CODE MAGUS

Code Magus NIP/FIP Pattern Generator Program Version 1

CML00095-01

Code Magus Limited (England reg. no. 4024745)
Number 6, 69 Woodstock Road
Oxford, OX2 6EY, United Kingdom
www.codemagus.com

Business Partner IBM

December 13, 2013

Contents

1 Introduction

The NIP/FIP Pattern Generator is a supplementary tool to the Eresia Visual Test Environment[4], as well as, the Network Injection[3] and File Injection Portals [2].

The concepts and theory surrounding the pattern are explained in the Pattern manual A Design Pattern for Flexible Script Solutions Using the Eresia FIP, NIP and XIP Portals [5].

The NIP/FIP Pattern Generator takes advantage of the pattern concepts and technologies provided by Code Magus and fully automates the generation of, in most cases, readily useable solutions.

Given the metadata and optionally a sample file, the NIP/FIP Pattern Generator is able to generate a full and working solution in a matter of seconds.

2 Code Magus NIP/FIP Pattern Generator **cmlptngn**

2.1 Introduction

The cmlptngn tool makes use of an objtypes [1] definition and optionally, an input file mapped by an objtypes [1] definition to produce a pattern-based solution.

2.2 Processing

The generated solution will be either a NIP type or FIP type solution, which are capable of producing either network messages or files respectively, depending on which style is selected.

2.3 Command line parameters

```
Usage: cmlptngn <access>(<object>[,<options>]) ...
  -p, --package-style={nip|fip}                      Style of the package
  -s, --solution-directory={|<solution-directory>}   The script solution
                                                     directory

  -j, --formats-directory={|<formats-directory>}     The metadata directory
  -n, --solution-name={|<solution-name>}             Solution name
  -g, --drive-letter={|<drive-letter>}               The target disk drive
  -k, --base-directory={|<base-directory>}           The base directory for
                                                     the solution

  -l, --name-map-style={single|ntuple}               Name-map generation style
  -a, --type-name={|<objtype-name>}                  Type name for name-map
  -o, --output-file={stdout|<output file>}           Optional output file name
  -r, --recv-edit={|<recv edit>}                     Optional buffer edit
                                                     receive edit

  -q, --send-edit={|<send edit>}                     Optional buffer edit send
                                                     edit

  -t, --objtypes={|<objtypes-name>}                  Name of objtypes member
                                                     for buffer typing

  -c, --access-method={binary|text|edit}             The access method for
                                                     file IO in fip solutions

  -e, --am-options={|<am-options>}                   The options for the
                                                     access method

  -v, --verbose                                      List symbol table and
                                                     parsed copybooks

Help options
  -?, --help                                         Show this help message
  --usage                                            Display brief usage
                                                     message
```

where:

- `-p, --package-style={nip|fip}`
  Style of the package
- `-s, --solution-directory={|<solution-directory>}`
  The script solution directory

- `-j, --formats-directory={|<formats-directory>}`
  The metadata directory

- `-n, --solution-name={|<solution-name>}`
  The Solution name

- `-g, --drive-letter={|<drive-letter>}`
  The target disk drive

- `-k, --base-directory={|<base-directory>}`
  The base directory for the solution

- `-l, --name-map-style={single|ntuple}`
  The name-map generation style

- `-a, --type-name={|<objtype-name>}`
  The type name for name-map

- `-o, --output-file={stdout|<output file>}`
  Optional output file name

- `-r, --recv-edit={|<recv edit>}`
  Optional buffer edit receive edit

- `-q, --send-edit={|<send edit>}`
  Optional buffer edit send edit

- `-t, --objtypes={|<objtypes-name>}`
  Name of objtypes member for buffer typing

- `-c, --access-method={binary|text|edit}`
  The access method for file IO in fip solutions

- `-e, --am-options={|<am-options>}`
  The options for the access method

- `-v, --verbose`
  List symbol table and parsed copybooks

3 Example

3.1 Solution Generation

Using the layouts provided in the specification Proprietary Financial Record Layout for Training and Demonstration Purposes [6], the `cmlptngn` program was able to generate the solution in a few seconds.

The only pre-configured artefact were the objtypes definitions and a sample file.

The following command was used create a ready-to-run FIP style solution.

```
cmlptngn --solution-directory=CMLDemoSettlement ^
         --formats-directory=CMLDemoSettlementFormats ^
         --solution-name=CMLStlmnt ^
         --package-style=fip ^
         --access-method=binary ^
         --am-options="recfm=f,reclen=190,mode=wb" ^
         --output-file=log.txt ^
         --objtypes=C:\CMLDemoSettlementFormats\testdata\objtypes\cml_settlement_demo.objtypes ^
         binary(C:\CMLDemoSettlement\testdata\data\cml_prop.bin,recfm=f,reclen=190,mode=rb)
```

This example will generate all required artefacts, mappings and logic for the solution to work off the shelf. The solutions are highly configurable and extensible, allowing developers to enhance the product as necessary, should they so wish.

3.2 Input Files used in the Example

3.2.1 The Input File

```
Start of File = binary(CMLS_SETTLEMENT_DEFAULT_VALUES.bin,recfm=f,mode=rb,reclen=190).

Seq = 1, Length = 190
File = binary(CMLS_SETTLEMENT_DEFAULT_VALUES.bin,recfm=f,mode=rb,reclen=190)

        00..__..__05..__..__10..__..__15..__..__20..__..__25..__..__30..
  0000: F1F5F0F1F0F0F0F13134333235323039303430 31F0F0F0F0F0F0F0F0F0F0F0F0
  0000: .1.5.0.1.0.0.0.11.4.3.2.5.2.0.9.0.4.0.1..0.0.0.0.0.0.0.0.0.0.0.0
  0032: F0F0F1C3D6C4C540D4C1C7E4E240D7D9D6C3C5E2E2C9D5C740E2E8E2E3C5D440
  0032: .0.0.1.C.O.D.E@ .M.A.G.U.S@ .P.R.O.C.E.S.S.I.N.G@ .S.Y.S.T.E.M@
  0064: 40C4D940E2D940C4D6D5C1D3C4E2D6D540F0F1F140F5F0F040F2F3F2F3404040
  0064: @ .D.R@ .S.R@ .D.O.N.A.L.D.S.O.N@ .0.1.1@ .5.0.0@ .2.3.2.3@ @ @
  0096: 4040404040404040404040404040404040403134333235323039303430 31D7
  0096: @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ 1.4.3.2.5.2.0.9.0.4.0.1..P
  0128: 999697998985A38199A840C6899581958389819340D9858396998440D381A896
  0128: .r.o.p.r.i.e.t.a.r.y@ .F.i.n.a.n.c.i.a.l@ .R.e.c.o.r.d@ .L.a.y.o
  0160: A4A340E58599A289969540F1C396848540D48187A4A240D3899489A38584
  0160: .u.t@ .V.e.r.s.i.o.n@ .1.C.o.d.e@ .M.a.g.u.s@ .L.i.m.i.t.e.d

Seq = 2, Length = 190
File = binary(CMLS_SETTLEMENT_DEFAULT_VALUES.bin,recfm=f,mode=rb,reclen=190)

        00..__..__05..__..__10..__..__15..__..__20..__..__25..__..__30..
  0000: F1F5F0F2F13130333030313039303231 30F7F1F0000000000000099999FF2F0F0F0
  0000: .1.5.0.2.11.0.3.0.0.1.0.9.0.2.1.0..7.1.0..........r.r...2.0.0.0
  0032: F0F0F0F0F0F0F0F0F0F0F0F241E20100A0452200D7F1F2F3F4D7D6E2C4D4C1D9
  0032: .0.0.0.0.0.0.0.0.0.0.0.2A..S......E."....P.1.2.3.4.P.O.S.D.M.A.R
  0064: C9D6E240D7C9E9E9C1D9C9C14040404040D9C5E2E3D9D5E340F2F540C8E4D5C4
  0064: .I.O.S@ .P.I.Z.Z.A.R.I.A@ @ @ @ @ .R.E.S.T.R.N.T@ .2.5@ .H.U.N.D
  0096: D9C5C4E3C840C1E5C54040404040404040404040E6C9C5D9C4C140E5C1D3D3C5
  0096: .R.E.D.T.H@ .A.V.E@ @ @ @ @ @ @ @ @ @ @ .W.I.E.R.D.A@ .V.A.L.L.E
  0128: E840404040404040E2C1D5C4E3D6D540404040404040404040404040F1F9F1F0
  0128: .Y@ @ @ @ @ @ @ .S.A.N.D.T.O.N@ @ @ @ @ @ @ @ @ @ @ @ @ .1.9.1.0
  0160: C4D9C4D6D4C5E2E3C9C340404040404EF2F1F3F24EF2F1F3F24EF2F1F3F2
  0160: .D.R.D.O.M.E.S.T.I.C@ @ @ @ @ N+.2.1.3.2N+.2.1.3.2N+.2.1.3.2

Seq = 3, Length = 190
File = binary(CMLS_SETTLEMENT_DEFAULT_VALUES.bin,recfm=f,mode=rb,reclen=190)

        00..__..__05..__..__10..__..__15..__..__20..__..__25..__..__30..
  0000: F1F5F0F2F23130333030313039303231 30F7F1F0000000000000099999FF2F0F0F0
  0000: .1.5.0.2.21.0.3.0.0.1.0.9.0.2.1.0..7.1.0..........r.r...2.0.0.0
  0032: F0F0F0F0F0F0F0F0F0F0F0F341E20100A0452200D7F1F2F3F6D7D6E2C4D4C1D9
  0032: .0.0.0.0.0.0.0.0.0.0.0.3A..S......E."....P.1.2.3.6.P.O.S.D.M.A.R
  0064: C9D6E240D7C9E9E9C1D9C9C14040404040D9C5E2E3D9D5E340F2F540C8E4D5C4
  0064: .I.O.S@ .P.I.Z.Z.A.R.I.A@ @ @ @ @ .R.E.S.T.R.N.T@ .2.5@ .H.U.N.D
  0096: D9C5C4E3C840C1E5C54040404040404040404040E6C9C5D9C4C140E5C1D3D3C5
  0096: .R.E.D.T.H@ .A.V.E@ @ @ @ @ @ @ @ @ @ @ .W.I.E.R.D.A@ .V.A.L.L.E
  0128: E840404040404040E2C1D5C4E3D6D540404040404040404040404040F1F9F1F0
  0128: .Y@ @ @ @ @ @ @ .S.A.N.D.T.O.N@ @ @ @ @ @ @ @ @ @ @ @ @ .1.9.1.0
  0160: C4D9C4D6D4C5E2E3C9C340404040404EF2F1F3F24EF2F1F3F24EF2F1F3F2
  0160: .D.R.D.O.M.E.S.T.I.C@ @ @ @ @ N+.2.1.3.2N+.2.1.3.2N+.2.1.3.2

Seq = 4, Length = 190
File = binary(CMLS_SETTLEMENT_DEFAULT_VALUES.bin,recfm=f,mode=rb,reclen=190)

        00..__..__05..__..__10..__..__15..__..__20..__..__25..__..__30..
  0000: F1F5F0F2F33130333030313039303231 30F7F1F0000000000000099999FF2F0F0F0
  0000: .1.5.0.2.31.0.3.0.0.1.0.9.0.2.1.0..7.1.0..........r.r...2.0.0.0
  0032: F0F0F0F0F0F0F0F0F0F0F0F441E20100A0452200D7F1F2F3F5D7D6E2C4D3E4C9
  0032: .0.0.0.0.0.0.0.0.0.0.0.4A..S......E."....P.1.2.3.5.P.O.S.D.L.U.I
  0064: C7C9E240E2E4D7D7D3C9C5E24040404040D9C5E3C1C9D3E240F2F540E2C1D3D3
  0064: .G.I.S@ .S.U.P.P.L.I.E.S@ @ @ @ @ .R.E.T.A.I.L.S@ .2.5@ .S.A.L.L
  0096: E840E2E3D9C5C5E3404040404040404040404040E6C9C5D9C4C140E5C1D3D3C5
  0096: .Y@ .S.T.R.E.E.T@ @ @ @ @ @ @ @ @ @ @ @ .W.I.E.R.D.A@ .V.A.L.L.E
  0128: E840404040404040E2C1D5C4E3D6D540404040404040404040404040F1F9F1F0
  0128: .Y@ @ @ @ @ @ @ .S.A.N.D.T.O.N@ @ @ @ @ @ @ @ @ @ @ @ @ .1.9.1.0
  0160: C4D9C4D6D4C5E2E3C9C340404040404EF2F1F3F24EF2F1F3F24EF2F1F3F2
  0160: .D.R.D.O.M.E.S.T.I.C@ @ @ @ @ N+.2.1.3.2N+.2.1.3.2N+.2.1.3.2

Seq = 5, Length = 190
File = binary(CMLS_SETTLEMENT_DEFAULT_VALUES.bin,recfm=f,mode=rb,reclen=190)

        00..__..__05..__..__10..__..__15..__..__20..__..__25..__..__30..
  0000: F1F5F0F3F0F0F0F0F0F0F0F0F0F0F0F0F9F9F97F9698004340A602F0F0F0F131
  0000: .1.5.0.3.0.0.0.0.0.0.0.0.0.0.0.0.9.9.9.".o.q..C.@ .w...0.0.0.11.
  0032: 3433323533303930343031C2C1D5D240D6C640C2C1D5D2E240404040404040F0
  0032: 4.3.2.5.3.0.9.0.4.0.1..B.A.N.K@ .O.F@ .B.A.N.K.S@ @ @ @ @ @ @ .0
  0064: F0F0F0F0F0F0F0F0F0F0F0F0F0F5F0F1F0F0F1F5F4F6F7F8F0F0F0F0F0F1F0F0
  0064: .0.0.0.0.0.0.0.0.0.0.0.0.0.5.0.1.0.0.1.5.4.6.7.8.0.0.0.0.0.1.0.0
  0096: F0F8F0F0F0F5F0F0F2F0F0F700102540045C0001245965812F00000124596581
  0096: .0.8.0.0.0.5.0.0.2.0.0.7....%.@ ..\*....$.Y.e..a/.......$.Y.e..a
  0128: 2F000001245965812FE38889A240868993854088881A24082858595408183838 5
  0128: /.......$.Y.e..a/..T.h.i.s@ .f.i.l.e@ .h.a.s@ .b.e.e.n@ .a.c.c.e
  0160: 97A385844B40C3D4D3A5F140D7D9D6C3C5E2E2C5C44040C1C3C3C5D7E3C4
  0160: .p.t.e.dK.@ .C.M.L.v.1@ .P.R.O.C.E.S.S.E.D@ @ .A.C.C.E.P.T.D
```

