```
// SelectCardData.cpp - SELECT CARD DATA
//
// MODULE INDEX
// NAME                         CONTENTS
// AnalyseSource                Analyse a source
// AnalyseSelection             Analyse a selection
// SelectCardData               Select card data
//
// MAINTENANCE HISTORY
// DATE         PROGRAMER AND DETAILS
// 18-09-13     JS      Original
//
//--------------------------------------------------------------------------

#include <string>               // C++ string declarations
using namespace std;            // Expand the standard namespace
#include "SelectCardData.h"     // Declarations

//--------------------------------------------------------------------------

// FORWARD DECLARATION OF ANALYSE SELECTION

CardData_t
AnalyseSelection (
        const char      *&p,                    // Parsing pointer
        const CardData_t *m1CardData,           // Magnetic track 1 data
        const CardData_t *m2CardData,           // Magnetic track 2 data
        const CardData_t *icCardData,           // IC with contacts data
        const CardData_t *rfCardData);          // RF identification data

//--------------------------------------------------------------------------

// ANALYSE A SOURCE

CardData_t
AnalyseSource (
        const char      *&p,                    // Parsing pointer
        const CardData_t *m1CardData,           // Magnetic track 1 data
        const CardData_t *m2CardData,           // Magnetic track 2 data
        const CardData_t *icCardData,           // IC with contacts data
        const CardData_t *rfCardData)           // RF identification data
{
        CardData_t      source;                 // Selected source

        while (*p == ' ') p ++ ;
        if (*p == 'M') {
                p ++ ;
                if (*p == '1')
                        source = *m1CardData;
                else if (*p == '2')
                        source = *m2CardData;
                else
                        goto ParameterStringError;
                p ++ ;
        } else if (*p == 'I') {
                p ++ ;
                if (*p != 'C') goto ParameterStringError;
                p ++ ;
                source = *icCardData;
        } else if (*p == 'R') {
```

```
                p ++ ;
                if (*p != 'F') goto ParameterStringError;
                p ++ ;
                source = *rfCardData;
        } else if (*p == '(') {
                p ++ ;
                source = AnalyseSelection (p, m1CardData, m2CardData,
                        icCardData, rfCardData);
                while (*p == ' ') p ++ ;
                if (*p != ')') goto ParameterStringError;
                p ++ ;
        } else {
                goto ParameterStringError;
        }
        return source;

ParameterStringError:
        source.errorCode = PARAMETER_STRING_ERROR;
        source.recording = "";
        return source;
}

//------------------------------------------------------------------------------

// ANALYSE A SELECTION

CardData_t
AnalyseSelection (
        const char      *&p,                    // Parsing pointer
        const CardData_t *m1CardData,           // Magnetic track 1 data
        const CardData_t *m2CardData,           // Magnetic track 2 data
        const CardData_t *icCardData,           // IC with contacts data
        const CardData_t *rfCardData)           // RF identification data
{
        char            operatorCode;           // Operator code
        CardData_t      firstOperand;           // First operand
        CardData_t      secondOperand;          // Second operand

        firstOperand = AnalyseSource (p, m1CardData, m2CardData,
                icCardData, rfCardData);
        while (*p == ' ') p ++ ;
        while (*p == '|' || *p == '&') {
                operatorCode = *p;
                p ++ ;
                secondOperand = AnalyseSource (p, m1CardData, m2CardData,
                        icCardData, rfCardData);
                if (
                        firstOperand.errorCode == NO_ERROR &&
                        secondOperand.errorCode == NO_ERROR
                ) {
                        if (
                                operatorCode == '&' &&
                                firstOperand.recording!=secondOperand.recording
                        ) {
                                firstOperand.errorCode = DATA_MISMATCH_ERROR;
                                firstOperand.recording = "";
                        }
                }
                else if (secondOperand.errorCode == NO_ERROR) {
                        firstOperand = secondOperand;
                }
```

```
                while (*p == ' ') p ++ ;
        }
        return firstOperand;
}

//------------------------------------------------------------------------

// SELECT CARD DATA

CardData_t
SelectCardData (
        const char      *parameterString,       // Parameter string
        const CardData_t *m1CardData,           // Magnetic track 1 data
        const CardData_t *m2CardData,           // Magnetic track 2 data
        const CardData_t *icCardData,           // IC with contacts data
        const CardData_t *rfCardData)           // RF identification data
{
        const char      *p;                     // Parsing pointer
        CardData_t      cardData;               // Selected card data

        p = parameterString;
        cardData = AnalyseSelection (p, m1CardData, m2CardData,
                icCardData, rfCardData);
        while (*p == ' ') p ++ ;
        if (*p != '\0') {
                cardData.errorCode = PARAMETER_STRING_ERROR;
                cardData.recording = "";
        }
        return cardData;
}
```