STS Association

STS 201-1

Edition1.2 April 2018

STANDARD TRANSFER SPECIFICATION –

Companion specification – Meter function object: RegisterTable for payment meters

CONTENTS

Revision History:

Revision	**Clause**	**Date**	**Change details from previous Edition**
CDV			Added new register definitions Altered some register names to align with new register names Added some explanatory notes Removed reference to specific TokenCarrierTypes and EncryptionAlgorithms since they are defined in other referenced documents Added CTS test default register (2029) Added mandatory support for Registers 2027 and 2028
CDV	general	June 2015	The document numbering was changed from STS201-15-x-x to STS201-1
Edition1 2015	general	July 2015	Changed format descriptor from 32 to 20 bit for registers 2019 to 201C and 2023 to 2026
Edition 1.1 2017	general	Aug 2017	Added all the new registers as defined in STS202-5
Edition 1.2 2018	7.43 Table40	April 2018	Added CTS reset for register 2029 with 'D' value 99

STANDARD TRANSFER SPECIFICATION ASSOCIATION

_____

COMPANION SPECIFICATION

STS 201-1: Standard transfer specification (STS) – Companion specification – Meter function object: RegisterTable for electricity payment meters

FOREWORD

The STS Association is a Not for Profit Company registered in terms of South African Law. The organisation holds an annual general meeting of members where the members elect nominated members to the board. The board consists of elected directors as well as one director each from the four founding organisations, Itron, Conlog, Landis+Gyr and Eskom in South Africa. The board is supported by a secretariat provided by the firm VdW&Co in Johannesburg, South Africa.

The Standard Transfer Specification (STS) has become recognized as the only globally accepted open standard for prepayment systems, ensuring inter-operability between system components from different manufacturers of prepayment systems. The application of the technology is licensed through the STS Association, thus ensuring that the appropriate encryption key management practices are applied to protect the security of the prepayment transactions of utilities operating STS systems.

It has become established as a worldwide standard for the transfer of electricity prepayment tokens since its introduction in South Africa in 1993 and subsequent publication by the International Electrotechnical Commission as the IEC62055 series of specifications.

Address: The STS Association, P.O. Box 868, Ferndale 2160, Republic of South Africa.
Tel: +27 061 5000
Fax: +27 86 680 7449
Email: email@sts.org.za
Website: http://www.sts.org.za

INTRODUCTION

STS is a secure message system for carrying information between a point-of-sale and a meter, and is currently finding wide application in electricity, water, gas, time, and currency metering and payment systems.

The STS series of companion Specifications have been introduced to formalise additional functionality available to those users requiring facilities not addressed in the IEC 62055 series of specifications

It is envisaged that the STS companion specifications will ultimately be adopted by the IEC in the IEC62055 series of specifications

The Standard Transfer Specification (STS) is a secure message protocol that allows information to be carried between point-of-sale (POS) equipment and payment meters and it caters for several message types such as credit, configuration control, display and test instructions. It further specifies devices and codes of practice that allows for the secure management (generation, storage, retrieval and transportation) of cryptographic keys used within the system.

1 Scope

This companion specification is intended to document and adopt the de-facto industry standard "credit reader port" functionality as implemented by existing prepayment meters.
The specification is for complementary use with companion specification STS 101-1 and is envisaged by IEC 62055-52 as an instance of a MeterFunctionObject.

It further expands the STS token carrier interface suite to include a virtual token carrier port and effectively records what is already a de-facto industry standard as used in millions of STS meters worldwide.

The specification aligns and maintains compatibility with the specifications: IEC 62055-41Ed2, IEC 62055-52. IEC 62055-31 and IEC 62055-51.

2 Normative references

IEC 62055-41 Ed2, Electricity metering – Payment systems Part 41: Standard Transfer Specification – Application Layer Protocol for one-way token carrier systems.

IEC 62055-52, Electricity metering – Payment systems Part 52: Standard Transfer Specification – Physical Layer Protocol for a two-way virtual token carrier for direct local connection.

STS202-3 - Addendum to IEC62055- 41: Standard transfer specification (STS)– EA11 (MISTY1) and DKGA04 (KDF-HMAC-SHA-256)

STS202-1 - Addendum to IEC62055-41: ELECTRICITY METERING PAYMENT SYSTEMS – Currency Token

STS203-1 - ELECTRICITY METERING – PAYMENT SYSTEMS – Method for default Payment Meter values for conformance testing

STS531-9-1-07 - COMPLIANCE TEST SPECIFICATION – Entity Type I - Physical layer protocol for a two-way virtual token carrier for direct local connection for Port types A and B for EA=07

STS531-9-1-11 - COMPLIANCE TEST SPECIFICATION – Entity Type I - Physical layer protocol for a two-way virtual token carrier for direct local connection for Port types A and B for EA=11

STS202-5 Addendum to IEC62055- 41: Payment Systems - Standard Transfer Specification (STS) - Class 2 token extention

3 Terms and definitions

3.1 Definitions

For the purposes of this document the Terms and Definitions given in IEC 60050-300, IEC 62055-41 Ed2, IEC 62055-52. IEC 62055-31 and IEC 62055-51shall generally apply.

Where there is a difference between the definitions in this standard and those contained in other referenced IEC standards, then those defined in this standard shall take precedence.

3.2 Abbreviated Terms

The following abbreviated terms are used throughout this STS Companion Specification:

ACK	Acknowledge
DL	Data Length
FOIN	FunctionObjectIdentificationNumber
GPS	Global Positioning System
IEC	International Electrotechnical Commission
I_{IN}	Current IN

I_{OH}	Current OUT High
I_{OL}	Current OUT Low
I_{OUT}	Current OUT
ISO	International Standards Organisation
KRN	KeyRevisionNumber
KT	KeyType
kWh	Kilo Watt Hour
NAK	Negative Acknowledge
RID	RegisterIdentifier
SGC	SupplyGroupCode
SC	Social Credit
STS	Standard Transfer Specification
TCDU	TokenCarrierDataUnit
TID	TokenIdentifier
V_{IN}	Voltage IN
V_{OH}	Voltage OUT High
V_{OL}	Voltage OUT Low

3.3 Notation and terminology

Throughout this standard the following rules are observed regarding the naming of terms.

1. Entity names, data element names, function names and process names are treated as generic object classes and are given names in terms of phrases in which the words are capitalized and joined without spaces. Examples are: SupplyGroupCode as a data element name, EncryptionAlgorithm07 as a function name and TransferCredit as a process name (see Note 1).
2. Direct (specific) reference to a named class of object uses the capitalized form, while general (non-specific) reference uses the conventional text i.e. lower case form with spaces. An example of a direct reference is: “The SupplyGroupCode is linked to a group of meters”, while an example of a general reference is: “A supply group code links to a vending key”.
3. Other terms use the generally accepted abbreviated forms like PSTN for Public Switched Telephone Network.

Note 1 The notation used for naming of objects has been aligned with the so-called “camel-notation” used in the common information model (CIM) standards prepared by IEC TC57, in order to facilitate future harmonization and integration of payment system standards with the CIM standards.

3.4 Numbering conventions

In this standard, the representation of numbers in binary strings uses the convention that the least significant bit is to the right and the most significant bit is to the left.
Numbering of bit positions start with bit position 0, which corresponds to the least significant bit of a binary number.
Numbers are generally in decimal format, unless otherwise indicated. Any digit without an indicator signifies decimal format.
Binary digit values range from 0-1.
Decimal digit values range from 0-9.
Hexadecimal digit values range from 0-9, A-F and are indicated by “hex”.

4 Reference model

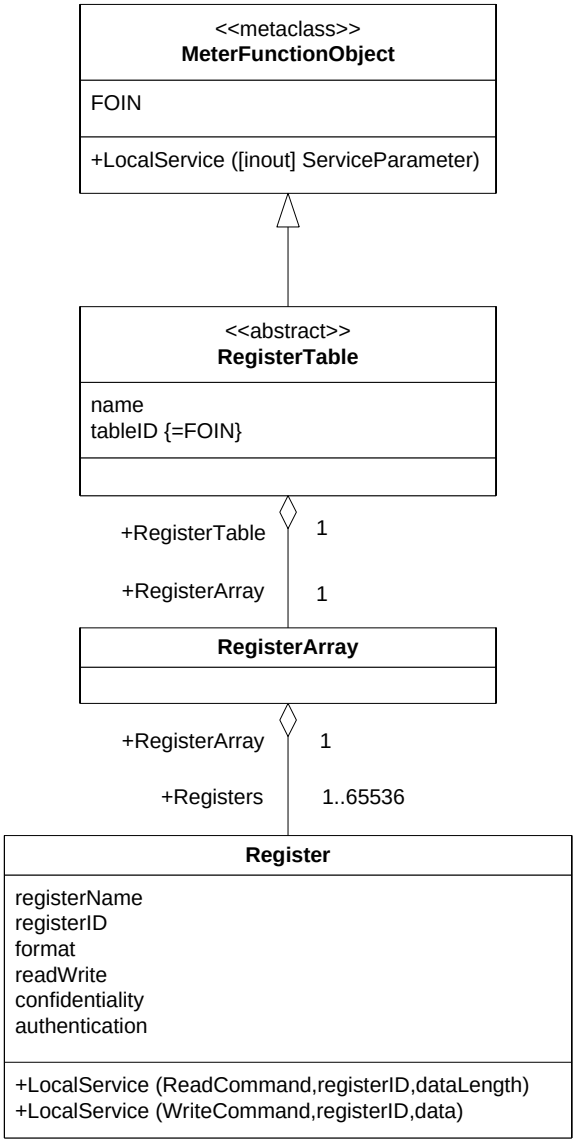

Figure .1 – UML representation of RegisterTable as an extension of MeterFunctionObject

RegisterTable is a kind of MeterFunctionObject and contains 1 RegisterArray.

RegisterTable has 2 attributes name and tableID (which has a value = FOIN).

RegisterArray has no attributes and is merely an abstraction container.

RegisterArray contains 1..65536 Register instances.

Each Register has 6 attributes and 2 methods associated with it.

5 MeterFunctionObject : RegisterTable

Table1 shows the MeterFunctionObject register table.

