XFS4IoT SP-Dev Workgroup

4th February 2025

XFS4IoT SP-Dev Workgroup agenda

- Recap from previous meeting
- Contactless card reader demo
- New specification version 2024-03
- Code generators
- What’s next?
- Next meeting

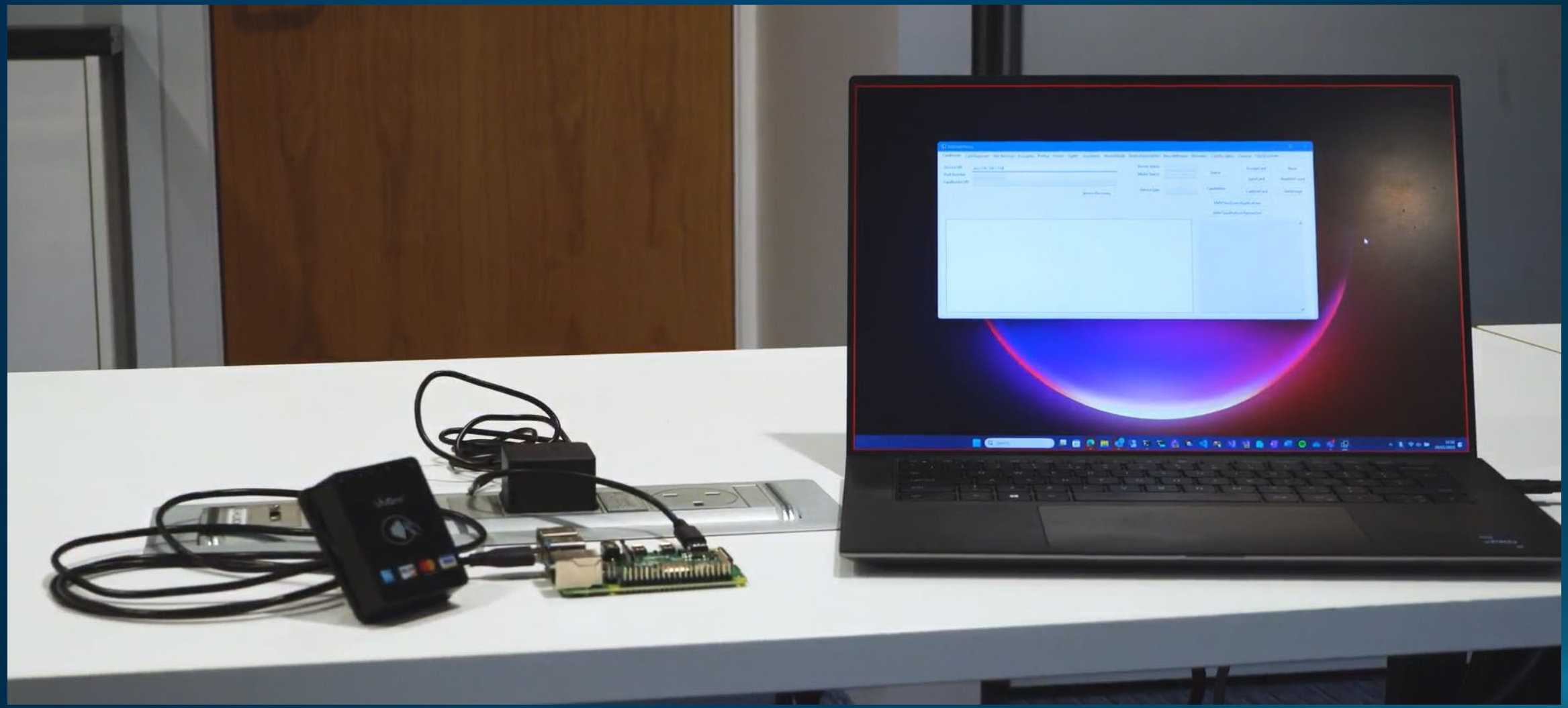
Recap from previous meeting

Recap from previous meeting

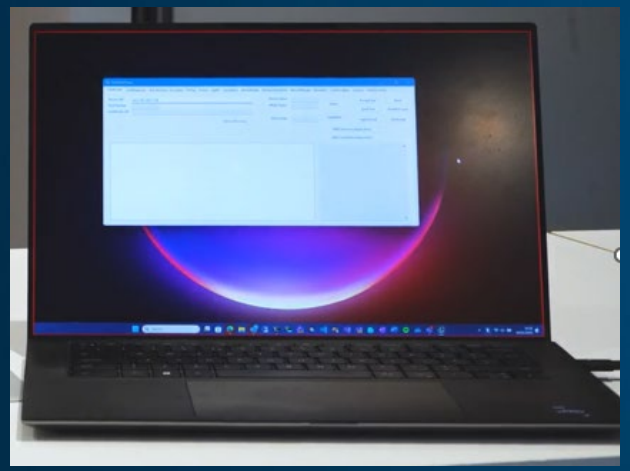
- TLS Authentication
  - Discussed options to authenticate initial http/s call
- Recap of topics presented in 2024
- Recap of all demos in 2024
- CEN specification update

