CONVERGENT SOFTWARE

Convergent Software & ISO 28560-2 Conformance Document

1. Introduction

The document lays out Convergent Software’s response to the document:

Guidelines for ISO 28560-2 conformant devices and processes
(direct link: http://biblstandard.dk/rfid/docs/conformance_28560-2.pdf).

In the remainder of this document, that document will be referred to as the “ISO 28560-2 Conformance Document”.

1.1 Definition of Terms

The following terms will are used throughout this document:

Term	Definition
CSL	Convergent Software Limited
The core software	CSL’s ISO 28560-2 set of .NET libraries focused on the encoding, decoding and editing of logical memory.
ISO 28560-2 Conformance Document	*Guidelines for ISO 28560-2 conformant devices and processes* (direct link: http://biblstandard.dk/rfid/docs/conformance_28560-2.pdf)

2. Scope of Software

Convergent Software’s ISO 28560-2 core software is a set of .NET libraries focused on the encoding, decoding and editing of logical memory. In each scenario, the software does not interact with the RFID interrogator. The core software is designed for integration in solutions, providing all of the conformance requirements for encoding, decoding and editing, off-the-shelf. As a result, all of the conformance requirements at the device interface layer are not met by the core software, and their implementation is the responsibility of systems integrators or software vendors embedding the core software in their own.

The decoding and encoding functions are also made available via web services, which make use of the core libraries and, therefore, have the same conformance levels, thereof.

Convergent Software also produces some tools that include the air interface layer but these are for quality control purposes only and work with a limited range of devices. These tools are not included in the scope of this document.

3. Conformance statements for ISO 28560-2

The *ISO 28560-2 Conformance Document* discusses the need for correct bit and byte ordering. This issue is outside of the scope of the core software and needs to be implemented by systems integrators.

3.1 System Support Using ISO/IEC 18000-3 Mode 1 RFID Tags

The majority of this section is outside of the scope of the core software. However, there are some features (e.g. block size) that have an impact of the encoding or decoding of logical memory and are, therefore, relevant to the core software.

- Support one or more block sizes (within the specified range of 1 to 32 bytes per block), by declaring specific block sizes that are supported.

  The core software supports all block sizes and the API allows calling applications to specify the block size in use.

- Support one or more memory sizes by declaring specific memory encoding capacity or the minimum and maximum capacity supported.

  The core software supports any memory size and the API allows calling applications to specify the memory capacity in use.

- Support the writing of the AFI using specified air interface commands. Declare whether AFI locking is supported.

  The air interface commands for writing and locking the AFI require no encoding rules to be applied. Invoking these commands need to be invoked by systems integrators (see 2. Scope of Software).

- Support the writing of the hard-coded DSFID using specified air interface commands. Declare whether DSFID locking is supported.

  The air interface commands for writing and locking the DSFID require no encoding rules to be applied. Invoking these commands need to be invoked by systems integrators (see 2. Scope of Software).

- Declare whether the system supports RFID tags that can only encode the soft-coded DSFID (where this has to be encoded to precede the encoded data).

  The core software supports soft-coded DSFID, encoding it as part of the logical memory. The API allows calling applications to specify whether the DSFID is hard-coded or soft-coded.

- Support the encoding and decoding of the AFI compliant with ISO 28560-1:
  - Using C2$_{\text{HEX}}$ for all loan items that leave the library;
  - Optionally, using 07$_{\text{HEX}}$ for in stock items if this is used for the security system.
  - Declare whether an AFI other than those specified by ISO 28560-2 will be processed as non-compliant and belonging to a different application. Optionally, provide or report a message for tags with such an AFI.

  The air interface commands for writing and locking the DSFID require no encoding rules to be applied. Invoking these commands need to be invoked by systems integrators (see 2. Scope of Software).

- Support the encoding and decoding of the DSFID compliant with ISO 28560-2, comprising:
  - The correct access method;

- The correct data format;
- Resulting in a DSFID value of 06_{HEX}.
- During the decoding procedure, a DSFID other than that declared by ISO 28560-2 shall be processed as non-compliant and belonging to a different application or encoding. Optionally, provide or report a message for tags with such a DSFID.

Where the tag requires a soft-coded DSFID, the core software deals with it appropriately, including it at the start of the logical memory when encoding, and expecting it at the start of logical memory when decoding. An error is reported on decode if a soft-coded DSFID is expected but none is found.