3.2.2 The Copybook Definitions

```
***************************************************************
*
* Code Magus Limited.
* Proprietry Data Record Layout.
*
* About: To be used with demonstrations and
*        as a training tool.
*
* Copyright (c) 2008 Code Magus Limited. All rights reserved.
* Contact: Stephen Donaldson [stephen@codemagus.com].
*
***************************************************************
*
* $Author: justin $
* $Date: 2013/12/06 16:52:23 $
* $Id: CMLDMDAT.cpy,v 1.1 2013/12/06 16:52:23 justin Exp $
* $Name:  $
* $Revision: 1.1 $
* $State: Exp $
*
* $Log: CMLDMDAT.cpy,v $
* Revision 1.1  2013/12/06 16:52:23  justin
* initial.
*
* Revision 1.7  2009/03/05 15:05:49  justin
* modified as per spec.
*
* Revision 1.6  2009/02/26 05:34:07  justin
* added binary fields
*
* Revision 1.5  2009/02/25 16:20:39  justin
* In line with spec changes.
*
* Revision 1.4  2009/02/23 16:33:58  justin
* in line with spec.
*
* Revision 1.3  2009/02/18 14:19:04  justin
* updated.
*
* Revision 1.2  2009/02/18 11:51:50  justin
* updated.
*
* Revision 1.1  2009/02/18 10:21:39  justin
* Initial.
*
*
***************************************************************
*
 01  CMLDATAR.
*
     03  TRANSACTION-CODE                       PIC 9(4).
*
*begin attribute TRANSACTION-CODE["DESCRIPTION"].
* Field: Transaction Code.
*
* The transaction code will be 1502 - Code Magus Data Record.
*
*end attribute.
*
         88  DATA-RECORD                        VALUE 1502.
*
     03  TRANSACTION-CLASSIFIER                 PIC 9(1).
*
```

```
*begin attribute TRANSACTION-CLASSIFIER ["DESCRIPTION"].
* Field: Transaction Classifier.
*
* The class of the given transaction.
*
* Values     Description
*
*  1         Payment Record
*  2         Repayment Record
*  3         Reject Record
*
*end attribute.
*
         88  PAYMENT-RECORD                     VALUE 1.
*
         88  REPAYMENT-RECORD                   VALUE 2.
*
         88  REJECT-RECORD                      VALUE 3.
*
     03  PROCESS-TIME                            PIC 9(6)
                                       COLLATING IS ASCII.
*
*attribute set PROCESS-TIME["MASK"]="HHMMSS".
*
*begin attribute PROCESS-TIME["DESCRIPTION"].
* Field: Process Time.
*
* The time at which the settlment file
* was processed.
*
*end attribute.
*
     03  PROCESS-DATE                            PIC 9(6)
                                       COLLATING IS ASCII.
*
*attribute set PROCESS-DATE["MASK"]="YYMMDD".
*
*begin attribute PROCESS-DATE["DESCRIPTION"].
* Field: Process Date.
*
* The date on which the settlment file
* was processed.
*
*end attribute.
*
     03  CURRENCY-CODE                           PIC 9(3).
*
*begin attribute CURRENCY-CODE["DESCRIPTION"].
* Field: Currency Code.
*
* A ISO three byte numeric currency code of the
* transaction.
*
*end attribute.
*
     03  TRANSACTION-AMOUNT                      PIC 9(15) COMP-3.
*
*begin attribute CURRENCY-CODE["DESCRIPTION"].
* Field: Transaction Amount.
*
* The transaction amount.
*
*end attribute.
*
     03  TRANSACTION-DECIMALIZATION              PIC 9(1).
*
```

```
*begin attribute TRANSACTION-DECIMALIZATION["DESCRIPTION"].
* Field: Transaction Decimalization.
*
* The number of decimal places implied in the transaction
* amount.
*
*end attribute.
*
     03  MESSAGE-NUMBER                          PIC 9(15).
*
*begin attribute MESSAGE-NUMBER["DESCRIPTION"].
* Field: Message Number.
*
* The sequential message number within the
* current file.
*
*end attribute.
*
     03  UNIQUE-ID                               PIC 9(6) COMP.
*
*begin attribute UNIQUE-ID["DESCRIPTION"].
* Field: Unique Identifier.
*
* The unique identifier assigned by the terminal to
* trace the record through the life cycle.
*
*end attribute.
*
     03  ENCRYPTION-KEY                          PIC X(8) COMP-X.
*
*begin attribute ENCRYPTION-KEY["DESCRIPTION"].
* Field: Encryption Key
*
* A public encryption key for the file.
*
*end attribute.
*
     03  TERMINAL-DESCRIPTION-DATA.
*
*begin attribute TERMINAL-DESCRIPTION-DATA["DESCRIPTION"].
* Field: Terminal Description Data.
*
* This field describes the device on which the
* original transaction was performed.
*
*end attribute.
*

         05  TERMINAL-ID                         PIC X(5).
*
*begin attribute TERMINAL-ID["DESCRIPTION"].
* Field: Terminal Identifier.
*
* This field describes the institution assigned terminal
* identifier on which the transaction was performed.
*
* This field may not be space filled or all zeroes.
*
*end attribute.
*
         05  TERMINAL-TYPE                       PIC X(4).
*
*begin attribute TERMINAL-TYPE["DESCRIPTION"].
* Field: Terminal Type.
*
* This field describes the type of terminal on which
* the transaction was performed.
*
* Values     Description
*
* POSD       Point-of-Sale Terminal.
* ATMD       ATM Machine.
* 3270       3270 Terminal.
*
*end attribute.
*
              88  POINT-OF-SALE                 VALUE "POSD".
*
              88  ATM-MACHINE                   VALUE "ATMD".
*
              88  3270-TERMINAL                 VALUE "3270".
*
     03  MERCHANT-DESCRIPTION-DATA.
*
*begin attribute MERCHANT-DESCRIPTION-DATA["DESCRIPTION"].
* Field: Merchant Description Data.
*
* This field describes the merchant at which
* the original transaction was performed.
*
*end attribute.
*
         05  MERCHANT-NAME                      PIC X(20).
*
*begin attribute MERCHANT-NAME["DESCRIPTION"].
* Field: Merchant Name.
*
* This field contains the name of the merchant at which
* transaction was performed.
*
*end attribute.
*
         05  MERCHANT-CLASSIFIER                 PIC X(8).
*
*begin attribute MERCHANT-CLASSIFIER["DESCRIPTION"].
* Field: Merchant Classifier.
*
* This field describes the class of merchant at which
* transaction was performed.
*
* Values     Description
*
* RESTRNT    Restaurant Merchant.
* RETAILS    Retail Merchant.
* AIRLINE    Airline Merchant.
*
*end attribute.
*
              88  RESTAURANT-MERCHANT           VALUE "RESTRNT".
*
              88  RETAIL-MERCHANT               VALUE "RETAILS".
*
              88  AIRLINE-MERCHANT              VALUE "AIRLINE".
*
         05  MERCHANT-ADDRESS.
*
*begin attribute MERCHANT-ADDRESS["DESCRIPTION"].
* Field: Merchant Address.
*
* The physical address of the merchant.
*
* This field contains the following subfields:
```

```
* Street Address
* Suburb
* Town/City
* Postal Code.
*
*end attribute.
*
             07  MERCH-STREET-ADDRESS           PIC X(27).
             07  MERCH-SUBURB                   PIC X(20).
             07  MERCH-TOWN                     PIC X(20).
             07  MERCH-POSTAL-CODE              PIC X(4).



     03   DEBIT-CREDIT-INDICATOR                PIC X(2).
*
*begin attribute DEBIT-CREDIT-INDICATOR["DESCRIPTION"].
* Field: Debit or Credit Indicator.
*
* Indicates whether the net effect of the transaction is a
* debit or credit. Valid Values are as follows:
*
* "DR" = Debit Transaction
* "CR" = Credit Transaction
*
*end attribute.
*

     03   LOCAL-INTERNATIONAL-FLAG              PIC X(13).
*
*begin attribute LOCAL-INTERNATIONAL-FLAG["DESCRIPTION"].
* Field: Local or International Flag.
* Flag indicating whether the transaction is local or
* international. Valid Values are as follows:
*
* "INTERNATIONAL" = International Transaction
* "DOMESTIC"      = Local or Domestic Transaction
*
*end attribute.
*
          88  INTERNATIONAL-TRAN
                                        VALUE "INTERNATIONAL".
          88  LOCAL-DOMESTIC-TRAN
                                        VALUE "DOMESTIC".


     03   ACCOUNT-SCORE                         PIC +99v99.
*
*begin attribute ACCOUNT-SCORE["DESCRIPTION"].
* Field: Account Score.
*
* A score assigned by crediting agencies to determine the
* credit worthiness of the account holder. The format of
* this field is as follows:
*
* * The first byte of this field must contain a "+" or "-".
* * The next two bytes of this field must contain an integer
*   between 01 and 99.
* * The next part of this field must contain a floating point.
* * The last two bytes of this field must contain an integer
*   between 00 and 99.
*
*end attribute.
*
```

```
     03   POINTS-SCORE                          PIC +99v99.
*
*begin attribute POINTS-SCORE["DESCRIPTION"].
* Field: Points Score.
*
* A score assigned by banking institution to determine the
* credit worthiness of the account holder. The format of
* this field is as follows:
*
* * The first byte of this field must contain a "+" or "-".
* * The next two bytes of this field must contain an integer
*   between 01 and 99.
* * The next part of this field must contain a floating point.
* * The last two bytes of this field must contain an integer
*   between 00 and 99.
*
*end attribute.
*



     03  ACCOUNT-POINT-RATIO                    PIC +99v99.
*
*begin attribute ACCOUNT-POINT-RATIO["DESCRIPTION"].
* Field: Account Point Ratio.

* This field is the ratio between fields 14 and 15, which
* represents the ratio between the banking and credit agency
* assigned score to determine the overall credit worthiness
* of the account holder.
*
* * The first byte of this field must contain a "+" or "-".
* * The next two bytes of this field must contain an integer
*   between 01 and 99.
* * The next part of this field must contain a floating point.
* * The last two bytes of this field must contain an integer
*   between 00 and 99.
*
*end attribute.
*

***************************************************************
*
* Code Magus Limited.
* Proprietry Header Record Layout.
*
* About: To be used with demonstrations and
*        as a training tool.
*
* Copyright (c) 2008 Code Magus Limited. All rights reserved.
* Contact: Stephen Donaldson [stephen@codemagus.com].
*
***************************************************************
*
* $Author: justin $
* $Date: 2013/12/06 16:52:23 $
* $Id: CMLDMHDR.cpy,v 1.1 2013/12/06 16:52:23 justin Exp $
* $Name:  $
* $Revision: 1.1 $
* $State: Exp $
*
* $Log: CMLDMHDR.cpy,v $
* Revision 1.1  2013/12/06 16:52:23  justin
* initial.
*
* Revision 1.5  2009/02/27 07:22:31  justin
* in line with specification.
*
* Revision 1.4  2009/02/25 16:20:39  justin
* In line with spec changes.
*
* Revision 1.3  2009/02/18 14:19:04  justin
* updated.
*
* Revision 1.2  2009/02/18 11:51:50  justin
* updated.
*
* Revision 1.1  2009/02/18 10:21:39  justin
* Initial.
*
*
***************************************************************
*
 01  CMLHEADR.
*
     03  TRANSACTION-CODE                       PIC 9(4).
*
*begin attribute TRANSACTION-CODE["DESCRIPTION"].
* Field: Transaction Code.
*
* The transaction code will be 1501 - Code Magus Header Record.
*
*end attribute.
*
         88  HEADER-RECORD                     VALUE 1501.
*
     03  BATCH-NUMBER                            PIC 9(4).
*
*begin attribute BATCH-NUMBER["DESCRIPTION"].
* Field: Batch Number.
*
* The batch number for the current day.
* This must be the same batch number
* as in the file trailer.
*end attribute.
*
     03  PROCESS-TIME                            PIC 9(6)
                                       COLLATING IS ASCII.
*
*attribute set PROCESS-TIME["MASK"]="HHMMSS".
*
*begin attribute PROCESS-TIME["DESCRIPTION"].
* Field: Process Time.
*
* The time at which the settlment file
* was created.
*
*end attribute.
*
     03  PROCESS-DATE                            PIC 9(6)
                                       COLLATING IS ASCII.
*
*attribute set PROCESS-DATE["MASK"]="YYMMDD".
*
*begin attribute PROCESS-DATE["DESCRIPTION"].
* Field: Process Date.
*
* The date on which the settlment file
* was created.
*
*end attribute.
*
```

```
     03  MESSAGE-NUMBER                          PIC 9(15)
                                                 VALUE 1.
*
*begin attribute MESSAGE-NUMBER["DESCRIPTION"].
* Field: Message Number.
*
* The sequential message number within the
* current file.
*
* For the header record this must be equal to 1.
*
*end attribute.
*
     03  PROCESSING-SYSTEM                       PIC X(30).
*
*begin attribute PROCESSING-SYSTEM["DESCRIPTION"].
* Field: Processing System
*
* A Description of the system on which this file is
* being processed.
*
*end attribute.
*
     03  FILE-SUBMITTED-BY                       PIC X(50).
*
*begin attribute FILE-SUBMITTED-BY["DESCRIPTION"].
* Field: File Submitted By
*
* The details of the person who submitted the file in the
* event of problems arising in which the file could not be
* processed.
*
*end attribute.
*
     03  SUBMISSION-TIME                        PIC 9(6)
                                       COLLATING IS ASCII.
*
*attribute set SUBMISSION-TIME["MASK"]="HHMMSS".
*
*begin attribute SUBMISSION-TIME["DESCRIPTION"].
* Field: Submission Time.
*
* The time at which the settlment file
* was submitted.
*
*end attribute.
*
     03  SUBMISSION-DATE                       PIC 9(6)
                                       COLLATING IS ASCII.
*
*attribute set SUBMISSION-DATE["MASK"]="YYMMDD".
*
*begin attribute SUBMISSION-DATE["DESCRIPTION"].
* Field: Submission Date.
*
* The date on which the settlment file
* was submitted.
*
*end attribute.
*
     03  SOFTWARE-VERSION                        PIC X(45).
*         VALUE "Proprietary Financial Record Layout Version 1".
*
*begin attribute SOFTWARE-VERSION["DESCRIPTION"].
* Field: Software Version
* The software version implied by the file. For this
* specification the software version must contain the
* following:
* "Proprietary Financial Record Layout Version 1"
*
*end attribute.
*
     03  SOFTWARE-VENDOR                        PIC X(18).
*                               VALUE "Code Magus Limited".
*
*begin attribute SOFTWARE-VENDOR["DESCRIPTION"].
* Field: Software Vendor
*
* The software vendor that supplied the software to process
* files of this specification. The software vendor must
* contain the following: "Code Magus Limited"
*
*end attribute.
*



****************************************************************
*
* Code Magus Limited.
* Proprietry Trailer Record Layout.
*
* About: To be used with demonstrations and
*        as a training tool.
*
* Copyright (c) 2008 Code Magus Limited. All rights reserved.
* Contact: Stephen Donaldson [stephen@codemagus.com].
*
****************************************************************
*
* $Author: justin $
* $Date: 2013/12/06 16:52:23 $
* $Id: CMLDMTLR.cpy,v 1.1 2013/12/06 16:52:23 justin Exp $
* $Name:  $
* $Revision: 1.1 $
* $State: Exp $
*
* $Log: CMLDMTLR.cpy,v $
* Revision 1.1  2013/12/06 16:52:23  justin
* initial.
*
* Revision 1.6  2009/02/26 05:27:35  justin
* *** empty log message ***
*
* Revision 1.5  2009/02/25 16:20:39  justin
* In line with spec changes.
*
* Revision 1.4  2009/02/23 16:33:58  justin
* in line with spec.
*
* Revision 1.3  2009/02/18 14:19:04  justin
* updated.
*
* Revision 1.2  2009/02/18 11:51:50  justin
* updated.
*
* Revision 1.1  2009/02/18 10:21:39  justin
* Initial.
*
*
****************************************************************
*
```

```
 01  CMLTRLRR.
*
     03  TRANSACTION-CODE                       PIC 9(4).
*
*begin attribute TRANSACTION-CODE["DESCRIPTION"].
* Field: Transaction Code.
*
* The transaction code will be 1503 - Code Magus Trailer Record.
*
*end attribute.
*
         88  TRAILER-RECORD                     VALUE 1503.
*
     03  RECORD-TOTAL                           PIC 9(15).
*
*begin attribute RECORD-TOTAL["DESCRIPTION"].
* Field: Record Total.
*
* The total number of records in the file.
* This total includes the header and trailer
* record.
*
*end attribute.
*
     03  CHECKSUM-TOTAL                          PIC 9(15) COMP.

*
*begin attribute CHECKSUM-TOTAL["DESCRIPTION"].
* Field: Checksum Total.
*
* The sum total of all transactions amounts
* for each data record in the file.
*
*end attribute.
*
     03  BATCH-NUMBER                            PIC 9(4).
*
*begin attribute BATCH-NUMBER["DESCRIPTION"].
* Field: Batch Number.
*
* The batch number for the current day.
* This must be the same batch number
* as in the file header.
*end attribute.
*
     03  PROCESS-TIME                            PIC 9(6)
                                       COLLATING IS ASCII.
*
*attribute set PROCESS-TIME["MASK"]="HHMMSS".
*
*begin attribute PROCESS-TIME["DESCRIPTION"].
* Field: Process Time.
*
* The time at which the settlment file
* was processed.
*
*end attribute.
*
     03  PROCESS-DATE                            PIC 9(6)
                                       COLLATING IS ASCII.
*
*attribute set PROCESS-DATE["MASK"]="YYMMDD".
*
*begin attribute PROCESS-DATE["DESCRIPTION"].
* Field: Process Date.
*
```

```
* The date on which the settlment file
* was processed.
*
*end attribute.
*
     03  INSTITUTION-ID                          PIC X(20).
*
*begin attribute INSTITUTION-ID["DESCRIPTION"].
* Field: Institution Identifier.
*
* The identifier of the institution for
* which the file was created.
*
*end attribute.
*
     03  MESSAGE-NUMBER                          PIC 9(15).
*
*begin attribute MESSAGE-NUMBER["DESCRIPTION"].
* Field: Message Number.
*
* The sequential message number within the
* current file.
*
*end attribute.
*
     03  COMPANY-REGISTRATION-NO               PIC 9(10).
*
*begin attribute COMPANY-REGISTRATION-NO["DESCRIPTION"].
* Field: Company Registration Number.
*
*  The registration number of the company that generated
*  the current file.
*
*end attribute.
*
     03  FILE-TOTAL-DEBITS                      PIC 9(10).
*
*begin attribute FILE-TOTAL-DEBITS["DESCRIPTION"].
* Field: File Total Debits.
*
* The sum total of the transaction amounts of records whose
* debit or credit indicator field contains the value "DR".
*
*end attribute.
*
     03  FILE-TOTAL-CREDITS                     PIC 9(10).
*
*begin attribute FILE-TOTAL-CREDITS["DESCRIPTION"].
* Field: File Total Credits.
*
* The sum total of the transaction amounts of records whose
* debit or credit indicator field contains the value "CR".
*
*end attribute.
*
     03  FILE-BALANCE-TOTAL                      PIC S9(10)
                                                     COMP-3.
*
*begin attribute FILE-BALANCE-TOTAL["DESCRIPTION"].
* Field: File Balance Total
*
* The difference between the File Total Debit and File Total
* Credit Fields contained in the file trailer record.
*
*end attribute.
*
```

```
     03  TOTAL-PAYMENT-RECORDS                   PIC 9(12)
                                                     COMP-3.
*
*begin attribute TOTAL-PAYMENT-RECORDS["DESCRIPTION"].
* Field: Total Payment Records
*
* The number of Data Records with Transaction Classifier = 1
* (Payment Record) present in the file.
*
*end attribute.
*
     03  TOTAL-REPAYMENT-RECORDS                 PIC 9(15)
                                                     COMP-3.
*
*begin attribute TOTAL-REPAYMENT-RECORDS["DESCRIPTION"].
* Field: Total Repayment Records
*
* The number of Data Records with Transaction Classifier = 2
* (Repayment Record) present in the file.
*
*end attribute.
*
     03  TOTAL-REJECT-RECORDS                    PIC 9(15)
                                                     COMP-3.
*
*begin attribute TOTAL-REJECT-RECORDS["DESCRIPTION"].
* Field: Total Repayment Records
*
* The number of Data Records with Transaction Classifier = 2
* (Repayment Record) present in the file.
*
*end attribute.
*
     03  FILE-ACCEPT-REJECT-MSG                  PIC X(46).
*
*begin attribute FILE-ACCEPT-REJECT-MSG["DESCRIPTION"].
* Field: File Accepted or Rejected Message.
*
* A message indicating whether or not the file has been
* accepted by the system. The processing system will
* populate this field.
*
*end attribute.
*
     03  ACCEPT-REJECT-CODE                      PIC X(7).
*
*begin attribute ACCEPT-REJECT-CODE["DESCRIPTION"].
* Field:  Accept/Reject Code.
*
* A code indicating the status of the file. Valid Values
* are :
* "ACCEPTD" - Accepted
* "REJECTD" - Rejected
*
*end attribute.
*
        88 ACCEPTED                             VALUE "ACCEPTD".
*
        88 REJECTED                             VALUE "REJECTD".
```