Contactless card reader

Contactless card reader demo



Laptop running TestClientForms



Laptop running SP-Dev TestClientForms application

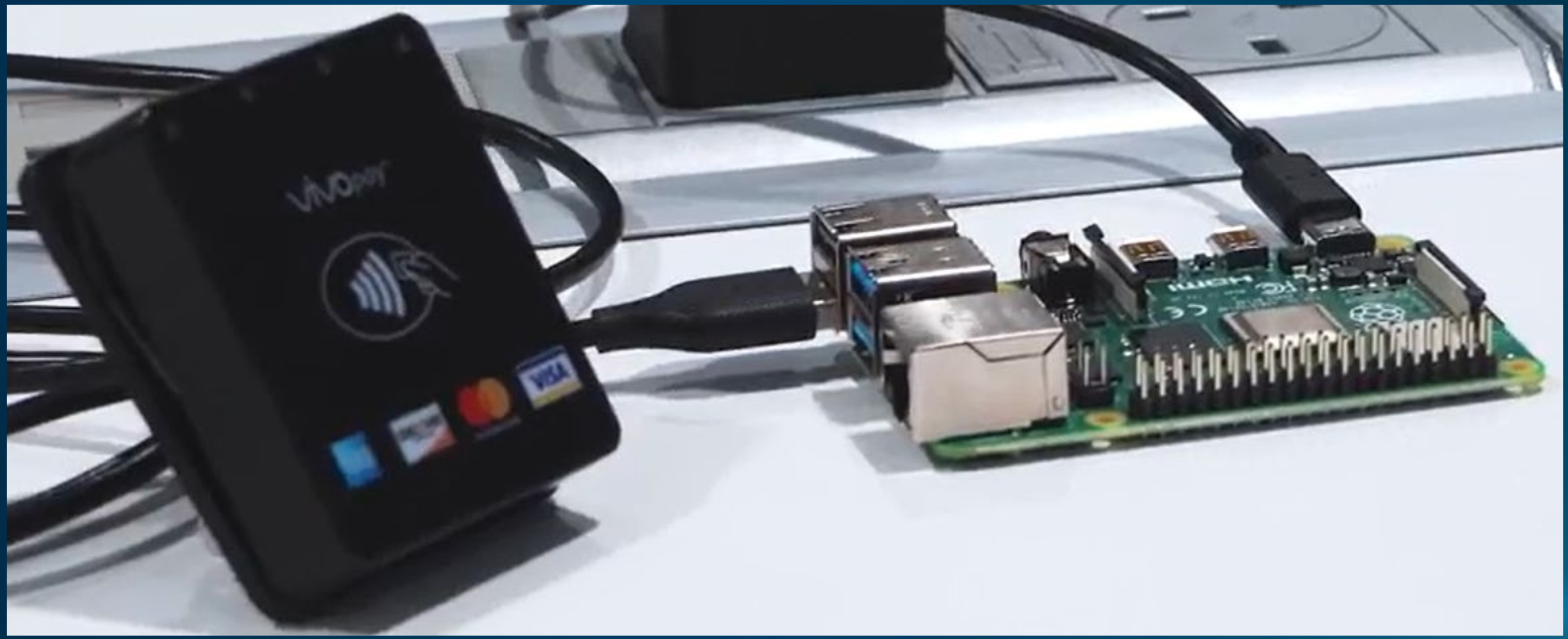
Found in SP-Dev Samples repository

Updated to support EMV Contactless commands

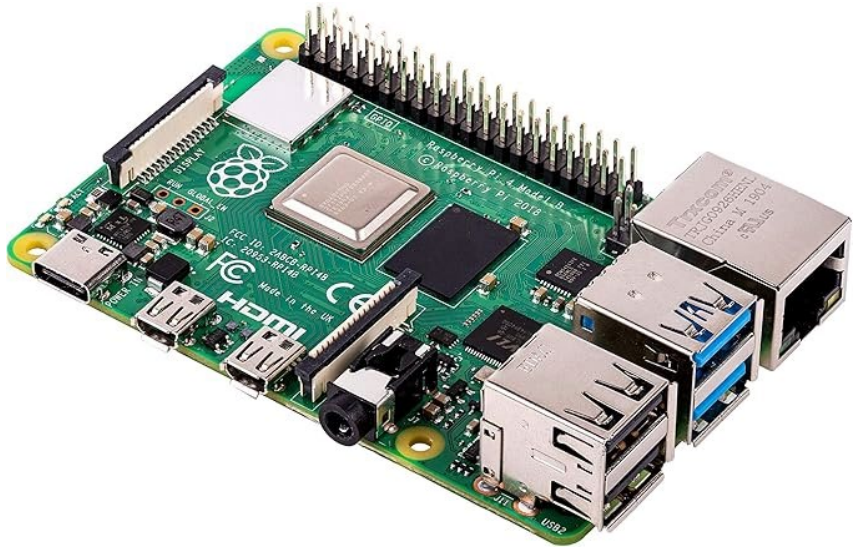
Will be updated on GitHub soon

Connection over Wi-Fi

Contactless card reader - SP



Contactless card reader - SP



Raspberry Pi 4

Raspberry Pi OS
IDTech C# SDK

SP can run on Windows or Linux

Connection over Wi-Fi

Contactless card reader demo

New specification version 2024-03

CEN – specification update

- 2024-03 finalised in December 2024
- Available on GitHub:
  - https://xfs4iot.github.io/Specifications-Preview.github.io
- Delivered to CEN for publication. Should be available in the next few months

CEN – specification update

- New banknote neutralisation interface
- New power management interface
- Support for German DK card specification
- Data sensitivity information on fields
- New JSON form format for text terminal and printer
- Many improvements, clarifications, and fixes

Code generators

XFS4IoT Specifications

- Specifications defined in YAML
- All machine readable
- PDF and HTML output generated from YAML

CardReader > Commands > ! ReadRawData.yaml

```yaml
xfs4iot: "1.0.0"

commands:
  CardReader.ReadRawData:
    description: |-
      For motor driven card readers, the card unit checks whether a card has been inserted. If so, all specified tracks
