabilities.

The TC-SC functionality is independent of the functionalities TC-BAS or TC-GEO; it does not require one of these to be supported in combination with the TC-SC functionality. For providing only the TC-SC functionality, there is no requirement to support processing of a data transfer file set or a connection between an FMIS and a TC.

The TC and TC client implementations according to the TC-SC generation 1 shall publish the Task Controller version number 3 in the Version message.

The DDIs that shall be supported by the TC-SC server and the TC client for the TC-SC functionality are listed in Table F.3.

Table F.3 — DDIs required to be supported for TC-SC functionality

DDIs (defined in ISO 11783-11)
67, 70, 134, 135, 141,142, 143, 160, 161 .. 176, 205, 206, 290 .. 305

A series of device descriptor object pools that meet the TC-SC functionality generation 1 are provided in F.3.

F.2.4 LOG generation 1 functionality

The functionality LOG is defined as a recording of lifetime totals, i.e. the reading of lifetime total values for one or more of the LOG client's device elements. Lifetime totals are accumulated values for input or output resources, distance, time, or rates of those values (e.g. lifetime total area, lifetime total harvest mass, etc.). Lifetime totals are not necessarily site-specific and therefore do not require time or position data.

All DDIs defined as totals in ISO 11783-11 and listed in the ISOBUS Data Dictionary shall be supported by a LOG service type in order for that LOG server to conform to the LOG functionality.

A LOG conform LOG client may provide a subset of the lifetime totals. At least DDI 274, lifetime total time, is required to be provided by the LOG client to be LOG conform. In case a rate DDI is defined within the DDOP also, the corresponding lifetime total value shall be provided. A list of additional recommended lifetime totals values per type of device is provided in Table F.4,

Table F.4 — Recommended Total DDIs per Device Class

Device Class		DDIs (defined in ISO 11783-11)
1	Tractor	272, 273, 274, 275, 276
2	Tillage	271, 272, 274

Table F.4 *(continued)*

Device Class		DDIs (defined in ISO 11783-11)
3	Secondary Tillage	271, 272, 274
4	Seeders/Planters	271, 272, 274, 266 or 267
5	Fertilizers	271, 272, 274, 266
6	Sprayers	271, 272, 274, 325
7	Harvesters	271, 272, 274 and 268 or 269 or 270
8	Root Harvesters	271, 272, 274
9	Forage	271, 272, 274
10	Irrigation	274, 325
11	Transport/Trailers	272, 274
12	Farmstead operations	274, 275
13	Powered auxiliary devices	271, 272, 274
14	Special crop	271, 272, 274
15	Earthworks	274
16	Skidders	274, 276
17	Sensor Systems	274
25	Slurry Applicators	325, 271, 272, 274

The FMIS, LOG server, and LOG client implementations according to the LOG functionality generation 1 shall publish the Task Controller version number 4 in the data transfer file and the Version message.

FMIS products shall consider the limitations of embedded implementations of LOG and vice versa, LOG implementations shall avoid generation of large task data transfer files. At a minimum, task data transfer files with 20000 XML elements in total and 2000 XML elements per element type shall be supported by FMIS and LOG implementations.

A series of device descriptor object pools that meet the LOG functionality generation 1 are provided in F.3.

F.3 Device descriptor object pools

For a number of devices that are commonly used in agricultural field operations, device descriptor object pool examples have been drafted to guide the implementation of the ISO 11783-10 standard. The device descriptor object pool examples are grouped by TC functionality and follow a progression from a basic device with few features to more complex devices. Alongside the examples, a set of design rules relevant for each step in this progression is defined. These design rules shall be followed when variants of the device descriptor object pools are implemented.

Some of the device descriptor object pool examples in this section are used across multiple functionalities. Where this is the case only, the attributes relevant for the functionality under which the examples are listed are present in the figures. This means that when a device supports more than one functionality, the device descriptor shall contain the combined set attributes from the corresponding examples from each functionality.

F.3.1 Boom device element definition

In many of the device descriptor object pools, the concept of a “boom” is mentioned. A boom is either a device element of type “Function” or a device element of type “Device”. In case the boom is a device element of type “Device” this is the root device element. The concept of a boom is used to indicate an individual operation of the device, which results in a single coverage layer, the worked area, for that device. In case a device has multiple booms, then multiple operations or coverage layers can be tracked and controlled simultaneously for that device. The device element that represents a boom can have a set of device elements of type “Section” or “Bin” as child device elements to describe a more detailed level of coverage control and to support devices that can apply multiple products through a single boom. Examples of booms are the header on a harvester, a sprayer boom with nozzles on a liquid applicator, or a planter bar with planting elements on a planter.

F.3.2 Device descriptor diagram notation legend

The notation used in the device descriptor object pool diagrams uses the legend depicted in Figure F.1 to distinguish device element types.

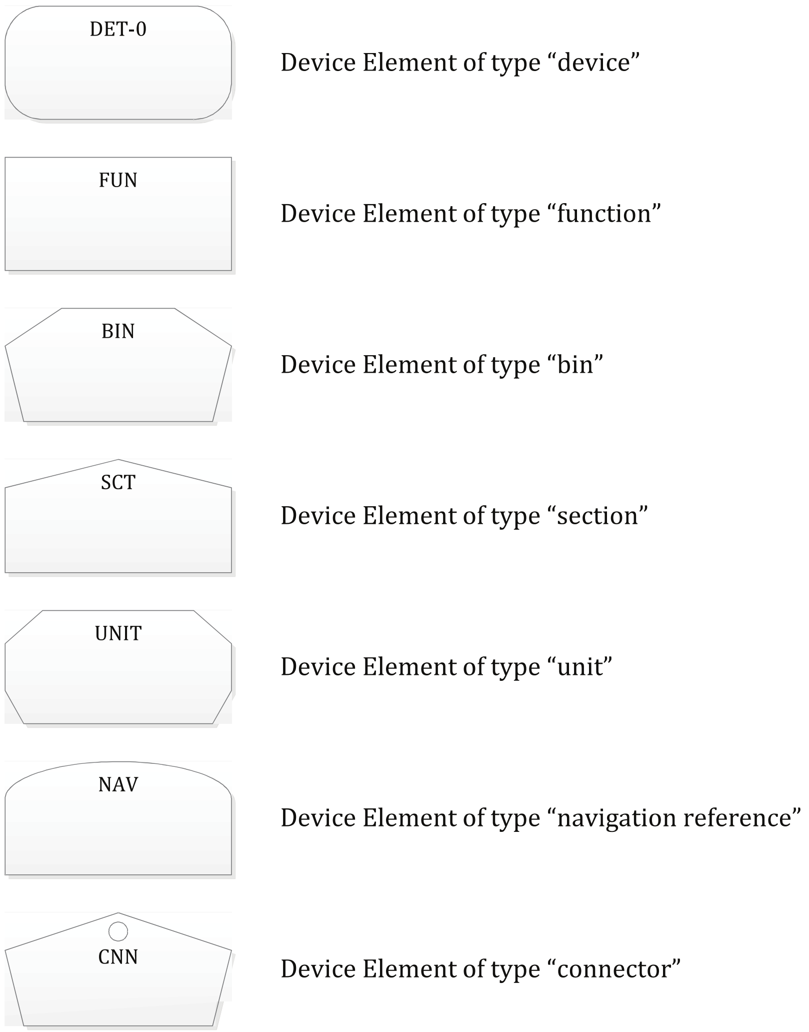


Figure F.1 — Device element type legend

Device elements contain device process data and device properties. The annotation and the symbols used at this device element attribute level of detail are provided in Figure F.2. In addition to the symbols, the background fill colour indicates whether process data are setable (blue background) or only readable (white background).

DDI		Property
DDI		Process Data (read only)
DDI		Process Data, Lifetime Total
DDI		Process Data, Total
DDI		Process Data, Setpoint
DDI		Process Data, Control Source
DDI		Time Interval Measurement
DDI		On Change Measurement
DDI		Distance Interval Measurement

Figure F.2 — Device element attribute legend

F.3.3 TC-BAS generation 1 device descriptor object pools

The most basic version of a TC-BAS generation 1 conform device descriptor object pool is shown in Figure F.3. This device descriptor contains only a single attribute which is the total time, indicated with a pencil symbol to be a setable total. The time interval measurement symbol indicates that this total DDI can be requested to be transmitted at a certain time interval.

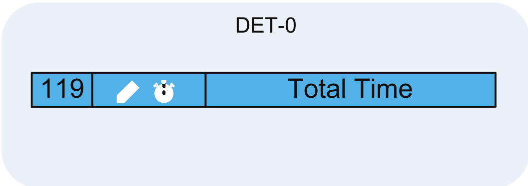


Figure F.3 — Single operation Device without a Function Device Element and no geometry information

Devices that include a rate DDI shall provide a corresponding total DDI for the rate (Figure F.4).

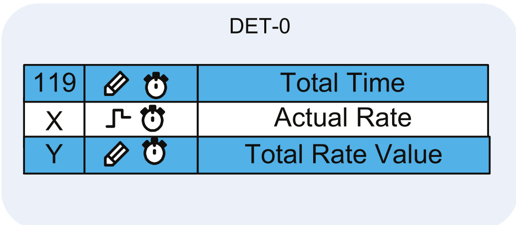


Figure F.4 — Single operation Device with an Actual Rate DDI without a Function Device Element and no geometry information

The example provided in Figure F.5 adds an optional connector device element to the device descriptor object pool and specifies the working width of the device in the root device element. Two variations regarding providing working width and connection offsets as properties or as process data are shown. When the value of these attributes is a constant on the device, then a property is recommended to be used. When the value of these attributes can vary during operation of the device, then communicating this through process data are mandatory. A connector type attribute is recommended to be part of a connector device element.

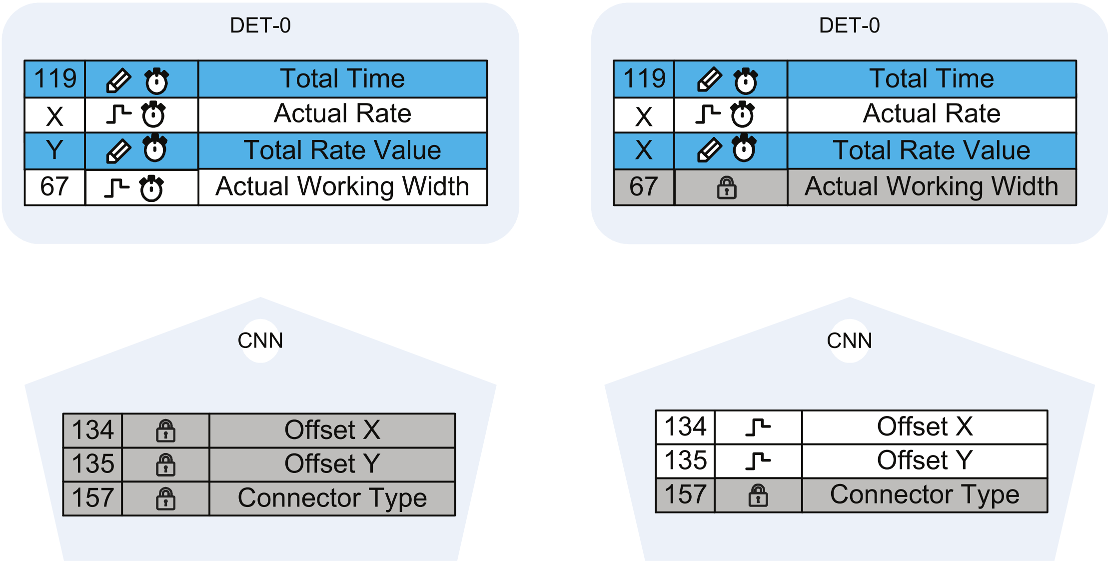


Figure F.5 — Single operation Device with an Actual Rate DDI and connector type and working width information

The device descriptor object pools depicted thus far did not have a separate functional device element from the device's root device element. This separation is shown in Figure F.6 for the most basic device descriptor on the left-hand side and with an optional connector device element on the right-hand side.

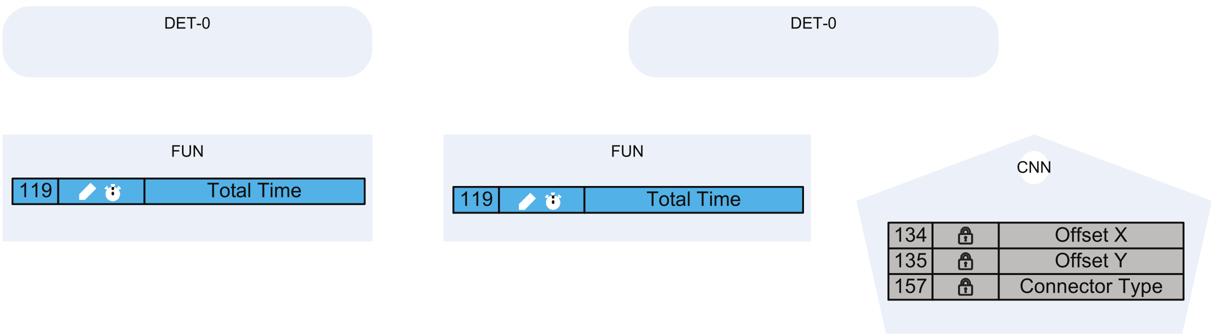


Figure F.6 — Single operation Device with a separate Function device element with and without a connector type device element

A similar separation of attributes embedded in a function for a slightly more complex device is provided in Figure F.7. Next to totals, also different variants of rate and working width attributes are specified.

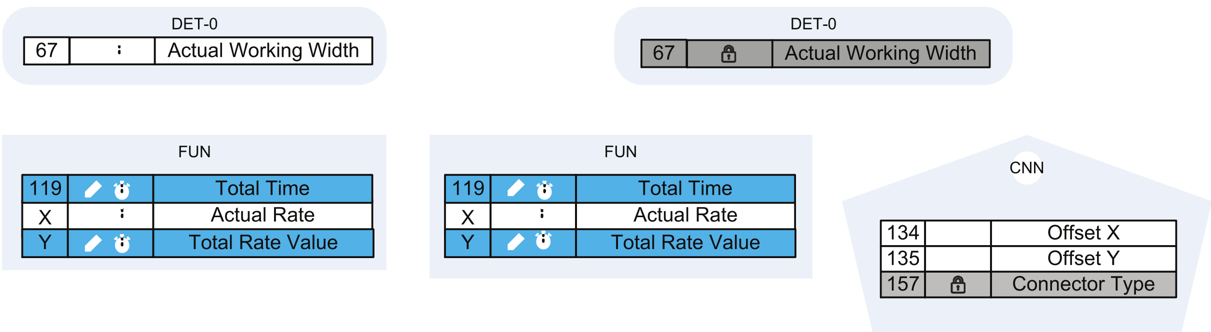


Figure F.7 — Single operation Device with actual rate and working width DDIs and a separate Function device element with and without a connector type device element

The totals can be provided in multiple device elements. In Figure F.8, an example is provided of a total time that is available for the Device, specified in the root device element, as well as a total time that is recorded for a specific device element, the Function. Each total time is updated based on the logic of the device element it is specified in. All totals from the Function level do not necessarily need to match the total of the Device level. For the TC-BAS generation 1functionality, a total time DDI at Device level is mandatory; any total time DDIs inside other device elements are optional.

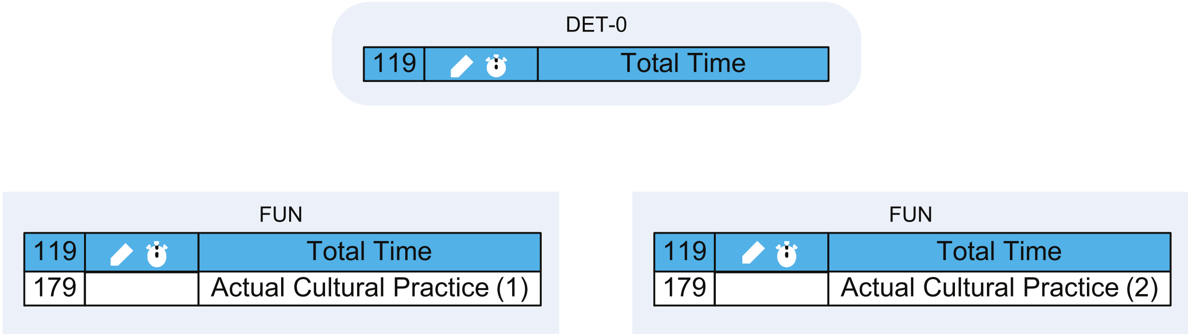


Figure F.8 — Total times present at Device level and Function level

A couple other optional attributes added to Figure F.8 are the Actual Cultural Practice DDIs. In this example they represent a fertilizer operation and a seeding or planting operation, both executed by the same device. The total times of each operation can thus be independently recorded.

