STS Association

STS200-1

Edition 1.2
Jan 2022

COMPANION SPECIFICATION -

Generic classes for meter function objects

CONTENTS

Revision History:

Revision	**Clause**	**Date**	**Change details from previous Edition**
1.1	Normative References	April 2015	Added reference to IEC62055-21 in Normative Refernces
1.2		Jan 2022	Changed to new STS Logo Updated foreword

STANDARD TRANSFER SPECIFICATION ASSOCIATION

_____

COMPANION SPECIFICATION

STS 200-1: Standard transfer specification (STS) – Companion specification – Generic classes for meter function objects

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 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 de facto worldwide standard for transfer of electricity prepayment tokens since its initial introduction in South Africa in 1993.

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 11 789 1384
Fax: +27 11 789 1385
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 metering and payment systems. STS is not limited to this application (see the section: Future enhancements), but because of the wide interest in this area, this document describes the electricity application.

STS is a South African industry specification described in *NRS 009-6, section 6 to 9, and part 7 which are being considered for publication as publicly available standards through IEC TC 13 working group 15 †

The STS series of companion Specifications have been introduced to formalise additional functional 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.

COMPANION SPECIFICATION

STS 200-1: Standard transfer specification (STS) – Companion specification – Generic classes for meter function objects

1 Scope

The Specification introduces, describes and defines the concept of Meter_Function_Objects, which provide for specifying the functionality of the meter without imposing design-specific or technology-specific constraints on the actual meter implementation.
A MeterFunctionObject (MFO) is an object-oriented specification that encapsulates a certain functionality of a payment meter. Each MFO is defined in a companion specification and allocated a unique FunctionObjectIdentificationNumber (FOIN).

2 Normative references

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 62055-41:2007, Electricity metering – Payment systems Part 41: Standard Transfer Specification – Application Layer Protocol for one-way token carrier systems

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

IEC 62055-21, Electricity Metering - Payment systems Part 21: Framework for standardisation

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, 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 the STS Companion Specifications

3DES	3rd Data Encryption Standard
ACK	Acknowledge
ALFO	Application_Layer_Function_Object
APDU	Application Protocol Data Unit
CRFO	Configuration_Registry_Function_Object
DES	Data Encryption Standard

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
MFO	MeterFunctionObject
NAK	Negative Acknowledge
PLFO	Physical_Layer_Function_Object
RID	RegisterIdentifier
Rx	Receiver
SG	Supply Group
SGC	SupplyGroupCode
SC	Social Credit
STA,	Standard Transfer Algorithm
STS	Standard Transfer Specification
STSA	Standard Transfer Specification Association
TCDU	TokenCarrierDataUnit
TI	Tarrif Index
TID	TokenIdentifier
Tx	Transmitter
USB	Universal Serial Bus
VFMO	Virtual_Meter_Function_Object
V_{IN}	Voltage IN
V_{OH}	Voltage OUT High
V_{OL}	Voltage OUT Low
V_{OUT}	Voltage OUT

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

The reference model to this specification is described in IEC 62055-52 Clause 4 and IEC62055-41 Clause 5

