RFID

With

SmartCodeStudio and SmartCodeDrivers

General Enquiry : info@technoriversoft.com
Sales : sales@technoriversoft.com

TechnoRiver
Outstanding Printing Software

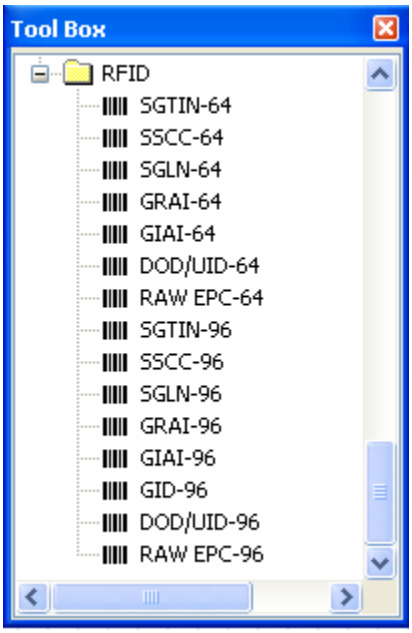
RFID (Radio Frequency and Identification)

RFID can be thought of simply as a type of electronic bar code. It does not have the requirement of an uninterrupted line-of-sight between the label and the human or machine reader. It has gain significant interest in many different industries including retail, manufacturing and health care. Many thermal printer manufacturers have also added support for RFID in their printers.

SmartCodeStudio and SmartCodeDrivers come with extremely good support for RFID. This document attempts to describe the setup and usage of RFID in SmartCodeStudio. It also describes in details the support for the different RFID Encoding Schemes.

Setting up SmartCodeStudio and SmartCodeDrivers for RFID

- Get ready your printer that supports RFID
- Install the latest version of SmartCodeStudio (Build 1416 onwards).
- Install the appropriate SmartCodeDriver for the printer that supports RFID.
- Both SmartCodeStudio and SmartCodeDrivers can be downloaded from **http://www.technoriversoft.com/downloads.html**
- Launch SmartCodeStudio and when prompted to create a label, make sure to choose the SmartCodeDriver Printer installed previously.
- The Tool Box will display the different RFID objects.



- Simply drag and drop an RFID encoding from the unique One-Touch object Toolbox of SmartCodeStudio and place it on the label.
- Double click on the RFID object and key in the values for the particular encoding scheme.
- Save and Print your label. SmartCodeStudio will automatically work with the drivers and printer to encode the label.

SmartCodeStudio supports the following RFID Encoding Schemes

- **SGTIN-64**
- **SSCC-64**
- **SGLN-64**
- **GRAI-64**
- **GIAI-64**
- **DOD/UID-64**
- **RAW EPC-64**
- **SGTIN-96**
- **SSCC-96**
- **SGLN-96**
- **GRAI-96**
- **GIAI-96**
- **GID-96**
- **DOD/UID-96**
- **RAW EPC-96**

SGTIN-64

Format: H:P1;P2;P3;P4
where H = Header (2 bits). Header is an optional field. SmartCodeStudio will automatically format the output with the appropriate header bits.

P1 = Filter Value (3 bits)
P2 = Company Prefix (14 bits)
P3 = Item Reference (20 bits)
P4 = Serial Number (25 bits)

Sample input for SGTIN-64:
Text encoding: 2;14501;200045294701
Expected output: 96062E6062000000

Dec encoding: 2;14501;200045294701;632637288
Expected output: 9714A1E8DBB54768

Hex encoding: 2;14501;200045294701;632637288
Expected output: 90A0328E02637288

SSCC-64

Format: H:P1;P2;P3
where H = Header (8 bits). Header is an optional field. SmartCodeStudio will automatically format the output with the appropriate header bits.

P1 = Filter Value (3 bits)
P2 = Company Prefix (14 bits)
P3 = Serial Number (39 bits)

Sample input for SSCC-64:
Text encoding: 2;14501;200045294701
Expected output: 085818B934373031

Dec encoding: 2;14501;200045294701
Expected output: 085C52AE93A0F46D

Hex encoding: 2;14501;200045294701
Expected output: 0842808045294701

SGLN-64

Format: H:P1;P2;P3;P4
where H = Header (8 bits). Header is an optional field. SmartCodeStudio will automatically format the output with the appropriate header bits.