Instead of a generic Function device element, the rate DDIs and associated totals can also be specified as attributes of a Bin device element. Two examples of these device descriptor object pools are provided in Figure F.9. In this example, the rate, working width, and connector information are recommended. The root device element represents the boom of the device. The location of the Bin device elements as a child of the boom specifies that the product from this bin is distributed through that boom. It is not allowed to have the same rate DDI attached to both the root device element and the bin device element. The total time in the root device element is mandatory and the total rate DDI shall be located in the same device element that has the actual rate DDI.

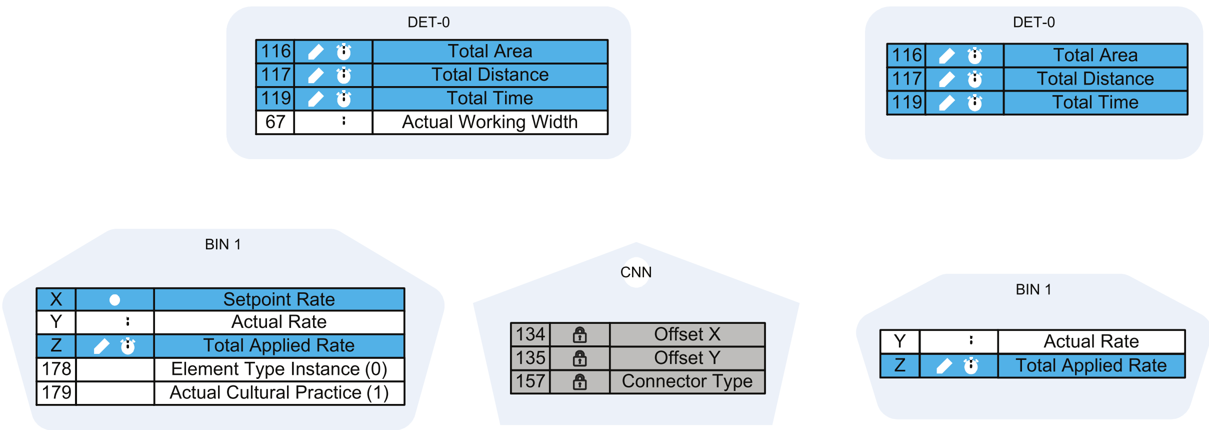


Figure F.9 — Single operation Device with a Bin device element

Figure F.10 expands on Figure F.9 by adding a secondary operation to the device and separate Function device elements that represent two independent product application booms. In this example, each Function device element represents a boom. The location of the Bin device elements as children of the booms specifies that the products from these bins are distributed through these booms. The same rate DDI shall not occur at both the Function and the Bin device elements, nor be attached to the root device element and the boom Function device element simultaneously.

The first function on the left-hand side in Figure F.10 represents a fertilizer operation (Actual Cultural Practice DDI has value 1), the second function represents a seeding or planting operation (Actual Cultural Practice DDI value is 2). The total values inside the root device element represent the overall device totals. The total values in the Function device elements represent just the total values for each operation. The sum of the function level total values is not necessarily the same as the overall device total values. The total time in the root device element is mandatory, total time DDIs in the Function type device elements are optional.

In case of multiple booms, the Actual Cultural Practice DDI (179) shall be placed either in the boom Function device element or in the underlying Bin device element.

The Element Type Instance DDI (178) shall be used to identify a bin to the operator to enable to select from which bin a product is applied. This implies that the Element Type Instance DDI shall be placed inside the Bin device element.

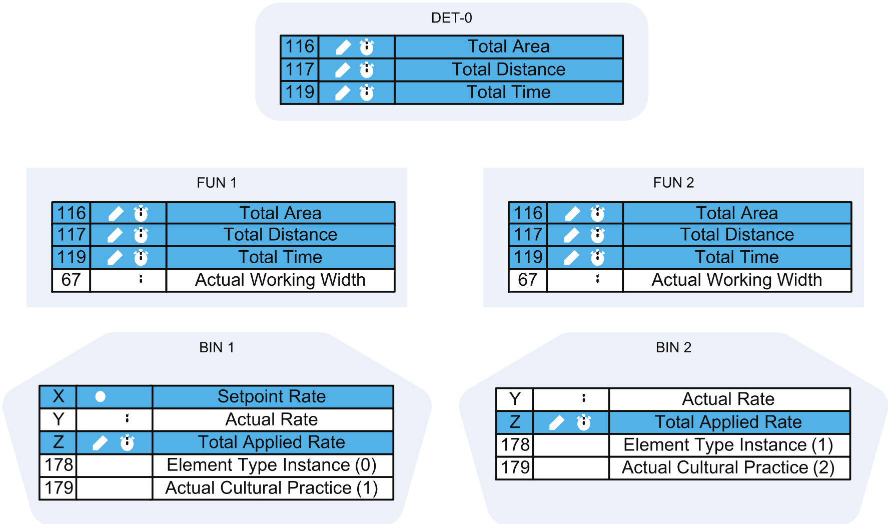


Figure F.10 — Multi operation Planter with two Functions and two Bins

The use of the Actual Cultural Practice DDI is further illustrated in Figure F.11. In this example of a baler device descriptor object pool, the Actual Cultural Practice DDI value 5 identifies this device as a baler rather than, for example, a mower. Without the use of the Actual Cultural Practice DDI, as the device class for balers and mowers is forage harvesting, these devices would not be distinguishable.

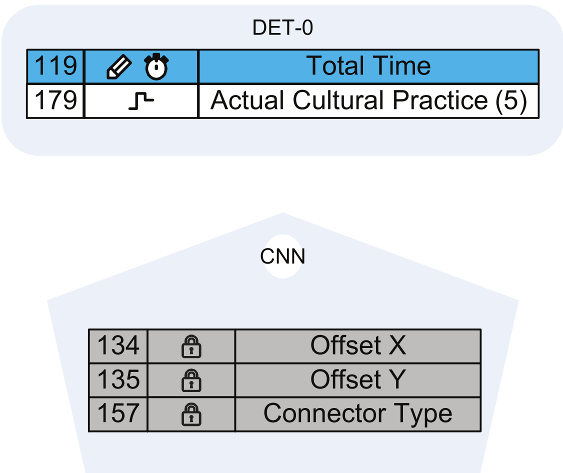


Figure F.11 — Baler device descriptor, device class forage harvesting

Figure F.12 shows an example of a baler device descriptor that includes two levels of bale counting. In this example, the total values of the precut and uncut bales shall equal the yield total count DDI value.

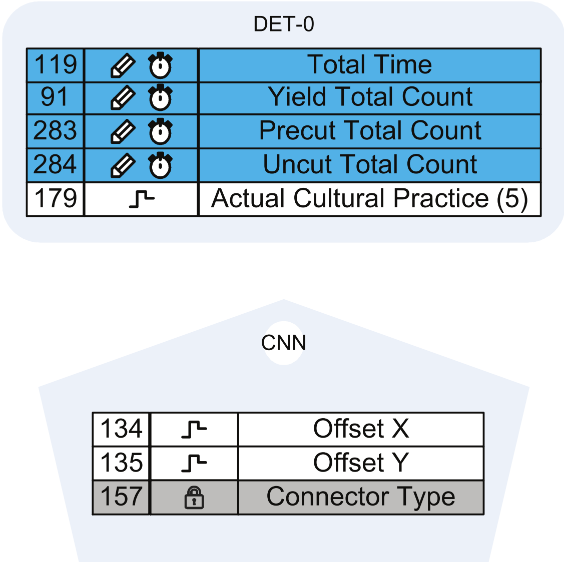


Figure F.12 — Baler device descriptor, multiple totals

Another example for a specific device class is provided in Figure F.13. In this example, a set of totals for a tractor is specified to be able to be recorded. The Connector device elements and Navigation device element are optional for the TC-BAS functionality.

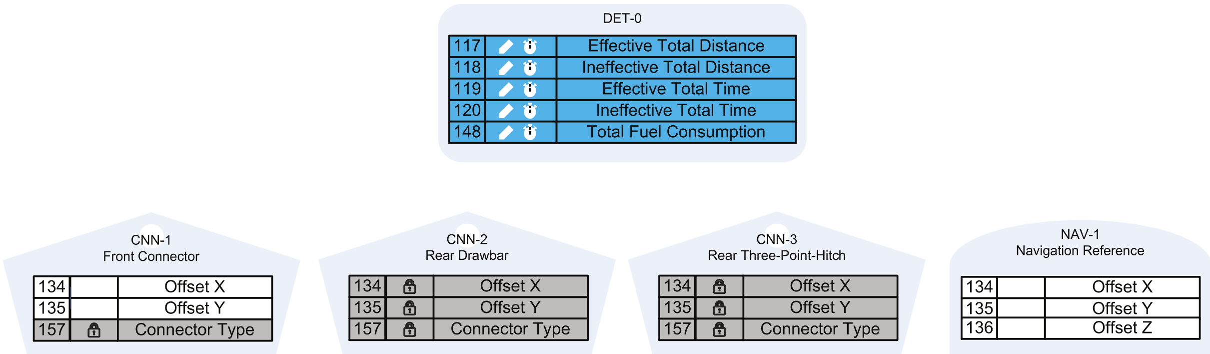


Figure F.13 — Tractor device descriptor, multiple totals

If a peer control capable device (e.g. online sensor) has a setpoint value source and a setpoint value user of the same setpoint DDI, then these DeviceProcessDataObjects shall not be referenced in a single device element. Figure F.14 illustrates this case: two separate device elements are required when the same DDI is present as both a setpoint value source and a setpoint value user.

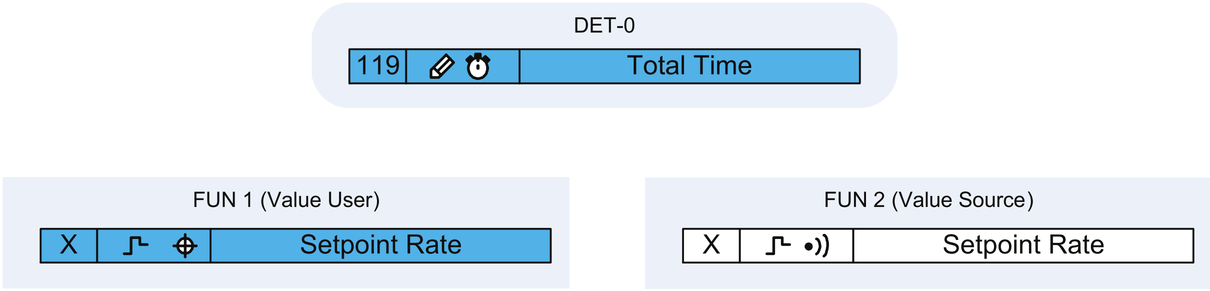


Figure F.14 — Sensor device description, set point value source/user of same DDI

F.3.4 TC-GEO generation 1 device descriptor object pools

The minimum device descriptor object pool structure that meets the TC-GEO functionality is shown in Figure F.15. This device includes a single boom. The minimum number of boom to be present in a TC-GEO conform device is one boom. The minimum functionality of a boom consists of providing actual values that can be recorded spatially. At a minimum, the measurement trigger "Time Interval" shall be supported for these actual value process data attributes.

F.3.4.1 Work state

The boom shall include an actual work state process data definition. This process data shall support the measurement trigger "Change Threshold".

F.3.4.2 Geometry

Each boom shall provide its full geometry definition to enable the TC and FMIS to derive the worked area. At a minimum, the geometry is defined by properties or process data for the offset X, offset Y, and maximum working width DDIs. The offset X and offset Y DDIs are required for each boom. The boom geometry shall not be derived from its child section device elements.

A mix between process data and property attributes is allowed in the device descriptor object pool. In Figure F.17, the Connector Type attribute is a property which will not change over the lifetime of the device descriptor object pool structure, whereas the rest of the geometry attributes may change upon e.g. Device configuration modifications.

The measurement trigger "On Change" is recommended for the geometry process data. Upon system startup, a TC that supports TC-GEO functionality shall query the Device for its geometry DDI values to ensure that the correct geometry is used to determine coverage and location based commands.

The TC-GEO functionality requires a connector that specifies the connection point to the tractor for a mounted or trailed implement. The definition of multiple connectors is allowed in a device descriptor object pool. Connector and Navigation type device elements shall only be placed directly below the root device element. The connector and navigation device elements shall include X and Y offset properties or process data, as well as a connector type definition.

F.3.4.3 Working width

The maximum working width DDI listed as required under the geometry represents the width that the prescription setpoint can be applied to. This maximum width can change dynamically on certain types of implements or can be a static property when the prescription setpoint is applied through a fixed width boom. Optionally, a boom may provide an actual working width. An actual working width at Device or Function level shall depend on the actual work states of its underlying sections. This enables a Device or Function to report the actual working width as the sum of the working widths of the sections that are "On", while the TC or FMIS uses the section working widths to determine the section geometry and to map where to sections were "On" or "Off".

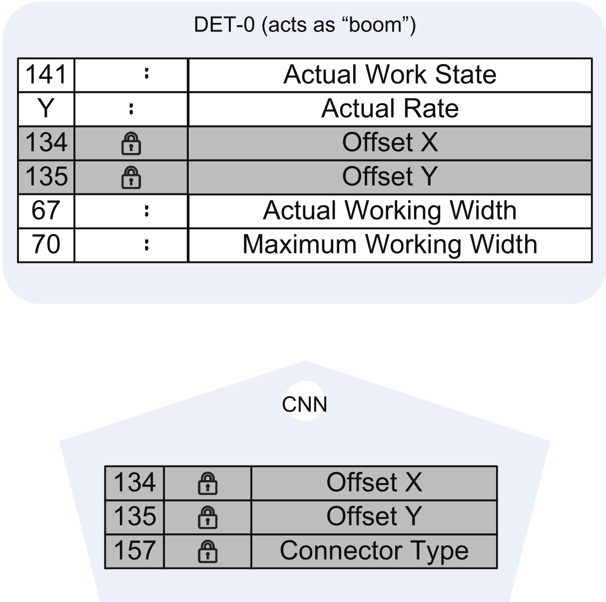


Figure F.15 — Actual rate and geometry present in device descriptor

F.3.4.4 Prescription control state

If a boom includes a setpoint value process data attribute, the boom shall also include a prescription control state process data attribute at the same level as the setpoint value process data. This prescription control state process data attribute shall be settable and shall support an on-change measurement trigger. In addition to the specification in the ISOBUS Data Dictionary for the prescription control state DDI (158), the following rules apply:

a) At task stop, the TC client shall internally set its prescription control state to DISABLED/OFF.

b) The TC shall set the prescription control state to ENABLED/ON before sending rate value commands to the TC client.

c) Upon reception of a prescription control state command to ENABLED/ON, the TC client shall reset the setpoint process data attribute that corresponds with this prescription control state to the determined safe state value for this setpoint process data attribute.

d) The TC shall use an on-change measurement command to receive prescription control state changes from the TC client.

A boom that includes a setpoint value process data attribute shall also include the corresponding actual value process data attribute. This variant of Figure F.15 is shown in Figure F.16

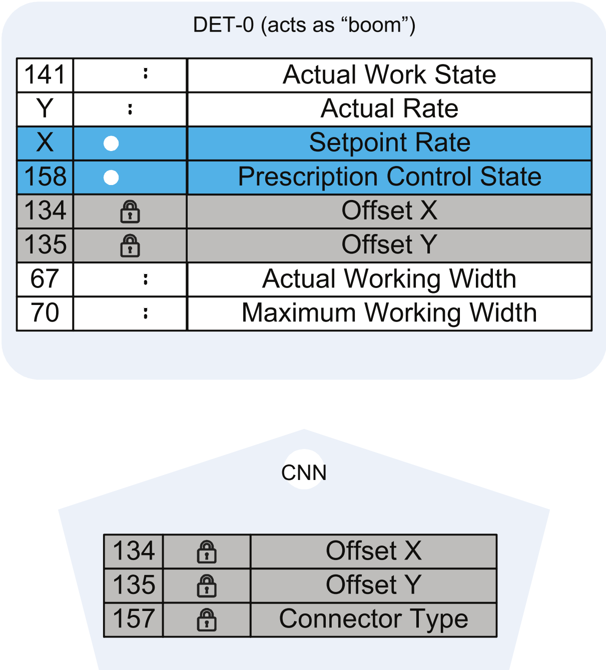


Figure F.16 — Setpoint rate added to a device descriptor

F.3.4.5 Boom sections

Figure F.17 shows the addition of section device elements to a boom. In this example, the TC-GEO conform boom includes the requirements for the TC-SC functionality. Both TC-GEO (Prescription Control State) and TC-SC (Section Control State) attributes are present in the boom device element.