      are read immediately. If reading the chip is requested, the chip will be contacted and reset and the ATR (Answer
      To Reset) data will be read. When this command completes the chip will be in contacted position. This command can
      also be used for an explicit cold reset of a previously contacted chip.

    command:
      $ref: "#/components/messages/CardReader.ReadRawDataCommand"
    events:
      CardReader.InsertCardEvent:
        $ref: "../Events/InsertCardEvent.yaml#/components/messages/CardReader.InsertCardEvent"
      CardReader.MediaInsertedEvent:
        $ref: "../Events/MediaInsertedEvent.yaml#/components/messages/CardReader.MediaInsertedEvent"
      CardReader.InvalidMediaEvent:
        $ref: "../Events/InvalidMediaEvent.yaml#/components/messages/CardReader.InvalidMediaEvent"
      CardReader.TrackDetectedEvent:
        $ref: "../Events/TrackDetectedEvent.yaml#/components/messages/CardReader.TrackDetectedEvent"
    completion:
      $ref: "#/components/messages/CardReader.ReadRawDataCompletion"

components:
  messages:
    CardReader.ReadRawDataCommand:
      version: "2.0"
      payload:
        type: object
        minProperties: 1
        properties:
          track1:
            description: |-
              Track 1 of the magnetic stripe will be read.
            type: boolean
            default: false
          track2:
            description: |-
              Track 2 of the magnetic stripe will be read.
            type: boolean
            default: false
```

