

PL/I to .NET Translation Tools

PL/I to C# Translator Version 3.0 Release 26

A Visual Studio Tool

Sample Conversion from PL/I to C#

MPS Dialect Type - 7 : PL/I full set source code

```

/* CHANGE A PESSIMIST TO OPTIMIST */
/*****
/* PROGRAM NAME: OPTIMIST */
/*
/*
/* DESCRIPTION: THIS PROGRAM DEMONSTRATES PL/I CHARACTER STRING */
/* PROCESSING BY TURNING A NEGATIVE SENTENCE INTO */
/* A POSITIVE ONE. */
/*
/* INPUT: SYSTEM INPUT */
/*
/* OUTPUT: SYSTEM OUTPUT */
/*
/*
/*****
OPTIMIST: PROCEDURE OPTIONS(MAIN); /* main */
  DECLARE ARRAY_INDEX FIXED BINARY(15,0);
  DECLARE MORE_RECORDS BIT(1) INITIAL ('1'B);
  DECLARE NEGATIVE (5) CHARACTER(8) VARYING
    INITIAL (' NEVER',
            ' NONE',
            ' NOTHING',
            ' NOT',
            ' NO');
  DECLARE NO BIT (1) INITIAL ('0'B);
  DECLARE POSITIVE (5) CHARACTER(10) VARYING
    INITIAL (' ALWAYS',
            ' ALL',
            ' SOMETHING',
            '',
            ' SOME');
  DECLARE SENTENCE CHARACTER(254) VARYING;
  DECLARE START_OF_WORD FIXED BINARY(15,0);
  DECLARE VALID_CHARACTERS CHARACTER(28)
    INITIAL (' ABCDEFGHIJKLMNOPQRSTUVWXYZ. ');
  DECLARE WORD CHARACTER(32) VARYING;

  DECLARE INDEX BUILTIN;
  DECLARE LENGTH BUILTIN;

```

```

DECLARE SUBSTR          BUILTIN;
DECLARE VERIFY          BUILTIN;

/*****
/*
/*          */
/* PROGRAM NUCLEUS
/*          */
/*          */
/*          */
/*****

ON ENDFILE (SYSIN)
  MORE_RECORDS = NO;

SENTENCE = ' ';
DO WHILE (MORE_RECORDS);
  DO WHILE (SUBSTR (SENTENCE, LENGTH (SENTENCE)) ^= ' ');
    GET LIST (WORD);
    SENTENCE = SENTENCE || ' ' || WORD;
  END;
  PUT SKIP (2) LIST ('WHAT'S UP? ' || SENTENCE);
  IF VERIFY (SENTENCE, VALID_CHARACTERS) > 0
    THEN PUT SKIP LIST ('ACTUALLY, THAT'S AN INTERESTING IDEA. ');
  ARRAY_INDEX = 1;
  DO WHILE (ARRAY_INDEX <= 5);
    START_OF_WORD = INDEX (SENTENCE, NEGATIVE (ARRAY_INDEX));
    IF START_OF_WORD ^= 0
      THEN SENTENCE = SUBSTR (SENTENCE, 1, START_OF_WORD - 1)
        || POSITIVE (ARRAY_INDEX)
        || SUBSTR (SENTENCE, START_OF_WORD +
          LENGTH (NEGATIVE (ARRAY_INDEX)));
    ARRAY_INDEX = ARRAY_INDEX + 1;
  END;
  PUT SKIP LIST ('ACTUALLY, ' || SENTENCE);
  GET LIST (WORD);
  SENTENCE = ' ' || WORD;
END;
END OPTIMST;

```

PL/I to C# Translator Listing File

```

PL1CS-7 PL/I TO C# TRANSLATOR Version 3.0R26.
(c) 1990 - 2008 Micro-Processor Services Inc.
0 1 1      /* CHANGE A PESSIMIST TO OPTIMIST
*/
0 0 1  /******
0 0 2  /*      PROGRAM NAME: OPTIMIST
*/
0 0 3  /*
                                */
0 0 4  /*      DESCRIPTION:  THIS PROGRAM DEMONSTRATES PL/I CHARACTER STRING      */
0 0 5  /*      PROCESSING BY TURNING A NEGATIVE SENTENCE INTO                      */
0 0 6  /*      A POSITIVE ONE.
                                */
0 0 7  /*
                                */
0 0 8  /*      INPUT:          SYSTEM INPUT
                                */
0 0 9  /*
                                */
0 0 10 /*      OUTPUT:         SYSTEM OUTPUT
                                */
0 0 11 /*
                                */
0 0 12
/*****
0 0 13 OPTIMIST: PROCEDURE OPTIONS(MAIN); /* main */
0 1 14 DECLARE ARRAY_INDEX      FIXED BINARY(15,0);
0 1 15 DECLARE MORE_RECORDS     BIT(1)          INITIAL ('1'B);
0 1 16 DECLARE NEGATIVE (5)     CHARACTER(8) VARYING
0 1 17     INITIAL (' NEVER,
0 1 18     ' NONE,
0 1 19     ' NOTHING,
0 1 20     ' NOT,
0 1 21     ' NO);
0 1 22 DECLARE NO                BIT (1)          INITIAL ('0'B);
0 1 23 DECLARE POSITIVE (5)     CHARACTER(10) VARYING
0 1 24     INITIAL (' ALWAYS,
0 1 25     ' ALL,
0 1 26     ' SOMETHING,
0 1 27     ' ,
0 1 28     ' SOME);
0 1 29 DECLARE SENTENCE         CHARACTER(254) VARYING;
0 1 30 DECLARE START_OF_WORD    FIXED BINARY(15,0);
0 1 31 DECLARE VALID_CHARACTERS CHARACTER(28)
0 1 32     INITIAL ('ABCDEFGHIJKLMNOPQRSTUVWXYZ. );
0 1 33 DECLARE WORD             CHARACTER(32) VARYING;
0 1 34
0 1 35 DECLARE INDEX            BUILTIN;
0 1 36 DECLARE LENGTH            BUILTIN;
0 1 37 DECLARE SUBSTR           BUILTIN;
0 1 38 DECLARE VERIFY          BUILTIN;
0 1 39
0 1 40
/*****
0 1 41 /*