3.2.3 The Object Types Definition

```
path "C:/CMLDemoSettlementFormats/testdata/copybooks/%s.cpy";
options ebcdic, omit_fillers;

--------------------------------------------------------------------------------
-- Code Magus Limited.
-- Proprietary Object Types.
--
-- About: To be used with demonstrations and
--        as a training tool.
--
-- Copyright (c) 2008 Code Magus Limited. All rights reserved.
-- Contact: Stephen Donaldson [stephen@codemagus.com].
--
--------------------------------------------------------------------------------

type CML_PROP_HEADER_RECORD
   title "Code Magus Limited: Proprietary Header Record Layout"
   book CMLDMHDR
   map CMLHEADR
      include CMLHEADR
      when (CMLHEADR.TRANSACTION_CODE = 1501);


type CML_PROP_DATA_RECORD_PAYMENT
   title "Code Magus Limited: Proprietary Data Record Layout - Payment"
   book CMLDMDAT
   map CMLDATAR
      include CMLDATAR
      when (CMLDATAR.TRANSACTION_CODE = 1502)
       and (CMLDATAR.TRANSACTION_CLASSIFIER = 1);

type CML_PROP_DATA_RECORD_REPAYMENT
   title "Code Magus Limited: Proprietary Data Record Layout - Repayment"
   book CMLDMDAT
   map CMLDATAR
      include CMLDATAR
      when (CMLDATAR.TRANSACTION_CODE = 1502)
       and (CMLDATAR.TRANSACTION_CLASSIFIER = 2);

type CML_PROP_DATA_RECORD_REJECT
   title "Code Magus Limited: Proprietary Data Record Layout - Reject"
   book CMLDMDAT
   map CMLDATAR
      include CMLDATAR
      when (CMLDATAR.TRANSACTION_CODE = 1502)
       and (CMLDATAR.TRANSACTION_CLASSIFIER = 3);


type CML_PROP_TRAILER_RECORD
   title "Code Magus Limited: Proprietary Trailer Record Layout"
   book CMLDMTLR
   map CMLTRLRR
      include CMLTRLRR
      when (CMLTRLRR.TRANSACTION_CODE = 1503);
```