SP-Dev Framework Core message classes

- Generated Command, Completion and Event classes
- Each command must be output so we can parse the data to/from JSON

```
15  namespace XFS4IoT.CardReader.Commands
16  {
17      //Original name = ReadRawData
18      [DataContract]
19      [XFS4Version(Version = "2.0")]
20      [Command(Name = "CardReader.ReadRawData")]
        30 references
21      public sealed class ReadRawDataCommand : Command<ReadRawDataCommand.PayloadData>
22      {
            2 references
23          public ReadRawDataCommand(int RequestId, ReadRawDataCommand.PayloadData Payload, int Timeout)
24              : base(RequestId, Payload, Timeout)
25          { }
26
27          [DataContract]
            9 references
28          public sealed class PayloadData : MessagePayload
29          {
30
                2 references
31              public PayloadData(bool? Track1 = null, bool? Track2 = null, bool? Track3 = null, bool? Chip = null, bool
49
50              /// <summary>
51              /// Track 1 of the magnetic stripe will be read.
52              /// </summary>
53              [DataMember(Name = "track1")]
                4 references
54              public bool? Track1 { get; init; }
55
56              /// <summary>
57              /// Track 2 of the magnetic stripe will be read.
58              /// </summary>
59              [DataMember(Name = "track2")]
                3 references
60              public bool? Track2 { get; init; }
61
62              /// <summary>
63              /// Track 3 of the magnetic stripe will be read.
64              /// </summary>
65              [DataMember(Name = "track3")]
                3 references
66              public bool? Track3 { get; init; }
67
```

SP-Dev Framework service providers

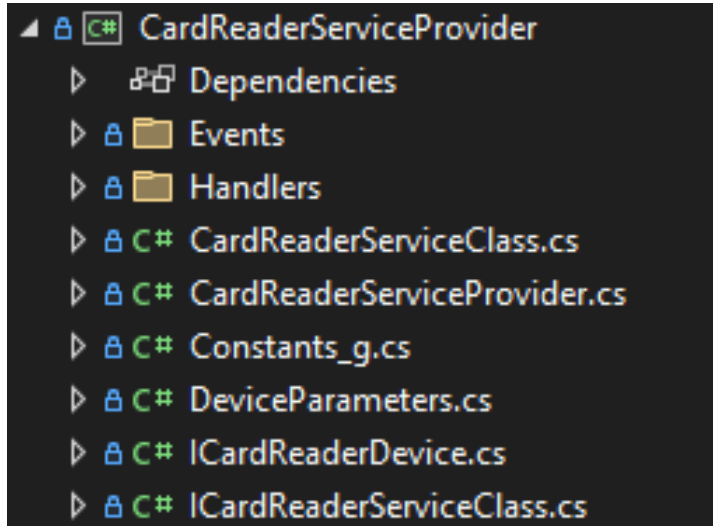
- Generated partial classes for each command
- Manual part with SP-Dev implementation
- Output into ServiceProvider project

```
21  namespace XFS4IoTFramework.CardReader
22  {
23      [CommandHandler(XFSConstants.ServiceClass.CardReader, typeof(ReadRawDataCommand))]
        6 references
24      public partial class ReadRawDataHandler : ICommandHandler
25      {
            0 references
26          public ReadRawDataHandler(IConnection Connection, ICommandDispatcher Dispatcher, ILogger logger)...
40
            3 references
41          public async Task Handle(object command, CancellationToken cancel)
42          {
43              var ReadRawDataCommand readRawDataCmd = command.IsA<ReadRawDataCommand>(message: $"Invalid parameter in the ReadRawData Handle met
44              readRawDataCmd.Header.RequestId.HasValue.IsTrue();
45
46              IReadRawDataEvents events = new ReadRawDataEvents(Connection, readRawDataCmd.Header.RequestId.Value);
47
48              var CommandResult<PayloadData> result = await HandleReadRawData(events, readRawDataCmd, cancel);
49              await Connection.SendMessageAsync(message: new ReadRawDataCompletion(readRawDataCmd.Header.RequestId.Value, result.Payload, r
50
51              await this.IsA<ICommandHandler>().CommandPostProcessing(result);
52          }
53
            8 references
54          public async Task HandleError(object command, Exception commandException)
55          {
56              var ReadRawDataCommand readRawDataCommand = command.IsA<ReadRawDataCommand>();
57              readRawDataCommand.Header.RequestId.HasValue.IsTrue();
58
59              MessageHeader.CompletionCodeEnum errorCode = commandException switch
60              {
61                  InvalidDataException => MessageHeader.CompletionCodeEnum.InvalidData,
62                  InternalErrorException => MessageHeader.CompletionCodeEnum.InternalError,
63                  UnsupportedDataException => MessageHeader.CompletionCodeEnum.UnsupportedData,
64                  SequenceErrorException => MessageHeader.CompletionCodeEnum.SequenceError,
65                  AuthorisationRequiredException => MessageHeader.CompletionCodeEnum.AuthorisationRequired,
66                  HardwareErrorException => MessageHeader.CompletionCodeEnum.HardwareError,
67                  UserErrorException => MessageHeader.CompletionCodeEnum.UserError,
68                  FraudAttemptException => MessageHeader.CompletionCodeEnum.FraudAttempt,
69                  DeviceNotReadyException => MessageHeader.CompletionCodeEnum.DeviceNotReady,
70                  InvalidCommandException => MessageHeader.CompletionCodeEnum.InvalidCommand,
71                  NotEnoughSpaceException => MessageHeader.CompletionCodeEnum.NotEnoughSpace,
72                  NotImplementedException or NotSupportedException => MessageHeader.CompletionCodeEnum.UnsupportedCommand,
73                  TimeoutCanceledException t when t.IsCancelRequested => MessageHeader.CompletionCodeEnum.Canceled,
                    TimeoutCanceledException => MessageHeader.CompletionCodeEnum.TimeOut
```