Table 1 – RegisterTable instance

Attributes	**Range**	**Context**
name	**RegisterTable**	The registered name of this FunctionObject in the Companion Specification.
tableID	FOIN = 15.1.2	FunctionObjectIdentificationNumber as registered in the Companion Specification.
Data Elements		**External data interface to the FunctionObject instance** (from the TokenCarrier perspective)
RegisterArray	Array of simple or complex registers; quantity between 0 – 65535 Data Elements are defined per Register instance (see clause 7)	Array of Register instances defined in this Companion Specification. There may be up to 65536 entries in the table. Each Register instance may have a simple or complex dataset structure
Methods		**External service interface to the FunctionObject instance** (from the TokenCarrier perspective)
none	Methods are defined per Register instance (see clause 7) The RegisterTable has no operational methods	The table is virtual and cannot be accessed.
Operation		**Internal functionality of the RegisterTable**
none	Operations are defined per Register instance (see clause 7) The RegisterTable has no operational functionality	This table merely defines a list of Register definitions, each of which provides a logical interface to the actual payment meter registers or functions.
Association		**Support services provided by other FunctionObjects**
none	Associations are defined per Register instance (see clause 7)	The RegisterTable does not make use of other specified services for it's functioning.

The definition of the RegisterArray data element is given in clause 6.

6 RegisterArray

Some of the main attributes are given in Table 2.

Table 2 – RegisterArray with main attribute values for each register instance

registerID (hex)	registerName	readWrite	confidentiality	authentication	Reference
0000 – 1FFF	Proprietary reserved registers	x	x	True	7.1
2000	ProtocolVersion	R	False	False	7.2
2001	TableID	R	False	False	7.3
2002	ServerStatus	R	False	False	7.4
2003	SoftwareVersion	R	False	False	7.5
2004	BinaryTokenEntry	W	False	False	7.6
2005	TokenLockoutTimeRemaining	R	False	False	7.7
2006	DecoderReferenceNumber(11 Digit)	R	False	False	7.8
2007	PrimaryTokenCarrierType	R	False	False	7.9
2008	EncryptionAlgorithm	R	False	False	7.10
2009	TariffIndex	R	False	False	7.11
200A	KeyRevisionKeyType	R	False	False	7.12
200B	KeyExpiryNumber	R	False	False	7.13
200C	MaximumPowerLimit	R	False	False	7.14
200D	MaximumPowerPhaseUnbalanceLimit	R	False	False	7.15
200E	TariffRate	R	False	False	7.16
200F	WaterMeterFactor	R	False	False	7.17
2010	AvailableElectricityCredit	R	False	False	7.18
2011	CumulativeElectricityEnergyConsumption	R	False	False	7.19
2012	LastCreditToken	R	False	False	7.20
2013	LastCreditTokenID	R	False	False	7.21
2014	TamperStatus	R	False	False	7.22
2015	GPSCoordinates	R,W	False	False	7.23
2016	SupplyGroupCode	R,W	False	False	7.24
2017	DecoderReferenceNumber(13 Digit)	R	False	False	7.25
2018	TIDBaseYear	R,W	False	False	7.26
2019	AvailableElectricityCurrency	R	False	False	7.27
201A	AvailableWaterCurrency	R	False	False	7.28
201B	AvailableGasCurrency	R	False	False	7.29
201C	AvailableTimeCurrency	R	False	False	7.30
201D	AvailableWaterCredit	R	False	False	7.31
201E	AvailableGasCredit	R	False	False	7.32
201F	AvailableTimeCredit	R	False	False	7.33
2020	CumulativeWaterConsumption	R	False	False	7.34
2021	CumulativeGasConsumption	R	False	False	7.35
2022	CumulativeTimeConsumption	R	False	False	7.36
2023	CumulativeElectricityCurrencyConsumption	R	False	False	7.37
2024	CumulativeWaterCurrencyConsumption	R	False	False	7.38
2025	CumulativeGasCurrencyConsumption	R	False	False	7.39
2026	CumulativeTimeCurrencyConsumption	R	False	False	7.40

2027	PowerLimitingState	R	False	False	7.41
2028	NumberOfKCTSupported	R	False	False	7.42
2029	SetCTSDefault (see Note1)	W	False	False	7.43
202A	FlagSettingsRegister	R	False	False	7.44
202B	ControlElementSettingsRegister	R	False	False	7.45
202C – 8FFF	STS reserved registers	x	x	x	7.46
9000 – FFFD	Proprietary reserved registers	x	x	True	7.47
FFFE	TokenStatus	R	False	False	7.48
FFFF	NumericTokenEntry	W	False	False	7.49

The definition of each Register instance is given in the clause indicated by the Reference column.

Note: Mandatory support for registers shall be as defined in IEC62055-52 Table15. Support for all other registers is optional, except for the following registers, support of which is mandatory:

Register 2027 - PowerLimitingState

Register 2028 - NumberOfKCTSupported

Note1: The default values to be set in the payment meter by writing to this register are found in STS531-9-1-07 Table 1 and STS531-9-1-11 Table 1.

7 Register instance definitions

7.1 Proprietary reserved registers at registerID values 0000 – 1FFF hex

Manufacturers may assign the registerID values 0000 – 1FFF hex to proprietary register implementations.

The standard does not prevent collision between register identifiers assigned by different manufacturers, and each manufacturer should thus ensure that the WriteCommand messages used are authenticated against a unique identifier such as the MfrCode (see 6.1.2.3.2 of IEC 62055-41) and/or the SoftwareVersion (see 6.8.3.5 of IEC 62055-52).

7.2 ProtocolVersion register

This shall be a predefined register as given in 6.8.3.2 of IEC 62055-52 with a pre-assigned registerID = 2000 hex.

7.3 TableID register

This shall be a predefined register as given in 6.8.3.3 of IEC 62055-52 with a pre-assigned registerID = 2001 hex.

7.4 ServerStatus register

This shall be a predefined register as given in 6.8.3.4 of IEC 62055-52 with a pre-assigned registerID = 2002 hex.

7.5 SoftwareVersion register

This shall be a predefined register as given in 6.8.3.5 of IEC 62055-52 an assigned registerID = 2003 hex.

7.6 BinaryTokenEntry register

This shall be a predefined register as given in 6.8.3.6 of IEC 62055-52 with an assigned registerID = 2004 hex.

7.7 TokenLockoutTimeRemaining register

This shall be a predefined register as given in 6.8.3.8 of IEC 62055-52 with an assigned registerID = 2005 hex.

7.8 DecoderReferenceNumber register – 2 digit Manufacturer Code (for 4 digit Manufacturer Code, see 7.25)

Table 3 – Composition of the DecoderReferenceNumber register

Attributes	Range	Context
registerName	**DecoderReferenceNumber**	Register name in this table
registerID	2006 hex	Register identifier in this table
format	11 decimal digits	Presentation format to the interface
readWrite	R	Read only
confidentiality	False	The content of this register is not confidential
authentication	False	Client authentication not required
Data Elements		**External data interface to the Register instance** (from the TokenCarrier perspective)
DecoderReferenceNumber	11-digit DRN as defined in 6.1.2.3 of IEC 62055-41	
Methods		**External service interface to the Register instance** (from the TokenCarrier perspective)
ReadCommand (RID, DL)	Returns the value of the DRN RID = registerID to be read DL = ignored by server	The data may be retrieved from the Register by sending a ReadCommand message to the server with the registerID value in the RID field. The DL field is ignored.
Operation		**Internal functionality of the Register**
Virtual record	Keeps virtual record of the value of the DRN	This object merely defines a virtual Register, which provides a logical interface to the actual payment meter register or function.
Association		**Support services provided by other FunctionObjects**
none		The register does not make use of other specified services for it's functioning.

Where the MfrCode part of the DRN (see 6.1.2.3.2 of IEC 62055-41) is smaller than 10, then the value shall be right justified and left padded with a decimal zero to always make two decimal digits. For example: 01, 02, - 09.

The DSN part of the DRN (see 6.1.2.3.3 of IEC 62055-41) shall be right justified and left padded with leading zeros to make up the full 8 digits. For example 00000078.

Thus the DRN shall always be presented as a full 11-digit decimal number.

If this register is read and the server is a 4 digit manufacturer code server (13 digit DRN), then the server will return a NAK when the client tries to read this register with an appropriate code in the ServerStatus register (see 6.6.3 of IEC 62055-52).

7.9 PrimaryTokenCarrierType register

Table 4 – Composition of the PrimaryTokenCarrierType register

Attributes	Range	Context
registerName	**PrimaryTokenCarrierType**	Register name in this table
registerID	2007 hex	Register identifier in this table
format	2 decimal digits	Presentation format to the interface

readWrite	R	Read only
confidentiality	False	The content of this register is not confidential
authentication	False	Client authentication not required
Data Elements		**External data interface to the Register instance** (from the TokenCarrier perspective)
TokenCarrierType	2-digit decimal number 00 – 99 as defined in 6.1.3 of IEC 62055-41	The primary token carrier interface the user will normally use to enter credit tokens.
Methods		**External service interface to the Register instance** (from the TokenCarrier perspective)
ReadCommand (RID, DL)	Returns the value of the PrimaryTokenCarrierType RID = registerID to be read DL = ignored by server	The data may be retrieved from the Register by sending a ReadCommand message to the server with the registerID value in the RID field. The DL field is ignored.
Operation		**Internal functionality of the Register**
Virtual record	Keeps virtual record of the value of the PrimaryTokenCarrierType	This object merely defines a virtual Register, which provides a logical interface to the actual payment meter register or function.
Association		**Support services provided by other FunctionObjects**
none		The register does not make use of other specified services for it's functioning.

Values less than 10 shall be right justified and left padded with a decimal zero to always make two decimal digits. For example: 01, 02, - 09.

7.10 EncryptionAlgorithm register

Table 5 – Composition of the EncryptionAlgorithm register

Attributes	Range	Context
registerName	**EncryptionAlgorithm**	Register name in this table
registerID	2008 hex	Register identifier in this table
format	2 decimal digits	Presentation format to the interface
readWrite	R	Read only
confidentiality	False	The content of this register is not confidential
authentication	False	Client authentication not required
Data Elements		**External data interface to the Register instance** (from the TokenCarrier perspective)
EncryptionAlgorithm	2-digit decimal number 00 – 99 as defined in 6.1.5 of IEC 62055-41	The encryption algorithm to be used for encrypting tokens for use with this payment meter.
Methods		**External service interface to the Register instance** (from the TokenCarrier perspective)
ReadCommand (RID, DL)	Returns the value of the EncryptionAlgorithm RID = registerID to be read DL = ignored by server	The data may be retrieved from the Register by sending a ReadCommand message to the server with the registerID value in the RID field. The DL field is ignored.
Operation		**Internal functionality of the Register**
Virtual record	Keeps virtual record of the value of the EncryptionAlgorithm	This object merely defines a virtual Register, which provides a logical interface to the actual payment meter register or function.
Association		**Support services provided by other FunctionObjects**
none		The register does not make use of other specified services for it's functioning.

Values less than 10 shall be right justified and left padded with a decimal zero to always make two decimal digits. For example: 01, 02, - 09.