5 Generic Meter_Function_Objects

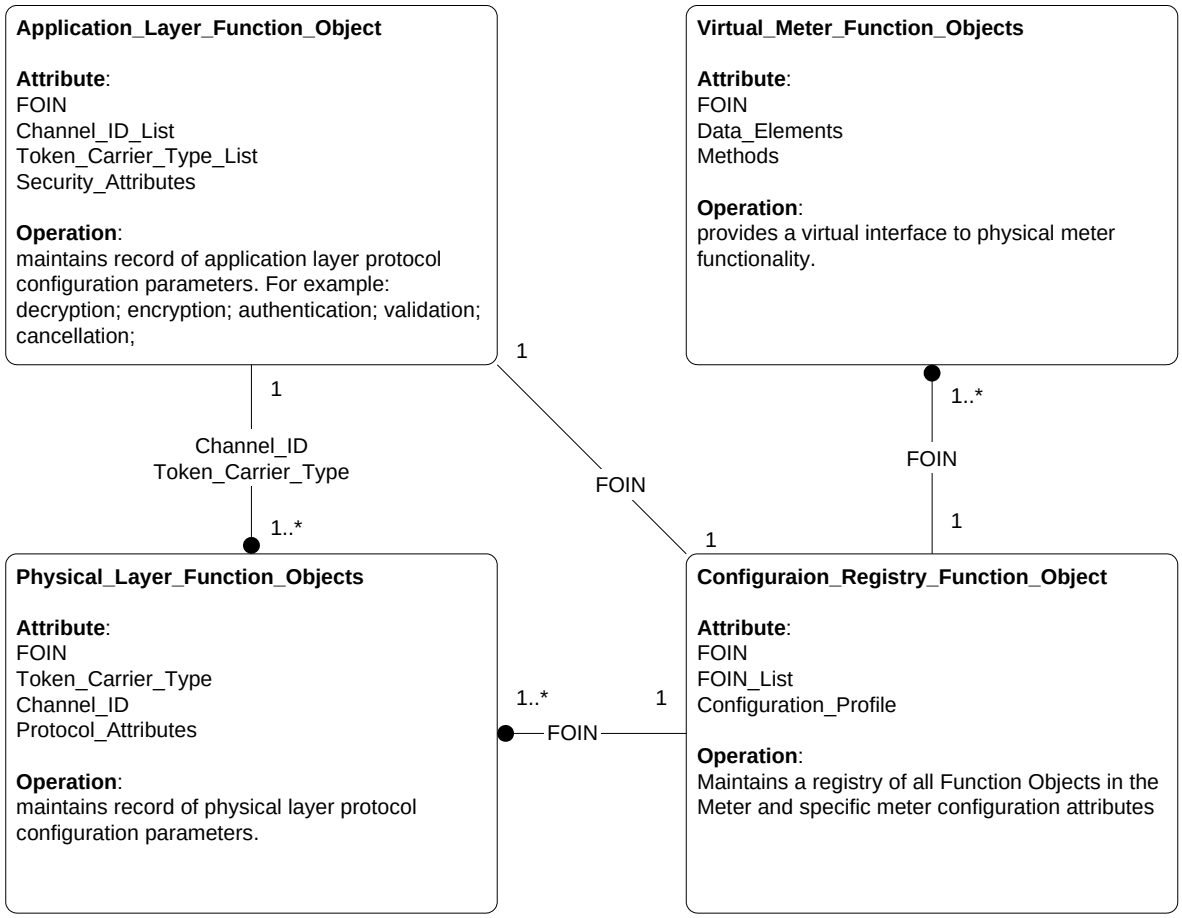

Figure 1 – Generic Meter_Function_Object reference model

Note 2- Only the key attributes are shown in Figure 1. See clause 5.1 to 5.4 for more detailed information on the attributes of each Function_Object.

While the STS describes how data is transferred between a POS and a meter in a standard way, it is also necessary to specify how the meter will behave in response to the receipt of such data or tokens. This Meter_Function_Object reference model introduces the concept of Function_Objects, which provides for specifying the functionality of the meter without imposing design-specific or technology-specific constraints on the actual meter implementation.

The Meter_Application_Process provides the essential underlying functionality of the Function_Object. The Function_Object thus in effect provides an abstracted interface between this underlying functionality in the Meter_Application_Process and the Meter_Application_Process utilising the STS as the means for data exchange and the Token as the actual data interface in the APDU. (Application Protocol Data Unit)

The relation between the Physical_Layer_Function_Object and the Application_Layer_Function_Object is uniquely keyed on the Channel_ID and Token_Carrier_Type data elements. The management of Channel_ID is entirely under the control of the Meter_Application_Process.

The relation between the Configuration_Registry_Function_Object and any other Function_Object is uniquely keyed on the FOIN (see 5.5).

Only one instance of a FOIN may be present in a meter at any one time.

There are essentially 4 Generic Function_Object types. While only their generic structures are defined in this International Standard, the specific detailed specifications for each instance are defined in a

Companion Specification, to each of which is allocated a globally unique FOIN. The following 4 Function_Object generic types are defined:

1) A meter will have **1** *Configuration_Registry_Function_Object* that contains a registered list of all Function_Objects as well as specific configuration information for the particular meter. A meter is thus effectively able to "report" on what it is capable of doing, either onto a token carrier for transfer to a management system or onto a display device of the meter to the user.
2) A meter will normally have **1 or more** *Virtual_Meter_Function_Objects*, which give the meter its particular flavour of functionality. Examples are: Token Credit, Emergency Credit, Lifeline Credit, Tariff Charges, Auxiliary Charges, Metering, Load Switch, Real-time Clock, Back-up Battery, etc.
3) A meter will have **1** *Application_Layer_Function_Object* that contains the variable configuration data for the Application Layer security functions and also a registered list of physical layer token carrier interfaces that the particular meter is configured to support. A meter is thus effectively able to "report" on which token carrier interfaces it supports, either onto a token carrier for transfer to a management system or onto a display device of the meter to the user. A meter is thus able to manage and support several token carrier interface channels concurrently.
4) A meter will have **1 or more** *Physical_Layer_Function_Objects,* one for each actual physical token carrier interface that the meter provides. Each Function_Object contains the variable protocol configuration data for the particular interface.

5.1 VMFO : Virtual_Meter_Function_Object

The generic structure for the Virtual_Meter_Function_Object is given in the table below.

Table 1 – Generic structure for the Virtual_Meter_Function_Object

Attributes	Range	Context
Name	variable	**{Virtual_Meter_Function_Object_Name}** The registered name of this Function_Object in the Companion Specification.
FOIN	22 bit binary	FunctionObjectIdentificationNumber as registered in the Companion Specification.
Data_Elements	x	These are particular data elements for the specific Function_Object, which are exchanged over the Function_Object data interface, and will vary from one Function_Object to the next.
Methods	x	The specific methods that may be applied at the Function_Object interface in order to invoke the specified functions or services of the specific object. These methods are in effect the various messages or tokens that are transferred from the POS to the meter on the Token_Carrier. They are always described from a token perspective (i.e. tokens that invoke actions to be performed in the Virtual_Meter_Function_Object or tokens that are generated by the Virtual_Meter_Function_Object in response to a previously given request via a token.
Operation	x	Description of what the specific Virtual_Meter_Function_Object does or the services that it provides across the interface.
Association	x	Associated support functions it calls on in order to perform its basic function. These are typically Display, Recording and Data_Exchange functions.

An example Meter_Function_Object is given in Annex A.

5.2 CRFO : Configuration_Registry_Function_Object

The generic structure for the Configuration_Registry_Function_Object is given in the table below.