3.3 Complete listing of generated solution

3.3.1 The Solution Package

```
package CMLStlmnt;
{ preamble }

   created by 'Pattern Generator';
   description 'Generate an CMLStlmnt File';
   date 2005-08-02T09:31:28;
   target 'Eresia File Portal';
   usecase GenFile : PackageScripts.CMLStlmnt;

begin

   {Set up non-optional initial parameters}
   typespath     := System.Root_Directory # System.Defined_Names.PackageTypes
                      # "cml_settlement_demo.objtypes";
   apdfile       := System.Root_Directory # System.Defined_Names.PackageScripts
                      # "CMLStlmnt.apd";
   access_method := "binary";
   am_options    := "recfm=f,reclen=190,mode=wb";

   {Create The File}
   GenFile(TYPESPATH     := typespath,
           ACCESS_METHOD := access_method,
           AM_OPTIONS    := am_options,
           APDFILE       := apdfile);
end.
```

3.3.2 The Solution Applparms Defintion

```
application CMLStlmnt;

   title "CMLStlmnt - File Generation.";
   description "The CMLStlmnt file generation script suite, "
               "facilitating the creation of CMLStlmnt files. "
     ;

   set LOGHOME = ${USERPROFILE};
   set TODAY = ${DATE_YYYYMMDD};
   set NOW = ${TIME_HHMMSS};
   set CODEMAGUS_APPLPARMS_OutputFileName_PATH="C:\CMLDemoSettlement\testdata\data\";
   set CODEMAGUS_APPLPARMS_InputSpreadsheet_PATH="C:\CMLDemoSettlement\testdata\spreadsheets\";

   store ${LOGHOME};

   interface ${CODEMAGUS_ERESIA_ROOT}"\lib\aptabgui.dll";
   entry aptabgui_init;

   parameter InputSpreadsheet
      title "Input Spreadsheet";
      default NULL;
      options filename;
      description
         "This is the name of the Excel spreadsheet that will be "
         "responsible for supplying input data to the current run "
         "of the script.  The spreadsheet consists of executable "
         "test definitions and scenarios that are the driving force "
         "in the creation of the online requests. ";
      constraint "^[^ ]\+$";
   end

   parameter OutputFileName
      title "Output File Name";
      default ${CODEMAGUS_APPLPARMS_OutputFileName_PATH}"GENFILE_D"${TODAY}"_T"${NOW}".bin";
      options filename;
      description
         "This is the name of the output file that will be "
         "created as a result of a success execution of this script. "
         "The file will be in a format suitable for directly copying "
         "into the concantenation of your file process.";
      constraint "^[^ ]\+$";
   end

   parameter PerRecordProcessingStepOne
      title "Perform Per Record Processing Step One.";
      default "YES";
      description
         "This option allows the user to decide whether "
         "or not they would like to use the logical cards "
         "stored as .icc or .mag files, or if not, populate "
         "all card details themselves. ";
      constraint "\(^YES$\|^NO$\)";
   end

   parameter PerRecordProcessingStepTwo
      title "Perform Per Record Processing Step Two.";
      default "YES";
      description
         "This option allows the user to decide whether "
         "or not the post process scripts should be invoked. "
         "The post process scripts are responsible for computing "
         "lengths, as well as adhoc tasks. ";
      constraint "\(^YES$\|^NO$\)";
   end

end.
```

3.3.3 The Solution Thistle Configuration

```
DefaultRootDirectory=C:\
PackageScripts=CMLDemoSettlement\testdata\scripts\usecases\
PackageSheets=CMLDemoSettlement\testdata\spreadsheets\
Binaries=bin\
PackageConfigs=CMLDemoSettlementFormats\testdata\configs\
PackageData=CMLDemoSettlement\testdata\data\
PackageTypes=CMLDemoSettlementFormats\testdata\objtypes\
LUHNCheckDigitScripts=LUHNCheckDigit\testdata\scripts\
CurrencyScripts=Currency\
CommonScripts=common\scripts\
CARDBASE=CardFiles\
```