7.11 TariffIndex register

Table 6 – Composition of the TariffIndex register

Attributes	Range	Context
registerName	**TariffIndex**	Register name in this table
registerID	2009 hex	Register identifier in this table
format	2 decimal digits	Presentation format to the interface
readWrite	R	Read only
confidentiality	False	The content of this register is not confidential
authentication	False	Client authentication not required
Data Elements		**External data interface to the Register instance** (from the TokenCarrier perspective)
TariffIndex	2-digit decimal number 00 – 99 as defined in 6.1.7 of IEC 62055-41	The TariffIndex value to which the payment meter is set.
Methods		**External service interface to the Register instance** (from the TokenCarrier perspective)
ReadCommand (RID, DL)	Returns the value of the TariffIndex RID = registerID to be read DL = ignored by server	The data may be retrieved from the Register by sending a ReadCommand message to the server with the registerID value in the RID field. The DL field is ignored.
Operation		**Internal functionality of the Register**
Virtual record	Keeps virtual record of the value of the TariffIndex	This object merely defines a virtual Register, which provides a logical interface to the actual payment meter register or function.
Association		**Support services provided by other FunctionObjects**
none		The register does not make use of other specified services for it's functioning.

Values less than 10 shall be right justified and left padded with a decimal zero to always make two decimal digits. For example: 01, 02, - 09.

7.12 KeyRevisionKeyType register

Table 7 – Composition of the KeyRevisionKeyType register

Attributes	Range	Context
registerName	**KeyRevisionKeyType**	Register name in this table
registerID	200A hex	Register identifier in this table
format	2 decimal digits	Presentation format to the interface
readWrite	R	Read only
confidentiality	False	The content of this register is not confidential
authentication	False	Client authentication not required
Data Elements		**External data interface to the Register instance** (from the TokenCarrier perspective)
KeyRevisionKeyType	2-digit composite decimal number 10 – 93 as defined in 6.1.8 and 6.1.9 of IEC 62055-41	The combined value of the concatenated KRN and KT digits

Methods ReadCommand (RID, DL)	 Returns the value of the KeyRevisionKeyType RID = registerID to be read DL = ignored by server	**External service interface to the Register instance** (from the TokenCarrier perspective) The data may be retrieved from the Register by sending a ReadCommand message to the server with the registerID value in the RID field. The DL field is ignored.
Operation Virtual record	 Keeps virtual record of the value of the KeyRevisionKeyType	**Internal functionality of the Register** This object merely defines a virtual Register, which provides a logical interface to the actual payment meter register or function.
Association none		**Support services provided by other FunctionObjects** The register does not make use of other specified services for it's functioning.

This register is a composite of two values: the most significant digit is the KRN value and the least significant digit is the KT value.

7.13 KeyExpiryNumber register

Table 8 – Composition of the KeyExpiryNumber register

Attributes	Range	Context
registerName	**KeyExpiryNumber**	Register name in this table
registerID	200B hex	Register identifier in this table
format	8-bit binary	Presentation format to the interface
readWrite	R	Read only
confidentiality	False	The content of this register is not confidential
authentication	False	Client authentication not required
Data Elements KeyExpiryNumber	 8-bit binary number 0 – 255 as defined in 6.1.10 of IEC 62055-41	**External data interface to the Register instance** (from the TokenCarrier perspective) The time period, during which the DecoderKey will remain valid.
Methods ReadCommand (RID, DL)	 Returns the value of the KeyExpiryNumber RID = registerID to be read DL = ignored by server	**External service interface to the Register instance** (from the TokenCarrier perspective) The data may be retrieved from the Register by sending a ReadCommand message to the server with the registerID value in the RID field. The DL field is ignored.
Operation Virtual record	 Keeps virtual record of the value of the KeyExpiryNumber	**Internal functionality of the Register** This object merely defines a virtual Register, which provides a logical interface to the actual payment meter register or function.
Association none		**Support services provided by other FunctionObjects** The register does not make use of other specified services for it's functioning.

7.14 MaximumPowerLimit register

Table 9 – Composition of the MaximumPowerLimit register

Attributes	Range	Context
registerName	**MaximumPowerLimit**	Register name in this table
registerID	200C hex	Register identifier in this table

format	16-bit binary	Presentation format to the interface
readWrite	R	Read only
confidentiality	False	The content of this register is not confidential
authentication	False	Client authentication not required
Data Elements		**External data interface to the Register instance** (from the TokenCarrier perspective)
MaximumPowerLimit	16-bit binary number as defined in 6.3.9 of IEC 62055-41	The maximum power that the load may draw from the payment meter as set in the payment meter.
Methods		**External service interface to the Register instance** (from the TokenCarrier perspective)
ReadCommand (RID, DL)	Returns the value of the MaximumPowerLimit RID = registerID to be read DL = ignored by server	The data may be retrieved from the Register by sending a ReadCommand message to the server with the registerID value in the RID field. The DL field is ignored.
Operation		**Internal functionality of the Register**
Virtual record	Keeps virtual record of the value of the MaximumPowerLimit	This object merely defines a virtual Register, which provides a logical interface to the actual payment meter register or function.
Association		**Support services provided by other FunctionObjects**
none		The register does not make use of other specified services for it's functioning.

The power limiting function is optional in a payment meter (see Table C.3 in C.11 of IEC 62055-41).

If the function is not implemented then the server will return a NAK when the client tries to read this register with an appropriate code in the ServerStatus register (see 6.6.3 of IEC 62055-52).

If the function is supported, but disabled then it shall give a FunctionDisabled status code indication in the ServerStatus register (see 7.4) and the server will return a NAK.

7.15 MaximumPhasePowerUnbalanceLimit register

Table 10 – Composition of the MaximumPhasePowerUnbalanceLimit register

Attributes	Range	Context
registerName	**MaximumPhasePowerUnbalanceLimit**	Register name in this table
registerID	200D hex	Register identifier in this table
format	16-bit binary	Presentation format to the interface
readWrite	R	Read only
confidentiality	False	The content of this register is not confidential
authentication	False	Client authentication not required
Data Elements		**External data interface to the Register instance** (from the TokenCarrier perspective)
MaximumPowerPhase Unbalance Limit	16-bit binary number as defined in 6.3.10 of IEC 62055-41	The maximum allowable power difference between phase loads as set in the payment meter.
Methods		**External service interface to the Register instance** (from the TokenCarrier perspective)

ReadCommand (RID, DL)	Returns the value of the MaximumPowerPhaseUnbalanceLimit RID = registerID to be read DL = ignored by server	The data may be retrieved from the Register by sending a ReadCommand message to the server with the registerID value in the RID field. The DL field is ignored.
Operation Virtual record	Keeps virtual record of the value of the MaximumPowerPhaseUnbalanceLimit	**Internal functionality of the Register** This object merely defines a virtual Register, which provides a logical interface to the actual payment meter register or function.
Association none		**Support services provided by other FunctionObjects** The register does not make use of other specified services for it's functioning.

The phase power unbalance limiting function is optional in a payment meter (see Table C.3 in C.11 of IEC 62055-41).

If the function is not implemented then the server will return a NAK when the client tries to read this register with an appropriate code in the ServerStatus register (see 6.6.3 of IEC 62055-52).

If the function is supported, but disabled then it shall give a FunctionDisabled status code indication in the ServerStatus register (see 7.4) and the server will return a NAK.

7.16 TariffRate register

This register is reserved for future definition by the STS Association (see 6.3.11 of IEC 62055-41).

For the sake of continuity, it is provisionally assigned a registerID value of 200E hex.

A payment meter shall not implement this register.

If a client tries to read this register then it shall give a registerIDInvalid status code indication in the ServerStatus register (see 7.4) and the server will return a NAK (see 6.6.3 of IEC 62055-52).

7.17 WaterMeterFactor register

This register is reserved for future definition by the STS Association (see 6.3.12 of IEC 62055-41).

For the sake of continuity, it is provisionally assigned a registerID value of 200F hex.

A payment meter shall not implement this register.

If a client tries to read this register then it shall give a registerIDInvalid status code indication in the ServerStatus register (see 7.4) and the server will return a NAK (see 6.6.3 of IEC 62055-52).

7.18 AvailablekWhCredit register

Table 11 – Composition of the AvailablekWhCredit register

Attributes	Range	Context
registerName	**AvailablekWhCredit**	Register name in this table
registerID	2010 hex	Register identifier in this table
format	32-bit binary	Presentation format to the interface
readWrite	R	Read only
confidentiality	False	The content of this register is not confidential

authentication	False	Client authentication not required
Data Elements AvailablekWhCredit	Sign-bit+31-bit integer as defined in 3.1.2 of IEC 62055-31 units are in 100 watt-hours	**External data interface to the Register instance** (from the TokenCarrier perspective) The amount of credit available for consumption as reflected by the accounting function in the payment meter.
Methods ReadCommand (RID, DL)	Returns the value of the AvailablekWhCredit RID = registerID to be read DL = ignored by server	**External service interface to the Register instance** (from the TokenCarrier perspective) The data may be retrieved from the Register by sending a ReadCommand message to the server with the registerID value in the RID field. The DL field is ignored.
Operation Virtual record	Keeps virtual record of the value of the AvailablekWhCredit	**Internal functionality of the Register** This object merely defines a virtual Register, which provides a logical interface to the actual payment meter register or function.
Association none		**Support services provided by other FunctionObjects** The register does not make use of other specified services for it's functioning.

The format of the data presentation for sending over the communication link is a 32-bit binary number.

The application interpretation of the 32-bit value is:

- Most significant bit = sign bit; (0 = plus ; 1 = minus)
- Least significant 31 bits = 31-bit integer value;
- The least significant bit in the integer represents 0.1 kWh unit;
- Any smaller fractional part of the register value is truncated.

7.19 CumulativekWhEnergyConsumption register

Table 12 – Composition of the CumulativekWhEnergyConsumption register

Attributes	Range	Context
registerName	**CumulativekWhEnergyConsumption**	Register name in this table
registerID	2011 hex	Register identifier in this table
format	32-bit binary	Presentation format to the interface
readWrite	R	Read only
confidentiality	False	The content of this register is not confidential
authentication	False	Client authentication not required
Data Elements CumulativekWhEnergyConsumption	Sign-bit+31-bit integer as defined in 5.11.4 of IEC 62055-31 units are in 100 watt-hours	**External data interface to the Register instance** (from the TokenCarrier perspective) The cumulative total consumption of kWh energy delivered to the customer's load as registered over the lifetime of the payment meter.
Methods ReadCommand (RID, DL)	Returns the value of the CumulativekWhEnergyConsumption RID = registerID to be read DL = ignored by server	**External service interface to the Register instance** (from the TokenCarrier perspective) The data may be retrieved from the Register by sending a ReadCommand message to the server with the registerID value in the RID field. The DL field is ignored.
Operation		**Internal functionality of the Register**

Virtual record	Keeps virtual record of the value of the CumulativekWhEnergyConsumption	This object merely defines a virtual Register, which provides a logical interface to the actual payment meter register or function.
Association none		**Support services provided by other FunctionObjects** The register does not make use of other specified services for it's functioning.