Table 2 – Generic structure for the Configuration_Registry_Function_Object

Attributes	Range	Context
Name	variable	{Configuration_Registry_Function_Object_ Name} The registered name of this Function_Object in the Companion Specification.
FOIN	22 bit binary	FunctionObjectIdentificationNumber as registered in the Companion Specification.
Data Elements		External data interface to the Function_Object instance (from the token perspective)
Function_Object_List	x	List of all the FOI Numbers that are present in the meter, effectively the list of Function_Objects in the meter including this one.
Configuration_Profile	x	Attributes defining the specific configuration of the meter. These are defined in the Companion Specification. Examples are: date of commissioning; firmware version; hardware version; meter type; Decoder_Reference_Number; Meter_Manufacturer; meter operational status; meter technical status; number of power up resets; meter operational life.
Methods		External service interface to the Function_Object instance (from the token perspective)
get_Function_Object_List()	x	Writes the list of FOI Numbers onto the Token_Carrier upon receiving the instruction via an appropriate token.
get_Configuration_Profile()	x	Writes the attribute values from the Configuration_Profile onto the Token_Carrier upon receiving the instruction via an appropriate token.
set_Configuration_Profile()	x	Writes the attribute values from the token as received from the Token_Carrier into the Configuration_Profile upon receiving the instruction via an appropriate token.
display_Function_Object_List	x	Displays the list of FOI Numbers plus associated information on the user interface upon receiving the instruction via an appropriate token.
display_Configuration_Profile	x	Displays the attribute values from the Configuration_Profile in appropriate format on the user interface upon receiving the instruction via an appropriate token.
Operation		Internal functionality of the Object Registry
Function_Object registration	x	This Function_Object maintains a register of FOI Numbers for all the Function_Objects that are present in the meter.
Configuration_Profile registration	x	This Function_Object maintains a register of attributes that define the configuration of and that control the operation of the meter.
Association		Support services provided by other Function_Objects
Display	x	Support service to display values on the user display interface.
Recording	x	Support service to write values to memory registers
Data_Exchange	x	Support service to write values onto a token carrier.

Note 3- Setting of the Function_Object_List values is performed by the Meter_Manufacturer and is thus left up to him to implement.

5.3 ALFO : Application_Layer_Function_Object

The generic structure for the Application_Layer_Function_Object in the Token_Carrier_Interface is given in the table below.

Table 3 – Generic structure for the Application_Layer_Function_Object

Attributes	Range	Context
Name	variable	{Application_Layer_Function_Object_Name} The registered name of this Function_Object in the Companion Specification.
FOIN	22 bit binary	FunctionObjectIdentificationNumber as assigned in the Companion Specification.
Data Elements		External data interface to the Function_Object instance (from the token perspective)
Channel_ID_List	x	A list of channel numbers, each of which is implicitly associated with each of the particular Token_Carrier_Types as listed in the Token_Carrier_Type_List.
Token_Carrier_Type_List	x	A list of Token_Carrier_Types, each of which is implicitly associated with one of the channels, each channel being associated with one particular Physical_Layer_Function_Object. The items in The Token_Carrier_Type_List implicitly map to the list of items in the Channel_ID_List, which together make up a unique foreign key of association to the Physical_Layer_Function_Objects.
Security_Attributes	x	A suite of attributes that determines the particular setup of the security functions in the Application Layer protocol. Examples are: Encryption_Algorithm (STA, DES, 3DES,etc); Decoder_Key; Key_Type; Key_Revision_Number; Supply_Group_Code; Tariff_Index; Key Change Count; Rollover_Key_Change Count; Credit Token Count; Recent Credit Token; Token ID Register; Recent Token ID; Youngest Token ID; Oldest Token ID.
Methods		External interface to the Function_Object instance (from the token perspective)
get_Channel_List	x	Writes the list of channel numbers mapped against the list of Token_Carrier_Types, mapped against the FOIN of each channel as supported in the meter onto the Token_Carrier upon receiving the request via an appropriate token. (see Note 4) and 5).
get_Security_Attributes()	x	Writes the particular requested attribute values onto the Token_Carrier as defined in the detailed Companion Specification.
set_Security_Attributes()	x	Writes the attribute values from the data in the Token as received from the Token_Carrier into the particular Security_Attributes upon receiving the instruction via an appropriate token as defined in the detailed Companion Specification.
display_Channel_List	x	Display the list of channel numbers, mapped against the list of Token_Carrier_Types as supported in the meter onto the user interface, the format of which is defined in the Companion Specification.
display_Security_Attributes	x	Various values from the Security_Attributes may be displayed on the meter user interface, the detail of which is defined in the Companion Specification.
Operation		Internal functionality of the Function_Object instance
Application Layer Protocol parameter configuration	x	Maintains a record of the Application Layer Protocol configuration parameters (See also examples in *Security_Attributes* above).
Associated Functions		Other associated functions and services provided by other Function_Objects (from the token perspective).
Display	x	Support service to display values on the user display interface.
Recording	x	Support service to write values to memory registers
Data_Exchange	x	Support service to write values onto a token carrier.

Note-

4) The information is returned back to the requesting channel if it supports two-way token carrier functionality.

5) The setting of the Channel_ID_List and the Token_Carrier_Type_List is left to the Meter_Manufacturer to implement.

5.4 PLFO : Physical_Layer_Function_Object

The generic structure for the Physical_Layer_Function_Object in the Token_Carrier_Interface is given in the table below.