Other uses for the generator

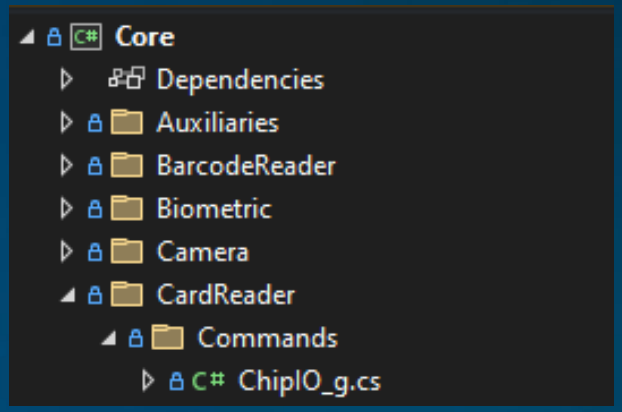
- Events classes used by SP implementation to raise command specific events
- ServiceProvider project constants
- Entire ServiceProvider project structure generated automatically with placeholders

```
16  namespace XFS4IoTFramework.CardReader
17  {
18      /// <summary>
19      /// Constants for only CardReader framework assembly
20      /// </summary>
        35 references
21      internal static class Constants
22      {
23          public const string Framework = "Framework";
24          public const string DeviceName = "CardReader";
25          public const string DeviceClass = "DevClass";
26      }
27  }
```