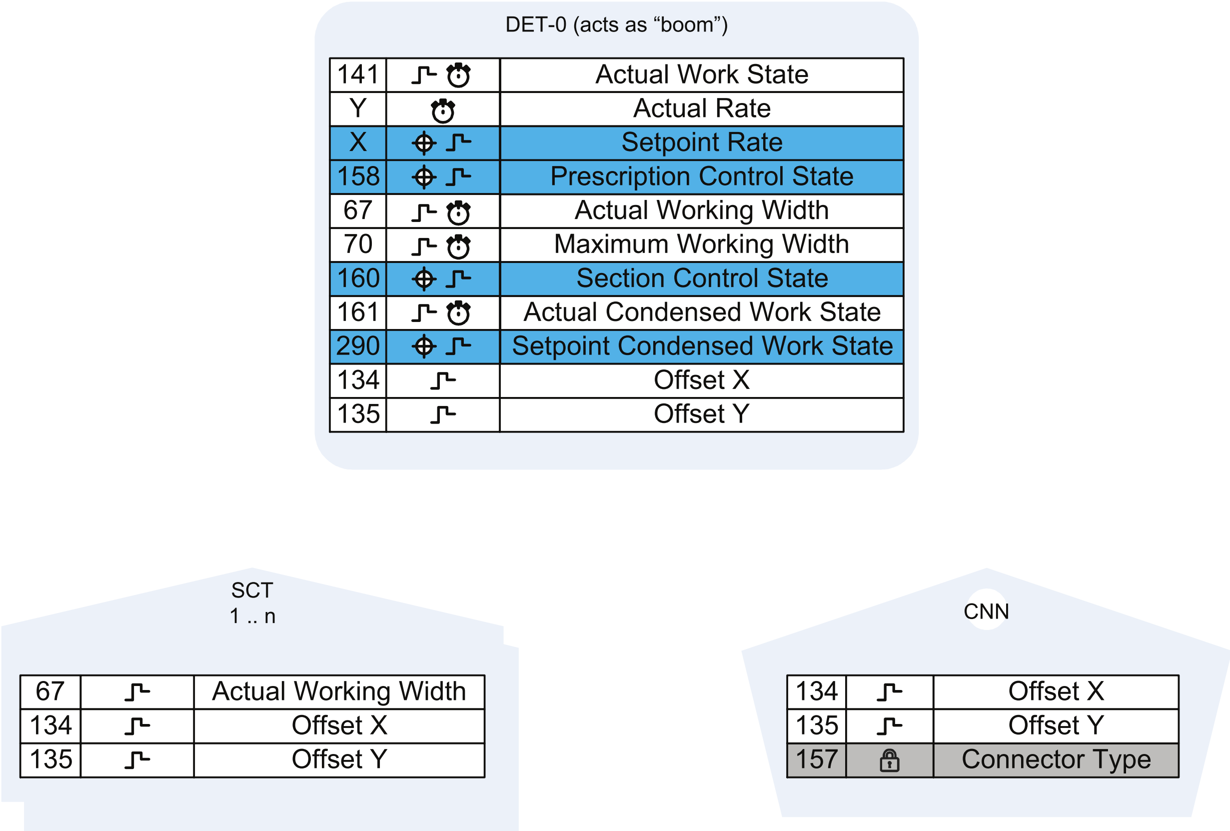


Figure F.17 — Single operation boom with sections

Figure F.18 shows an example device descriptor object pool with a Function device element defined as boom. If the boom is defined by a Function device element, then the actual and setpoint values, the X and Y offset and maximum working width, and the prescription control state shall all be placed in this device element. In this case, the root device element (DET-0) shall not contain any actual or setpoint values except an actual work state attribute.

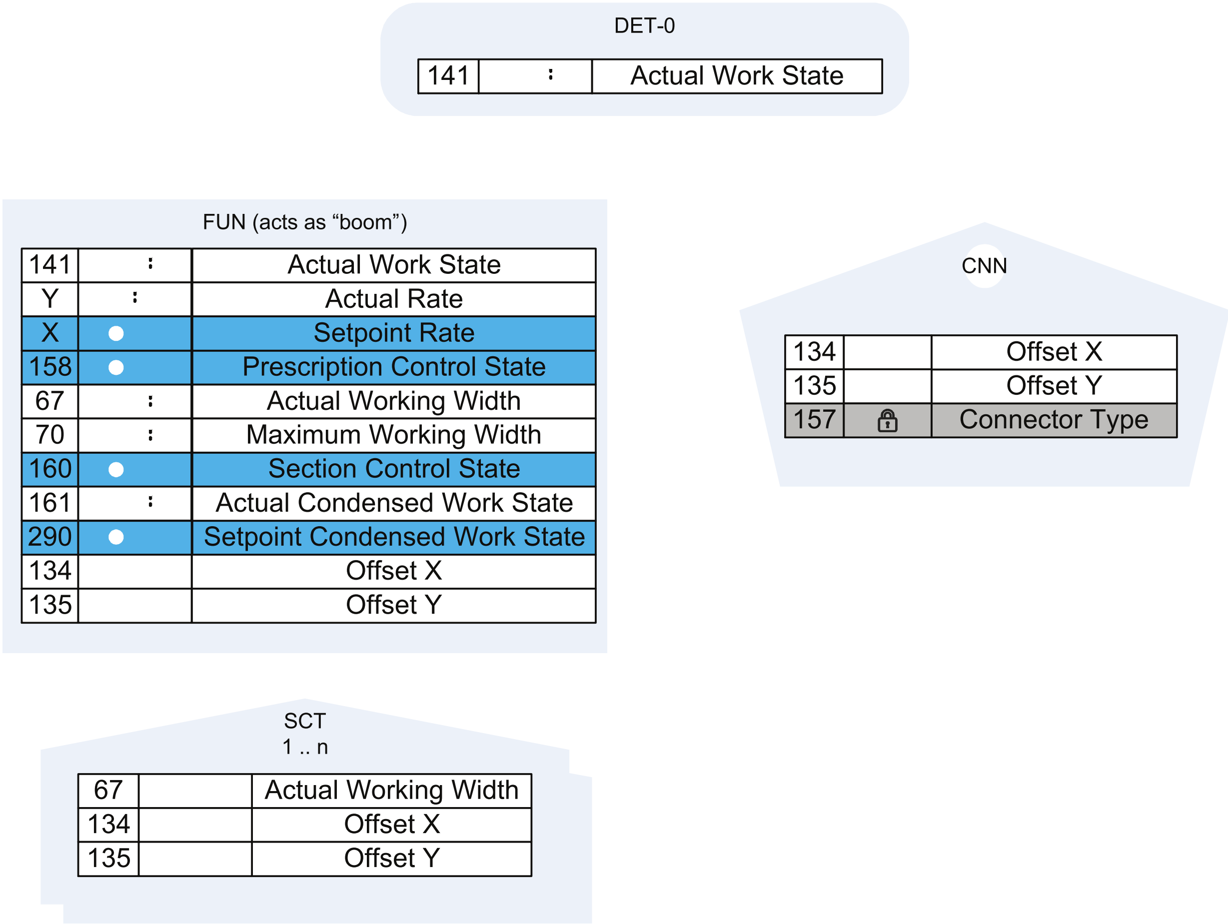


Figure F.18 — Single operation Device with a Function acting as boom

As an alternative to Figure F.18, it is allowed to add a Bin device element below the device element that represents the boom when a product is being applied through that boom. In case of only a single operation device, the use of the Element Type Instance DDI inside the Bin device element is optional. This device descriptor object pool is shown in Figure F.19.

The Actual Cultural Practice DDI is not mandatory in this example, but is recommended to specify the type of cultural practice that the Bin device element is configured for.

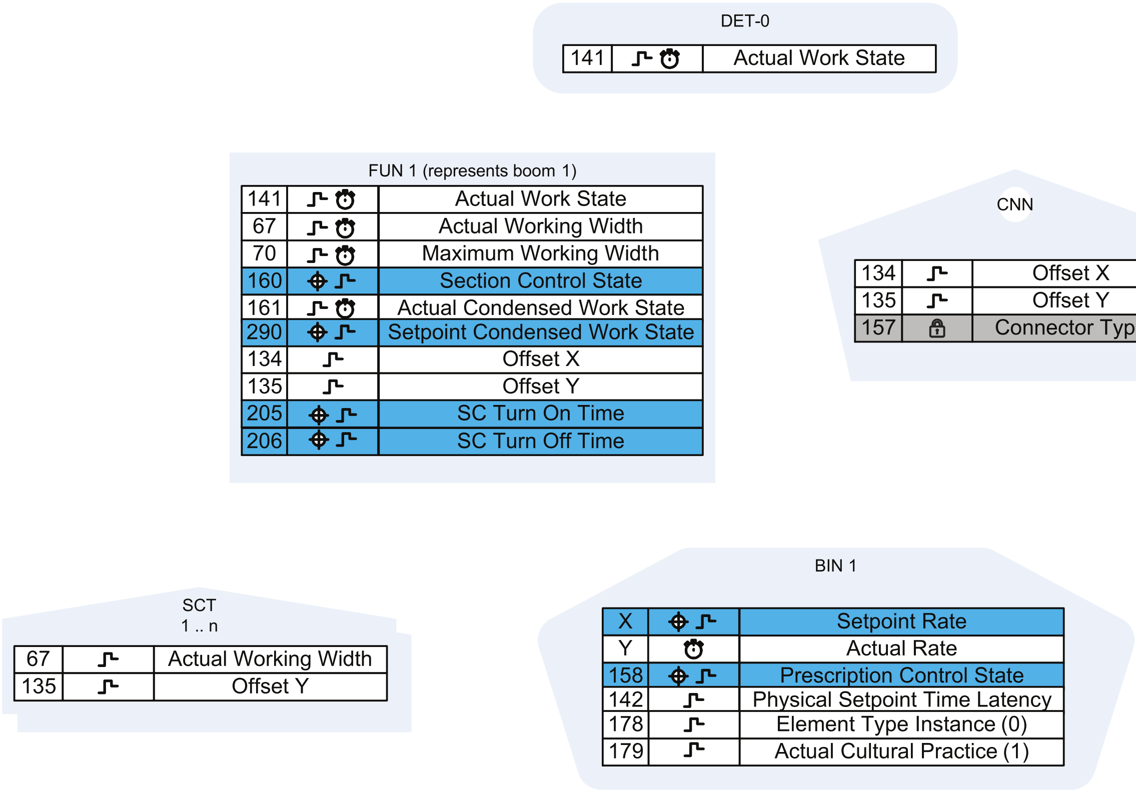


Figure F.19 — Single operation Device with a separate Bin device element, the boom is represented by a Function device element

F.3.4.6 Control latency

The effect of the physical setpoint time latency process data values is shown in Figure F.20. When the physical setpoint time latency attribute is present in a device element, then its value shall be used by the TC to adjust the sending of setpoint commands for actuation delays in the TC client. Figure F.20 shows the message flow between a TC, a connected TC client, and the device internal communication with a rate control module. The TC client in this example transmits actual rate updates based on a time interval at times T1, T3, and T4. The TC receives a position update and transmits a rate change command that results in the rate starting to change at time T2. Between time marks T2 and T4, the rate is being updated by the rate controller and the time it takes for this updated rate to be realized by the device is T4 – T2 ms. This time is reported by the TC client as the physical setpoint time latency DDI value. A TC compensating for the physical setpoint time latency shall project the position received at T2 to the estimated position of the device element at T4 and transmit the setpoint rate value for the estimated position at T4 to the TC client. Note that time interval based actual rate updates may be transmitted during the rate change.

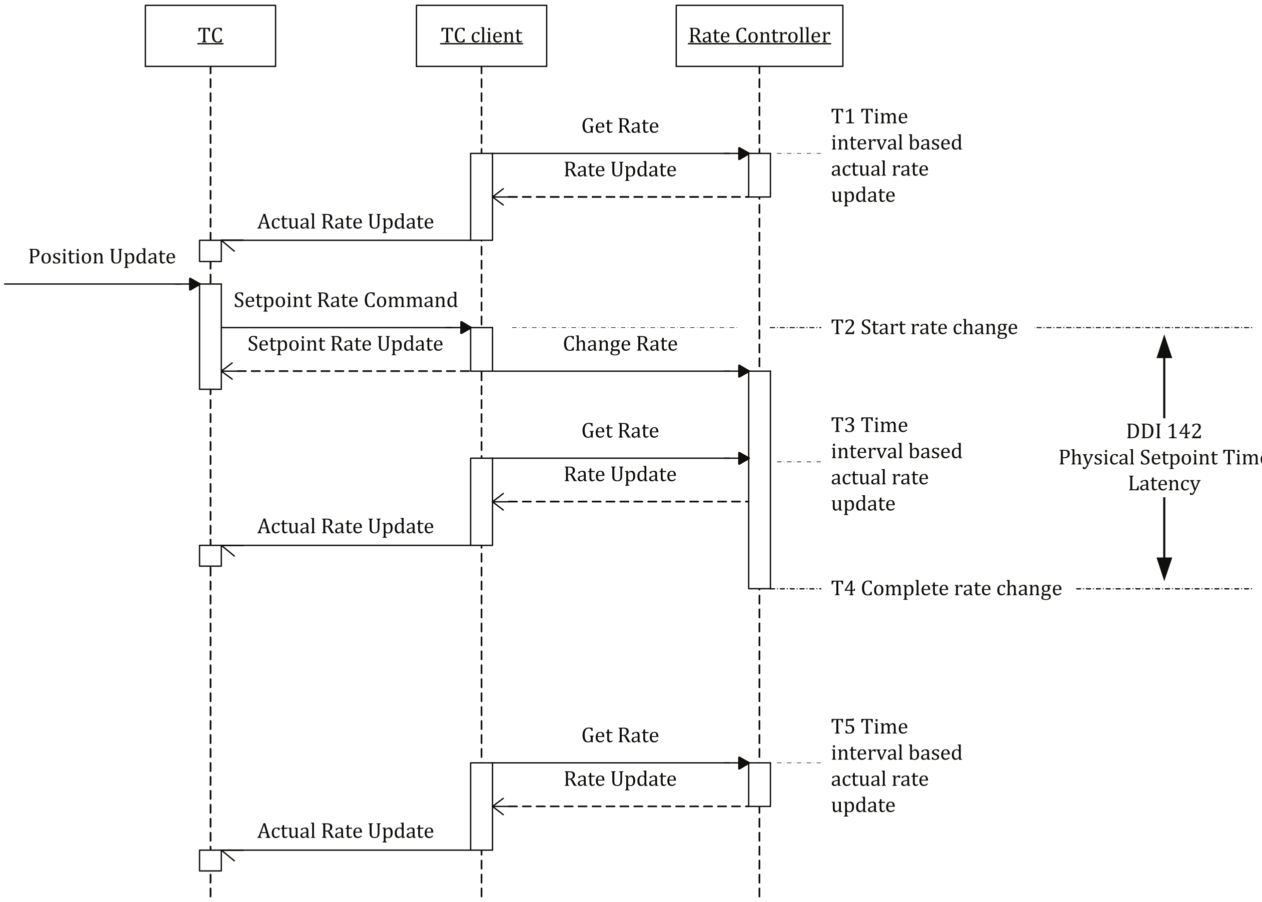


Figure F.20 — Physical Setpoint Time Latency and Rate Update sequence diagram

If the physical actual time latency attribute (DDI 143) is present, then its value shall be used by the TC and by the FMIS to adjust the values on, for example, a map to the correct locations. Note that the physical actual time latency value can be both positive and negative. Examples of these cases are defined in the DDI definition in ISO 11783-11. The physical setpoint time latency can only have a positive value.

A variant on the device descriptor object pool depicted in Figure F.19 with the same functionality is shown in Figure F.21. In this example, there is no separate function that represents the boom but instead the boom is represented by the root device element. In case of a single boom device, the use of the element type instance attribute inside the Bin device element is optional. The rate process data attribute shall be present either in the root device element or in the Bin device element, never in both device elements simultaneously.