Table 4 – Generic structure for the Physical_Layer_Function_Object

Attributes	Range	Context
Name	variable	{Physical_Layer_Function_Object_Name} The registered name of this Function_Object in the Companion Specification.
FOIN	22 bit binary	FunctionObjectIdentificationNumber as assigned in the Companion Specification (see also Note 6).
Data Elements		External data interface to the Function_Object instance (from the token perspective)
Token_Carrier_Type	x	Token carriers are either physically transportable token carrier devices or virtual by nature. Examples of physical token carriers are: magnetic cards; barcodes; numeric strings; smart cards; memory keys and other memory devices. Examples of virtual token carriers are: modem devices coupled to PSTN, Radio, ISDN, GSM, GPRS, PLC, LAN, WAN and other networks.
Channel_ID	x	The particular channel number as assigned by the manufacturer of the meter and recorded in the Application_Layer_Function_Object. The channel number shall always start from 1 and logically increment by one in any given meter implementation. This data entity is essentially managed inside the meter between the Physical_Layer_Function_Objects and the Application_Layer_Function_Object.
Protocol_Attributes	x	A suite of attributes that determines the particular setup of the communication functions in the Physical Layer communication protocol, mainly in the case of a virtual token carrier. Examples are: Baud rate, error correction and compression algorithms, etc.
Methods		External interface to the Function_Object instance (from the token perspective)
get_Protocol_Attributes()	x	Writes the particular requested attribute values onto the Token_Carrier as defined in the detailed Companion Specification.
set_Protocol_Attributes()	x	Writes the attribute values from the data in the Token as received from the Token_Carrier into the particular Protocol_Attributes upon receiving the instruction via an appropriate token as defined in the detailed Companion Specification.
display_Channel	x	Display the channel number and the Token_Carrier_Type and the FOIN of this channel onto the user interface, the format of which is defined in the Companion Specification.
display_Protocol_Attributes	x	Various values from this Protocol_Attributes may be displayed on the meter user interface, the detail of which is defined in the Companion Specification.
Operation		Internal functionality of the Function_Object instance
Physical Layer Protocol parameter configuration	x	Maintains a record of the Physical Layer Protocol configuration parameters (see also examples in *Protocol_Attributes* above).
Associated Functions		Other associated functions and services provided by other Function_Objects (from the token perspective).

Display	x	Support service to display values on the user display interface.
Recording	x	Support service to write values to memory registers
Data_Exchange	x	Support service to write values onto a token carrier.

Note 6- The FOIN and the Token_Carrier_Type uniquely keys the Function_Object. It is the task of the Companion Specification to ensure that this is the case.

5.5 FOIN : FunctionObjectIdentificationNumber

The STS Association allocates FunctionObjectIdentificationNumber values to uniquely identify FunctionObject entities in order to ensure interoperability of STS-compliant equipment. This includes the management and publication of Companion Specifications.

The FunctionObjectIdentificationNumber is a 22-bit binary number that shall be used to uniquely identify FunctionObject entities, which are defined in Companion Specifications.

The FOIN comprises a concatenation of 3 codes.

Table 5 – Data elements in the FOIN

FunctionClass	**DefinitionID**	**DefinitionVersion**
5-bit binary	12-bit binary	5-bit binary
0-31	0-4095	0-31

The FOIN presentation format is : FunctionClass**.** DefinitionID**.** DefinitionVersion

For example FOIN = 9**.**5**.**3 means:

- This object belongs to the Accounting function class (see Table 6);
- This is the 5th object instance defined in the Accounting class;
- This is the 3rd version of this object definition.

The FunctionClass is a 5-bit binary code with values defined as follows:

Table 6 – Coding of the FunctionClass

Function_Class	The FunctionClass that this FunctionObject is associated with in accordance with IEC 62055-21
0	Reserved
1	Supplier
2	Customer
3	Contract
4	Generation
5	Transmission
6	Distribution
7	Metering
8	Delivery
9	Accounting
10	Receipting
11	Settlement

12	Time
13	Test
14	Display
15	Recording
16	DataExchange
17	Security
18-31	Reserved for future assignment

The DefinitionID is a 12-bit binary code with values defined as follows:

Table 7 – Coding of the DefinitionID

DefinitionID	The sequential ID number of the object definition as allocated in the Companion Specification.
0	Reserved for future assignment
1-4095	Reserved for assignment by the STS Association

The DefinitionVersion is a 5-bit binary code with values defined as follows:

Table 8 – Coding of the DefinitionVersion

DefinitionVersion	The sequential Version number of the object definition as allocated in the Companion Specification.
0-31	Reserved for assignment by the STS Association

Annex A
(informative)
Example of a Companion Specification for a Social_Credit_Function_Object

A.1 Scope

This informative example serves to illustrate an implementation of the Meter_Function_Object concept as defined in clause 5.1 Virtual_Meter_Function_Objects.

This Social_Credit_Function_Object provides a controlled mechanism for dispensing free service under certain conditions. It is typically used in indigent populations where consumers are entitled to a small amount of electrical energy per month, generally granted by the government. See clause A.5 for more detail of the operation of this function.

The service is generic and this Social_Credit_Function_Object can thus be implemented in a payment meter for any Utility service such as electricity, gas or water.

It is intended for this Social_Credit_Function_Object to operate under the control of the meter accounting process.

This Social_Credit_Function_Object requires advanced functionality of the STS, such as two-way token carriers and wider variety of tokens for example. These are still to be defined in a future enhancement to the standard.