The value returned by reading from this register shall be the content of the cumulative kWh energy consumption register as defined in 5.11.4 of IEC 62055-31.

In the case where more than one cumulative kWh energy consumption register is implemented in the payment meter, then the value returned from reading the CumulativekWhEnergyConsumption register shall be the cumulative total energy that is registered over the lifetime of the payment meter.

The format of the data presentation for sending over the communication link is a 32-bit binary number.

The application interpretation of the 32-bit value is:

- Most significant bit = sign bit; (0 = plus ; 1 = minus)
- Least significant 31 bits = 31-bit integer value;
- The least significant bit in the integer represents 0.1 kWh unit;
- Any smaller fractional part of the register value is truncated.

7.20 LastCreditToken register

Table 13 – Composition of the LastCreditToken register

Attributes	**Range**	**Context**
registerName	**LastCreditToken**	Register name in this table
registerID	2012 hex	Register identifier in this table
format	66-bit binary	Presentation format to the interface
readWrite	R	Read only
confidentiality	False	The content of this register is not confidential
authentication	False	Client authentication not required
Data Elements LastCreditToken	 66-bit binary as defined in 6.1.2 of IEC 62055-52 as TokenData in the TCDU	**External data interface to the Register instance** (from the TokenCarrier perspective) The most recent credit token that was accepted by the payment meter.
Methods ReadCommand (RID, DL)	 Returns the value of the LastCreditToken RID = registerID to be read DL = ignored by server	**External service interface to the Register instance** (from the TokenCarrier perspective) The data may be retrieved from the Register by sending a ReadCommand message to the server with the registerID value in the RID field. The DL field is ignored.
Operation Virtual record	 Keeps virtual record of the value of the LastCreditToken	**Internal functionality of the Register** This object merely defines a virtual Register, which provides a logical interface to the actual payment meter register or function.
Association none		**Support services provided by other FunctionObjects** The register does not make use of other specified services for it's functioning.

NOTE This is not necessarily the token with the most recent TID, as tokens may be entered in a different order to that, in which they were created.

This value is the last token that was accepted by the payment meter irrespective of which token carrier interface was used to enter it into the payment meter.

7.21 LastCreditTokenID register

Table 14 – Composition of the LastCreditTokenID register

Attributes	Range	Context
registerName	**LastCreditTokenID**	Register name in this table
registerID	2013 hex	Register identifier in this table
format	24-bit binary	Presentation format to the interface
readWrite	R	Read only
confidentiality	False	The content of this register is not confidential
authentication	False	Client authentication not required
Data Elements		**External data interface to the Register instance** (from the TokenCarrier perspective)
LastCreditTokenID	24-bit binary as defined in 6.2.2 and 6.3.5 of IEC 62055-41	The TID of the most recent credit token that was accepted by the payment meter.
Methods		**External service interface to the Register instance** (from the TokenCarrier perspective)
ReadCommand (RID, DL)	Returns the value of the LastCreditTokenID RID = registerID to be read DL = ignored by server	The data may be retrieved from the Register by sending a ReadCommand message to the server with the registerID value in the RID field. The DL field is ignored.
Operation		**Internal functionality of the Register**
Virtual record	Keeps virtual record of the value of the LastCreditTokenID	This object merely defines a virtual Register, which provides a logical interface to the actual payment meter register or function.
Association		**Support services provided by other FunctionObjects**
none		The register does not make use of other specified services for it's functioning.

This value is the TID of the token stored in LastCreditToken register (see 7.20).

7.22 TamperStatus register

Table 15 – Composition of the TamperStatus register

Attributes	Range	Context
registerName	**TamperStatus**	Register name in this table
registerID	2014 hex	Register identifier in this table
format	16-bit binary	Presentation format to the interface
readWrite	R	Read only
confidentiality	False	The content of this register is not confidential
authentication	False	Client authentication not required
Data Elements		**External data interface to the Register instance** (from the TokenCarrier perspective)

TamperStatus	16-bit status value as encoded in Table 16	
Methods ReadCommand (RID, DL)	 Returns the value of the status of the tamper function RID = registerID to be read DL = ignored by server	**External service interface to the Register instance** (from the TokenCarrier perspective) The data may be retrieved from the Register by sending a ReadCommand message to the server with the registerID value in the RID field. The DL field is ignored.
Operation Virtual record	 Keeps virtual record of the value of the status of the tamper function	**Internal functionality of the Register** This object merely defines a virtual Register, which provides a logical interface to the actual payment meter register or function.
Association none		**Support services provided by other FunctionObjects** The register does not make use of other specified services for it's functioning.

The implementation of this register is mandatory only if the payment meter supports a tamper detection function as agreed between the manufacturer and the utility.

The register definition does not model the complete functioning of the tamper detection function, but merely the interface to read the status of the function.

If the function is not implemented then the server will return a NAK when the client tries to read this register with an appropriate code in the ServerStatus register (see 6.6.3 of IEC 62055-52).

If the function is supported, but disabled then it shall give a FunctionDisabled status code indication in the ServerStatus register (see 7.4) and the server will return a NAK.

The encoded value of the TamperStatus register is given in Table 16

Table 16 – Bit-encoding of the TamperStatus register value

Value	Interpretation	
Bit 0 = 0	The payment meter has not detected any tamper condition	
Bit 0 = 1	The payment meter has detected a tamper condition	
Bit 1 =0	The payment meter has not detected any bypass condition	
Bit 1 =1	The payment meter has detected a bypass condition	
Bit 2 =0	The payment meter has not detected any consumption irregularities	
Bit 2 =1	The payment meter has detected consumption irregularities	
Bit 3 – 7	Reserved for STS assignment	
Bit 8 – 15	Assigned for proprietary use	

The TamperStatus register merely reports the status of these values and is not concerned with the operational functions in the payment meter that control the values. The operation and control of the tamper functions are defined in other relevant specifications.

7.23 GPSCoordinates register

Table 17 – Composition of the GPSCoordinates register

Attributes	Range	Context
registerName	**GPSCoordinates**	Register name in this table
registerID	2015 hex	Register identifier in this table
format	20 decimal digits	Presentation format to the interface

readWrite	R, W	Read and Write
confidentiality	False	The content of this register is not confidential
authentication	False	Client authentication not required
Data Elements		**External data interface to the Register instance** (from the TokenCarrier perspective)
GPSCoordinates	20 composite decimal digit string	The X and Y coordinates with reference to geographical Global Positioning System
Methods		**External service interface to the Register instance** (from the TokenCarrier perspective)
ReadCommand (RID, DL) WriteCommand (RID, D)	Returns the value of the GPSCoordinates RID = registerID to be read or written DL = ignored by server D = 20 digit GPS coordinates	The data may be retrieved from the Register by sending a ReadCommand message to the server with the registerID value in the RID field. The DL field is ignored.
Operation		**Internal functionality of the Register**
Virtual record	Keeps virtual record of the value of the GPSCoordinates	This object merely defines a virtual Register, which provides a logical interface to the actual payment meter register or function.
Association		**Support services provided by other FunctionObjects**
none		The register does not make use of other specified services for it's functioning.

This register is optional and implementation is subject to agreement between the manufacturer and the utility.

If the function is not implemented then the server will return a NAK when the client tries to read this register with an appropriate code in the ServerStatus register (see 6.6.3 of IEC 62055-52).

The dataset is 20 decimal digits in the following format:

XDDDmmss(.)ssYDDDmmss(.)ss

The position of the decimal point is implicit and is not part of the transmitted dataset in the server response message.

All elements are left padded with leading zeros to make up the full 20-digit dataset.

The dataset interpretation key is given in Table 18

Table 18 – Dataset interpretation key

Element	Range	Key		
X	0	"+"	Longitude	East
X	9	"-"	Longitude	West
Y	0	"+"	Latitude	North
Y	9	"-"	Latitude	South
DDD	000 – 180		degrees	
mm	00 – 59		minutes	
ss	00 – 59		seconds	
.ss	00 – 99	decimal fraction	seconds	(position of decimal point is implicit)

7.24 SupplyGroupCode register

Table 19 – Composition of the SupplyGroupCode register

Attributes	Range	Context
registerName	**SupplyGroupCode**	Register name in this table
registerID	2016 hex	Register identifier in this table
format	6 decimal digits	Presentation format to the interface
readWrite	R, W	Read and Write
confidentiality	False	The content of this register is not confidential
authentication	False	Client authentication not required
Data Elements		**External data interface to the Register instance** (from the TokenCarrier perspective)
SupplyGroupCode	6-digit decimal number as defined in 6.1.6 of IEC 62055-41	An uncontrolled code placed in this register by a client.
Methods		**External service interface to the Register instance** (from the TokenCarrier perspective)
ReadCommand (RID, DL) WriteCommand (RID, D)	Returns the value of the SupplyGroupCode RID = registerID to be read or written DL = ignored by server D = 6-digit SGC	The data may be retrieved from the Register by sending a ReadCommand message to the server with the registerID value in the RID field. The DL field is ignored.
Operation		**Internal functionality of the Register**
Virtual record	Keeps virtual record of the value of the SupplyGroupCode	This object merely defines a virtual Register, which provides a logical interface to the actual payment meter register or function.
Association		**Support services provided by other FunctionObjects**
none		The register does not make use of other specified services for it's functioning.

A client may set this register to any desired value at any time, but whenever the DecoderKey in the payment meter is changed, then the payment meter shall set the value of this register to 000000 (see also 6.2.7, 6.2.8, 8.9 and 8.10 of IEC 62055-41) if the payment meter supports only 2 keychange tokens in a keychange token set.

If the payment meter supports 3 or 4 keychange tokens in a keychange token set, then it shall set this value to the value of the Supply Group Code transmitted with the keychange token set (see STS202-3).

The value is right justified and left padded with 0 to make a full 6-digit code.

The placing of a value into this register by a client is an uncontrolled event and does not necessarily reflect the actual value used in the POS database to generate the DecoderKey for the particular payment meter (see 6.5.3 of IEC 62055-41).