```

```

0 1 42  /*      PROGRAM NUCLEUS
                                */
0 1 43  /*
                                */
0 1 44
/*****/
0 1 45
0 1 46      ON ENDFILE (SYSIN)
0 1 47          MORE_RECORDS = NO;
0 1 48
0 1 49      SENTENCE = ' ;
0 1 50      DO WHILE (MORE_RECORDS);
0 2 51          DO WHILE(SUBSTR(SENTENCE,LENGTH(SENTENCE))^='.);
0 3 52              GET LIST(WORD);
0 3 53              SENTENCE = SENTENCE || ' || WORD;
0 3 54          END;
PL1CS-7 PL/I TO C# TRANSLATOR Version 3.0R26.
(c) 1990 - 2008 Micro-Processor Services Inc.
0 2 55          PUT SKIP(2) LIST ('WHAT'S UP? || SENTENCE);
0 2 56          IF VERIFY(SENTENCE, VALID_CHARACTERS) > 0
0 2 57              THEN PUT SKIP LIST('ACTUALLY, THAT'S AN INTERESTING IDEA.);
0 2 58          ARRAY_INDEX = 1;
0 2 59          DO WHILE (ARRAY_INDEX <= 5);
0 3 60              START_OF_WORD = INDEX(SENTENCE,NEGATIVE(ARRAY_INDEX));
0 3 61              IF START_OF_WORD ^= 0
0 3 62                  THEN SENTENCE = SUBSTR(SENTENCE,1,START_OF_WORD - 1)
0 3 63                      || POSITIVE(ARRAY_INDEX)
0 3 64                      || SUBSTR(SENTENCE,START_OF_WORD +
0 3 65                          LENGTH(NEGATIVE(ARRAY_INDEX)));
0 3 66              ARRAY_INDEX = ARRAY_INDEX + 1;
0 3 67          END;
0 2 68          PUT SKIP LIST ('ACTUALLY, || SENTENCE);
0 2 69          GET LIST (WORD);
0 2 70          SENTENCE = ' || WORD;
0 2 71      END;
0 1 72  END OPTIMST;
0 0 73
THERE WERE 0 ERRORS FOUND

```

PL/I to C# Translator, C# source code output

```

/* Module Name: C:\MPS\PL1C\TestData\test.pl1 */
/*Translated by PL1CS-7 PL/I to C# Translator Ver 3.0R26.*/
/*(c) 1990-2008 Micro Processor Services Date:05/17/08 Time:17:03:13 */
/*      Translator Option List (1=ON,0=OFF) :
ansii style declaration- 0  include format      - 0  temp_drv_valid      - 0
include select          - 0  index increment    - 0  move nested function - 1
complete struct member - 0  comment trans stop- 0  error by pass        - 0
indent # of spaces     - 4  include file name -' ' include file ext   -'HH'
character set          48   - 0  pound character(#)- 2
output select          - 3  when numeric      - 1  not used            -'0'
*/
using System;
using System.Collections;
using System.Text;
class test
{
/* CHANGE A PESSIMIST TO OPTIMIST */
/*****/
/* PROGRAM NAME: OPTIMIST */
/*
*/
/* DESCRIPTION: THIS PROGRAM DEMONSTRATES PL/I CHARACTER STRING */
/* PROCESSING BY TURNING A NEGATIVE SENTENCE INTO */
/* A POSITIVE ONE. */
/*
*/
/* INPUT: SYSTEM INPUT */
/*
*/
/* OUTPUT: SYSTEM OUTPUT */
/*
*/
/*****/

int Main()
{
/* main */
short array_index;
bool more_records = 0x1;
string []negative = new string[5] { " NEVER", " NONE", " NOTHING", " NOT", "
NO"};
bool no = 0x0;
string []positive = new string[5] { " ALWAYS", " ALL", " SOMETHING", "", "
SOME"};
string sentence;
short start_of_word;
string valid_characters = "ABCDEFGHIJKLMNOPQRSTUVWXYZ. ";
string word;
/*****/
/*
*/
/* PROGRAM NUCLEUS

```

```
/*
    */
    */
/*****
signal(endfile, endfile_handler);

sentence = ' ';
while(more_records){
    while(substr(sentence, strlen(sentence)) != '.'){
        Console.WriteLine(word);
        sentence = sentence + ' ' + word;
    }
    Console.WriteLine("\n\nWHAT'S UP? " + sentence);
    if (verify(sentence, valid_characters) > 0)
Console.WriteLine("\nACTUALLY, THAT'S AN INTERESTING IDEA.");

    array_index = 1;
    while(array_index <= 5){
        start_of_word = index(sentence, negative[array_index]);
        if (start_of_word != 0) sentence = substr(sentence, 1, (start_of_word
- 1) + positive[array_index] + substr(sentence, (start_of_word +
strlen(negative[array_index])));
        array_index = array_index + 1;
    }
    Console.WriteLine("\nACTUALLY, " + sentence);
    Console.WriteLine(word);
    sentence = ' ' + word;
}

void endfile_handler()
{
    more_records = no;
}

}
```