Note:

7 This specification does not prescribe how the function is to be implemented or how it obtains the information other than that provided at the Methods interface.

8 This specification is not an Object Oriented Design, but rather an Object Oriented Model for defining the functionality requirements of the meter application.

9 The Methods describe the external interface purely from the Token perspective and does not concern itself with other methods that may exist from the application perspective.

10 Where registers carry values measured in service units, the value range and unit of measure are normalized to, and is the same as, that of the 16-bit STS transfer_Credit token Amount for the particular service type.

A.2 Social_Credit_Function_Object

Table A 1 – Definition of the Social_Credit_Function_Object

Attributes	Bits	Value	Ref
Name		Social_Credit (SC)	
FOIN	22	09.001.01	A.3.1
SC_Profile_Revision	16	4 hex digits	A.3.2
Priority	4	Binary; range 1 to 15;	A.3.3
Availability	8	Bit position control per functionality	A.3.4
Accounting_Mode	1	Units-Based; constant	A.3.5
SC_Accounting_Register	24	Integer	A.3.6
SC_Service_Delivery_Register	16	16-bit STS	A.3.7
Date_Of_Last_Read	32	8 hex digits: ddmmyyyy	A.3.8
Activation_Criteria	1	Auto, Manual	A.3.9
SC_Quantity	16	16-bit STS	A.3.10
SC_Cycle	4	Binary; 0 to 15 months	A.3.11
Release_Quantity	16	16-bit STS	A.3.12

Release_Cycle	4	Binary; hours, daily, weekly, monthly	A.3.13
Available_Service_Level	8	Binary; 0-100%	A.3.14
Low_Credit_Warning_Level	16	16-bit STS	A.3.15
Low_Credit_Warning	1	True, False	A.3.16
Credit_Expiry_Level	16	Integer	A.3.17
Credit_Expired	1	True, False	A.3.18
Interrupt_Service_On_Credit_Expiry	1	True, False	A.3.19
Restore_Service_On_Credit_Replenishment	1	Auto, Manual	A.3.20
Methods			A.4
set_SC_Profile(Profile_Data)		Token	A.4.1
get_SC_Profile()		Token	A.4.2
display_SC_Profile(Attribute_Name)		Token	A.4.3
clear_SC_Service_Delivery_Register()		Token	A.4.4
get_SC_Service_Delivery_Register()		Token	A.4.5
display_SC_Service_Delivery_Register()		Token	A.4.6
clear_SC_Accounting_Register()		Token	A.4.7
display_SC_Accounting_Register()		Token	A.4.8
decrement_SC_Accounting_Register()		Token	A.4.9
Operation			A.5
Controlled release of credit for free service		Time schedule	
Control status		Available service level; Low credit warning; Credit expiry; Interrupt service; Restore service;	
Associations			A.6
Display function		Support service to display values on the user display interface.	A.6.1
Recording function		Support service to read/write values from/to memory registers	A.6.2
Data_Exchange function		Support service to read/write values from/onto a token carrier.	A.6.3
Security function		Support service to assure the secure functioning of the Function_Object	A.6.4
Time function		Support service to provide date and time information to the Function_Object	A.6.5
Test function		Support service to test for the correct operation of the Function_Object	A.6.6

A.3 Attributes

A.3.1 FOIN

The FunctionObjectIdentificationNumber is composed as follows.

Table A 2 – Composition of the FOIN

Attributes	Value	Format	Context	Ref
Function_Class	09	5 binary bits	Accounting function class	5.5
Definition_ID	001	10 binary bits	Specific object identifier in this Function_Class	5.5
Definition_Version	01	7 binary bits	First instance of this object	5.5

A.3.2 SC_Profile_Revision

The SC_Profile_Revision is a sequential number that keeps track of changes to the attribute values and is under the control of the management system. A particular Utility will configure his SC attributes in different ways and with different values over time. This number is entirely under the management control of the Utility.

Range: 0000 to 9999, each decimal digit represented by hexadecimal digits 0h to 9h

A.3.3 Priority

The SC_Function_Object may be used in combination with other Credit_Function_Objects, in which case it needs to be assigned a particular priority for activation in an ordered structure. The meter accounting process will attempt to extract credit from the highest priority Credit_Function_Object and when such credit depleted, it will revert to the next lower priority Credit_Function_Object, working its way down the priority chain.

Range: 1 to 15

A.3.4 Availability

This attribute controls the availability of the Function_Object.

A Function_Object may be present, but may be programmed to be available or unavailable under certain conditions.

Control of the availability of this Function_Object is by bit position. If the bit is set to logical 1, the function is available.

Table A 3 – Control bits for the Availability attribute

Bit	Availability
0	Reserved
1	Never available
2	Available while in tamper state
3	Available when tariff expired
4	Available when commissioning period has expired
5	Available when feedback latency period has expired
6-7	Reserved

A.3.5 Accounting_Mode

Social_Credit is accounted for in service units. This is a constant set to logical 0.

Table A 4 – Control values for Accounting_Mode attribute

Bit value	Context
0	Units-based
1	Currency-based

A.3.6 SC_Accounting_Register

This register maintains a balance of the difference between quantities of released SC and quantities of delivered service per SC_Cycle. It is cleared of any un-used SC at the end of each SC_Cycle.