7.25 DecoderReferenceNumber register – 4 digit Manufacturer Code (for 4 digit Manufacturer Code, see 7.8)

Table 20 – Composition of the DecoderReferenceNumber register

Attributes	Range	Context
registerName	**DecoderReferenceNumber**	Register name in this table
registerID	2017 hex	Register identifier in this table
format	13 decimal digits	Presentation format to the interface
readWrite	R	Read only
confidentiality	False	The content of this register is not confidential

authentication	False	Client authentication not required
Data Elements DecoderReferenceNumber	 13-digit DRN as defined in 6.1.2.3 of IEC 62055-41	**External data interface to the Register instance** (from the TokenCarrier perspective)
Methods ReadCommand (RID, DL)	 Returns the value of the DRN RID = registerID to be read DL = ignored by server	**External service interface to the Register instance** (from the TokenCarrier perspective) The data may be retrieved from the Register by sending a ReadCommand message to the server with the registerID value in the RID field. The DL field is ignored.
Operation Virtual record	 Keeps virtual record of the value of the DRN	**Internal functionality of the Register** This object merely defines a virtual Register, which provides a logical interface to the actual payment meter register or function.
Association none		**Support services provided by other FunctionObjects** The register does not make use of other specified services for it's functioning.

Where the MfrCode part of the DRN (see 6.1.2.3.2 of IEC 62055-41) is smaller than 1000 and greater than 99, then the value shall be right justified and left padded with a decimal zero to always make four decimal digits. For example: 0101, 0102, - 0999.

The DSN part of the DRN (see 6.1.2.3.3 of IEC 62055-41) shall be right justified and left padded with leading zeros to make up the full 8 digits. For example 00000078.

Thus the DRN shall always be presented as a full 13-digit decimal number.

If this register is read and the server is a 2 digit manufacturer code server (11 digit DRN), then the server will return a NAK when the client tries to read this register with an appropriate code in the ServerStatus register (see 6.6.3 of IEC 62055-52).

7.26 TIDBaseYear register

Table 21 - Composition of the TIDBaseYear register

Attributes	Range	Context
registerName	**TIDBaseYear**	Register name in this table
registerID	2018 hex	Register identifier in this table
format	4 decimal digits	Presentation format to the interface
readWrite	R, W	Read and Write
confidentiality	False	The content of this register is not confidential
authentication	False	Client authentication not required
Data Elements TID Base year	 4 decimal digit string for base dates as defined in 6.3.5.1 of IEC 62055-41	**External data interface to the Register instance** (from the TokenCarrier perspective) The year part of the TID base date and time.
Methods ReadCommand (RID, DL) WriteCommand (RID, D)	 Returns the value of the TIDBaseYear RID = registerID to be read or written DL = ignored by server Loads the 4-digit decimal number into the TIDBaseYear data element as if it were being	External service interface to the Register instance (from the TokenCarrier perspective) The data may be retrieved from the Register by sending a ReadCommand message to the server with the registerID value in the RID field. The DL field is ignored. The data may be loaded into the payment meter

	selected by means of a user interface such as a keypad. RID = registerID of Register to be written D = 4-digit decimal number	via the Register interface by sending a WriteCommand message to the Register with the registerID value in the RID field and the dataset in the D field.
Operation Virtual record	Keeps virtual record of the value of the TIDBaseyear	**Internal functionality of the Register** This object merely defines a virtual Register, which provides a logical interface to the actual payment meter register or function.
Association none		**Support services provided by other FunctionObjects** The register does not make use of other specified services for its functioning.

This register is optional and implementation is subject to agreement between the manufacturer and the utility.

If the function is not implemented then the server will return a NAK when the client tries to read this register with an appropriate code in the ServerStatus register (see 6.6.3 of IEC 62055-52).

The dataset is 4 decimal digits in the following format:
YYYY, where
YYYY is defined as the follows in 6.3.5.1 of IEC 62055-41:

Table 22 - TID Base Years

TIDBaseYear	TID Base date and time represented
1993	01 January 1993, 00:00:00
2014	01 January 2014, 00:00:00
2035	01 January 2035, 00:00:00

The placing of a value into this register by a client is an uncontrolled event and does not necessarily reflect the actual value used in the POS database to generate a Token relying on a TID value for the particular payment meter.

7.27 AvailableElectricityCurrencyCredit register

Table 23 – Composition of the AvailableElectricityCurrencyCredit register

Attributes	Range	Context
registerName	**AvailableElectricityCurrency Credit**	Register name in this table
registerID	2019 hex	Register identifier in this table
format	20-bit binary	Presentation format to the interface
readWrite	R	Read only
confidentiality	False	The content of this register is not confidential
authentication	False	Client authentication not required
Data Elements		**External data interface to the Register instance** (from the TokenCarrier perspective)

AvailableElectricityCurrencyCredit	1 bit sign+5 bit exponent + 14 bit integer as defined in 5.6 of STS202-1 units are in 10^{-5} base currency units	The amount of credit available for consumption as reflected by the accounting function in the payment meter.
Methods ReadCommand (RID, DL)	 Returns the value of the AvailableElectricityCurrencyCredit RID = registerID to be read DL = ignored by server	**External service interface to the Register instance** (from the TokenCarrier perspective) The data may be retrieved from the Register by sending a ReadCommand message to the server with the registerID value in the RID field. The DL field is ignored.
Operation Virtual record	 Keeps virtual record of the value of the AvailableElectricityCurrencyCredit	**Internal functionality of the Register** This object merely defines a virtual Register, which provides a logical interface to the actual payment meter register or function.
Association none		**Support services provided by other FunctionObjects** The register does not make use of other specified services for it's functioning.

The format of the data presentation for sending over the communication link is a 20bit binary number.

The application interpretation of the 20-bit value is:

- Most significant bit = sign bit; (0 = plus ; 1 = minus)
- Next 5 most significant bits = 5 bit exponent, from 10^0 - 10^{31};
- The least 14 significant bits represent 10^{-5} base currency units each, from 2^0 - 2^{13};

If the function is not implemented then the server will return a NAK when the client tries to read this register with an appropriate code in the ServerStatus register (see 6.6.3 of IEC 62055-52).

7.28 AvailableWaterCurrencyCredit register

Table 24 – Composition of the AvailableWaterCurrencyCredit register

Attributes	Range	Context
registerName	**AvailableWaterCurrencyCredit**	Register name in this table
registerID	201A hex	Register identifier in this table
format	20-bit binary	Presentation format to the interface
readWrite	R	Read only
confidentiality	False	The content of this register is not confidential
authentication	False	Client authentication not required
Data Elements AvailableWaterCurrencyCredit	 1 bit sign+5 bit exponent + 14 bit integer as defined in 5.6 of STS202-1 units are in 10^{-5} base currency units	**External data interface to the Register instance** (from the TokenCarrier perspective) The amount of credit available for consumption as reflected by the accounting function in the payment meter.
Methods ReadCommand (RID, DL)	 Returns the value of the AvailableWaterCurrencyCredit RID = registerID to be read DL = ignored by server	**External service interface to the Register instance** (from the TokenCarrier perspective) The data may be retrieved from the Register by sending a ReadCommand message to the server with the registerID value in the RID field. The DL field is ignored.

Operation Virtual record	Keeps virtual record of the value of the AvailableWaterCurrencyCredit	**Internal functionality of the Register** This object merely defines a virtual Register, which provides a logical interface to the actual payment meter register or function.
Association none		**Support services provided by other FunctionObjects** The register does not make use of other specified services for it's functioning.

The format of the data presentation for sending over the communication link is a 20bit binary number.

The application interpretation of the 20-bit value is:

- Most significant bit = sign bit; (0 = plus ; 1 = minus)
- Next 5 most significant bits = 5 bit exponent, from 10^0 - 10^{31};
- The least 14 significant bits represent 10^{-5} base currency units each, from 2^0 - 2^{13};

If the function is not implemented then the server will return a NAK when the client tries to read this register with an appropriate code in the ServerStatus register (see 6.6.3 of IEC 62055-52).

7.29 AvailableGasCurrencyCredit register

Table 25 – Composition of the AvailableGasCurrencyCredit register

Attributes	Range	Context
registerName	**AvailableGasCurrencyCredit**	Register name in this table
registerID	201B hex	Register identifier in this table
format	20-bit binary	Presentation format to the interface
readWrite	R	Read only
confidentiality	False	The content of this register is not confidential
authentication	False	Client authentication not required
Data Elements AvailableGasCurrency Credit	1 bit sign+5 bit exponent + 14 bit integer as defined in 5.6 of STS202-1 units are in 10^{-5} base currency units	**External data interface to the Register instance** (from the TokenCarrier perspective) The amount of credit available for consumption as reflected by the accounting function in the payment meter.
Methods ReadCommand (RID, DL)	Returns the value of the AvailableGasCurrencyCredit RID = registerID to be read DL = ignored by server	**External service interface to the Register instance** (from the TokenCarrier perspective) The data may be retrieved from the Register by sending a ReadCommand message to the server with the registerID value in the RID field. The DL field is ignored.
Operation Virtual record	Keeps virtual record of the value of the AvailableGasCurrencyCredit	**Internal functionality of the Register** This object merely defines a virtual Register, which provides a logical interface to the actual payment meter register or function.
Association none		**Support services provided by other FunctionObjects** The register does not make use of other specified services for it's functioning.

The format of the data presentation for sending over the communication link is a 20bit binary number.

The application interpretation of the 20-bit value is:

- Most significant bit = sign bit; (0 = plus ; 1 = minus)
- Next 5 most significant bits = 5 bit exponent, from 10^0 - 10^{31};
- The least 14 significant bits represent 10^{-5} base currency units each, from 2^0 - 2^{13};

If the function is not implemented then the server will return a NAK when the client tries to read this register with an appropriate code in the ServerStatus register (see 6.6.3 of IEC 62055-52).

7.30 AvailableTimeCurrencyCredit register

Table 26 – Composition of the AvailableTimeCurrencyCredit register

Attributes	Range	Context
registerName	**AvailableTimeCurrencyCredit**	Register name in this table
registerID	201C hex	Register identifier in this table
format	20-bit binary	Presentation format to the interface
readWrite	R	Read only
confidentiality	False	The content of this register is not confidential
authentication	False	Client authentication not required
Data Elements		**External data interface to the Register instance** (from the TokenCarrier perspective)
AvailableTimeCurrencyCredit	1 bit sign+5 bit exponent + 14 bit integer as defined in 5.6 of STS202-1 units are in 10^{-5} base currency units	The amount of credit available for consumption as reflected by the accounting function in the payment meter.
Methods		**External service interface to the Register instance** (from the TokenCarrier perspective)
ReadCommand (RID, DL)	Returns the value of the AvailableTimeCurrencyCredit RID = registerID to be read DL = ignored by server	The data may be retrieved from the Register by sending a ReadCommand message to the server with the registerID value in the RID field. The DL field is ignored.
Operation		**Internal functionality of the Register**
Virtual record	Keeps virtual record of the value of the AvailableTimeCurrencyCredit	This object merely defines a virtual Register, which provides a logical interface to the actual payment meter register or function.
Association		**Support services provided by other FunctionObjects**
none		The register does not make use of other specified services for it's functioning.