P1 = Filter Value (3 bits)
P2 = Company Prefix (14 bits)
P3 = Location Reference (20 bits)
P4 = Serial Number (19 bits)

Sample input for SGLN-64:
Text encoding: 2;14501;200045294701
Expected output: 095818B981880000

Dec encoding: 2;14501;200045294701
Expected output: 095C5287A3680000

Hex encoding: 2;14501;200045294701;632637288
Expected output: 094280CA380B7288

GRAI-64

Format: H:P1;P2;P3;P4
where H = Header (8 bits). Header is an optional field. SmartCodeStudio will automatically format the output with the appropriate header bits.

P1 = Filter Value (3 bits)
P2 = Company Prefix (14 bits)
P3 = Asset Type (20 bits)
P4 = Serial Number (19 bits)

Sample input for GRAI-64:
Text encoding: 2;14501;200045294701
Expected output: 0A5818B981880000

Dec encoding: 2;14501;200045294701
Expected output: 0A5C5287A3680000

Hex encoding: 2;14501;200045294701;632637288
Expected output: 0A4280CA380B7288

GIAI-64

Format: H:P1;P2;P3
where H = Header (8 bits). Header is an optional field. SmartCodeStudio will automatically format the output with the appropriate header bits.

P1 = Filter Value (3 bits)
P2 = Company Prefix (14 bits)
P3 = Asset Reference (39 bits)

Sample input for GIAI-64:
Text encoding: 2;14501;200045294701
Expected output: 0B5818B934373031

Dec encoding: 2;14501;200045294701
Expected output: 0B5C52AE93A0F46D

Hex encoding: 2;14501;200045294701
Expected output: 0B42808045294701

DOD/UID-64

Format: H:P1;P2;P3
where H = Header (8 bits). Header is an optional field. SmartCodeStudio will automatically format the output with the appropriate header bits.

P1 = Filter Value (2 bits)
P2 = Government Managed Identifier (30 bits)
P3 = Serial Number (24 bits)

Sample input for DOD/UID-64:
Text encoding: 2;14501;200045294701
Expected output: CEB1D35C31373031

Dec encoding: 2;14501;200045294701
Expected output: CEB1D35C31A0F46D

Hex encoding: 2;14501;200045294701
Expected output: CEB1D35C31294701

RAW EPC-64

No format. Any input string can be converted.

Sample input for RAW EPC-64:
Text encoding: is a test string
Expected output: 7420737472696E67

Dec encoding: 9987
Expected output: 0000000000002703

Hex encoding: ABCDEF1234567890
Expected output: ABCDEF1234567890

SGTIN-96

Format: H:P1;P2;P3;P4;P5
where H = Header (8 bits). Header is an optional field. SmartCodeStudio will automatically format the output with the appropriate header bits.

P1 = Filter Value (3 bits)
P2 = Partition (3 bits). This field is generated by SmartCodeStudio. There is no need to input this field.
P3 = Company Prefix (20-40 bits, depending on the Partition bit)
P4 = Item Reference (24-4 bits, depending on the Partition bit)
P5 = Serial Number (38 bits)

Sample input for SGTIN-96:
Text encoding: 12345678;9087654;64782922
Expected output: 3000E0DCD8D4D08000000000

Dec encoding: 2;14501;765432;56743265
Expected output: 30580E2942EB7E000361D561

Hex encoding: 2;14501;98576342
Expected output: 3058514055D8D08000000000

SSCC-96

Format: H:P1;P2;P3;P4;P5
where H = Header (8 bits). Header is an optional field. SmartCodeStudio will automatically format the output with the appropriate header bits.

P1 = Filter Value (3 bits)
P2 = Partition (3 bits). This field is generated by SmartCodeStudio. There is no need to input this field.
P3 = Company Prefix (20-40 bits, depending on the Partition bit)
P4 = Serial Reference (37-17 bits, depending on the Partition bit)
P5 = Unallocated (25 bits)

Sample input for SSCC-96:
Text encoding: Text1;Text2;Text3;Text4
Expected output: 31215195E1D0C87433787434

Dec encoding: 2;14501;765432;56743265
Expected output: 31580E2940000BADF861D561

Hex encoding: 2;14501;98576342
Expected output: 315851404098576342000000

SGLN-96

Format: H:P1;P2;P3;P4;P5
where H = Header (8 bits). Header is an optional field. SmartCodeStudio will automatically format the output with the appropriate header bits.