Available_Social_Credit is thus the accumulated sum of unused SC still available within the current SC_Cycle

The value of this register represents Available_Social_Credit as follows:

Table A 5 – Format of the SC_Accounting_Register

Bit number	Context	Value
23	Sign bit	
0 to 22	Mantissa	Range: 0 to ± 83886.07 service units

A.3.7 SC_Service_Delivery_Register

Registration of actual service units delivered while the social credit function is active. The register accumulates until it is transferred to a token carrier for transportation to the management system, at which point it is cleared. See also A.4.4 and A.4.5

This register duplicates the function of the equivalent register that would be present in the Tariff_&_Auxiliary_Charge function. See also A.5.

The value range and unit of measure is the same as that of the 16-bit STS transfer_Credit token Amount for the particular service type.

A.3.8 Date_Of_Last_Read

Date, at which the SC_Service_Delivery_Register value was last retrieved and transferred onto a token carrier for transportation to the management system. See also A.4.4 and A.4.5

Format: 8 hex digits; ddmmyyyy

A.3.9 Activation_Criteria

If set to Auto, then the Meter Accounting Process shall automatically activate this function when conditions dictate.

When set to Manual, user intervention shall be required each time, with the option to activate this function or not.

Table A 6 – Control values for Activation_Criteria attribute

Bit value	Context
0	Auto
1	Manual

A.3.10 SC_Quantity

The quantity of Social_Credit that the consumer is entitled to per SC_Cycle. (e.g. 60 kWh per month).

The value range and unit of measure is the same as that of the 16-bit STS transfer_Credit token Amount for the particular service type.

A.3.11 SC_Cycle

The period for which the entitlement of SC is valid, and at the end of which any unused accumulated SC is cleared from the SC_Accounting_Register. Typically the SC_Cycle is 1 month.

The month starts on 00h00m00s on the 01mmyyyy every month of every year

Table A 7 – Values for the SC_Cycle attribute

Value	Frequency	Context
0	none	Never happens
1 to 15	months	Every x months (where x = 1 to 15)

A.3.12 Release_Quantity

The quantity of Social Credit to be released per Release_Cycle. SC may be released in smaller quantities over shorter time periods within the SC_Cycle and any unused portions may be accumulated up to the end of the SC_Cycle, when such unused credit is cleared and the cycle resets to zero. (e.g 200 liter per day).

The value range and unit of measure is the same as that of the 16-bit STS transfer_Credit token Amount for the particular service type.

A.3.13 Release_Cycle

Frequency, at which smaller quantities of the SC_Quantity is released for use to the consumer. (e.g daily). The 4-bit binary number shall be interpreted as follows:

Table A 8 – Frequency values for the Release_Cycle attribute

Value	Frequency	Context
0	none	Never happens
1 to 12	hours	Starts on real time xxh00m00s (where xx = 1 to 12)
13	daily	Starts on real time 00h00m00s every day
14	weekly	Starts on 00h00m00s every Monday
15	monthly	Starts on 00h00m00s on the 01mmyyyy every month of every year

A.3.14 Available_Service_Level

The consumer may be compelled to receive a reduced level of service during the time that this credit function is active.

It is expressed as a % of the maximum service demand level that the consumer is contractually entitled to as determined by the applicable tariff.

The Delivery function regulates the service delivery in accordance with this value.

Note;

11) In electricity meters this would translate to a % of the power limit setting.

12) This particular reduced Service_Level constraint is lifted once the consumer has used his particular quantity of Social Credit within the particular Release_Cycle and temporarily drops to the next lower priority level function, until the next release event. (Also remember that the re-activation of this SC function may be subject to user confirmation each time)

The number format is binary, but the range is limited to between 0 and 100. Values from 101 to 255 are not allowed.

A.3.15 Low_Credit_Warning_Level

When the Available_Social_Credit reaches or drops below this level, the Low_Credit_Warning attribute is set true.

The value range and unit of measure is the same as that of the 16-bit STS transfer_Credit token Amount for the particular service type.

A.3.16 Low_Credit_Warning

Indicates to the Meter Accounting Process that the Available_Social_Credit is about to expire.

Low_Credit_Warning is true while Available_Social_Credit $\leq$ Low_Credit_Warning_Level.

Table A 9 – Indicator values for the Low_Credit_Warning attribute

Bit value	Context
0	False
1	True

A.3.17 Credit_Expiry_Level

When the Available_Social_Credit reduces to or drops below this level, the Credit_Expired attribute is set true. The Credit_Expiry_Level may be a positive or a negative number.

Table A 10 – Format for the Credit_Expiry_Level attribute

Bit number	Context	Value
15	Sign bit	
0 to 14	Mantissa	Range 0 to ± 3276.7 service units

Note 13 : This range should be sufficient given that normal operation is at zero units remaining of available credit.

A.3.18 Credit_Expired

When Available_Social_Credit ≤ Credit_Expiry_Level, then the Meter Accounting Process shall set the Credit_Expired attribute true. It shall then de-activate the Social_Credit_Function_Object and activate the next lower priority Function_Object if it is configured to be available.

Table A 11 – Indicator values for the Credit_Expired attribute

Bit value	Context
0	False
1	True

A.3.19 Interrupt_Service_On_Credit_Expiry

Dictates to the Delivery function whether the service delivery should be automatically interrupted when the Credit_Expired attribute is set true.

Table A 12 – Control values for the Interrupt_Service_On_Credit_Expiry attribute

Bit value	Context
0	False
1	True

A.3.20 Restore_Service_On_Credit_Replenishment

Dictates to the Delivery function whether the service delivery should be Automatically or Manually restored when the Credit_Expired attribute changes from true to false.