3.3.4 The Solution Control Script

```
usecase CMLStlmnt(
          TYPESPATH     := "",
          ACCESS_METHOD := "",
          AM_OPTIONS    := "",
          APDFILE       := "");

{ preamble }
   created by 'Pattern Generator';
   description 'Create a CMLStlmnt file';
   date 2005-08-04T10:51:18;
   target 'Eresia File Portal';
   interface Portal.Types : CodeMagus.Types;
   interface Portal.Excel : CodeMagus.Excel;
   interface Recio : CodeMagus.RECIO;
   interface ApplParms : CodeMagus.ApplParms;
   Library CMLStlmntLib : PackageScripts.CMLStlmntLibrary;
   usecase GetColumn : CommonScripts.GetNextColumn;
   usecase GetHeaders : CommonScripts.GetHeaders;
   {Spreadsheet IO Management}
   usecase GetColumnlist : CommonScripts.SheetMgmtGetColumnlist;
   usecase GetNextInputField : CommonScripts.SheetMgmtGetNextInputField;
   usecase GetNextOutputField : CommonScripts.SheetMgmtGetNextOutputField;
   usecase GetNextCompareField : CommonScripts.SheetMgmtGetNextCompareField;
   usecase GetResultField : CommonScripts.SheetMgmtGetResultField;
   usecase CompareFields : CommonScripts.SheetMgmtCompareFields;
   usecase SheetMgmtInit : CommonScripts.SheetMgmtInit;
   usecase SheetMgmtReInit : CommonScripts.SheetMgmtReInit;
   usecase CML_PROP_DATA_RECORD_PAYMENT_INST
        : PackageScripts.CML_PROP_DATA_RECORD_PAYMENT_REC;
   usecase CML_PROP_DATA_RECORD_REPAYMENT_INST
        : PackageScripts.CML_PROP_DATA_RECORD_REPAYMENT_REC;
   usecase CML_PROP_TRAILER_RECORD_INST
        : PackageScripts.CML_PROP_TRAILER_RECORD_REC;
   usecase CML_PROP_HEADER_RECORD_INST
        : PackageScripts.CML_PROP_HEADER_RECORD_REC;
   usecase CML_PROP_DATA_RECORD_REJECT_INST
        : PackageScripts.CML_PROP_DATA_RECORD_REJECT_REC;
   {Post Processing Scripts - Process 1}
   usecase CML_PROP_DATA_RECORD_PAYMENT_PROC_ONE
        : PackageScripts.CML_PROP_DATA_RECORD_PAYMENT_POST_PROCESS_ONE;
   usecase CML_PROP_DATA_RECORD_REPAYMENT_PROC_ONE
        : PackageScripts.CML_PROP_DATA_RECORD_REPAYMENT_POST_PROCESS_ONE;
   usecase CML_PROP_TRAILER_RECORD_PROC_ONE
        : PackageScripts.CML_PROP_TRAILER_RECORD_POST_PROCESS_ONE;
   usecase CML_PROP_HEADER_RECORD_PROC_ONE
        : PackageScripts.CML_PROP_HEADER_RECORD_POST_PROCESS_ONE;
   usecase CML_PROP_DATA_RECORD_REJECT_PROC_ONE
        : PackageScripts.CML_PROP_DATA_RECORD_REJECT_POST_PROCESS_ONE;
   {Post Processing Scripts - Process 2}
   usecase CMLStlmnt_PROC_TWO_DEFAULT
        : PackageScripts.CMLStlmnt_POST_PROCESS_TWO_DEFAULT;


begin
   stream      := Recio.Connect();
   types       := Portal.Types.Connect(TYPESPATH);

   {Set up optional initial parameters}
   ParameterData := ApplParms.Connect();
   ParameterData.open(apdfile,"/VERBOSE");

   OutputFileName    := ParameterData.OutputFileName;
   InputSpreadsheet  := ParameterData.InputSpreadsheet;
```

```
  PostProcOne       := ParameterData.PerRecordProcessingStepOne;
  PostProcTwo       := ParameterData.PerRecordProcessingStepTwo;

  {Write all applparm values to the log}
  System.WriteLn("Applparms configuration for current run:\n");
  System.WriteLn("OutputFileName       : " # ParameterData.OutputFileName);
  System.WriteLn("InputSpreadsheet     : " # ParameterData.InputSpreadsheet);
  System.WriteLn("PostProcOne          : " # ParameterData.PerRecordProcessingStepOne);
  System.WriteLn("PostProcTwo          : " # ParameterData.PerRecordProcessingStepTwo);
  System.WriteLn("\n");

  {Connect to Excel}
  [thisInstance].TESTDATA := Portal.Excel.Connect(InputSpreadsheet);

  {Define the Spreadsheet worksheet name Here, so that scripts can be more generalized}
  InputWorkSheet := "Sheet1";

  {Get Header Information}
  HeaderColumns :=
         GetHeaders(TESTDATA.WorkSheet.[InputWorkSheet],"MASKLIST","TYPECOLUMN","STARTCOLUMN");

  open_spec_str := ACCESS_METHOD # "(" # OutputFileName # "," # AM_OPTIONS # ")";

  {Open the output stream to write the file}
  outputstream := stream.open(open_string:=open_spec_str,mode:="SEQ_OUTPUT",flags:="/VERBOSE");

  IndexFrom   := TESTDATA.WorkSheet.[InputWorkSheet].B[7];
  IndexTo     := TESTDATA.WorkSheet.[InputWorkSheet].B[8];

  {Initialize the Spreadsheet manangement routines}
  SheetMgmtInit(InputWorkSheet, HeaderColumns);

  {GLOBALS}
  {-- Define global variables here. counters, sequence numbers etc --}

  {An empty structure to house all definition data}
  CurrDef := "";

  CML_PROP_DATA_RECORD_PAYMENT_STRUC
     := CML_PROP_DATA_RECORD_PAYMENT_INST();
  CML_PROP_DATA_RECORD_REPAYMENT_STRUC
     := CML_PROP_DATA_RECORD_REPAYMENT_INST();
  CML_PROP_TRAILER_RECORD_STRUC
     := CML_PROP_TRAILER_RECORD_INST();
  CML_PROP_HEADER_RECORD_STRUC
     := CML_PROP_HEADER_RECORD_INST();
  CML_PROP_DATA_RECORD_REJECT_STRUC
     := CML_PROP_DATA_RECORD_REJECT_INST();

  {Load Aliases From Library Artifact}
  CML_PROP_DATA_RECORD_PAYMENT_MAPPING
     := CMLStlmntLib.CML_PROP_DATA_RECORD_PAYMENT_FIELD_MAP();
  CML_PROP_DATA_RECORD_REPAYMENT_MAPPING
     := CMLStlmntLib.CML_PROP_DATA_RECORD_REPAYMENT_FIELD_MAP();
  CML_PROP_TRAILER_RECORD_MAPPING
     := CMLStlmntLib.CML_PROP_TRAILER_RECORD_FIELD_MAP();
  CML_PROP_HEADER_RECORD_MAPPING
     := CMLStlmntLib.CML_PROP_HEADER_RECORD_FIELD_MAP();
  CML_PROP_DATA_RECORD_REJECT_MAPPING
     := CMLStlmntLib.CML_PROP_DATA_RECORD_REJECT_FIELD_MAP();

 {Load Tables From Library Artifact}
  ALIAS      := CMLStlmntLib.ALIAS_TABLE();
  POST_ALIAS_ONE := CMLStlmntLib.POST_PROCESS_ALIAS_TABLE_ONE();
  POST_ALIAS_TWO := CMLStlmntLib.POST_PROCESS_ALIAS_TABLE_TWO();
```

```
{Get necessary data from the spreadsheet as many times as necessary}
for Index := IndexFrom to IndexTo do
begin

    status := TESTDATA.WorkSheet.[InputWorkSheet].A[Index];

    {If the row is definition collect the data and     }
    {allow it to persist until a new definition occurs }
    if (status = "COLUMNLIST") then begin
      CurrDef := GetColumnlist(TESTDATA, Index);
    end

    if (status = "EXECUTABLE") then
    begin
         {determine the transaction type and specialization applied}
         specialization := CurrDef.TypeName;
         desc           := CMLStlmntLib.GetBaseType(specialization);
         {apply the correct field mapping}
         fieldmapping   := desc # "_MAPPING";

       repeat
          SheetField := GetNextInputField(TESTDATA, Index);

            {Start pre-evaluation spreadsheet items}
             {-- insert pre-evaluation items --}
             {End pre-evaluation items}

          if SheetField.name <> "" then
             [ALIAS[desc]][[fieldmapping][SheetField.name]] :=
                      SheetField.value;

            {Start post-evaluation}
             {-- insert post-evaluation items --}
            {End post-evaluation}
        until (SheetField.name = "");

       {Apply Post Processing Rules.}

       if (PostProcOne = "YES") then begin
         [ALIAS[desc]] := [POST_ALIAS_ONE[desc]]([ALIAS[desc]]);
       end
       if (PostProcTwo = "YES") then begin
         [ALIAS[desc]] := [POST_ALIAS_TWO[desc]]([ALIAS[desc]]);
       end

       {Re-initialize the manangement routines}
       SheetMgmtReInit(InputWorkSheet, HeaderColumns);

       {
        Spreadsheet second application - The spreadsheet needs to be reapplied
        so that any processing rules that overwrote something from the spreadsheet
        can be re-instated.  That way if some negative scenarios need creating, it
        will be possible to do so
       }

       repeat
          SheetField := GetNextInputField(TESTDATA, Index);

          {Start pre-evaluation spreadsheet items}
             {-- insert pre-evaluation items --}
          {End pre-evaluation items}

          if SheetField.name <> "" then
             [ALIAS[desc]][[fieldmapping][SheetField.name]] :=
                      SheetField.value;
             {Start post-evalution}
                {-- insert post-evaluation items --}
             {End post-evalution}
          until (SheetField.name = "");

          data.buffer := types.GetBuffer([ALIAS[desc]], desc);
          stream.write(outputstream, data.buffer);

       end{end if EXECUTABLE}
    end{for}

   {Reload Default Values}
    CML_PROP_DATA_RECORD_PAYMENT_STRUC := CML_PROP_DATA_RECORD_PAYMENT_INST();
    CML_PROP_DATA_RECORD_REPAYMENT_STRUC := CML_PROP_DATA_RECORD_REPAYMENT_INST();
    CML_PROP_TRAILER_RECORD_STRUC := CML_PROP_TRAILER_RECORD_INST();
    CML_PROP_HEADER_RECORD_STRUC := CML_PROP_HEADER_RECORD_INST();
    CML_PROP_DATA_RECORD_REJECT_STRUC := CML_PROP_DATA_RECORD_REJECT_INST();

end.
```