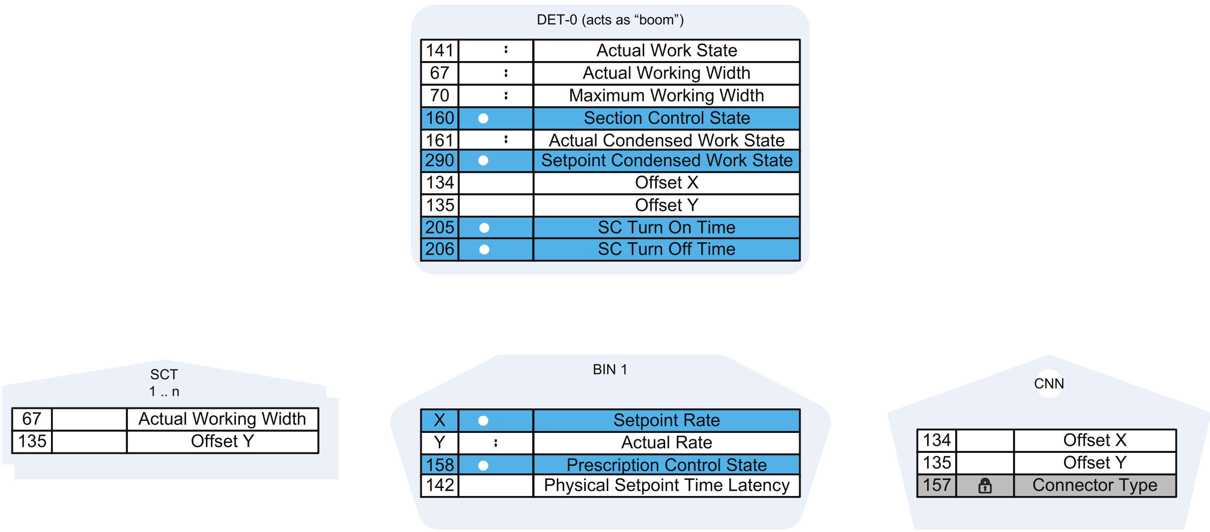


Figure F.21 — Single operation Device with a separate Bin device element, the boom is represented by the root device element

F.3.4.7 Multiple control channels

A more complex device, having two independent application control operations, is depicted in Figure F.22. The first function in this example represents a fertilizer operation with the Actual Cultural Practice attribute has a value of one. The second function represents a seeding or planting operation with the Actual Cultural Practice set to value two. In case of multiple booms, the Actual Cultural Practice attribute (DDI 179) shall be placed in the boom or in the underlying Bin device element. The TC can differentiate the boom definitions based on the Actual Cultural Practice attribute values.

The Element Type Instance attribute (DDI 178) shall be used to identify each bin to the operator. This enables an operator to select from which bin a product is applied. Hence, the Element Type Instance attribute shall be placed inside the Bin device element.

Similar to the previous examples, it is not allowed to place the same rate attribute in the root device element and the boom simultaneously or in a Function device element acting as a boom and a Bin device element.

The Actual Cultural Practice and Element Type Instance attributes also assist in the process of changing the execution of a TC-GEO task from one device to another device. When a device with another structure is used in-field or when there is a mismatch between planned-for and actual device descriptor object pool, then the "groups" of PDVs need to be reassigned to the device that is available. TCs shall be able to handle device reassignments and the mapping of groups of setpoint process data values to the device elements that have the corresponding device process data attributes to process the setpoints.

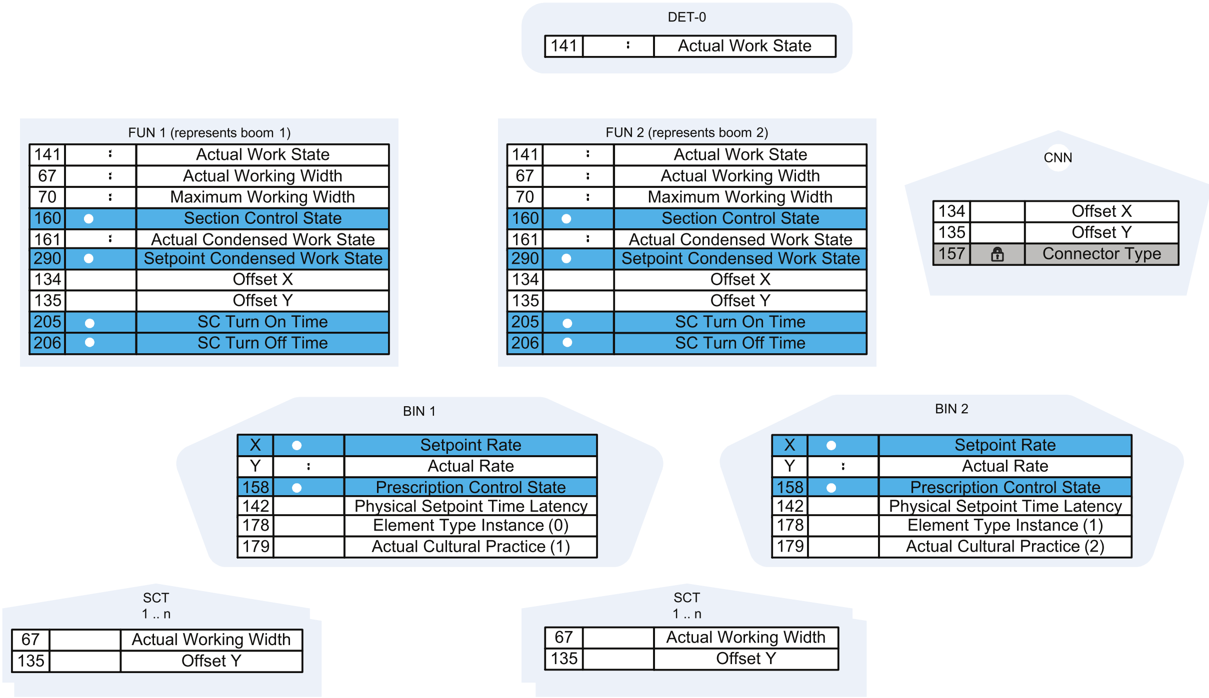


Figure F.22 — Dual operation Device with separate Bin device elements, the booms are represented by Function device elements

A variant of a multi-operation device is a single boom multi bin applicator. This example is given in Figure F.23. More than one Bin device element can be present under one boom, indicating that all products from these bins are applied through that single boom.

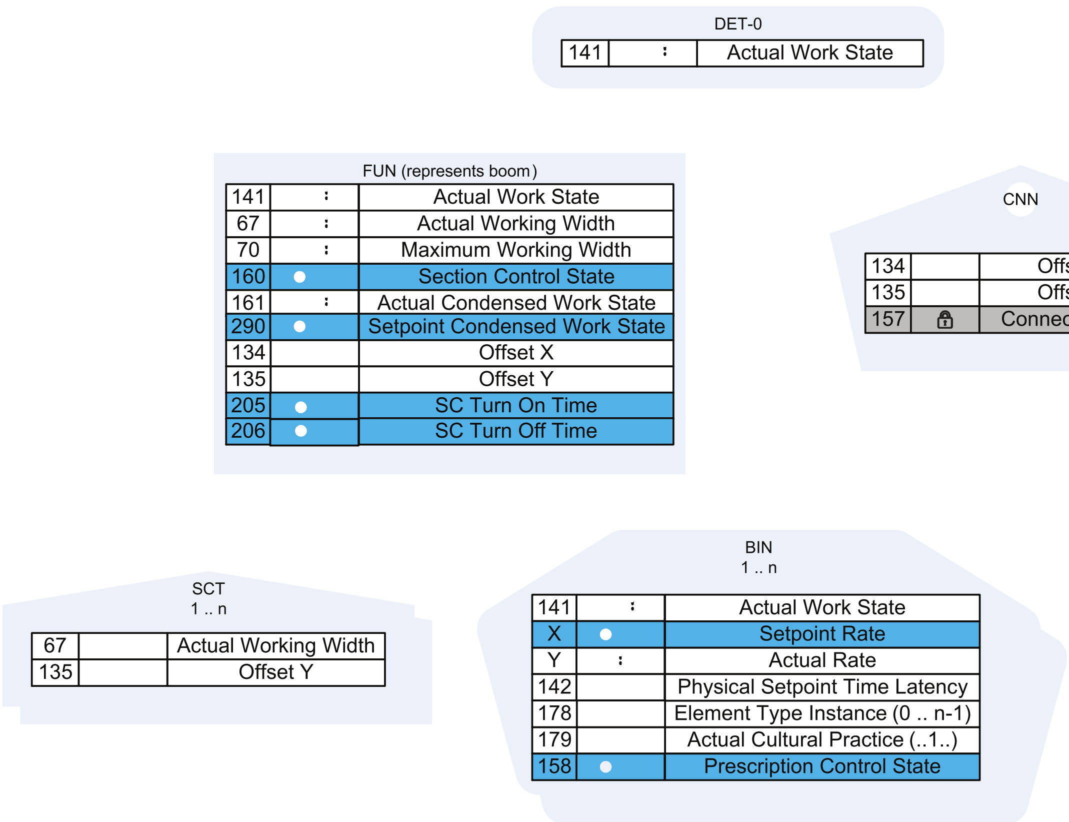


Figure F.23 — Multi operation Device with a single boom and multiple Bin device elements

Devices that can vary an application rate at a sub-boom or section level detail shall model this capability through the definition of sub-boom device elements or Section device elements with rate attributes. The setpoint and actual rate attributes that in this case may occur at two different device element levels specify that this is a single operation device with two sub-controllable rates. Rate control commands received at the topmost boom level shall be automatically forwarded by the device to its child functions. Rate control commands received at the most detailed level only apply to the addressed function. The actual rate at the topmost boom level is an average of the actual rates of its child functions. A device descriptor object pool shall have no more than one level of sub-booms, as depicted in Figures F.24 and F.25. The sub-booms shall include a complete geometry definition, including attributes for X and Y offsets and for a maximum working width.

Figure F.24 shows an example of a device with sub-booms with multiple sections per sub-boom. Figure F.25 has an alternative layout where each section has a controllable rate.

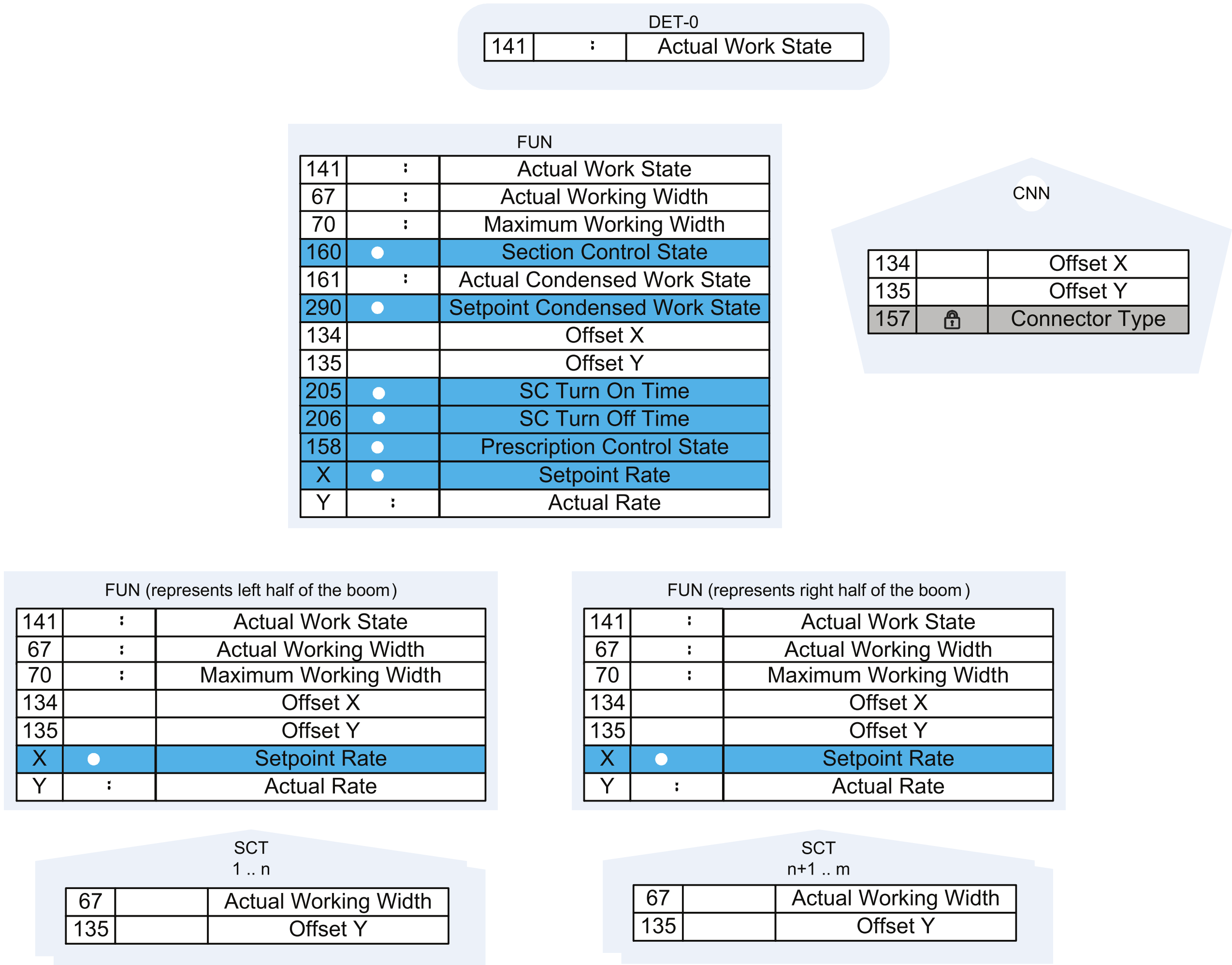


Figure F.24 — Single operation Device with multiple rate controllers as sub-booms

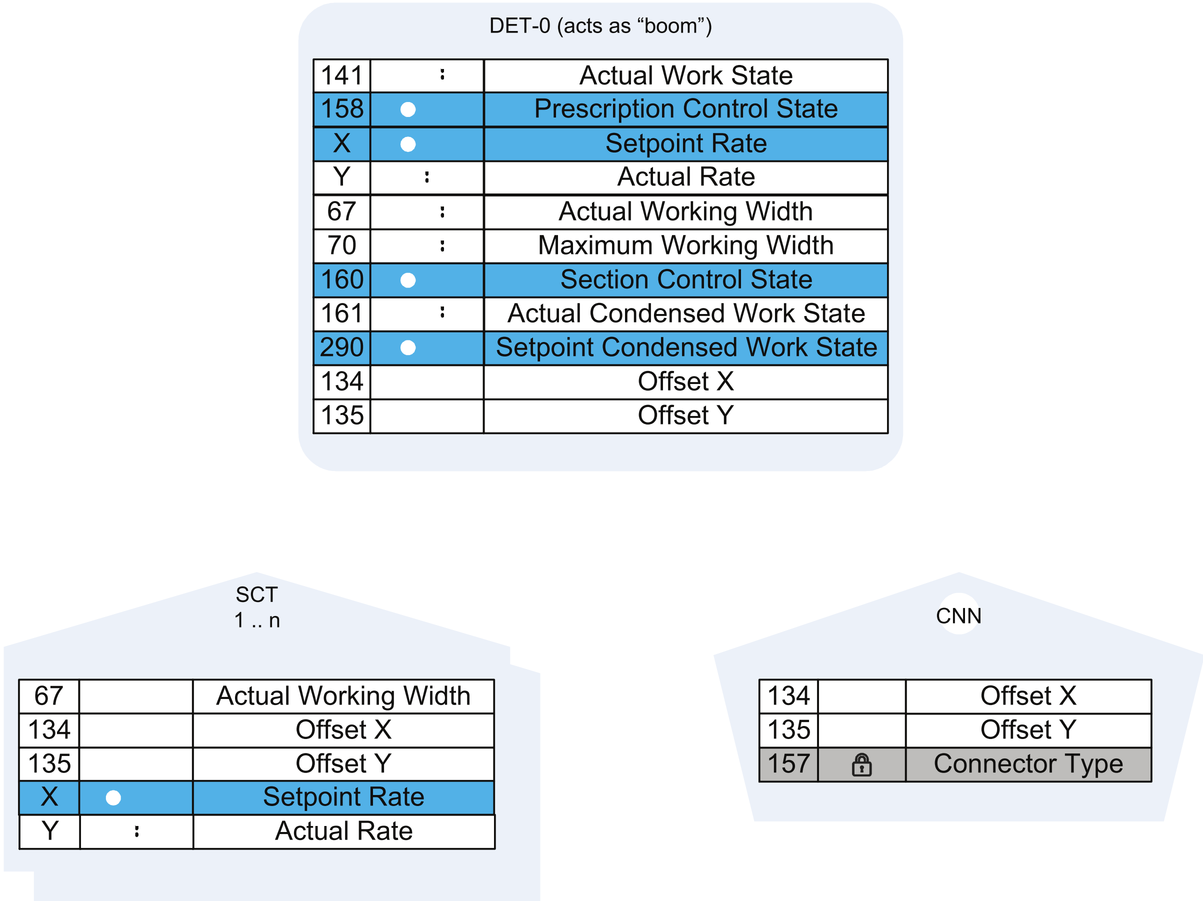


Figure F.25 — Single operation Device with multiple rate controllers as Section device elements

F.3.5 TC-SC generation 1 device descriptor object pools

The first device descriptor object pool example that meets the requirements for the TC-SC generation 1 is provided in Figure F.26. In this example, the root device element acts as the boom and the required Section and Connector type device elements plus the geometry attributes as device properties are present. Due to that, the geometry in this example is present as a set of device properties, the TC-SC controller does not need to query these upon connection or monitor them for changes during operation.