The hard-coded DSFID is written and read at the device interface using air interface commands (see 2. Scope of Software).

3.2 The Data Encoding Process

This deals explicitly with encoding data into user memory using an encoding engine, or similar device. The first two conformance features require an explicit declaration of whether they are supported based on the types of RFID tags that are supported.

❑ Declare whether the encoding process is capable of encoding a soft-coded DSFID, where a separate DSFID memory and commands are not supported by the RFID tag.

Yes

❑ Declare whether the encoding process is capable of selectively locking the soft-coded DSFID if used on the tag and specified to be locked by the user.

ENCODING ADVICE: The issues of locking or not locking the soft-coded DSFID depend on these four states:

A. If the soft-coded DSFID and the Primary Item Identifier are both intended to be locked, then at least the first block needs to be locked.
B. If the soft-coded DSFID and the Primary Item Identifier are both intended to be unlocked then at least the first block needs to be unlocked.
C. If the soft-coded DSFID is intended to be locked and the Primary Item Identifier intended to be unlocked, then the remaining bytes in the first block shall be encoded with the null byte (80_{HEX}) and this first block shall be locked.
D. If the soft-coded DSFID is intended to be unlocked and the Primary Item Identifier intended to be locked, then the remaining bytes in the first block shall be encoded with the null byte (80_{HEX}) and this first block shall be unlocked. Locking is then invoked for the blocks that contain the Primary Item Identifier.

Yes, selectively locking the soft-coded DSFID is supported and all four states are supported.

The following are mandatory requirements:

❑ Fully comply with ISO 28560-1 data elements.

Yes, all data elements are supported.

❑ Support the selection of a sub-set of data elements to comply with a national data model, with further selection (or de-selection) of optional data elements.

Yes, the API allows the calling application to pass any combination of ISO 28560-1 data elements.

- Enable the sequence of data objects to be defined by the library for a given scenario, so that the most important data objects are encoded in the lowest numbered memory blocks.

  Yes, the API allows the calling application to pass the ISO 28560-1 data elements in any sequence, with the exception of Relative-OID 1 (see below).

- Support variable length data input where this is permitted by ISO 28560-1 and 28560-2.

  Yes

- Validate input data objects to comply with ISO 28560-1.

  Yes. All data elements are checked for a valid format as defined in ISO 28560-1. Some data elements have additional checks, e.g. the proper construction of set information.

- Automatically construct the OID Index (Content Parameter) if called for by the application and encode this in the correct second position as defined in ISO 28560-2.

  Yes

- Prevent changes in the encoded data that would breach the ISO 28560-2 standard, specifically:
  - Avoid the removal of Relative-OID 1 (Primary Item Identifier);

    Yes

  - Avoid a change from position 1 of Relative-OID 1 (Primary Item Identifier);

    Yes

  - If encoded, avoid a change from position 2 of Relative-OID 2 (Content Parameter - OID Index).

    The CSL software provides more flexibility by supporting the OID index being in a different position. For example a library may choose to encode the Primary Item Identifier followed by the ISIL of the Owner Institute and then lock these two data elements. In this case the OID index may be third in sequence.

- Encode data to the ISO/IEC 15962 rules called out in ISO 28560-2.

  Yes

- Exclude ISO/IEC 15962 encoding rules not specified in ISO 28560-2, for example access methods [encoding schemes] not referred to in ISO 28560-2.

  Yes

- Encode data to the specific ISO 28560-2 rules, for example the ISIL, OID index (Content Parameter), and Set Information.

  Yes

- Correctly format the encoding when a data set is specified by the user to be locked, including the implications for adjacent data sets.

  Yes

- Provide or record a message when the intended encoding exceeds the memory capacity of the RFID tag.

  Yes, based on the size of memory specified by the calling application.

- Encode data such that a potential data overflow issue is resolved without permitting the partial programming of any data element on the tag.

  Yes

- Optionally, provide or report messages for any input errors.

  Yes

3.3 The Data Decoding Process

This deals explicitly with decoding data from user memory using a decoding engine, or similar device.

- Declare whether the decoding process is capable of decoding a soft-coded DSFID, where a separate DSFID memory and commands are not supported by the RFID tag.

  Yes

- Declare whether the decoding process supports the selective reading of a sequence of blocks across the air interface (typically for fast reading operations) where all the data sets might not be included in the air interface transmission. In this case ensure that any truncated and incomplete data set, either at the beginning and / or end of the transmission, is ignored and not treated as errors.