3.3.5 The Solution Library

```
Library CMLStlmntLibrary;

   Created by ’Pattern Generator’;
   Description ’CMLStlmnt Library Artifact’;
   Date 2007-01-08T09:20:29;
   Target ’ALPHA’;

method CML_PROP_TRAILER_RECORD_FIELD_MAPPING();
begin
   { Field map for: Code Magus Limited: Proprietary Trailer Record Layout }
   MAP[’TRANSACTION_CODE’] := ’CMLTRLRR.TRANSACTION_CODE’;
   MAP[’RECORD_TOTAL’] := ’CMLTRLRR.RECORD_TOTAL’;
   MAP[’CHECKSUM_TOTAL’] := ’CMLTRLRR.CHECKSUM_TOTAL’;
   MAP[’BATCH_NUMBER’] := ’CMLTRLRR.BATCH_NUMBER’;
   MAP[’PROCESS_TIME’] := ’CMLTRLRR.PROCESS_TIME’;
   MAP[’PROCESS_DATE’] := ’CMLTRLRR.PROCESS_DATE’;
   MAP[’INSTITUTION_ID’] := ’CMLTRLRR.INSTITUTION_ID’;
   MAP[’MESSAGE_NUMBER’] := ’CMLTRLRR.MESSAGE_NUMBER’;
   MAP[’COMPANY_REGISTRATION_NO’] := ’CMLTRLRR.COMPANY_REGISTRATION_NO’;
   MAP[’FILE_TOTAL_DEBITS’] := ’CMLTRLRR.FILE_TOTAL_DEBITS’;
   MAP[’FILE_TOTAL_CREDITS’] := ’CMLTRLRR.FILE_TOTAL_CREDITS’;
   MAP[’FILE_BALANCE_TOTAL’] := ’CMLTRLRR.FILE_BALANCE_TOTAL’;
   MAP[’TOTAL_PAYMENT_RECORDS’] := ’CMLTRLRR.TOTAL_PAYMENT_RECORDS’;
   MAP[’TOTAL_REPAYMENT_RECORDS’] := ’CMLTRLRR.TOTAL_REPAYMENT_RECORDS’;
   MAP[’TOTAL_REJECT_RECORDS’] := ’CMLTRLRR.TOTAL_REJECT_RECORDS’;
   MAP[’FILE_ACCEPT_REJECT_MSG’] := ’CMLTRLRR.FILE_ACCEPT_REJECT_MSG’;
   MAP[’ACCEPT_REJECT_CODE’] := ’CMLTRLRR.ACCEPT_REJECT_CODE’;
  return MAP;
end;

method CML_PROP_DATA_RECORD_REJECT_FIELD_MAPPING();
begin
   { Field map for: Code Magus Limited: Proprietary Data Record Layout - Reject }
   MAP[’TRANSACTION_CODE’] := ’CMLDATAR.TRANSACTION_CODE’;
   MAP[’TRANSACTION_CLASSIFIER’] := ’CMLDATAR.TRANSACTION_CLASSIFIER’;
   MAP[’PROCESS_TIME’] := ’CMLDATAR.PROCESS_TIME’;
   MAP[’PROCESS_DATE’] := ’CMLDATAR.PROCESS_DATE’;
   MAP[’CURRENCY_CODE’] := ’CMLDATAR.CURRENCY_CODE’;
   MAP[’TRANSACTION_AMOUNT’] := ’CMLDATAR.TRANSACTION_AMOUNT’;
   MAP[’TRANSACTION_DECIMALIZATION’] := ’CMLDATAR.TRANSACTION_DECIMALIZATION’;
   MAP[’MESSAGE_NUMBER’] := ’CMLDATAR.MESSAGE_NUMBER’;
   MAP[’UNIQUE_ID’] := ’CMLDATAR.UNIQUE_ID’;
   MAP[’ENCRYPTION_KEY’] := ’CMLDATAR.ENCRYPTION_KEY’;
   MAP[’TERMINAL_ID’] := ’CMLDATAR.TERMINAL_DESCRIPTION_DATA.TERMINAL_ID’;
   MAP[’TERMINAL_TYPE’] := ’CMLDATAR.TERMINAL_DESCRIPTION_DATA.TERMINAL_TYPE’;
   MAP[’MERCHANT_NAME’] := ’CMLDATAR.MERCHANT_DESCRIPTION_DATA.MERCHANT_NAME’;
   MAP[’MERCHANT_CLASSIFIER’] := ’CMLDATAR.MERCHANT_DESCRIPTION_DATA.MERCHANT_CLASSIFIER’;
   MAP[’MERCH_STREET_ADDRESS’]
         := ’CMLDATAR.MERCHANT_DESCRIPTION_DATA.MERCHANT_ADDRESS.MERCH_STREET_ADDRESS’;
   MAP[’MERCH_SUBURB’] := ’CMLDATAR.MERCHANT_DESCRIPTION_DATA.MERCHANT_ADDRESS.MERCH_SUBURB’;
   MAP[’MERCH_TOWN’] := ’CMLDATAR.MERCHANT_DESCRIPTION_DATA.MERCHANT_ADDRESS.MERCH_TOWN’;
   MAP[’MERCH_POSTAL_CODE’]
         := ’CMLDATAR.MERCHANT_DESCRIPTION_DATA.MERCHANT_ADDRESS.MERCH_POSTAL_CODE’;
   MAP[’DEBIT_CREDIT_INDICATOR’] := ’CMLDATAR.DEBIT_CREDIT_INDICATOR’;
   MAP[’LOCAL_INTERNATIONAL_FLAG’] := ’CMLDATAR.LOCAL_INTERNATIONAL_FLAG’;
   MAP[’ACCOUNT_SCORE’] := ’CMLDATAR.ACCOUNT_SCORE’;
   MAP[’POINTS_SCORE’] := ’CMLDATAR.POINTS_SCORE’;
   MAP[’ACCOUNT_POINT_RATIO’] := ’CMLDATAR.ACCOUNT_POINT_RATIO’;
  return MAP;
end;

method CML_PROP_DATA_RECORD_REPAYMENT_FIELD_MAPPING();
begin
```

```
   { Field map for: Code Magus Limited: Proprietary Data Record Layout - Repayment }
   MAP[’TRANSACTION_CODE’] := ’CMLDATAR.TRANSACTION_CODE’;
   MAP[’TRANSACTION_CLASSIFIER’] := ’CMLDATAR.TRANSACTION_CLASSIFIER’;
   MAP[’PROCESS_TIME’] := ’CMLDATAR.PROCESS_TIME’;
   MAP[’PROCESS_DATE’] := ’CMLDATAR.PROCESS_DATE’;
   MAP[’CURRENCY_CODE’] := ’CMLDATAR.CURRENCY_CODE’;
   MAP[’TRANSACTION_AMOUNT’] := ’CMLDATAR.TRANSACTION_AMOUNT’;
   MAP[’TRANSACTION_DECIMALIZATION’] := ’CMLDATAR.TRANSACTION_DECIMALIZATION’;
   MAP[’MESSAGE_NUMBER’] := ’CMLDATAR.MESSAGE_NUMBER’;
   MAP[’UNIQUE_ID’] := ’CMLDATAR.UNIQUE_ID’;
   MAP[’ENCRYPTION_KEY’] := ’CMLDATAR.ENCRYPTION_KEY’;
   MAP[’TERMINAL_ID’] := ’CMLDATAR.TERMINAL_DESCRIPTION_DATA.TERMINAL_ID’;
   MAP[’TERMINAL_TYPE’] := ’CMLDATAR.TERMINAL_DESCRIPTION_DATA.TERMINAL_TYPE’;
   MAP[’MERCHANT_NAME’] := ’CMLDATAR.MERCHANT_DESCRIPTION_DATA.MERCHANT_NAME’;
   MAP[’MERCHANT_CLASSIFIER’] := ’CMLDATAR.MERCHANT_DESCRIPTION_DATA.MERCHANT_CLASSIFIER’;
   MAP[’MERCH_STREET_ADDRESS’]
       := ’CMLDATAR.MERCHANT_DESCRIPTION_DATA.MERCHANT_ADDRESS.MERCH_STREET_ADDRESS’;
   MAP[’MERCH_SUBURB’] := ’CMLDATAR.MERCHANT_DESCRIPTION_DATA.MERCHANT_ADDRESS.MERCH_SUBURB’;
   MAP[’MERCH_TOWN’] := ’CMLDATAR.MERCHANT_DESCRIPTION_DATA.MERCHANT_ADDRESS.MERCH_TOWN’;
   MAP[’MERCH_POSTAL_CODE’]
       := ’CMLDATAR.MERCHANT_DESCRIPTION_DATA.MERCHANT_ADDRESS.MERCH_POSTAL_CODE’;
   MAP[’DEBIT_CREDIT_INDICATOR’] := ’CMLDATAR.DEBIT_CREDIT_INDICATOR’;
   MAP[’LOCAL_INTERNATIONAL_FLAG’] := ’CMLDATAR.LOCAL_INTERNATIONAL_FLAG’;
   MAP[’ACCOUNT_SCORE’] := ’CMLDATAR.ACCOUNT_SCORE’;
   MAP[’POINTS_SCORE’] := ’CMLDATAR.POINTS_SCORE’;
   MAP[’ACCOUNT_POINT_RATIO’] := ’CMLDATAR.ACCOUNT_POINT_RATIO’;
  return MAP;
end;

method CML_PROP_DATA_RECORD_PAYMENT_FIELD_MAPPING();
begin
   { Field map for: Code Magus Limited: Proprietary Data Record Layout - Payment }
   MAP[’TRANSACTION_CODE’] := ’CMLDATAR.TRANSACTION_CODE’;
   MAP[’TRANSACTION_CLASSIFIER’] := ’CMLDATAR.TRANSACTION_CLASSIFIER’;
   MAP[’PROCESS_TIME’] := ’CMLDATAR.PROCESS_TIME’;
   MAP[’PROCESS_DATE’] := ’CMLDATAR.PROCESS_DATE’;
   MAP[’CURRENCY_CODE’] := ’CMLDATAR.CURRENCY_CODE’;
   MAP[’TRANSACTION_AMOUNT’] := ’CMLDATAR.TRANSACTION_AMOUNT’;
   MAP[’TRANSACTION_DECIMALIZATION’] := ’CMLDATAR.TRANSACTION_DECIMALIZATION’;
   MAP[’MESSAGE_NUMBER’] := ’CMLDATAR.MESSAGE_NUMBER’;
   MAP[’UNIQUE_ID’] := ’CMLDATAR.UNIQUE_ID’;
   MAP[’ENCRYPTION_KEY’] := ’CMLDATAR.ENCRYPTION_KEY’;
   MAP[’TERMINAL_ID’] := ’CMLDATAR.TERMINAL_DESCRIPTION_DATA.TERMINAL_ID’;
   MAP[’TERMINAL_TYPE’] := ’CMLDATAR.TERMINAL_DESCRIPTION_DATA.TERMINAL_TYPE’;
   MAP[’MERCHANT_NAME’] := ’CMLDATAR.MERCHANT_DESCRIPTION_DATA.MERCHANT_NAME’;
   MAP[’MERCHANT_CLASSIFIER’] := ’CMLDATAR.MERCHANT_DESCRIPTION_DATA.MERCHANT_CLASSIFIER’;
   MAP[’MERCH_STREET_ADDRESS’]
        := ’CMLDATAR.MERCHANT_DESCRIPTION_DATA.MERCHANT_ADDRESS.MERCH_STREET_ADDRESS’;
   MAP[’MERCH_SUBURB’] := ’CMLDATAR.MERCHANT_DESCRIPTION_DATA.MERCHANT_ADDRESS.MERCH_SUBURB’;
   MAP[’MERCH_TOWN’] := ’CMLDATAR.MERCHANT_DESCRIPTION_DATA.MERCHANT_ADDRESS.MERCH_TOWN’;
   MAP[’MERCH_POSTAL_CODE’]
        := ’CMLDATAR.MERCHANT_DESCRIPTION_DATA.MERCHANT_ADDRESS.MERCH_POSTAL_CODE’;
   MAP[’DEBIT_CREDIT_INDICATOR’] := ’CMLDATAR.DEBIT_CREDIT_INDICATOR’;
   MAP[’LOCAL_INTERNATIONAL_FLAG’] := ’CMLDATAR.LOCAL_INTERNATIONAL_FLAG’;
   MAP[’ACCOUNT_SCORE’] := ’CMLDATAR.ACCOUNT_SCORE’;
   MAP[’POINTS_SCORE’] := ’CMLDATAR.POINTS_SCORE’;
   MAP[’ACCOUNT_POINT_RATIO’] := ’CMLDATAR.ACCOUNT_POINT_RATIO’;
  return MAP;
end;

method CML_PROP_HEADER_RECORD_FIELD_MAPPING();
begin
   { Field map for: Code Magus Limited: Proprietary Header Record Layout }
   MAP[’TRANSACTION_CODE’] := ’CMLHEADR.TRANSACTION_CODE’;
   MAP[’BATCH_NUMBER’] := ’CMLHEADR.BATCH_NUMBER’;
   MAP[’PROCESS_TIME’] := ’CMLHEADR.PROCESS_TIME’;
   MAP['PROCESS_DATE'] := 'CMLHEADR.PROCESS_DATE';
   MAP['MESSAGE_NUMBER'] := 'CMLHEADR.MESSAGE_NUMBER';
   MAP['PROCESSING_SYSTEM'] := 'CMLHEADR.PROCESSING_SYSTEM';
   MAP['FILE_SUBMITTED_BY'] := 'CMLHEADR.FILE_SUBMITTED_BY';
   MAP['SUBMISSION_TIME'] := 'CMLHEADR.SUBMISSION_TIME';
   MAP['SUBMISSION_DATE'] := 'CMLHEADR.SUBMISSION_DATE';
   MAP['SOFTWARE_VERSION'] := 'CMLHEADR.SOFTWARE_VERSION';
   MAP['SOFTWARE_VENDOR'] := 'CMLHEADR.SOFTWARE_VENDOR';
  return MAP;
end;

method CMLStlmnt_ALIAS_TABLE;
begin
   TABLE["CML_PROP_DATA_RECORD_PAYMENT"] := "CML_PROP_DATA_RECORD_PAYMENT_STRUC";
   TABLE["CML_PROP_DATA_RECORD_REPAYMENT"] := "CML_PROP_DATA_RECORD_REPAYMENT_STRUC";
   TABLE["CML_PROP_TRAILER_RECORD"] := "CML_PROP_TRAILER_RECORD_STRUC";
   TABLE["CML_PROP_HEADER_RECORD"] := "CML_PROP_HEADER_RECORD_STRUC";
   TABLE["CML_PROP_DATA_RECORD_REJECT"] := "CML_PROP_DATA_RECORD_REJECT_STRUC";
   return TABLE;
end;

{
  This table has been implemented in order to identify
  the correct post processing script based on the type.
}
method CMLStlmnt_POST_PROCESS_ALIAS_TABLE_ONE;
begin
   TABLE["CML_PROP_DATA_RECORD_PAYMENT"] := "CML_PROP_DATA_RECORD_PAYMENT_PROC_ONE";
   TABLE["CML_PROP_DATA_RECORD_REPAYMENT"] := "CML_PROP_DATA_RECORD_REPAYMENT_PROC_ONE";
   TABLE["CML_PROP_TRAILER_RECORD"] := "CML_PROP_TRAILER_RECORD_PROC_ONE";
   TABLE["CML_PROP_HEADER_RECORD"] := "CML_PROP_HEADER_RECORD_PROC_ONE";
   TABLE["CML_PROP_DATA_RECORD_REJECT"] := "CML_PROP_DATA_RECORD_REJECT_PROC_ONE";
   return TABLE;
end;

{
  This table has been implemented in order to identify
  the correct post processing script based on the type.
}
method CMLStlmnt_POST_PROCESS_ALIAS_TABLE_TWO;
begin

   {Default Processing Scripts}
   TABLE["CML_PROP_DATA_RECORD_PAYMENT"] := "CMLStlmnt_PROC_TWO_DEFAULT";
   TABLE["CML_PROP_DATA_RECORD_REPAYMENT"] := "CMLStlmnt_PROC_TWO_DEFAULT";
   TABLE["CML_PROP_TRAILER_RECORD"] := "CMLStlmnt_PROC_TWO_DEFAULT";
   TABLE["CML_PROP_HEADER_RECORD"] := "CMLStlmnt_PROC_TWO_DEFAULT";
   TABLE["CML_PROP_DATA_RECORD_REJECT"] := "CMLStlmnt_PROC_TWO_DEFAULT";
   return TABLE;
end;

{-- Add type specializations here --}
method CMLStlmntGetBaseType(specialization);
begin

   { -- the specialization must be a product of the type name --}
   { -- example: TYPE = XXXX, SPECIALIZATION = XXXX_YYYY --}

   RE_CML_PROP_DATA_RECORD_PAYMENT := "CML_PROP_DATA_RECORD_PAYMENT";
   RE_CML_PROP_DATA_RECORD_REPAYMENT := "CML_PROP_DATA_RECORD_REPAYMENT";
   RE_CML_PROP_TRAILER_RECORD := "CML_PROP_TRAILER_RECORD";
   RE_CML_PROP_HEADER_RECORD := "CML_PROP_HEADER_RECORD";
   RE_CML_PROP_DATA_RECORD_REJECT := "CML_PROP_DATA_RECORD_REJECT";
   if System.REMatch(specialization, RE_CML_PROP_DATA_RECORD_PAYMENT) then
      basetype := "CML_PROP_DATA_RECORD_PAYMENT";
   else if System.REMatch(specialization, RE_CML_PROP_DATA_RECORD_REPAYMENT) then
```

```
      basetype := "CML_PROP_DATA_RECORD_REPAYMENT";
   else if System.REMatch(specialization, RE_CML_PROP_TRAILER_RECORD) then
      basetype := "CML_PROP_TRAILER_RECORD";
   else if System.REMatch(specialization, RE_CML_PROP_HEADER_RECORD) then
      basetype := "CML_PROP_HEADER_RECORD";
   else if System.REMatch(specialization, RE_CML_PROP_DATA_RECORD_REJECT) then
      basetype := "CML_PROP_DATA_RECORD_REJECT";
   else begin
      System.WriteLn("Cannot map specialization or base type " # specialization # " to a known type");
      basetype := 0;
   end

   System.WriteLn("Specialization Type: " # specialization # ".  Base Type " # basetype);
   return basetype;
end;

{Export}
begin
  thisLibrary.CML_PROP_TRAILER_RECORD_FIELD_MAP := CML_PROP_TRAILER_RECORD_FIELD_MAPPING;
  thisLibrary.CML_PROP_DATA_RECORD_REJECT_FIELD_MAP := CML_PROP_DATA_RECORD_REJECT_FIELD_MAPPING;
  thisLibrary.CML_PROP_DATA_RECORD_REPAYMENT_FIELD_MAP := CML_PROP_DATA_RECORD_REPAYMENT_FIELD_MAPPING;
  thisLibrary.CML_PROP_DATA_RECORD_PAYMENT_FIELD_MAP := CML_PROP_DATA_RECORD_PAYMENT_FIELD_MAPPING;
  thisLibrary.CML_PROP_HEADER_RECORD_FIELD_MAP := CML_PROP_HEADER_RECORD_FIELD_MAPPING;
  thisLibrary.ALIAS_TABLE := CMLStlmnt_ALIAS_TABLE;
  thisLibrary.POST_PROCESS_ALIAS_TABLE_ONE := CMLStlmnt_POST_PROCESS_ALIAS_TABLE_ONE;
  thisLibrary.POST_PROCESS_ALIAS_TABLE_TWO := CMLStlmnt_POST_PROCESS_ALIAS_TABLE_TWO;
  thisLibrary.GetBaseType := CMLStlmntGetBaseType;
  return thisLibrary;
end.
```