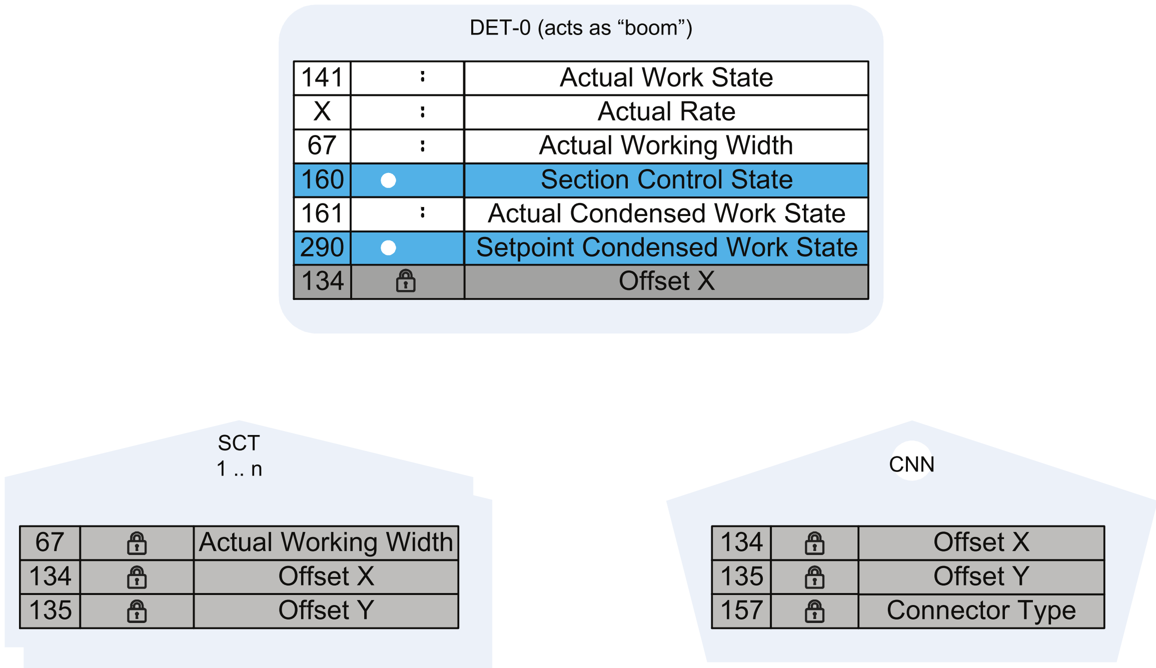


Figure F.26 — Single operation Device, geometry in Properties

F.3.5.1 Work state

The Actual and Setpoint Condensed Work States are the only workstates that are allowed in the DDOP to report and control the section work states. Individual Actual Work States (DDI 141) or individual Setpoint Work States (DDI 289) are not allowed in the Section device elements. The individual Actual Work State attribute in the boom device element represents a master work switch for this operation and can be reported on change and on a time interval basis.

The measurement type "On Change" is recommended to be supported by the Setpoint Condensed Work State attribute. This enables a TC-SC controller to set up a measurement for receiving a confirmation of each transmitted Setpoint Condensed Work State command.

The measurement types "On Change" and "Time Interval" are required to be supported by the Actual Work State(s) and Actual Condensed Work State(s) attributes.

The actual work state values along a hierarchy of Device Elements shall be combined to determine the work state of each section. In the above example, if the Actual Work State in the root device element has the value "Off", then even when the actual work states of the sections have a value "On", the resulting work state for these sections is processed as "Off" by the TC-SC controller.

F.3.5.2 Working width

Each Section device element shall at least provide one type of working width. If more than one type of working width is provided, the Section Controller shall be capable to use the different working width types with the following priority:

a) Actual Working Width (DDI 67)

b) Maximum Working Width (DDI 70)

c) Default Working Width (DDI 68)

The Actual Working Width of a Section shall not depend on the actual work state of that Section. An Actual Working Width at Device or Function level (the boom) shall depend on the actual work states of its underlying Sections. This enables a Device or Function to report the Actual Working Width as the sum of the working width of the sections that are “On”, while the Section Controller uses the section working width values to determine the section geometry and to control where to turn sections “On” or “Off”.

It is recommended that the Element Numbers of the Section device elements increase from left to right across a machine. In determining the order of the Sections, the geometry definitions of the Section device elements have priority over their Element Numbering.

Section device elements that belong to one boom shall geometrically be positioned in a row, exactly next to each other, without overlaps or gaps.

F.3.5.3 Section control state

For the TC-SC functionality, the boom shall include a Section Control State process data attribute at the same level as the setpoint condensed work state process data. This Section Control State process data attribute shall be settable and shall support an on-change measurement trigger. A boom that is not TC-SC capable shall not contain the Section Control State process data attribute. In addition to the specification in the ISOBUS Data Dictionary for the Section Control State DDI (160), the following rules apply:

a) At task stop, the TC client shall internally set its Section Control State to DISABLED/OFF.

b) The TC shall set the Section Control State to ENABLED/ON before sending setpoint condensed work state value commands to the TC client.

c) Upon reception of a Section Control State command to ENABLED/ON, the TC client shall reset the work state process data attributes that correspond with this Section Control State to the determined safe state value for the work state process data attributes.

d) The TC shall use an on-change measurement command to receive Section Control State changes from the TC client.

F.3.5.4 Device geometry

The second example, provided in Figure F.27, is very similar to the previous example except for that, the geometry is specified in process data attributes. Depending on whether the geometry of a device element can change during the operation of the device, a mix of device properties and device process data are allowed to be used for these attributes.

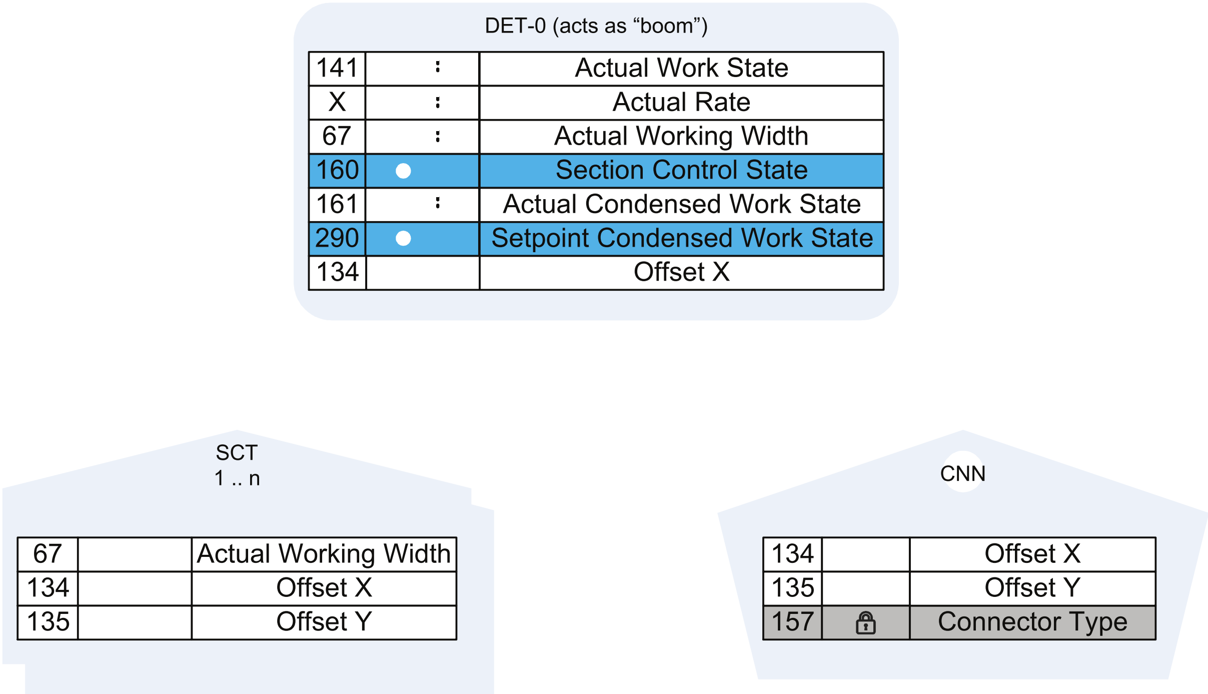


Figure F.27 — Single operation Device, geometry in Process Data

The measurement type "On Change" is recommended to be supported for the geometry process data. Upon system startup, a TC-SC shall query the Device for its initial geometry DDI values to ensure that the correct geometry is used to determine coverage and section control commands.

A mix between process data and properties is allowed in the DDOP. In this example, the Connection Type attribute is a property which will not change over the lifetime of the DDOP structure, whereas the rest of the geometry attributes may change upon, for example, device configuration modifications.

TC-SC requires a connector that indicates the connection point to the tractor. Multiple connectors are allowed. Connector and Navigation device elements shall only be placed directly below the root device element (DET-0). The connector and navigation device elements shall include device properties or device process data for X and Y offsets, as well as a connector type definition.

All Sections and Device Elements that act as a boom shall provide a width and offsets to specify their geometry. At a minimum, geometry is defined by properties or process data for Offset X, Offset Y and Width attributes. For the relation between a boom and sections, only one exception from this minimum definition of geometry is allowed:

a) A Section type device element does not need to provide an Offset X if its parent device element has an Offset X defined. In this case, the Offset X from the parent device element is valid for all its Section type child device elements. This optimizes the size of the Section type device elements. The device descriptor object pool in Figure F.28 shows a variant with this exception.

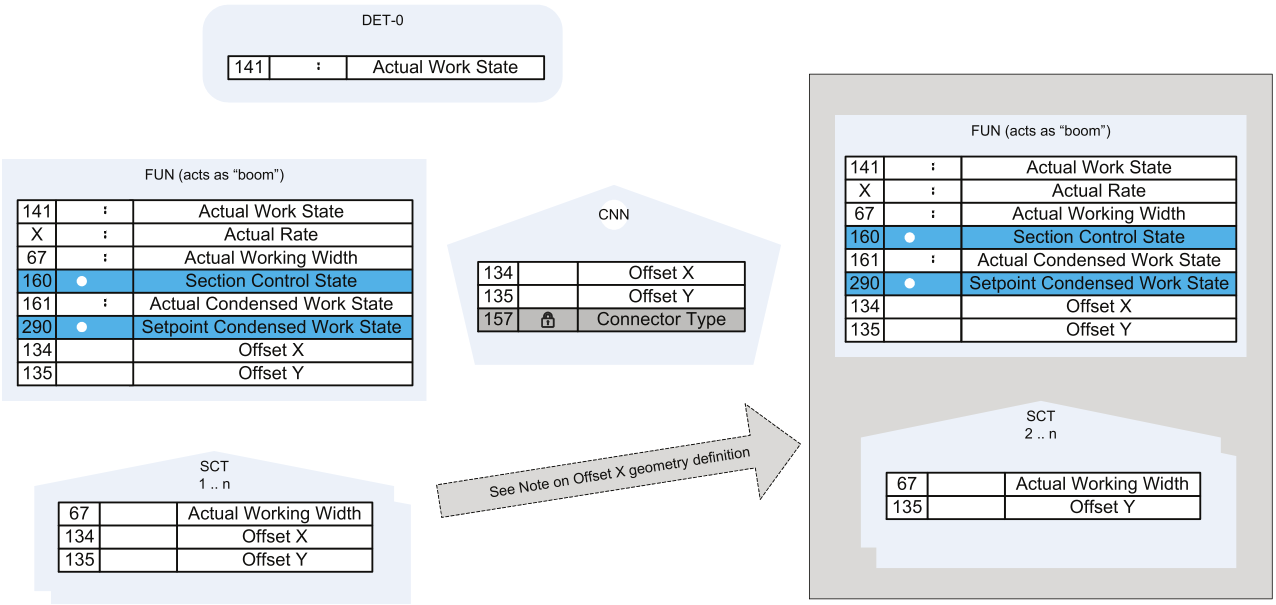


Figure F.28 — Single operation Device, offset X attribute variants

F.3.5.5 Control latency

Next to the requirements regarding the handling of workstate, working width, and geometry attributes, the handling of control latencies is clarified in the following examples.

Figure F.29 shows on the left-hand side an example device descriptor with the separate Section Control Turn On and Section Control Turn Off Time attributes specified at the boom level. This is the recommended method to specify the section control turn on and turn off delays in new designs of TC-SC clients. For backwards compatibility, a TC-SC server shall be able to use the physical setpoint latency DDI, as depicted on the right-hand side in Figure F.29.

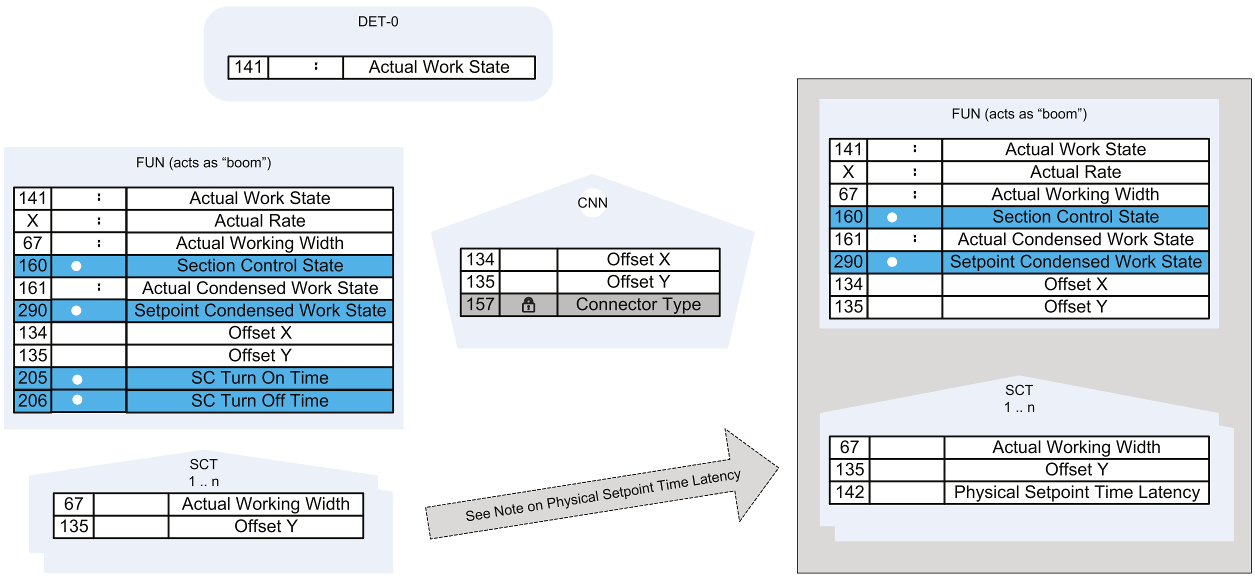


Figure F.29 — Section Control delay definitions at boom level (new designs recommendation) and physical setpoint time latency at section level (backwards compatible requirement)

The SC Turn On and Turn Off Time values are recommended to be used in the device descriptor and are required to be handled by the TC-SC controller. The TC-SC controller shall calculate the section turn on/off locations based upon the received Turn On and Turn Off Time values from the TC client. The TC client shall not change section on/off timing from the reported Turn On or Turn Off times. The Turn On and Turn Off times are based on physical performance of the sections.

A sequence diagram of a section turn on and turn off cycle is provided in Figure F.30. In this diagram, both the section work state updates transmitted due to a time interval (at time marks T1, T3, T5, and T8) and due to a state change (at time marks T2 and T6) are present. Note that the Turn On and Turn Off times for a section can be different; in this example, the time it takes to turn on a section (T4-T2) is longer than the time it takes to turn off a section (T7-T6). Also visible in this diagram is the acknowledgement of the setpoint condensed workstate (Setpoint CWS) command by a return message, the update. This response message and the update to the actual state to be sent by the TC-SC client immediately when a section state change starts are mandatory. The TC-SC server has the responsibility to use the SC Turn On or SC Turn Off time to adjust the reported workstate changes for, for example, updating coverage on a map display. The first section turn on sequence in this diagram is triggered by position update 1. The following position update 2 did not result in the need for a section state change from on to off and thus did not lead to the transfer of a Setpoint CWS command. The third position update triggers the section turn off sequence.