When set to Manual, the intervention of the user shall be required before the service may be restored, such as by pushing a button or by inserting a token carrier.

When set to Auto, the intervention of the user shall not be required for the service to be restored.

Table A 13 – Control values for the Restore_Service_On_Credit_Replenishment attribute

Bit value	Context
0	Auto
1	Manual

A.4 Methods

This is the external interface to this Function_Object instance from the Token perspective only and methods are thus only activated and operable by tokens that have been received by the meter.

All methods are pre-qualified in the APDU of the POS_to_Token_Carrier_Interface against the registered properties of the unique FOIN of the Function_Object.

A.4.1 set_SC_Profile(Profile_Data)

- *Configure the variable attributes of the entire profile of the Social_Credit_Function_Object, obtaining the values from the Token_Carrier as set by the management system.*

The inbound token that activates this method shall be authenticated using a unique or common decoder Key_Type.

The following attribute values are transferred from the Token_Carrier.

Table A 14 – Attribute values transferred from the Token_Carrier

Attributes	Bits	Value	Ref
SC_Profile_Revision	16	4 hex digits	A.3.2
Priority	4	Binary; range 1 to 15;	A.3.3
Availability	8	Bit position control per functionality	A.3.4
Activation_Criteria	1	Auto, Manual	A.3.9
SC_Quantity	16	16-bit STS	A.3.10
SC_Cycle	4	Binary; 0 to 15 months	A.3.11
Release_Quantity	16	16-bit STS	A.3.12
Release_Cycle	4	Binary; hours, daily, weekly, monthly	A.3.13
Available_Service_Level	8	Binary; 0-100%	A.3.14
Low_Credit_Warning_Level	16	16-bit STS	A.3.15
Credit_Expiry_Level	16	Integer	A.3.17
Interrupt_Service_On_Credit_Expiry	1	True, False	A.3.19
Restore_Service_On_Credit_Replenishment	1	Auto, Manual	A.3.20

A.4.2 get_SC_Profile()

- *Read the specified attribute values from the Social_Credit_Function_Object and write them onto the Token_Carrier.*

The inbound and the outbound Token shall be authenticated using a unique or common decoder Key_Type.

The following attribute values are transferred onto the Token_Carrier.

Table A 15 – Attribute values transferred to the Token_Carrier

Attributes	Bits	Value	Ref
SC_Profile_Revision	16	4 hex digits	A.3.2
Priority	4	Binary; range 1 to 15;	A.3.3
Availability	8	Bit position control per functionality	A.3.4
Accounting_Mode	1	Units-Based; constant	A.3.5

Activation_Criteria	1	Auto, Manual	A.3.9
SC_Quantity	16	16-bit STS	A.3.10
SC_Cycle	4	Binary; 0 to 15 months	A.3.11
Release_Quantity	16	16-bit STS	A.3.12
Release_Cycle	4	Binary; hours, daily, weekly, monthly	A.3.13
Available_Service_Level	8	Binary; 0-100%	A.3.14
Low_Credit_Warning_Level	16	16-bit STS	A.3.15
Credit_Expiry_Level	16	Integer	A.3.17
Interrupt_Service_On_Credit_Expiry	1	True, False	A.3.19
Restore_Service_On_Credit_Replenishment	1	Auto, Manual	A.3.20

A.4.3 display_SC_Profile(Attribute_Name)

- *Read the specified attribute values from the Social_Credit_Function_Object and display them onto the user interface for a suitable duration of time and in a format suitable for human interpretation.*

The inbound Token that activates this method shall be of STS Class 1.

The following attributes shall be displayed on the user interface.

Table A 16 – Attribute values displayed on the user interface

Attributes	Bits	Format	Ref
SC_Profile_Revision	16	4 decimal digits	A.3.2
Priority	4	Decimal value	A.3.3
Availability	8	2 hex digits	A.3.4
Accounting_Mode	1	‘0’ or ‘1’	A.3.5
Activation_Criteria	1	‘0’ or ‘1’	A.3.9
SC_Quantity	16	Decimal value	A.3.10
SC_Cycle	4	Decimal value	A.3.11
Release_Quantity	16	Decimal value	A.3.12
Release_Cycle	4	Decimal value	A.3.13
Available_Service_Level	8	Decimal value	A.3.14
Low_Credit_Warning_Level	16	Decimal value	A.3.15
Credit_Expiry_Level	16	Decimal value	A.3.17
Interrupt_Service_On_Credit_Expiry	1	‘0’ or ‘1’	A.3.19
Restore_Service_On_Credit_Replenishment	1	‘0’ or ‘1’	A.3.20

A.4.4 clear_SC_Service_Delivery_Register()

- *Clear the contents of the SC_Service_Delivery_Register to zero.*
- *Set the Date_Of_Last_Read to today’s date as per the real time clock in the meter.*

The inbound Token that activates this method shall be authenticated using a unique or common decoder Key_Type.

A.4.5 get_SC_Service_Delivery_Register()

- *Set Date_Of_Last_Read to today's date as per the real time clock in the meter.*
- *Retrieve the value of the SC_Service_Delivery_Register and write it onto the Token_Carrier together with the value of Date_Of_Last_Read (i.e. today's date).*
- *Clear the contents of the SC_Service_Delivery_Register to zero.*

The inbound and the outbound Token shall be authenticated using a unique or common decoder Key_Type.