3.3.6 The Solution Record Layout Scripts

```
usecase CML_PROP_DATA_RECORD_PAYMENT_REC();

{ preamble }

   created by 'Generator';
   description 'Code Magus Limited: Proprietary Data Record Layout - Payment';
   date 2007-06-12T16:23:41;
   target 'Eresia Network Injection Portal';

begin
   with CML_PROP_DATA_RECORD_PAYMENT do begin
      with CMLDATAR do begin
         TRANSACTION_CODE := 1502;
         TRANSACTION_CLASSIFIER := 1;
         PROCESS_TIME := 103001;
         PROCESS_DATE := 090210;
         CURRENCY_CODE := 710;
         TRANSACTION_AMOUNT := 99999;
         TRANSACTION_DECIMALIZATION := 2;
         MESSAGE_NUMBER := 000000000000002;
         UNIQUE_ID := 123457;
         ENCRYPTION_KEY := "A0452200";
         with TERMINAL_DESCRIPTION_DATA do begin
            TERMINAL_ID := "P1234";
            TERMINAL_TYPE := "POSD";
         end
         with MERCHANT_DESCRIPTION_DATA do begin
            MERCHANT_NAME := "MARIOS PIZZARIA     ";
            MERCHANT_CLASSIFIER := "RESTRNT ";
            with MERCHANT_ADDRESS do begin
               MERCH_STREET_ADDRESS := "25 HUNDREDTH AVE           ";
               MERCH_SUBURB := "WIERDA VALLEY       ";
               MERCH_TOWN := "SANDTON             ";
               MERCH_POSTAL_CODE := "1910";
            end
         end
         DEBIT_CREDIT_INDICATOR := "DR";
         LOCAL_INTERNATIONAL_FLAG := "DOMESTIC     ";
         ACCOUNT_SCORE := "+2132";
         POINTS_SCORE := "+2132";
         ACCOUNT_POINT_RATIO := "+2132";
      end
   end
   return CML_PROP_DATA_RECORD_PAYMENT;
end.
```

```
usecase CML_PROP_DATA_RECORD_REJECT_REC();

{ preamble }

   created by 'Generator';
   description 'Code Magus Limited: Proprietary Data Record Layout - Reject';
   date 2007-06-12T16:23:41;
   target 'Eresia Network Injection Portal';

begin
   with CML_PROP_DATA_RECORD_REJECT do begin
      with CMLDATAR do begin
         TRANSACTION_CODE := 1502;
         TRANSACTION_CLASSIFIER := 3;
         PROCESS_TIME := 103001;
         PROCESS_DATE := 090210;
         CURRENCY_CODE := 710;
         TRANSACTION_AMOUNT := 99999;
         TRANSACTION_DECIMALIZATION := 2;
         MESSAGE_NUMBER := 000000000000004;
         UNIQUE_ID := 123457;
         ENCRYPTION_KEY := "A0452200";
         with TERMINAL_DESCRIPTION_DATA do begin
            TERMINAL_ID := "P1235";
            TERMINAL_TYPE := "POSD";
         end
         with MERCHANT_DESCRIPTION_DATA do begin
            MERCHANT_NAME := "LUIGIS SUPPLIES     ";
            MERCHANT_CLASSIFIER := "RETAILS ";
            with MERCHANT_ADDRESS do begin
               MERCH_STREET_ADDRESS := "25 SALLY STREET            ";
               MERCH_SUBURB := "WIERDA VALLEY       ";
               MERCH_TOWN := "SANDTON             ";
               MERCH_POSTAL_CODE := "1910";
            end
         end
         DEBIT_CREDIT_INDICATOR := "DR";
         LOCAL_INTERNATIONAL_FLAG := "DOMESTIC     ";
         ACCOUNT_SCORE := "+2132";
         POINTS_SCORE := "+2132";
         ACCOUNT_POINT_RATIO := "+2132";
      end
   end
   return CML_PROP_DATA_RECORD_REJECT;
end.
```

```
usecase CML_PROP_DATA_RECORD_REPAYMENT_REC();

{ preamble }

   created by 'Generator';
   description 'Code Magus Limited: Proprietary Data Record Layout - Repayment';
   date 2007-06-12T16:23:41;
   target 'Eresia Network Injection Portal';

begin
   with CML_PROP_DATA_RECORD_REPAYMENT do begin
      with CMLDATAR do begin
         TRANSACTION_CODE := 1502;
         TRANSACTION_CLASSIFIER := 2;
         PROCESS_TIME := 103001;
         PROCESS_DATE := 090210;
         CURRENCY_CODE := 710;
         TRANSACTION_AMOUNT := 99999;
         TRANSACTION_DECIMALIZATION := 2;
         MESSAGE_NUMBER := 000000000000003;
         UNIQUE_ID := 123457;
         ENCRYPTION_KEY := "A0452200";
         with TERMINAL_DESCRIPTION_DATA do begin
            TERMINAL_ID := "P1236";
            TERMINAL_TYPE := "POSD";
         end
         with MERCHANT_DESCRIPTION_DATA do begin
            MERCHANT_NAME := "MARIOS PIZZARIA     ";
            MERCHANT_CLASSIFIER := "RESTRNT ";
            with MERCHANT_ADDRESS do begin
               MERCH_STREET_ADDRESS := "25 HUNDREDTH AVE           ";
               MERCH_SUBURB := "WIERDA VALLEY       ";
               MERCH_TOWN := "SANDTON             ";
               MERCH_POSTAL_CODE := "1910";
            end
         end
         DEBIT_CREDIT_INDICATOR := "DR";
         LOCAL_INTERNATIONAL_FLAG := "DOMESTIC     ";
         ACCOUNT_SCORE := "+2132";
         POINTS_SCORE := "+2132";
         ACCOUNT_POINT_RATIO := "+2132";
      end
   end
   return CML_PROP_DATA_RECORD_REPAYMENT;
end.
```

```
usecase CML_PROP_HEADER_RECORD_REC();

{ preamble }

   created by 'Generator';
   description 'Code Magus Limited: Proprietary Header Record Layout';
   date 2007-06-12T16:23:41;
   target 'Eresia Network Injection Portal';

begin
   with CML_PROP_HEADER_RECORD do begin
      with CMLHEADR do begin
         TRANSACTION_CODE := 1501;
         BATCH_NUMBER := 0001;
         PROCESS_TIME := 143252;
         PROCESS_DATE := 090401;
         MESSAGE_NUMBER := 000000000000001;
         PROCESSING_SYSTEM := "CODE MAGUS PROCESSING SYSTEM  ";
         FILE_SUBMITTED_BY := "DR SR DONALDSON 011 500 2323                      ";
         SUBMISSION_TIME := 143252;
         SUBMISSION_DATE := 090401;
         SOFTWARE_VERSION := "Proprietary Financial Record Layout Version 1";
         SOFTWARE_VENDOR := "Code Magus Limited";
      end
   end
   return CML_PROP_HEADER_RECORD;
end.
```

```
usecase CML_PROP_TRAILER_RECORD_REC();

{ preamble }

   created by 'Generator';
   description 'Code Magus Limited: Proprietary Trailer Record Layout';
   date 2007-06-12T16:23:41;
   target 'Eresia Network Injection Portal';

begin
   with CML_PROP_TRAILER_RECORD do begin
      with CMLTRLRR do begin
         TRANSACTION_CODE := 1503;
         RECORD_TOTAL := 000000000000999;
         CHECKSUM_TOTAL := 9999999;
         BATCH_NUMBER := 0001;
         PROCESS_TIME := 143253;
         PROCESS_DATE := 090401;
         INSTITUTION_ID := "BANK OF BANKS       ";
         MESSAGE_NUMBER := 000000000000005;
         COMPANY_REGISTRATION_NO := 0100154678;
         FILE_TOTAL_DEBITS := 0000010008;
         FILE_TOTAL_CREDITS := 0005002007;
         FILE_BALANCE_TOTAL := 102540045;
         TOTAL_PAYMENT_RECORDS := 1245965812;
         TOTAL_REPAYMENT_RECORDS := 1245965812;
         TOTAL_REJECT_RECORDS := 1245965812;
         FILE_ACCEPT_REJECT_MSG := "This file has been accepted. CMLv1 PROCESSED  ";
         ACCEPT_REJECT_CODE := "ACCEPTD";
      end
   end
   return CML_PROP_TRAILER_RECORD;
end.
```