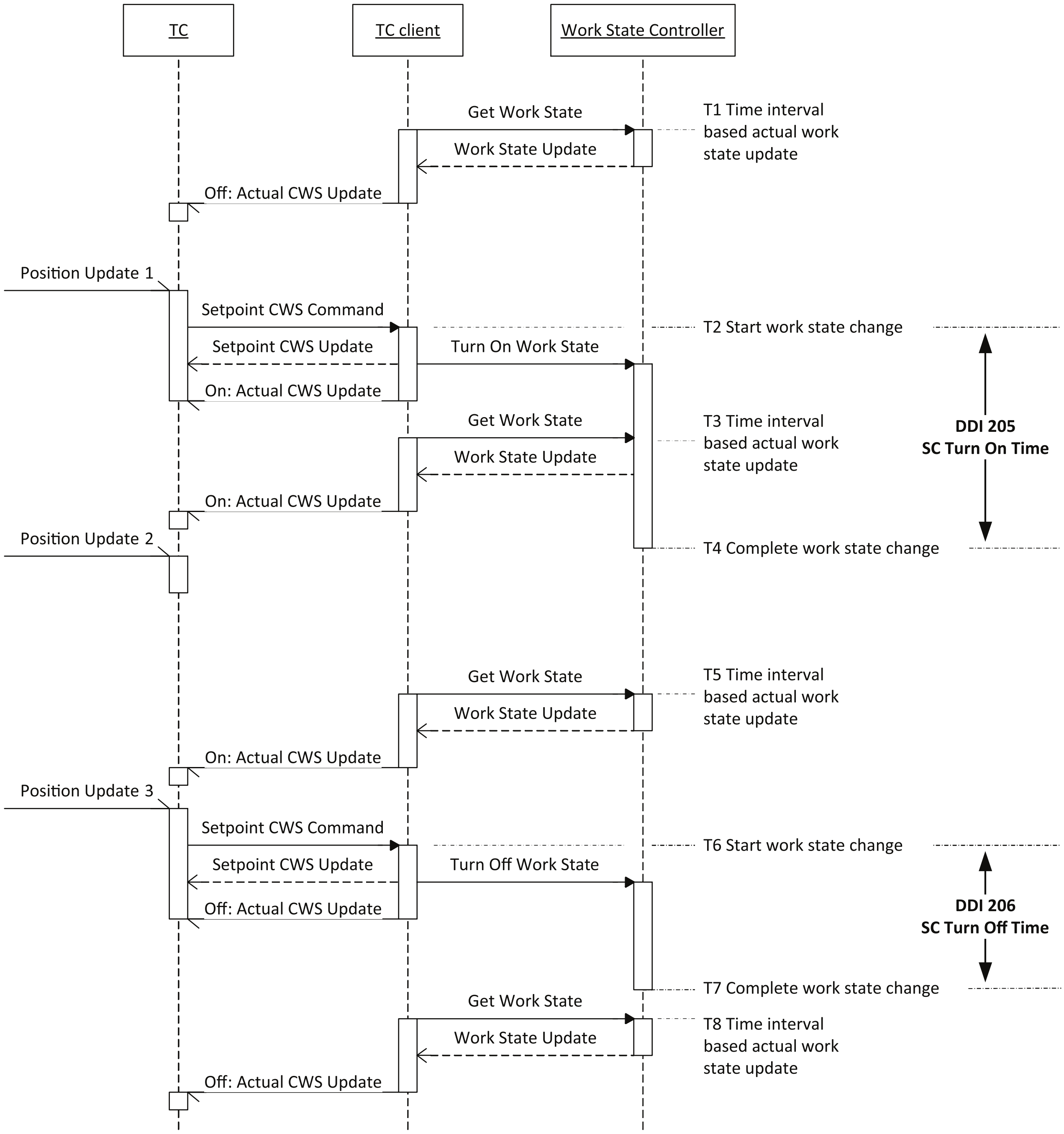


Figure F.30 — TC-SC section Turn On and Turn Off sequence diagram

The SC Turn On and Turn Off Time values shall be defined as settable Process Data in the boom device element settable and shall support an on-change measurement trigger. Thus, the values of these parameters can be adjusted in the TC-SC operator interface and stored in the TC-SC capable TC client.

The TC-SC server shall use the SC Turn On Time to advance its section Turn On control command by the amount of time specified in this process data value. A positive value causes the TC-SC server to send its section Turn On command earlier to the TC client to compensate for the TC client's Turn On latency.

The TC-SC server shall use the SC Turn Off Time to advance its section Turn Off control command by the amount of time specified in this process data value. A positive value causes the TC-SC server to send its section Turn Off command earlier to the TC client to compensate for the TC client's Turn Off latency.

For TC clients that provide SC Turn On/Turn Off Time configuration means on their own operator interface, it is recommended to support an On Change measurement type for these values and transmit updates to the TC-SC server when the operator adjusts these values. A TC-SC server shall use on-change measurement commands to receive value updates from TC-SC capable TC clients that allow the operator to change the SC Turn On and Turn Off values in their operator interface.

To ensure backwards compatibility, in case section device elements also contain the Physical Setpoint Time Latency (DDI 142), then the TC Turn On and Turn Off Time values have a higher priority and the Physical Setpoint Time Latency value shall be ignored for section control commands. If only the Physical Setpoint Time Latency is specified in a section device element, then it shall be used by the TC-SC server to advance the section Turn On and Off control commands by the specified latency. This alternative control delay handling scenario is depicted in Figure F.31. In this case, the control delays for turning a section on or off are equal and cannot be differentiated.

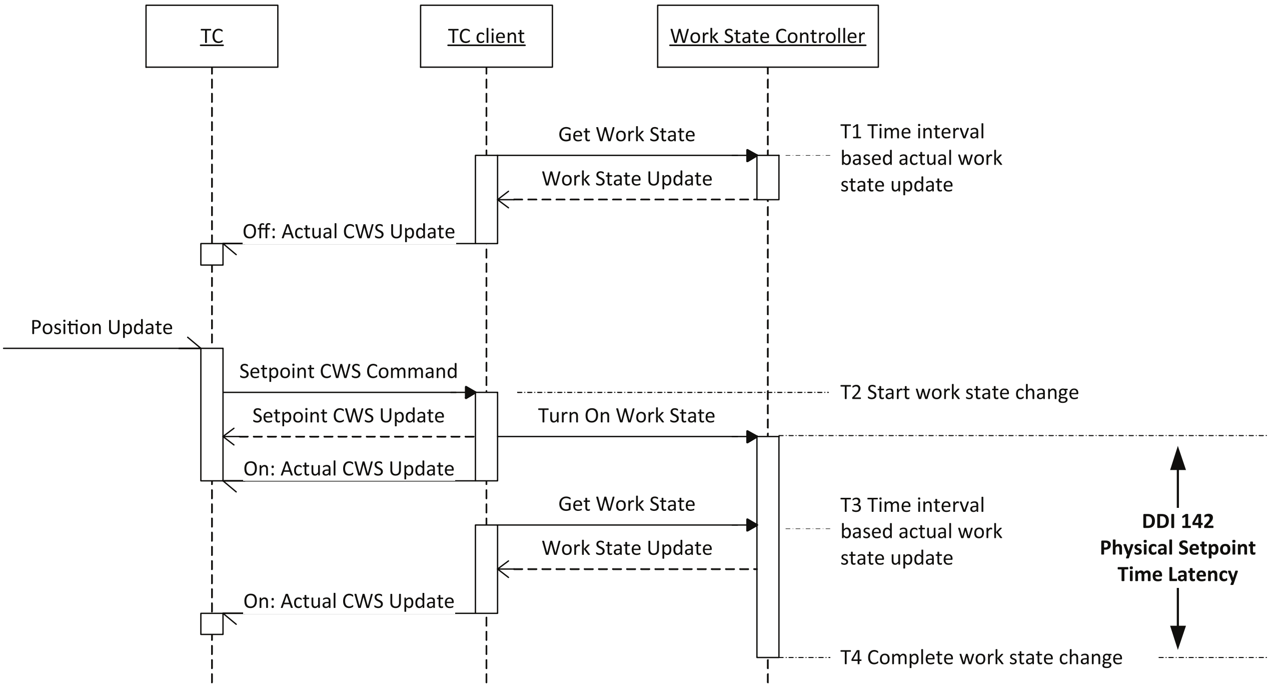


Figure F.31 — Physical Setpoint Time Latency control delay handling alternative

F.3.5.6 Multi-boom and multi product

An example of a multi-boom TC-SC compatible device descriptor is provided in Figure F.32. The first function in this example represents a fertilizer spreading operation, the value of the Actual Cultural Practice attribute is 1. The second function represents a seeding or planting operation with the value set to 2 for the Actual Cultural Practice attribute. In case of multiple booms, the Actual Cultural Practice attribute (DDI 179) shall be placed in the Boom or in the underlying Bin device element so that the TC can identify the various boom definitions.

In this multi-boom example, both functions have their own physical boom and set of sections. The number of sections of the first boom can be different from the number of sections of the second boom.

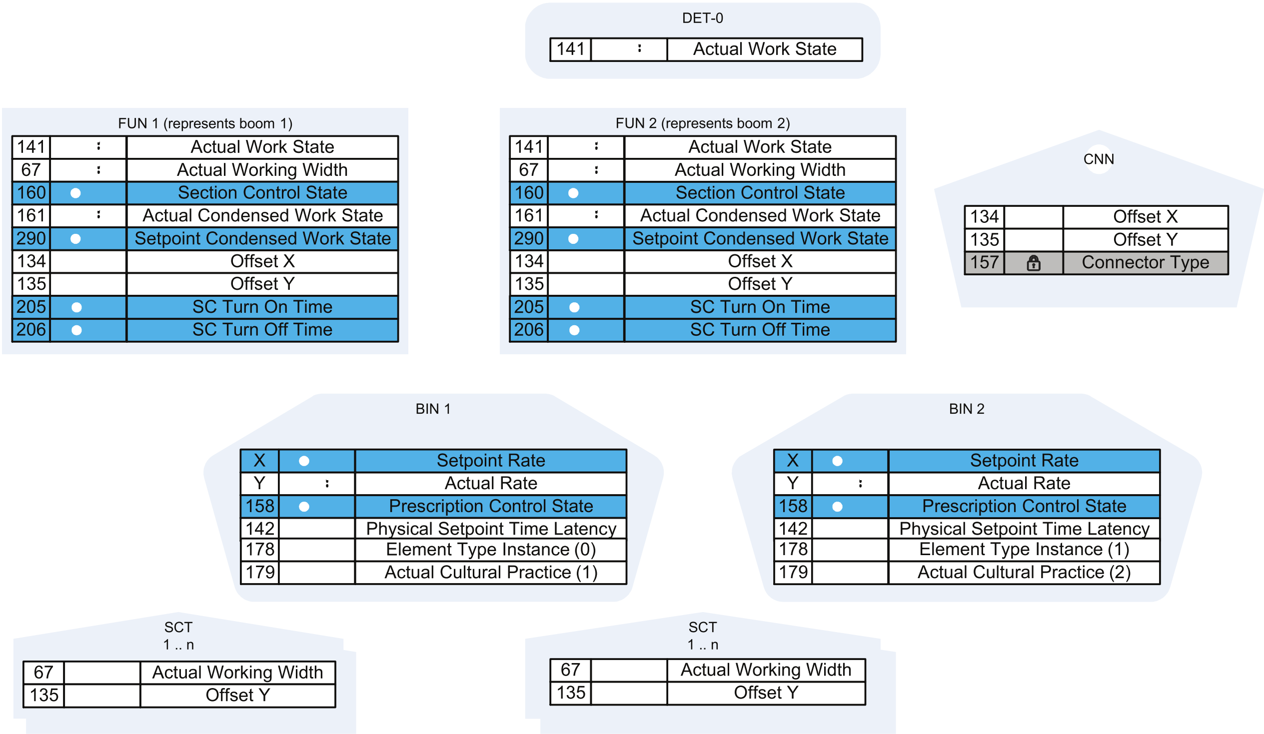


Figure F.32 — Multi-boom multi product section control device descriptor

An example of a device descriptor that models a device capable of applying multiple products through a single boom is provided in Figure F.33. From a TC-SC perspective, this example has a single boom which shall contain all attributes required for a single boom to be a TC-SC compatible device.

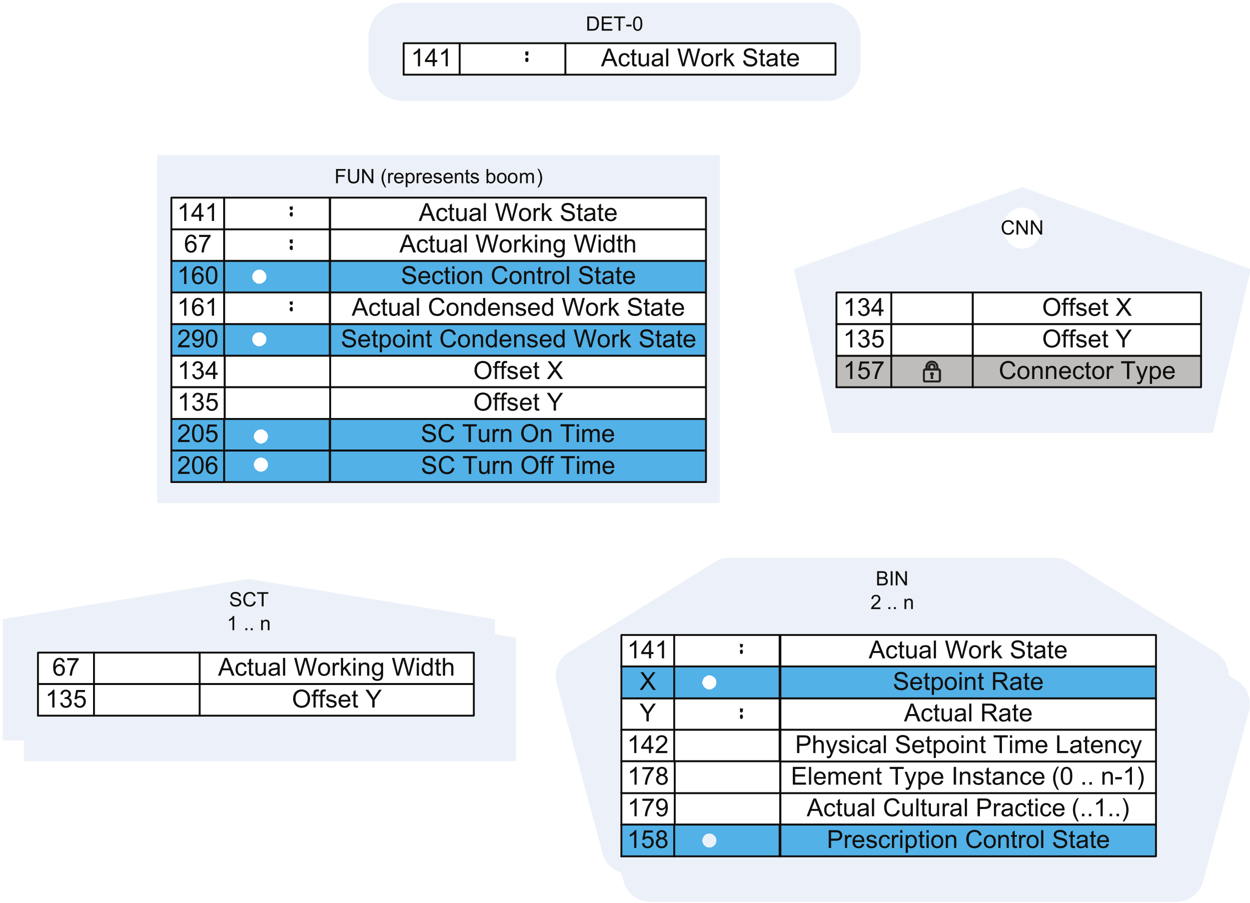


Figure F.33 — Single boom multi product section control device descriptor

The adjustment down of a multi-boom device to a single boom TC-SC configuration is represented in Figure F.34. This example has a similar structure as the example provided in Figure F.32, the difference being that the second boom in Figure F.34 is not a TC-SC boom. Adjusting down the number of TC-SC booms is required in order to not exceed the number of TC-SC booms that the TC-SC server supports. The total number of sections defined in TC-SC booms shall not exceed the number of sections the TC-SC server supports. The difference between a TC-SC controllable boom and a non-TC-SC controllable boom is that a TC-SC controllable boom shall include the Section Control State attribute in the boom as settable and supporting at least the on-change measurements. Non-TC-SC controllable booms shall not include the Section Control State attribute.