The format of the data presentation for sending over the communication link is a 20bit binary number.

The application interpretation of the 20-bit value is:

- Most significant bit = sign bit; (0 = plus ; 1 = minus)
- Next 5 most significant bits = 5 bit exponent, from 10^0 - 10^{31};
- The least 14 significant bits represent 10^{-5} base currency units each, from 2^0 - 2^{13};

If the function is not implemented then the server will return a NAK when the client tries to read this register with an appropriate code in the ServerStatus register (see 6.6.3 of IEC 62055-52).

7.31 AvailableWaterCredit register

Table 27 – Composition of the AvailableWaterCredit register

Attributes	Range	Context
registerName	**AvailableWaterCredit**	Register name in this table
registerID	201D hex	Register identifier in this table
format	32-bit binary	Presentation format to the interface
readWrite	R	Read only
confidentiality	False	The content of this register is not confidential
authentication	False	Client authentication not required
Data Elements		**External data interface to the Register instance** (from the TokenCarrier perspective)
AvailableWaterCredit	1 bit sign+31-bit integer as defined in 3.1.2 of IEC 62055-31 units are in 0.1 kl	The amount of credit available for consumption as reflected by the accounting function in the payment meter.
Methods		**External service interface to the Register instance** (from the TokenCarrier perspective)
ReadCommand (RID, DL)	Returns the value of the AvailableWaterCredit RID = registerID to be read DL = ignored by server	The data may be retrieved from the Register by sending a ReadCommand message to the server with the registerID value in the RID field. The DL field is ignored.
Operation		**Internal functionality of the Register**
Virtual record	Keeps virtual record of the value of the AvailableWaterCredit	This object merely defines a virtual Register, which provides a logical interface to the actual payment meter register or function.
Association		**Support services provided by other FunctionObjects**
none		The register does not make use of other specified services for it's functioning.

The format of the data presentation for sending over the communication link is a 32-bit binary number.

The application interpretation of the 32-bit value is:

- Most significant bit = sign bit; (0 = plus ; 1 = minus)
- Least significant 31 bits = 31-bit integer value;
- The least significant bit in the integer represents 0.1 kl unit;
- Any smaller fractional part of the register value is truncated.

7.32 AvailableGasCredit register

Table 28 – Composition of the AvailableGasCredit register

Attributes	Range	Context
registerName	**AvailableGasCredit**	Register name in this table
registerID	201E hex	Register identifier in this table
format	32-bit binary	Presentation format to the interface
readWrite	R	Read only
confidentiality	False	The content of this register is not confidential
authentication	False	Client authentication not required

Data Elements AvailableGasCredit	 1 bit sign+31-bit integer as defined in 3.1.2 of IEC 62055-31 units are in 0.1m^3	**External data interface to the Register instance** (from the TokenCarrier perspective) The amount of credit available for consumption as reflected by the accounting function in the payment meter.
Methods ReadCommand (RID, DL)	 Returns the value of the AvailableGasCredit RID = registerID to be read DL = ignored by server	**External service interface to the Register instance** (from the TokenCarrier perspective) The data may be retrieved from the Register by sending a ReadCommand message to the server with the registerID value in the RID field. The DL field is ignored.
Operation Virtual record	 Keeps virtual record of the value of the AvailableGasCredit	**Internal functionality of the Register** This object merely defines a virtual Register, which provides a logical interface to the actual payment meter register or function.
Association none		**Support services provided by other FunctionObjects** The register does not make use of other specified services for it's functioning.

The format of the data presentation for sending over the communication link is a 32-bit binary number.

The application interpretation of the 32-bit value is:

- Most significant bit = sign bit; (0 = plus ; 1 = minus)
- Least significant 31 bits = 31-bit integer value;
- The least significant bit in the integer represents 0.1 m^3 unit;
- Any smaller fractional part of the register value is truncated.

7.33 AvailableTimeCredit register

Table 29 – Composition of the AvailableTimeCredit register

Attributes	**Range**	**Context**
registerName	**AvailableTimeCredit**	Register name in this table
registerID	201F hex	Register identifier in this table
format	32-bit binary	Presentation format to the interface
readWrite	R	Read only
confidentiality	False	The content of this register is not confidential
authentication	False	Client authentication not required
Data Elements AvailableTimeCredit	 1 bit sign+31-bit integer as defined in 3.1.2 of IEC 62055-31 units are in 0.1 minutes	**External data interface to the Register instance** (from the TokenCarrier perspective) The amount of credit available for consumption as reflected by the accounting function in the payment meter.
Methods ReadCommand (RID, DL)	 Returns the value of the AvailableTimeCredit RID = registerID to be read DL = ignored by server	**External service interface to the Register instance** (from the TokenCarrier perspective) The data may be retrieved from the Register by sending a ReadCommand message to the server with the registerID value in the RID field. The DL field is ignored.
Operation		**Internal functionality of the Register**

Virtual record	Keeps virtual record of the value of the AvailableTimeCredit	This object merely defines a virtual Register, which provides a logical interface to the actual payment meter register or function.
Association none		**Support services provided by other FunctionObjects** The register does not make use of other specified services for it's functioning.

The format of the data presentation for sending over the communication link is a 32-bit binary number.

The application interpretation of the 32-bit value is:

- Most significant bit = sign bit; (0 = plus ; 1 = minus)
- Least significant 31 bits = 31bit integer value;
- The least significant bit in the integer represents 0.1 minutes unit;
- Any smaller fractional part of the register value is truncated.

7.34 CumulativeWaterConsumption register

Table 30 – Composition of the CumulativeWaterConsumption register

Attributes	Range	Context
registerName	**CumulativeWaterConsumption**	Register name in this table
registerID	2020 hex	Register identifier in this table
format	32-bit binary	Presentation format to the interface
readWrite	R	Read only
confidentiality	False	The content of this register is not confidential
authentication	False	Client authentication not required
Data Elements CumulativeWaterConsumption	 1 bit sign+31-bit integer as defined in 5.11.4 of IEC 62055-31 units are in 100 litres	**External data interface to the Register instance** (from the TokenCarrier perspective) The cumulative total water consumption delivered to the customer as registered over the lifetime of the payment meter.
Methods ReadCommand (RID, DL)	 Returns the value of the CumulativeWaterConsumption RID = registerID to be read DL = ignored by server	**External service interface to the Register instance** (from the TokenCarrier perspective) The data may be retrieved from the Register by sending a ReadCommand message to the server with the registerID value in the RID field. The DL field is ignored.
Operation Virtual record	 Keeps virtual record of the value of the CumulativeWaterConsumption	**Internal functionality of the Register** This object merely defines a virtual Register, which provides a logical interface to the actual payment meter register or function.
Association none		**Support services provided by other FunctionObjects** The register does not make use of other specified services for it's functioning.

The value returned by reading from this register shall be the content of the cumulative water consumption register as defined for electricity in 5.11.4 of IEC 62055-31.

In the case where more than one cumulative water consumption register is implemented in the payment meter, then the value returned from reading the CumulativeWaterConsumption register shall be the cumulative total consumption that is registered over the lifetime of the payment meter.

The format of the data presentation for sending over the communication link is a 32-bit binary number.

The application interpretation of the 32-bit value is:

- Most significant bit = sign bit; (0 = plus ; 1 = minus)
- Least significant 31 bits = 31-bit integer value;
- The least significant bit in the integer represents 0.1 kl unit;
- Any smaller fractional part of the register value is truncated.

7.35 CumulativeGasConsumption register

Table 31 – Composition of the CumulativeGasConsumption register

Attributes	**Range**	**Context**
registerName	**CumulativeGasConsumption**	Register name in this table
registerID	2021 hex	Register identifier in this table
format	32-bit binary	Presentation format to the interface
readWrite	R	Read only
confidentiality	False	The content of this register is not confidential
authentication	False	Client authentication not required
Data Elements		**External data interface to the Register instance** (from the TokenCarrier perspective)
CumulativeGasConsumption	1 bit sign+31-bit integer as defined in 5.11.4 of IEC 62055-31 units are in m^3	The cumulative total gas consumption delivered to the customer as registered over the lifetime of the payment meter.
Methods		**External service interface to the Register instance** (from the TokenCarrier perspective)
ReadCommand (RID, DL)	Returns the value of the CumulativeGasConsumption RID = registerID to be read DL = ignored by server	The data may be retrieved from the Register by sending a ReadCommand message to the server with the registerID value in the RID field. The DL field is ignored.
Operation		**Internal functionality of the Register**
Virtual record	Keeps virtual record of the value of the CumulativeGasConsumption	This object merely defines a virtual Register, which provides a logical interface to the actual payment meter register or function.
Association		**Support services provided by other FunctionObjects**
none		The register does not make use of other specified services for it's functioning.

The value returned by reading from this register shall be the content of the cumulative gas consumption register as defined for electricity in 5.11.4 of IEC 62055-31.

In the case where more than one cumulative gas consumption register is implemented in the payment meter, then the value returned from reading the CumulativeGasConsumption register shall be the cumulative total consumption that is registered over the lifetime of the payment meter.

The format of the data presentation for sending over the communication link is a 32-bit binary number.

The application interpretation of the 32-bit value is:

- Most significant bit = sign bit; (0 = plus ; 1 = minus)
- Least significant 31 bits = 31-bit integer value;
- The least significant bit in the integer represents $0.1m^3$ unit;
- Any smaller fractional part of the register value is truncated.

7.36 CumulativeTimeConsumption register

Table 32 – Composition of the CumulativeTimeConsumption register

Attributes	Range	Context
registerName	**CumulativeTimeConsumption**	Register name in this table
registerID	2022 hex	Register identifier in this table
format	32-bit binary	Presentation format to the interface
readWrite	R	Read only
confidentiality	False	The content of this register is not confidential
authentication	False	Client authentication not required
Data Elements		**External data interface to the Register instance** (from the TokenCarrier perspective)
CumulativeTimeConsumption	1 bit sign+31-bit integer as defined in 5.11.4 of IEC 62055-31 units are in minutes	The cumulative total time consumption delivered to the customer as registered over the lifetime of the payment meter.
Methods		**External service interface to the Register instance** (from the TokenCarrier perspective)
ReadCommand (RID, DL)	Returns the value of the CumulativeTimeConsumption RID = registerID to be read DL = ignored by server	The data may be retrieved from the Register by sending a ReadCommand message to the server with the registerID value in the RID field. The DL field is ignored.
Operation		**Internal functionality of the Register**
Virtual record	Keeps virtual record of the value of the CumulativeTimeConsumption	This object merely defines a virtual Register, which provides a logical interface to the actual payment meter register or function.
Association		**Support services provided by other FunctionObjects**
none		The register does not make use of other specified services for it's functioning.