P1 = Filter Value (3 bits)
P2 = Partition (3 bits). This field is generated by SmartCodeStudio. There is no need to input this field.
P3 = Company Prefix (20-40 bits, depending on the Partition bit)
P4 = Location Reference (21-1 bits, depending on the Partition bit)
P5 = Serial Number (41 bits)

Sample input for SGLN-96:
Text encoding: Text1;Text2;Text3;Text4
Expected output: 32215195E1D0CA5465787434

Dec encoding: 2;14501;765432;56743265
Expected output: 32580E29575BF0000361D561

Hex encoding: 2;14501;98576342
Expected output: 325851406EC6840000000000

GRAI-96

Format: H:P1;P2;P3;P4;P5
where H = Header (8 bits). Header is an optional field. SmartCodeStudio will automatically format the output with the appropriate header bits.

P1 = Filter Value (3 bits)
P2 = Partition (3 bits). This field is generated by SmartCodeStudio. There is no need to input this field.
P3 = Company Prefix (20-40 bits, depending on the Partition bit)
P4 = Asset Type (24-4 bits, depending on the Partition bit)
P5 = Serial Number (38 bits)

Sample input for GRAI-96:
Text encoding: Text1;Text2;Text3;Text4
Expected output: 33215195E1D0C8D465787434

Dec encoding: 2;14501;765432;56743265
Expected output: 33580E2942EB7E000361D561

Hex encoding: 2;14501;98576342
Expected output: 3358514055D8D08000000000

GIAI-96

Format: H:P1;P2;P3;P4
where H = Header (8 bits). Header is an optional field. SmartCodeStudio will automatically format the output with the appropriate header bits.

P1 = Filter Value (3 bits)
P2 = Partition (3 bits). This field is generated by SmartCodeStudio. There is no need to input this field.
P3 = Company Prefix (20-40 bits, depending on the Partition bit)
P4 = Asset Reference (62-42 bits, depending on the Partition bit)

Sample input for GIAI-96:
Text encoding: Text1;Text2;Text3;Text4
Expected output: 34215195E1D0C85465787433

Dec encoding: 2;14501;765432;56743265
Expected output: 34580E2940000000000BADF8

Hex encoding: 2;14501;98576342
Expected output: 345851404000000098576342

GID-96

Format: H:P1;P2;P3
where H = Header (8 bits). Header is an optional field. SmartCodeStudio will automatically format the output with the appropriate header bits.

P1 = General Manager Number (28 bits)
P2 = Object Class (24 bits).
P3 = Serial Number (36 bits)

Sample input for GID-96:
Text encoding: 2;14501;200045294701
Expected output: 350000032353031934373031

Dec encoding: 2;14501;200045294701
Expected output: 3500000020038A5E93A0F46D

Hex encoding: 2;14501;200045294701
Expected output: 350000002014501O452947010

DOD/UID-96

Format: H:P1;P2;P3
where H = Header (8 bits). Header is an optional field. SmartCodeStudio will automatically format the output with the appropriate header bits.

P1 = Filter Value (4 bits)
P2 = DODAAC/CAGE (48 bits)
P3 = Serial Number (36 bits)

Sample input for DOD/UID-96:
Text encoding: 2;14501;200045294701
Expected output: CF2003134353031934373031

Dec encoding: 2;14501;200045294701
Expected output: CF2003134353031E93A0F46D

Hex encoding: 2;14501;200045294701
Expected output: CF2003134353031045294701

RAW EPC-96

No format. Any input string can be converted.

Sample input for RAW EPC-96:
Text encoding: Text123
Expected output: 000000000054657874313233

Dec encoding: 9991
Expected output: 000000000000000000002707

Hex encoding: 34567890ABCDEF1234567890
Expected output: 34567890ABCDEF1234567890