The following attributes are thus transferred onto the Token_Carrier.

Table A 17 – Attribute values transferred to the Token_Carrier

Attributes	Bits	Value	Ref
SC_Service_Delivery_Register	16	16-bit STS	A.3.7
Date_Of_Last_Read	32	8 hex digits: ddmmyyyy	A.3.8

A.4.6 display_SC_Service_Delivery_Register()

- *Read the attribute value from the SC_Service_Delivery_Register and display it onto the user interface for a suitable duration of time and in a format suitable for human interpretation.*

The inbound Token that activates this method shall be of STS Class 1.

A.4.7 clear_SC_Accounting_Register()

- *Clear the contents of the SC_Accounting_Register to zero.*

The inbound Token that activates this method shall be authenticated using a unique or common decoder Key_Type.

A.4.8 display_SC_Accounting_Register()

- *Read the attribute value from the SC_Accounting_Register and display it onto the user interface for a suitable duration of time and in a format suitable for human interpretation.*

The inbound Token that activates this method shall be of STS Class 1.

A.4.9 decrement_SC_Accounting_Register()

- *Decrement the value in the SC_Accounting_Register with the amount on the Token.*

The Token shall be encrypted.

Note 14: This is a test token for using when evaluating the functionality of the Low_Credit_Warning_Level and the Credit_Expiry_Level above and below their set threshold values. (see A.6.6)

A.5 Operation

The internal functionality of the Social Credit function is as follows.

- Makes available a limited amount of credit for the delivery of FREE service to the consumer per a defined time cycle. The credit may be released in smaller quantities over shorter time periods.
- Accounts for the balance of Social Credit made available for use and actual service units delivered to the consumer from the available Social Credit.
- Operates in service units.

- Issues control and status signals and information in accordance with various attribute values (see A.3.14, A.3.16, A.3.18, A.3.19 and A.3.20).
- Tariff & Auxiliary Charge functions shall be inoperative while the Social Credit function is active. (See Note 15)

Note 15 It is the responsibility of the meter accounting process to ensure that Tariff & Auxiliary Charge functions are rendered inoperative.

Examples:

- Free basic water, typically available as 200 liter/day, but limited to 6 kiloliter/month;
- Free basic electricity, typically 60 kWh per month, available in one go from the start of the month;

A.6 Associations

These are services provided by support functions that are required to implement the Methods.

Other associations and interfaces may exist between this Social_Credit_Function_Object and other Function_Objects, but such interfaces are not under consideration in this specification.

A.6.1 Display functions

As per the relevant STS APDU tokens and presented at the Methods interface as defined in this specification.

The value of the SC_Accounting_Register contents shall normally be displayed on the user interface of the meter while the Social Credit function is active.

If the display mode of the meter changes for the purpose of displaying other values, it shall return to the normal display mode after a time-out or after manual intervention.

When displaying multiple attribute values on the user interface, a reference indicator shall indicate which attribute value is currently being displayed.

A.6.2 Recording functions

Setting, getting, clearing, incrementing and decrementing of register values shall be under the control of the Function_Object and as per the Methods interface defined in this specification.

Other methods may exist that provide more internal functionality, but these are outside of the scope of this specification.

A.6.3 Data_Exchange functions

Data shall be exchanged with the Function_Object by means of a token carrier via the Token_Carrier_to_Meter_Interface.

The data import and export to/from the Function_Object shall be via the Methods interface as defined in this specification.

A.6.4 Security functions

Data security shall be vested in the STS data exchange functions and the data integrity mechanisms employed by the recording functions.

It shall not be possible to alter any of the data attributes in this Function_Object other than through the Methods interface defined in this specification.

Management processes in the application domain shall control the generation and issuing of tokens that activate the Methods.

A.6.5 Time functions

A supporting time function shall provide date and time information in the format required by this specification.

A.6.6 Test functions

The Social_Credit_Function_Object operation shall be testable by means of using a decrement_SC_Accounting_Register token that activates the method to decrement the value in the SC_Accounting_Register with the amount on the Token.

The Low_Credit_Warning_Level and the Credit_Expiry_Level may be evaluated above and below their threshold values using several combinations of such tokens.

The Token shall be encrypted using a unique key type.

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 62055-41:2007, Electricity metering – Payment systems Part 41: Standard Transfer Specification – Application Layer Protocol for one-way token carrier systems

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

Version control

NP is issued as D1.00 for circulation and comments

Revision	Date	Author	Comments
D1.00	18/12/2005	DM Taylor	Copy of clause 9 from IEC 62055-41/CDV draft 2.00 in preparation for editing to release as STS 31-001/NP for comments Complementary information documents: • Change_log_STS_31-001_NP_D1.00doc No editing performed
D1.01	17/01/2006	DM Taylor	Updated figure 1 Update of FOIN to the 22-bit format Update of cross reference clause numbers Add Annex A example of Companion Specification
D1.02	23/03/2007	DM Taylor	Renumbered document to STS 200-1
	16/04/2007	DM Taylor	Adjusted FOIN structure to 5+12+5 bits to align with STSE
D1.03	17/04/2007	DM Taylor	Also changed underscore to camel notation for clause 5.5. only
D1.04	20 May 2012	Borton	Checked References, Added: Foreword, Introduction, Scope, Normative References, Definitions, Abbreviated Terms, Notation And Terminology, Numbering Conventions, Reference Model - Renumbered “Notes”