The value returned by reading from this register shall be the content of the cumulative time consumption register as defined for electricity in 5.11.4 of IEC 62055-31.

In the case where more than one cumulative time consumption register is implemented in the payment meter, then the value returned from reading the CumulativeTimeConsumption register shall be the cumulative total consumption that is registered over the lifetime of the payment meter.

The format of the data presentation for sending over the communication link is a 32-bit binary number.

The application interpretation of the 32-bit value is:

- Most significant bit = sign bit; (0 = plus ; 1 = minus)
- Least significant 31 bits = 31-bit integer value;
- The least significant bit in the integer represents 1 min unit;
- Any smaller fractional part of the register value is truncated.

7.37 CumulativeElectricityCurrencyConsumption register

Table 33 – Composition of the CumulativeElectricityCurrencyConsumption register

Attributes	Range	Context
registerName	**CumulativeElectricityCurrency Consumption**	Register name in this table
registerID	2023 hex	Register identifier in this table
format	20-bit binary	Presentation format to the interface
readWrite	R	Read only
confidentiality	False	The content of this register is not confidential
authentication	False	Client authentication not required
Data Elements		**External data interface to the Register instance** (from the TokenCarrier perspective)
CumulativeElectricity CurrencyConsumption	1 bit sign+5 bit exponent + 14 bit integer as defined in 5.6 of STS202-1 units are in 10^{-5} base currency units	The cumulative total electricity currency consumption delivered to the customer as registered over the lifetime of the payment meter.
Methods		**External service interface to the Register instance** (from the TokenCarrier perspective)
ReadCommand (RID, DL)	Returns the value of the CumulativeElectricityCurrencyConsumption RID = registerID to be read DL = ignored by server	The data may be retrieved from the Register by sending a ReadCommand message to the server with the registerID value in the RID field. The DL field is ignored.
Operation		**Internal functionality of the Register**
Virtual record	Keeps virtual record of the value of the CumulativeElectricityCurrencyConsumption	This object merely defines a virtual Register, which provides a logical interface to the actual payment meter register or function.
Association		**Support services provided by other FunctionObjects**
none		The register does not make use of other specified services for it's functioning.

The format of the data presentation for sending over the communication link is a 20 bit binary number.

The application interpretation of the 20-bit value is:

- Most significant bit = sign bit; (0 = plus ; 1 = minus)
- Next 5 most significant bits = 5 bit exponent, from 10^{0} - 10^{31};
- The least 14 significant bits represent 10^{-5} base currency units each, from 2^{0} - 2^{13};

If the function is not implemented then the server will return a NAK when the client tries to read this register with an appropriate code in the ServerStatus register (see 6.6.3 of IEC 62055-52).

7.38 CumulativeWaterCurrencyConsumption register

Table 34 – Composition of the CumulativeWaterCurrencyConsumption register

Attributes	Range	Context
registerName	**CumulativeWaterCurrencyConsumption**	Register name in this table
registerID	2024 hex	Register identifier in this table
format	20-bit binary	Presentation format to the interface
readWrite	R	Read only
confidentiality	False	The content of this register is not confidential
authentication	False	Client authentication not required
Data Elements		**External data interface to the Register instance** (from the TokenCarrier perspective)
CumulativeWaterCurrencyConsumption	1 bit sign+5 bit exponent + 14 bit integer as defined in 5.6 of STS202-1 units are in 10^{-5} base currency units	The cumulative total water currency consumption delivered to the customer as registered over the lifetime of the payment meter.
Methods		**External service interface to the Register instance** (from the TokenCarrier perspective)
ReadCommand (RID, DL)	Returns the value of the CumulativeWaterCurrencyConsumption RID = registerID to be read DL = ignored by server	The data may be retrieved from the Register by sending a ReadCommand message to the server with the registerID value in the RID field. The DL field is ignored.
Operation		**Internal functionality of the Register**
Virtual record	Keeps virtual record of the value of the CumulativeWaterCurrencyConsumption	This object merely defines a virtual Register, which provides a logical interface to the actual payment meter register or function.
Association		**Support services provided by other FunctionObjects**
none		The register does not make use of other specified services for it's functioning.

The format of the data presentation for sending over the communication link is a 20 bit binary number.

The application interpretation of the 20-bit value is:

- Most significant bit = sign bit; (0 = plus ; 1 = minus)
- Next 5 most significant bits = 5 bit exponent, from 10^0 - 10^{31};
- The least 14 significant bits represent 10^{-5} base currency units each, from 2^0 - 2^{13};

If the function is not implemented then the server will return a NAK when the client tries to read this register with an appropriate code in the ServerStatus register (see 6.6.3 of IEC 62055-52).

7.39 CumulativeGasCurrencyConsumption register

Table 35 – Composition of the CumulativeGasCurrencyConsumption register

Attributes	Range	Context
registerName	**CumulativeGasCurrencyConsumption**	Register name in this table

registerID	2025 hex	Register identifier in this table
format	20-bit binary	Presentation format to the interface
readWrite	R	Read only
confidentiality	False	The content of this register is not confidential
authentication	False	Client authentication not required
Data Elements		**External data interface to the Register instance** (from the TokenCarrier perspective)
CumulativeGasCurrencyConsumption	1 bit sign+5 bit exponent + 14 bit integer as defined in 5.6 of STS202-1 units are in 10^{-5} base currency units	The cumulative total Gas currency consumption delivered to the customer as registered over the lifetime of the payment meter.
Methods		**External service interface to the Register instance** (from the TokenCarrier perspective)
ReadCommand (RID, DL)	Returns the value of the CumulativeGasCurrencyConsumption RID = registerID to be read DL = ignored by server	The data may be retrieved from the Register by sending a ReadCommand message to the server with the registerID value in the RID field. The DL field is ignored.
Operation		**Internal functionality of the Register**
Virtual record	Keeps virtual record of the value of the CumulativeGasCurrencyConsumption	This object merely defines a virtual Register, which provides a logical interface to the actual payment meter register or function.
Association		**Support services provided by other FunctionObjects**
none		The register does not make use of other specified services for it's functioning.

The format of the data presentation for sending over the communication link is a 20 bit binary number.

The application interpretation of the 20-bit value is:

- Most significant bit = sign bit; (0 = plus ; 1 = minus)
- Next 5 most significant bits = 5 bit exponent, from 10^0 - 10^{31};
- The least 14 significant bits represent 10^{-5} base currency units each, from 2^0 - 2^{13};

If the function is not implemented then the server will return a NAK when the client tries to read this register with an appropriate code in the ServerStatus register (see 6.6.3 of IEC 62055-52).

7.40 CumulativeTimeCurrencyConsumption register

Table 36 – Composition of the CumulativeTimeCurrencyConsumption register

Attributes	Range	Context
registerName	**CumulativeTimeCurrencyConsumption**	Register name in this table
registerID	2026 hex	Register identifier in this table
format	20-bit binary	Presentation format to the interface
readWrite	R	Read only
confidentiality	False	The content of this register is not confidential
authentication	False	Client authentication not required

Data Elements		**External data interface to the Register instance** (from the TokenCarrier perspective)
CumulativeTimeCurrencyConsumption	1 bit sign+5 bit exponent + 14 bit integer as defined in 5.6 of STS202-1 units are in 10^{-5} base currency units	The cumulative total time currency consumption delivered to the customer as registered over the lifetime of the payment meter.
Methods		**External service interface to the Register instance** (from the TokenCarrier perspective)
ReadCommand (RID, DL)	Returns the value of the CumulativeTimeCurrencyConsumption RID = registerID to be read DL = ignored by server	The data may be retrieved from the Register by sending a ReadCommand message to the server with the registerID value in the RID field. The DL field is ignored.
Operation		**Internal functionality of the Register**
Virtual record	Keeps virtual record of the value of the CumulativeTimeCurrencyConsumption	This object merely defines a virtual Register, which provides a logical interface to the actual payment meter register or function.
Association		**Support services provided by other FunctionObjects**
none		The register does not make use of other specified services for it's functioning.

The format of the data presentation for sending over the communication link is a 20 bit binary number.

The application interpretation of the 20-bit value is:

- Most significant bit = sign bit; (0 = plus ; 1 = minus)
- Next 5 most significant bits = 5 bit exponent, from 10^0 - 10^{31};
- The least 14 significant bits represent 10^{-5} base currency units each, from 2^0 - 2^{13};

If the function is not implemented then the server will return a NAK when the client tries to read this register with an appropriate code in the ServerStatus register (see 6.6.3 of IEC 62055-52).

7.41 PowerLimitingState register

Table 37 – Composition of the PowerLimitingState register

Attributes	**Range**	**Context**
registerName	**PowerLimitingState**	Register name in this table
registerID	2027 hex	Register identifier in this table
format	16-bit binary	Presentation format to the interface
readWrite	R	Read only
confidentiality	False	The content of this register is not confidential
authentication	False	Client authentication not required
Data Elements		**External data interface to the Register instance** (from the TokenCarrier perspective)
TamperStatus	16-bit status value as encoded in Table 38	
Methods		**External service interface to the Register instance** (from the TokenCarrier perspective)

ReadCommand (RID, DL)	Returns the value of the status of the power limiting state of the payment meter RID = registerID to be read DL = ignored by server	The data may be retrieved from the Register by sending a ReadCommand message to the server with the registerID value in the RID field. The DL field is ignored.
Operation Virtual record	Keeps virtual record of the value of the power limiting state of the payment meter	**Internal functionality of the Register** This object merely defines a virtual Register, which provides a logical interface to the actual payment meter register or function.
Association none		**Support services provided by other FunctionObjects** The register does not make use of other specified services for it's functioning.

If the function is not supported (legacy meters) then the server will return a NAK when the client tries to read this register with an appropriate code in the ServerStatus register (see 6.6.3 of IEC 62055-52).

The encoded value of the PowerLimitingState register is given in Table 38.

Table 38 – Bit-encoding of the PowerLimitingState register value

Value	Interpretation	
Bit 0 = 0	The payment meter is not in a power limiting state	
Bit 0 = 1	The payment meter is in a power limiting state	
Bit 1 -7	Reserved for STS assignment	
Bit 8 – 15	Assigned for proprietary use	

The PowerLimitingState register merely reports the status of these values and is not concerned with the operational functions in the payment meter that control the values. The operation and control of the power limiting functions are defined in other relevant specifications.