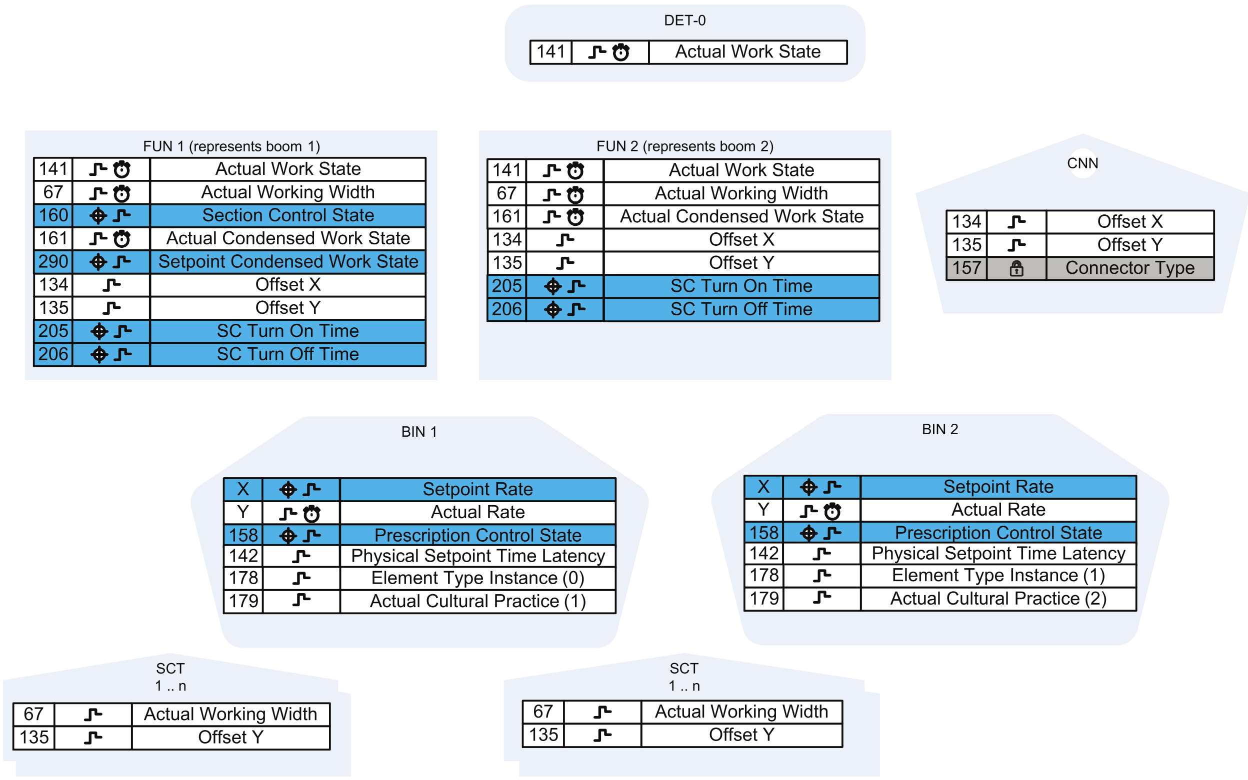


Figure F.34 — Multi-boom adjusted to single boom section control device descriptor

Another product application variant in which a product can be applied with different rates across a single TC-SC boom is depicted in Figure F.35. In this example, an intermediate function device element level is required to represent the different rates of each half of the boom. In order to use the condensed work states effectively across the entire boom, the TC-SC boom definition shall be located in the parent device element of the rate control boom halves. The Section Control State, Actual Condensed Work State and Setpoint Condensed Work State process data shall be specified in the topmost boom level. At a maximum, 1 level of sub-booms as depicted in this example is allowed.

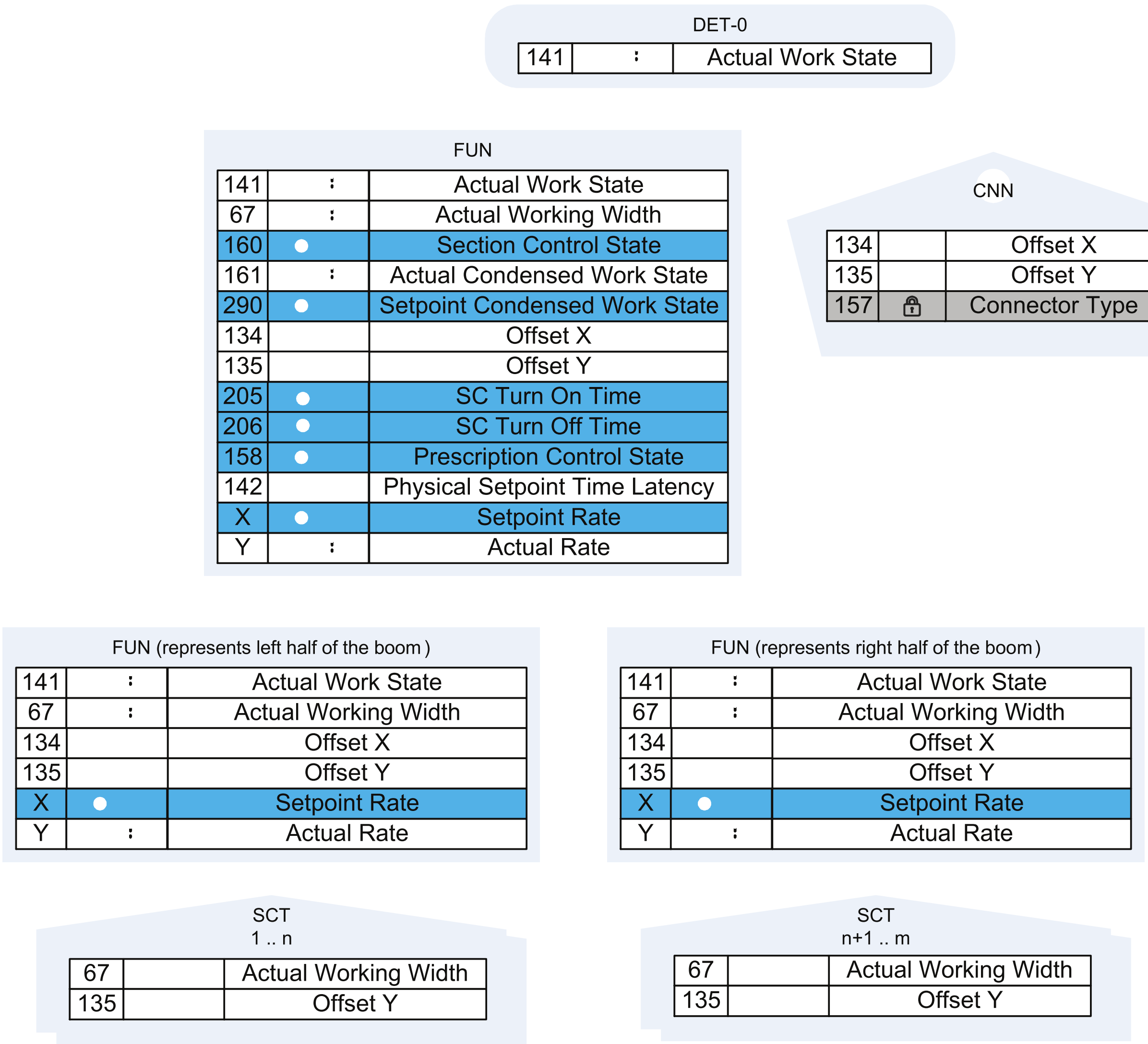


Figure F.35 — Single boom multi product section control device descriptor

F.3.5.7 Dynamic geometry

The Offset Y attribute of a TC-SC boom with multiple sections shall not change if the underlying sections are turned On or Off. The example depicted in Figure F.29 contains Offset Y attributes at both the TC-SC boom level, as well as in the section device elements. The value of the Offset Y attribute shall in this case be the centre of the boom and shall not be changed when, for example, the left half of the sections are turned Off.

An example that requires dynamic updates to the Offset Y attribute value to correctly represent the covered area is the device descriptor provided in Figure F.36. This is, for example, for a device that cuts crop with a cutter bar that can be partially used and can cut crop from either the left side or right side onwards. In this case, if only the left half of the cutter bar is cutting crop, the Actual Working Width value shall be reduced to half the maximum working width and the Offset Y value is adjusted to the centre of the cutter bar area that is cutting.

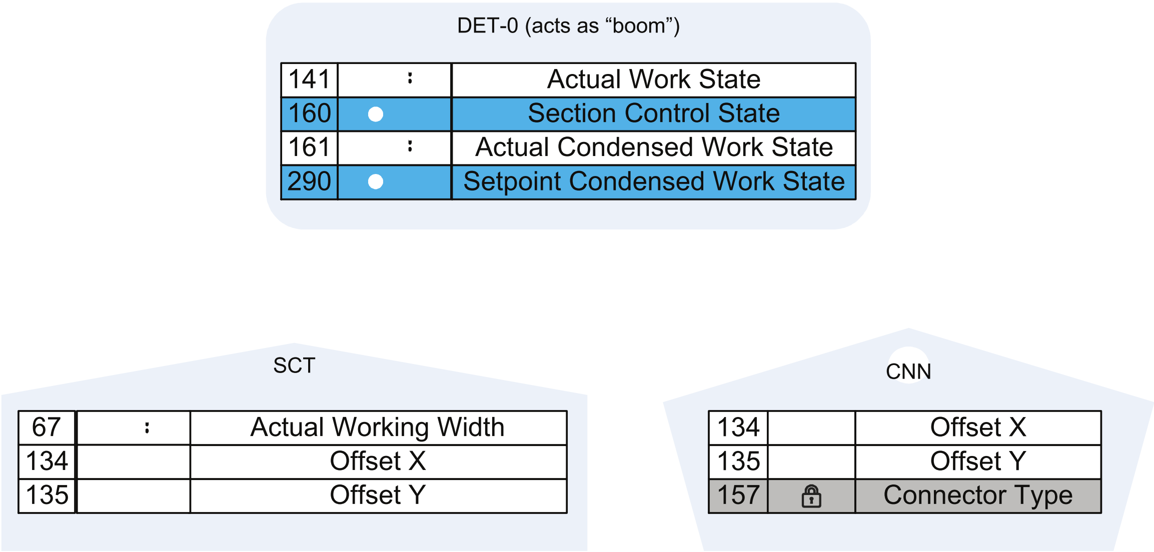


Figure F.36 — Single section, dynamic Offset Y attribute, device descriptor

F.3.5.8 Actual working length

The Actual Working Length attribute (DDI 226) can be used to define the working length of an overall operation or a device element such as a section. In case that the working length is not centred, the appropriate offsets on the same device element shall be used to define the value of the shift.

In case that an Actual Working Length attribute is present in the boom, care shall be taken that the device geometry and the inheritance of the X-Offset value are correctly represented. Values for the Actual Working Length and Offset X present in the boom apply to all underlying sections. In case individual sections have different Actual Working Length or Offset X values, then these attributes shall be present in the section device elements instead of in the boom device element.

Figure F.37 represents a multi-boom device descriptor in which each boom contains one Actual Working Length value valid for the entire boom.

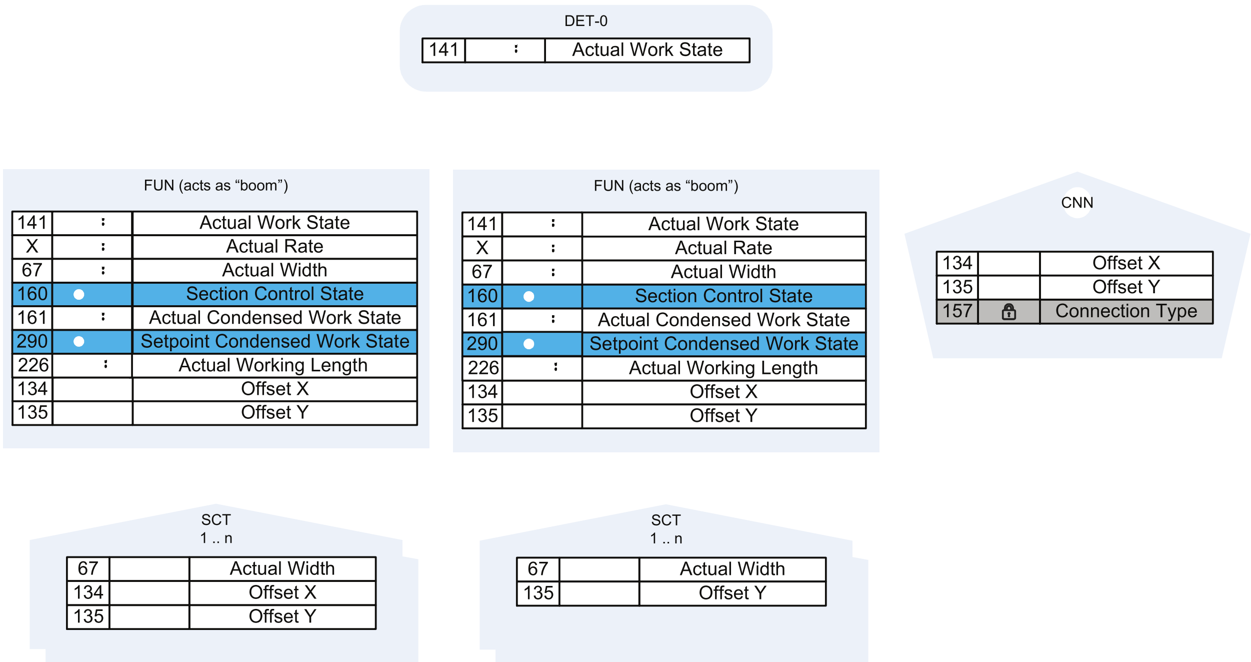


Figure F.37 — Actual working length attribute addition example

F.3.6 LOG generation 1 device descriptor object pools

The most simple version of a LOG generation 1 conform device descriptor object pool is shown in Figure F.38. This device descriptor contains only a single attribute which is the lifetime effective total time. Note that the lifetime total DDI cannot be set by a TC or DL server and hence does not have the setable background associated with it. The time interval measurement symbol indicates that this total DDI can be requested to be transmitted at a certain time interval.

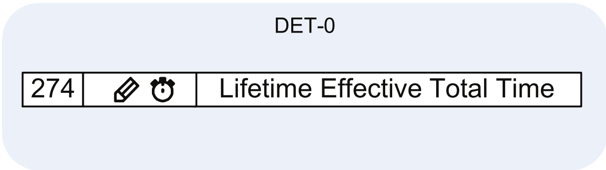


Figure F.38 — Single operation Device without a Function Device Element and no geometry information

Devices that include a rate DDI shall provide a corresponding lifetime total DDI for the rate (Figure F.39).

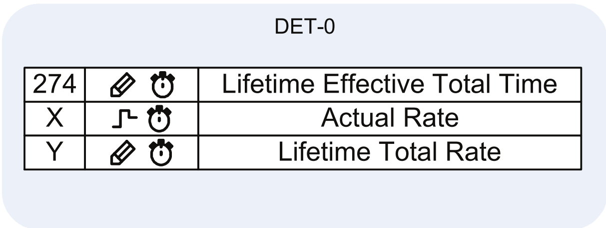


Figure F.39 — Single operation Device with an Actual Rate DDI without a Function Device Element and no geometry information

The device descriptor object pools depicted thus far did not have a separate functional device element from the device's root device element. This separation is shown in Figure F.40 for the most basic device descriptor with a separate functional device element.

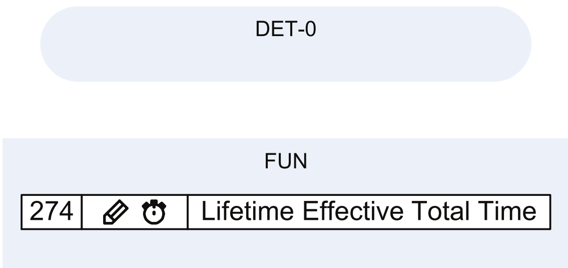


Figure F.40 — Single operation Device with a separate Function device element

A similar separation of attributes embedded in a function for a slightly more complex device is provided in Figure F.41. Next to totals, also different variants of rate and working width attributes are specified.