3.3.7 The Solution Post Processing Scripts

```
usecase CML_PROP_DATA_RECORD_PAYMENT_POST_PROCESS_ONE(message:="");

{ preamble }

   created by 'Pattern Generator';
   description 'CML_PROP_DATA_RECORD_PAYMENT Post Process';
   date 2007-06-12T10:12:55;
   target 'Eresia Network Injection Portal';

begin
   {
   All post processing actions to be included in this script.
   Post processing occurs after defaults have been loaded and
   spreadsheet data has been populated.

   Post processing is any updates required just before a message
   will be sent or a record written.
   }
   return message;
end.

usecase CML_PROP_DATA_RECORD_REJECT_POST_PROCESS_ONE(message:="");

{ preamble }

   created by 'Pattern Generator';
   description 'CML_PROP_DATA_RECORD_REJECT Post Process';
   date 2007-06-12T10:12:55;
   target 'Eresia Network Injection Portal';

begin
   {
   All post processing actions to be included in this script.
   Post processing occurs after defaults have been loaded and
   spreadsheet data has been populated.

   Post processing is any updates required just before a message
   will be sent or a record written.
   }
   return message;
end.

usecase CML_PROP_DATA_RECORD_REPAYMENT_POST_PROCESS_ONE(message:="");

{ preamble }

   created by 'Pattern Generator';
   description 'CML_PROP_DATA_RECORD_REPAYMENT Post Process';
   date 2007-06-12T10:12:55;
   target 'Eresia Network Injection Portal';

begin
   {
   All post processing actions to be included in this script.
   Post processing occurs after defaults have been loaded and
   spreadsheet data has been populated.

   Post processing is any updates required just before a message
   will be sent or a record written.
   }
   return message;
end.

usecase CML_PROP_HEADER_RECORD_POST_PROCESS_ONE(message:="");
```

```
{ preamble }

   created by 'Pattern Generator';
   description 'CML_PROP_HEADER_RECORD Post Process';
   date 2007-06-12T10:12:55;
   target 'Eresia Network Injection Portal';

begin
   {
   All post processing actions to be included in this script.
   Post processing occurs after defaults have been loaded and
   spreadsheet data has been populated.

   Post processing is any updates required just before a message
   will be sent or a record written.
   }
   return message;
end.

usecase CML_PROP_TRAILER_RECORD_POST_PROCESS_ONE(message:="");

{ preamble }

   created by 'Pattern Generator';
   description 'CML_PROP_TRAILER_RECORD Post Process';
   date 2007-06-12T10:12:55;
   target 'Eresia Network Injection Portal';

begin
   {
   All post processing actions to be included in this script.
   Post processing occurs after defaults have been loaded and
   spreadsheet data has been populated.

   Post processing is any updates required just before a message
   will be sent or a record written.
   }
   return message;
end.

usecase CMLStlmnt_POST_PROCESS_TWO_DEFAULT(message:="");

{ preamble }

   created by 'Pattern Generator';
   description 'Specialization Processing Script.';
   date 2007-06-12T10:12:55;
   target 'Eresia Network Injection Portal';
   Library CMLStlmntLib   : PackageScripts.CMLStlmntLibrary;

begin
{ -- Add post processing actions here -- }
   return message;
end.
```

References

[1] **objtypes**: Configuring for Object Recognition, Generation and Manipulation. CML Document CML00018-01, Code Magus Limited, July 2008. http://www.codemagus.com/documents/objtpuref_CML0001801.pdf .

[2] Eresia File Injection Portal (FIP) Version 2.1. CML Document CML00037-21, Code Magus Limited, January 2009. http://www.codemagus.com/documents/fipusrgd_CML0003721.pdf .

[3] Eresia Network Injection Portal (NIP) Version 2.1. CML Document CML00038-21, Code Magus Limited, January 2009. http://www.codemagus.com/documents/nipusrgd_CML0003821.pdf .

[4] Code Magus Eresia User Guide Version 2.1. CML Document CML00040-21, Code Magus Limited, January 2009. http://www.codemagus.com/documents/eresiaug_CML0004021.pdf .

[5] Code Magus Limited. A Design Pattern for Flexible Script Solutions Using the Eresia FIP, NIP and XIP Portals. (CML09005-01), March 2009. http://www.codemagus.com/documents/design_pattern_training.pdf .

[6] Code Magus Limited. Proprietary Financial Record Layout for Training and Demonstration Purposes. (CML09001-01), March 2009. http://www.codemagus.com/documents/proprietary_training_specification.pdf .