  Yes. The CSL software supports these different read types:

  - *Read 1st Objects – This is intended for fast reading operations. At the device interface layer sufficient blocks need to be read. We can provide advice on making this efficient. The application layer needs to provide a list of Relative-OIDs (this could just be the one for the Primary Item identifier in the first position). If insufficient blocks have been read, The CSL software issues a "short read" return value so that the calling application can read more blocks of memory.*
  - *Read Multiple Objects – This read type is different in that the Relative-OIDs do not need to be in sequence. As such the air interface transactions can be longer and require iterative read cycles. Otherwise the process defined for Read 1st Objects applies.*

- *Read All Objects – The device layer process invoked by the systems integrator should read all encoded memory blocks, i.e. impose a device layer constraint on reading memory that is not encoded. We can provide advice on making this efficient. In principle, the CSL decoding process should be delivered with all the encoded memory blocks. Sometimes this might not happen (e.g. because of an air interface time-out). In the instances where the CSL decoding process receives a partial data set, an error is raised.*

As the number of blocks being transferred across the air interface is increased, then the OID Index and other subsequent data elements can be decoded.

❑ Decode data to the ISO/IEC 15962 rules called out in ISO 28560-2.

Yes

❑ Decode data to the specific ISO 28560-2 rules, for example the ISIL, OID index and Set Information.

Yes

❑ Correctly decode a data set specified by the user to be locked.

Yes; this includes correctly decoding a correctly encoded locked data set and discarding any pad bytes.

❑ Any Relative-OID that is not defined in ISO 28560-2 or by extensions agreed by the relevant ISO committee shall not be decoded and interpreted.

Yes, an error is reported if any Relative-OID values are encountered that are not defined in ISO 28560-2.

❑ Any RFID tag with data using ISO/IEC 15962 encoding rules not specified in ISO 28560-2 (for example access methods [encoding schemes] not referred to in ISO 28560-2) shall be treated as non-compliant and not be processed as a properly encoded tag. The tag should be rejected.

Yes

3.4 The Editing Process

The editing process needs to be built on a fully functional encoding and decoding engine as defined above to enable users to decode existing tag data and edit the results by deleting, modifying (including locking) or adding data objects. Specifically:

❑ System features that support editing shall identify all existing locked memory blocks, and interpret this into information of locked data elements.

Yes, the information of locked blocks is interpreted into information of locked data elements. Obtaining the information from the tag about which blocks are locked is obtained using specific air interface commands (see 2. Scope of Software). CSL can provide advice about how to use this feature.

- System features that support editing shall provide a mechanism for identifying which memory blocks have been changed by the editing operation·

  Yes

- System features that modify existing data elements on a tag shall perform that modification, taking into account that the new data might require more or less encoding space and encode to the rules of ISO/IEC 15962 and ISO 28560-2.

  Yes. The CSL software makes use of existing memory space where the new encoding requires the same or less memory; and appends data where the new encoding is longer. The CSL software also takes into account the additional complexity where there are locked blocks.

- System features that delete existing data elements on a tag shall perform that deletion, providing a resultant encoding compliant with the rules of ISO/IEC 15962.

  Yes. Data can be deleted without re-writing the entire tag memory.

- System features that append any additional ISO 28560 data elements not encoded on the RFID tag shall perform that modification, encoding to the rules of ISO/IEC 15962 and ISO 28560-2 and take into account the rule that duplicate data elements are not permitted.

  Yes

- System features that support editing shall selectively lock any modified or appended data element, as defined by the user, ensuring that the locking process results in a compliant encoding.

  Yes

- System features that support editing shall automatically update the OID index within the encoding capacity of the prevailing OID index and its encoded position in memory. That is, the rule for the sequence of the OID index is paramount and, therefore, the OID index should not be nullified and appended if it needs to grow larger than the space currently available to it. One approach to avoid this problem is to always encode the OID index to be large enough to accommodate the library's potential list of data elements or the current complete list of data elements.

  Yes

- Provide or report relevant messages where the intended modified or appended data will not fit in the specified memory size.

  Yes

- Append data to a tag such that a potential data overflow issue is resolved without permitting the partial programming of any data element on the tag

  Yes

❑ Optionally, provide or record messages for any input errors.

Yes