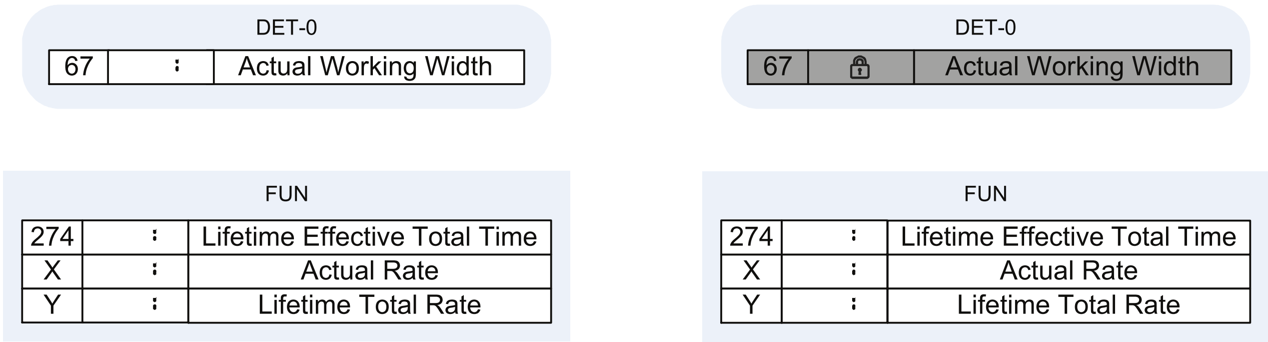


Figure F.41 — Single operation Device with actual rate and working width DDIs and a separate Function device element

The totals can be provided in multiple device elements. In Figure F.42, an example is provided of a lifetime total time that is available for the Device, specified in the root device element, as well as a lifetime total time that is recorded for a specific device element, the Function. Each lifetime total time is updated based on the logic of the device element it is specified in. All lifetime totals from the Function level do not necessarily need to match the lifetime total of the Device level. For the LOG generation 1 functionality, a lifetime total time DDI at Device level is mandatory, any lifetime total time DDIs inside other device elements are optional.

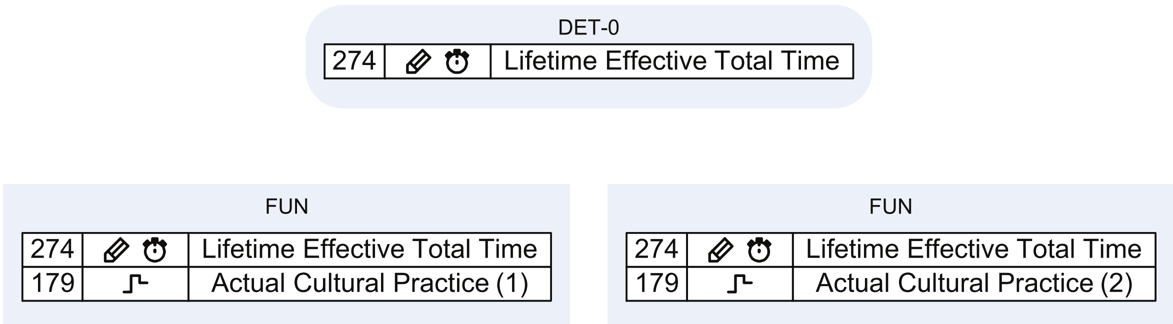


Figure F.42 — Total times present at Device level and Function level

A couple other optional attributes added to Figure F.42 are the Actual Cultural Practice DDIs. In this example, they represent a fertilizer operation and a seeding or planting operation, both executed by the same device. The total times of each operation can thus be independently recorded.

Instead of a generic Function device element, the rate DDIs and associated lifetime totals can also be specified as attributes of a Bin device element. Two examples of these device descriptor object pools are provided in Figure F.43. In this example, the rate and working width information are recommended. The root device element represents the boom of the device. The location of the Bin device elements as a child of the boom specifies that the product from this bin is distributed through that boom. It is not allowed to have the same rate DDI attached to both the root device element and the bin device element. The lifetime total time in the root device element is mandatory and the lifetime total rate DDI shall be located in the same device element that has the actual rate DDI.

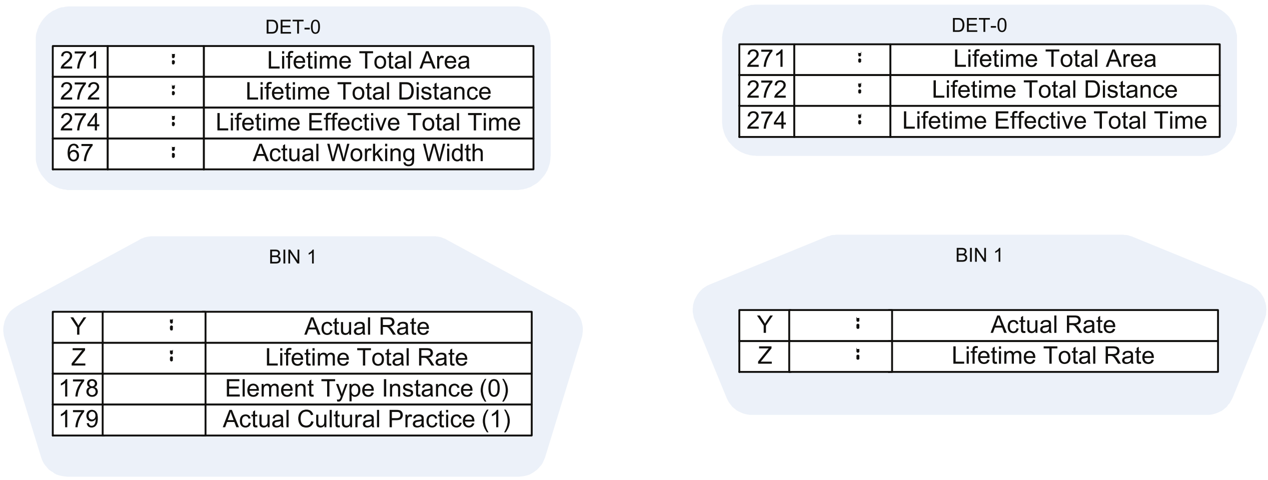


Figure F.43 — Single operation Device with a Bin device element

Figure F.44 expands on Figure F.43 by adding a secondary operation to the device and separate Function device elements that represent two independent product application booms. In this example, each Function device element represents a boom. The location of the Bin device elements as children of the booms specifies that the products from these bins are distributed through these booms. The same rate DDI shall not occur at both the Function and the Bin device elements, nor be attached to the root device element and the boom Function device element simultaneously.

The first function on the left-hand side in Figure F.44 represents a fertilizer operation (Actual Cultural Practice DDI has value 1), the second function represents a seeding or planting operation (Actual Cultural Practice DDI value is 2). The lifetime total values inside the root device element represent the overall device lifetime totals. The lifetime total values in the Function device elements represent just the total values for each operation. The sum of the function level total values is not necessarily the same as the overall device total values. The lifetime total time in the root device element is mandatory, lifetime total time DDIs in the Function type device elements are optional.

In case of multiple booms, the Actual Cultural Practice DDI (179) shall be placed either in the boom Function device element or in the underlying Bin device element.

The Element Type Instance DDI (178) shall be used to identify a bin to the operator to enable to select from which bin a product is applied. This implies that the Element Type Instance DDI shall be placed inside the Bin device element.

DET-0

271	:	Lifetime Total Area
272	:	Lifetime Total Distance
274	:	Lifetime Effective Total Time

FUN 1

271	:	Lifetime Total Area
272	:	Lifetime Total Distance
274	:	Lifetime Effective Total Time
67	:	Actual Working Width

BIN 1

Y	:	Actual Rate
Z	:	Lifetime Total Rate
178		Element Type Instance (0)
179		Actual Cultural Practice (1)

FUN 1

271	:	Lifetime Total Area
272	:	Lifetime Total Distance
274	:	Lifetime Effective Total Time
67	:	Actual Working Width

BIN 1

Y	:	Actual Rate
Z	:	Lifetime Total Rate
178		Element Type Instance (1)
179		Actual Cultural Practice (2)

Figure F.44 — Multi operation Planter with two Functions and two Bins

The use of the Actual Cultural Practice DDI is further illustrated in Figure F.45. In this example of a baler device descriptor object pool, the Actual Cultural Practice DDI value 5 identifies this device as a baler rather than, for example, a mower. Without the use of the Actual Cultural Practice DDI, as the device class for balers and mowers is forage harvesting, these devices would not be distinguishable.

DET-0

274		Lifetime Effective Total Time
179		Actual Cultural Practice (5)

Figure F.45 — Baler device descriptor, device class forage harvesting

Figure F.46 shows an example of a baler device descriptor that includes two levels of bale counting. In this example, the total values of the precut and uncut bales shall equal the yield total count DDI value.

DET-0

270	:	Lifetime Yield Total Count
274	:	Lifetime Effective Total Time
285	:	Lifetime Precut Total Count
286	:	Lifetime Uncut Total Count
179		Actual Cultural Practice (5)

Figure F.46 — Baler device descriptor, multiple totals

Another example for a specific device class is provided in Figure F.47. In this example, a set of lifetime totals for a tractor is specified to be able to be recorded.

DET-0

272	:	Lifetime Effective Total Distance
273	:	Lifetime Ineffective Total Distance
274	:	Lifetime Effective Total Time
275	:	Lifetime Ineffective Total Time
276	:	Lifetime Total Fuel Consumption

Figure F.47 — Tractor device descriptor, multiple totals

Annex G
(normative)

Task Based Time Registration

G.1 Time registration levels

Two levels of time registration can be used per the ISO 11783-10 standard. The first level is at Task level and can be used to register the occurrence of different types of times that a task can go through. The second level is at device data logging level. Each device can report states such as working or in-transport to a TC or DL and the state changes can be registered in a timelog.

G.2 Task level time registration

Within each task, Time XML elements can be added for the registration of the start, duration, and/or stop of different time types. Depending on the level of detail of the time registration implementation in a TC, different sets of time types may be used. Table G.1 lists the time type definitions that can be recorded. Table G.2 defines the levels of detail that can be distinguished in time registration implementations in this standard. Figure G.1 provides an example of a set of recorded time types for the minimal and intermediate time registration levels.

Table G.1 — Time Type definitions

Time Type	Definition
2. Preliminary	Time spent on preparing on the farm site, travelling to the field and preparing at the field. “Preliminary” time starts as soon as a Task is activated and stops when either “Effective” work or another time type is started on a Task. When recording of this specific timetype is not available, this time would fall under the “Ineffective” timetype.
4. Effective	Effective work covers the activities absolutely and directly required to fulfill the process of the work. “Effective” time of a Task starts when the main work of a Task started and contains the time spent on the main work. This timetype is the minimal requirement for the MICS to provide.
5. Ineffective	Is the time when the main/effective work of a Task is not active. The “Ineffective” time can be used between periods of “Effective” time.
6. Repair	Is the time allocated to repair work during a Task. When recording of this specific timetype is not available, this time would fall under the “Ineffective” timetype.
7. Clearing	Starts as soon as the “Effective” work is stopped for the last time during a Task and stops when the Task is stopped. When recording of this specific timetype is not available, this time would fall under the “Ineffective” timetype.
8. Powered Down	Starts as soon as the machine on which the time registration operates is powered down. When recording of this specific timetype is not available, this time would fall under the “Ineffective” timetype.

Table G.2 — Time Registration Levels

Level	MICS supported TimeTypes	Description
1. Minimal	4. Effective	Minimum requirement is to provide a Time recording with only this timetype for a Task. The sum of all effective Time XML elements in a Task is the total working time of that Task.
2. Intermediate	2. Preliminary 4. Effective 5. Ineffective 6. Repair 7. Clearing 8. Powered Down	Detection of main work allows for automatic recording of timetypes 2, 4, 5, and 7. The detection of main work may use work status information from the devices allocated to a Task. Timetype 6 may require operator entry and timetype 8 may be recorded automatically based upon information broadcasted on the implement bus.

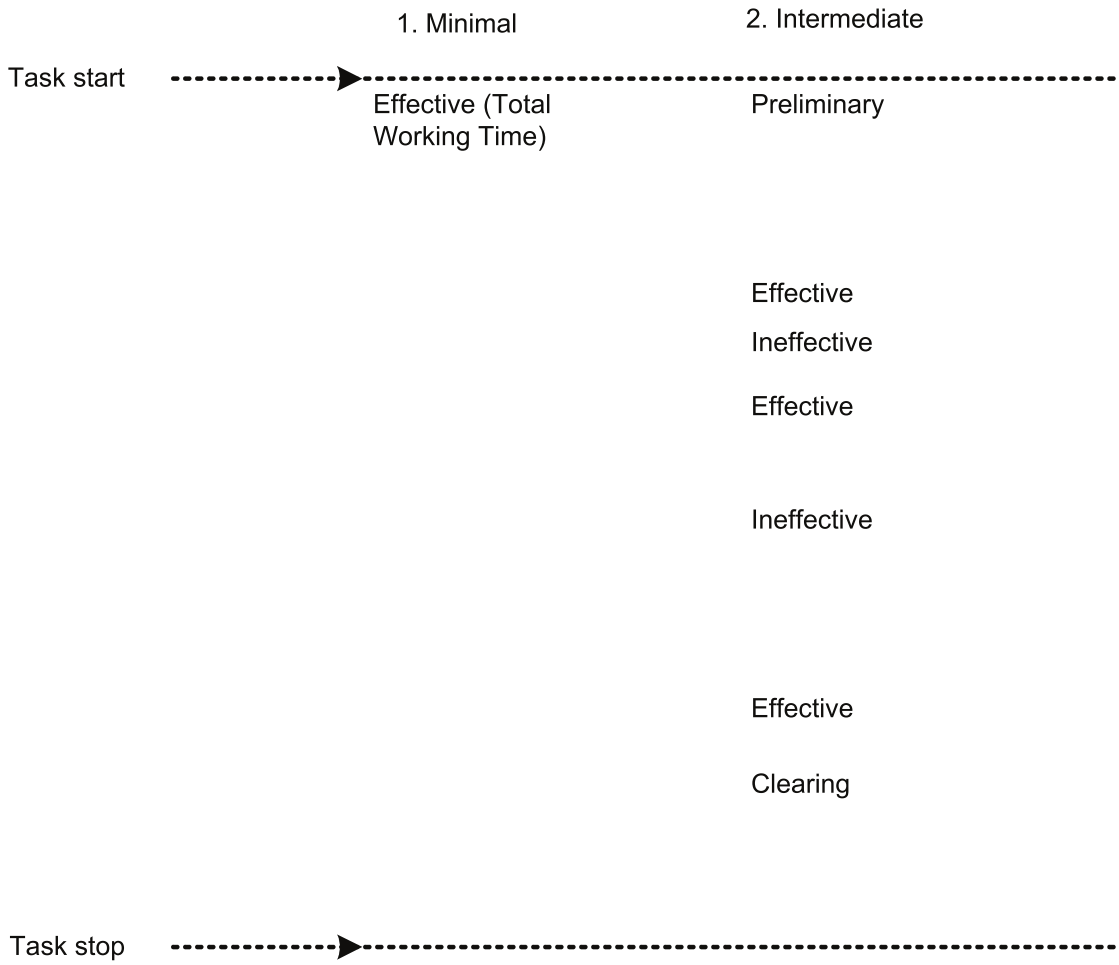


Figure G.1 — Example of time registration at two levels of detail

G.3 Device level time registration

All Device related data logging is based on either the data definitions present in the Data Dictionary Entities or the existence of Parameter Groups from which data can be logged. Two total time entities are defined in the Data Dictionary:

Table G.3 — Data dictionary time entities

DDI	Entity	Definition	Unit
119	Effective Total Time	Accumulated Time in working position and for nearly all field operations with a forward speed. A tractor driving an irrigation pump is an example of effective time without forward speed.	s
120	Ineffective Total Time	Accumulated Time out of working position or in standstill.	s

Remark: A plough can be in the working position while the tractor is not driving. When he does not drive, it is not an Effective time, but ineffective.

These two definitions enable logging of the total time that Devices or parts of Devices were either in an Effective or Ineffective work state.

The ISO 11783-7 and SAE-J1939 standards also list Parameter Groups from which information relevant for time recording can be obtained. Examples of these Parameter Groups are the “Wheel-based speed and distance” and “Implement operating state command”.

Bibliography

[1] SAE J1939, *Recommended Practice for a Serial Control and Communications Vehicle Network*

[2] AEF. (Ag Industry Electronics Foundation, http://www.aef-online.org), guidelines on the definition of ISOBUS functionalities.

[3] ISO 19136:2007, *Geographic information — Geography Markup Language (GML)*

ICS 65.060.01; 35.240.99

Price based on 205 pages

This page deliberately left blank