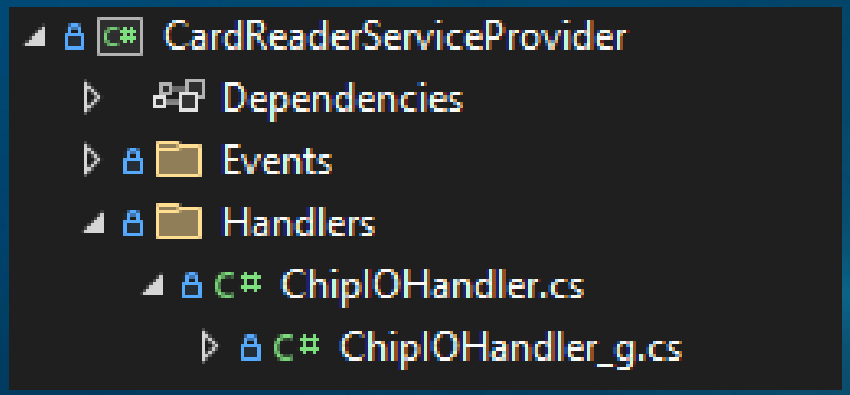


Determining if a file is generated in the Framework

- It ends with _g.cs
- It has the generated copyright header
- It is hidden under a manually edited partial class in Visual Studio
- Don’t edit generated files manually!




```
/**************************************************************************************\
 * (C) KAL ATM Software GmbH, 2025
 * KAL ATM Software GmbH licenses this file to you under the MIT license.
 * See the LICENSE file in the project root for more information.
 *
 * This file was created automatically as part of the XFS4IoT CardReader interface.
 * ChipIOHandler_g.cs uses automatically generated parts.
\**************************************************************************************/
```



Adding context from XFS4IoT Specification

```
/// <summary>
/// Track 1 of the magnetic stripe will be read.
/// </summary>
[DataMember(Name = "track1")]
4 references
public bool? Track1 { get; init; }

/// <summary>
/// Track 2 of the magnetic stripe will be read.
/// </summary>
[DataMember(Name = "track2")]
3 references
public bool? Track2 { get; init; }

/// <summary>
/// Track 3 of the magnetic stripe will be read.
/// </summary>
[DataMember(Name = "track3")]
3 references
public bool? Track3 { get; init; }

/// <summary>
/// The chip will be read.
/// </summary>
[DataMember(Name = "chip")]
3 references
public bool? Chip { get; init; }
```

```yaml
components:
  messages:
    CardReader.ReadRawDataCommand:
      version: "2.0"
      payload:
        type: object
        minProperties: 1
        properties:
          track1:
            description: |-
              Track 1 of the magnetic stripe will
            type: boolean
            default: false
          track2:
            description: |-
              Track 2 of the magnetic stripe will
            type: boolean
            default: false
          track3:
            description: |-
              Track 3 of the magnetic stripe will
            type: boolean
            default: false
          chip:
            description: |-
              The chip will be read.
            type: boolean
            default: false
```

DataTypes attribute

- Constraints taken from the specification

- Covers
  - Patterns
  - Min/Max Length
  - Min/Max value
  - Sensitive

```
[DataContract]
5 references
public sealed class CardDataClass
{
    1 reference
    public CardDataClass(CardDataStatusEnum? Status = null, List<byte> Data = null)
    {
        this.Status = Status;
        this.Data = Data;
    }

    [DataMember(Name = "status")]
    1 reference
    public CardDataStatusEnum? Status { get; init; }

    /// <summary>
    /// Base64 encoded representation of the data. This property is null if not read.
    /// <example>QmFzZTY0IGVuY29kZWQg ...</example>
    /// </summary>
    [DataMember(Name = "data")]
    [DataTypes(Pattern = @"^[A-Za-z0-9+/]+={0,2}$", Sensitive = true)]
    1 reference
    public List<byte> Data { get; init; }

}
```

Sensitive DataType

- Included in XFS4IoT Specification 2024-03
- For properties which may contain data which should not be logged
- E.g. CardReader track data

- Ilogger exposes a ISensitiveDataFormatter interface which can be used to control how these types are logged
- LogSensitive, WarningSensitive, TraceSensitive all call the SensitiveInterpolatedStringHandler

What's next?

What’s next?

- Framework updates and roadmap
- Migrating to new specification versions
- DK
- More guest speakers
- More demos (biometrics, and more)

Next call

Zoom

- First Tuesday of each month at 1300 UK time for 30 mins

Next call: 4th March 2025
1300 UK, 0800 US EST, 2200 Tokyo time

Calls are 30 mins long

We will continue to use Zoom
(Interpretation in Japanese, Chinese and Spanish is available using Zoom's interpretation feature)