7.42 NumberOfKCTSupported register

Table 39 – Composition of the NumberOfKCTSupported register

Attributes	Range	Context
registerName	**NumberOfKCTSupported**	Register name in this table
registerID	2028 hex	Register identifier in this table
format	2 decimal digits	Presentation format to the interface
readWrite	R	Read only
confidentiality	False	The content of this register is not confidential
authentication	False	Client authentication not required
Data Elements NumberOfKCTSupported	2-digit decimal number 02 – 99 Note: only 02, 03, 04 are allocated. The rest are reserved by the STS Association	**External data interface to the Register instance.** The number of key-change tokens the payment meter supports.
Methods		**External service interface to the Register instance** (from the TokenCarrier perspective)

ReadCommand (RID, DL)	Returns the value of the number of Keychange tokens supported RID = registerID to be read DL = ignored by server	The data may be retrieved from the Register by sending a ReadCommand message to the server with the registerID value in the RID field. The DL field is ignored.
Operation Virtual record	Keeps virtual record of the value of the number of Keychange tokens supported	**Internal functionality of the Register** This object merely defines a virtual Register, which provides a logical interface to the actual payment meter register or function.
Association none		**Support services provided by other FunctionObjects** The register does not make use of other specified services for it's functioning.

If the function is not supported (legacy meters) then the server will return a NAK when the client tries to read this register with an appropriate code in the ServerStatus register (see 6.6.3 of IEC 62055-52).

Values less than 10 shall be right justified and left padded with a decimal zero to always make two decimal digits. For example: 02, 03, - 09. Note that values less than 2 are not allowed.

7.43 SetCTSDefault register

Table 40 – Composition of the SetCTSDefault register

Attributes	Range	Context
registerName	**SetCTSDefault**	Register name in this table
registerID	2029 hex	Register identifier in this table
format	2 decimal digits	Presentation format to the interface
readWrite	W	Write only
confidentiality	False	The content of this register is not confidential
authentication	False	Client authentication not required
Data Elements SetCTSDefault or ExitCTSMode	2-digit decimal number 00 - 99 Ex: 01 = UUT01, 03 = UUT03…. 98 = UUT98 99 = exit CTS mode (Note3)	**External data interface to the Register instance.** The CTS UUT number to set the default values to (see Note1, Note2, Note3). Note that a value of 99 causes the payment meter to immediately exit CTS test mode irrespective of any other CTS exit mode implemented in the payment meter.
Methods WriteCommand (RID, D)	Resets the payment meter to its CTS default values for the specified UUT or exits CTS mode. RID = registerID to be written D = data to set the payment meter to a default specified by the two digits contained in D.	**External service interface to the Register instance** (from the TokenCarrier perspective) No data is retrieved from the Register by sending a WriteCommand message to the server with the registerID value in the RID field. The DL field is ignored.
Operation Virtual record	Not applicable	**Internal functionality of the Register** This object merely defines a virtual Register, which provides a logical interface to the actual payment meter register or function.
Association none		**Support services provided by other FunctionObjects** The register does not make use of other specified services for it's functioning.

Note1: see STS531-9-1-07 and STS531-9-1-11 Table1 for the default values for the payment meter. If the function is not implemented then the server will return a NAK when the client tries to write to this register with an appropriate code in the ServerStatus register (see 6.6.3 of IEC 62055-52).

Note2: This command is only effective in setting the default states if the payment meter conforms to the requirements specified in STS203-1 for entering CTS test mode, otherwise this write command shall have no effect on the state of the payment meter.

Note3: Value 99 is reserved for immediate termination of the CTS mode for the payment meter irrespective of any other exit mode implemented (see STS203-1).

7.44 FlagSettings register

Table 41 – Composition of the FlagsConfigurationSettings register

Attributes	Range	Context
registerName	**FlagSettings**	Register name in this table
registerID	202A hex	Register identifier in this table
format	512 Bytes (ASCII)	Presentation format to the interface
readWrite	R	Read only
confidentiality	False	The content of this register is not confidential
authentication	False	Client authentication not required
Data Elements		**External data interface to the Register instance.**
Configured function	512 byte ASCII representation of the binary value of each flag. See Note 1. '0' = function OFF/Disabled '1' = function ON/Enabled '-' = function not supported	Each byte in the 512 byte value represents the state of the corresponding bit value as specified in STS202-5 4.2and 4.4, with the rightmost bit representing index 0 of the bit array.
Methods		**External service interface to the Register instance** (from the TokenCarrier perspective)
ReadCommand (RID, DL)	Returns the value of the configuration of the payment meter RID = registerID to be read DL = ignored by server	The data may be retrieved from the Register by sending a ReadCommand message to the server with the registerID value in the RID field. The DL field is ignored.
Operation		**Internal functionality of the Register**
Virtual record	Not applicable	This object merely defines a virtual Register, which provides a logical interface to the actual payment meter register or function.
Association		**Support services provided by other FunctionObjects**
none		The register does not make use of other specified services for it's functioning.

If the function is not implemented then the server will return a NAK when the client tries to read this register with an appropriate code in the ServerStatus register (see 6.6.3 of IEC 62055-52).

Note1: It is not necessary to send back all 512 bytes, but only the bit settings supported as long as all the supported and unsupported bits are sent back starting from bit 0 and ending at the last supported bit.

For example:

If bits 3,5,6,7, and 12 are supported and have a value '1', then send back "1----111-1---"

7.45 ControlElementSettings register

Table 42 – Composition of the ControlElementSettings register

Attributes	Range	Context
registerName	**ControlElementSettings**	Register name in this table
registerID	202B hex	Register identifier in this table
format	10 Bits	Presentation format to the interface
readWrite	R	Read only
confidentiality	False	The content of this register is not confidential
authentication	False	Client authentication not required
Data Elements		**External data interface to the Register instance.**
Configured function	10 bit representation of the value of the control element	10 bit value representing the value of the control element as specified in STS202-5 4.5
Methods		**External service interface to the Register instance** (from the TokenCarrier perspective)
ReadCommand (RID, AI,DL)	Returns the value of the configuration of the payment meter RID = registerID to be read AI = array index (0x0-0x3F) DL = ignored by server	The data may be retrieved from the Register by sending a ReadCommand message to the server with the registerID value in the RID field and the array index as specified in STS202-5 4.3 and 4.5. The DL field is ignored.
Operation		**Internal functionality of the Register**
Virtual record	Not applicable	This object merely defines a virtual Register, which provides a logical interface to the actual payment meter register or function.
Association		**Support services provided by other FunctionObjects**
none		The register does not make use of other specified services for it's functioning.

If the function is not implemented then the server will return a NAK when the client tries to read this register with an appropriate code in the ServerStatus register (see 6.6.3 of IEC 62055-52).

7.46 STS reserved registers at registerID values 202C – 8FFF hex

The STS Association reserves registerID values 202C – 8FFF hex for future assignment to this RegisterTable.

7.47 Proprietary registers at registerID values 9000 – FFFD hex

Manufacturers may assign the registerID values 9000 – FFFD hex to proprietary register implementations.

The standard does not prevent collision between register identifiers assigned by different manufacturers, and each manufacturer should thus ensure that the WriteCommand messages used are authenticated against a unique identifier such as the MfrCode (see 6.1.2.3.2 of IEC 62055-41).

7.48 TokenStatus register

This shall be a predefined register as given in 6.8.3.7 of IEC 62055-52 with an assigned registerID = FFFE hex.

7.49 NumericTokenEntry register

Table 43 – Composition of the NumericTokenEntry register

Attributes	**Range**	**Context**
registerName	**NumericTokenEntry**	Register name in this table
registerID	FFFF hex	Register identifier in this table
format	20 decimal digits; 0 – 9 Defined in 5.3 of IEC 62055-51	Presentation format to the interface
readWrite	W	Write only
confidentiality	False	The content of this register is not confidential
authentication	False	Client authentication not required
Data Elements		**External data interface to the Register instance** (from the TokenCarrier perspective)
NumericTokenEntry	20-digit decimal number Represents a numeric token carrier	Actual data that is passed across the Register interface and that a client may “enter” by means of a WriteCommand
Methods		**External service interface to the Register instance** (from the TokenCarrier perspective)
WriteCommand (RID, D)	Loads the 20-digit decimal number into the NumericTokenEntry data element as if it were being entered by means of a user interface such as a keypad. RID = registerID of Register to be written D = 20-digit decimal number	The data may be loaded into the payment meter via the Register interface by sending a WriteCommand message to the Register with the registerID value in the RID field and the dataset in the D field
Operation		**Internal functionality of the Register**
Virtual token entry	Provides a virtual point of entry for a token to the payment meter as another token carrier interface	This object defines a virtual Register, which provides a logical interface to the actual payment meter register or function.
Association		**Support services provided by other FunctionObjects**
none		The register does not make use of other specified services for it’s functioning.

This Register serves to provide a means of loading a token into the payment meter via the virtual token carrier.

The client may load a token into the payment meter by sending a WriteCommand message (see 6.4.5 of IEC 62055-52) with the “20-digit numeric token” in the D field and the registerID value for the NumericTokenEntry register in the RID field of the message

The format of the “20-digit numeric token” is as defined for a numeric token carrier in 5.3 of IEC 62055-51.

The received 20 digits obtained from the included data are converted to a 66-bit binary number in accordance with 6.3.2 of IEC 62055-51 and then transferred to the TokenData field of the TCDU, which is further processed by the Application Layer Protocol (see 7.2 of IEC 62055-41) and also by the MeterApplicationProcess (see 8 of IEC 62055-41), where it is executed. The result is then returned in the TokenResult field of the TCDU (see also 6.8.3.7 of IEC 62055-52) for the reading of the result from the TokenStatus register).

At the same time when the 66-bit binary number is transferred to the TCDU, the TokenStatusNotReady code (see Table 24 of IEC 62055-52) shall also be set in the TokenStatus register.

Bibliography

IEC 60050-300, International Electrotechnical Vocabulary – Electrical and electronic measurements and measuring instruments

IEC 62051:1999, – Electricity metering – Glossary of terms

IEC 62055-31:2005, – Electricity metering – Payment systems Part 31: Particular requirements – Static payment meters for active energy (classes 1 and 2)

IEC 62055-51:2007, Electricity metering – Payment systems Part 51: Standard Transfer Specification – Physical Layer Protocol for one-way numeric and magnetic card token carriers

IEC 62055-52, Electricity metering – Payment systems Part 52: Standard Transfer Specification – Physical Layer Protocol for a two-way virtual token carrier for direct local connection