




<b>Publication Year</b>	2000
<b>Acceptance in OA</b>	2023-03-03T13:52:54Z
<b>Title</b>	A tool for an automated conversion of F77 code into F90 code
<b>Authors</b>	MARIS, Michele
<b>Handle</b>	<a href="http://hdl.handle.net/20.500.12386/33979">http://hdl.handle.net/20.500.12386/33979</a>
<b>Volume</b>	PL-LFI-OAT-TN-012

	<b>A Tool for the Automated Conversion of F77 code Into F90 Code</b>	Document No. Issue/Rev. No. Date Page	PL-LFI-OAT-TN-012 0.0 23 May 2000 1 of 12
---	--	--	--

## A Tool for the Automated Conversion of F77 Code into F90 Code

**Document No.:** PL-LFI-OAT-TN-012

**Issue/Rev. No.:** 0.0

**Date:** 2000-May-23

**Prepared by:** Michele Maris

**Authorized by:** Fabio Pasian

### Abstract

A tool written in PERL 5.001 which partially automates the porting of F77 code into F90 code is presented. Despite it allows only simple operations, the tool reduces largely the amount of trivial editing in order to transform a F77 code into a F90 code which follows the *Fortran 90 Guidelines* prescriptions.

	<b>A Tool for the Automated Conversion of F77 code Into F90 Code</b>	Document No. Issue/Rev. No. Date Page	PL-LFI-OAT-TN-012 0.0 23 May 2000 2 of 12
---	--	--	--

## Document Approval

Name	Organization	Signature	Date
F.Pasian LFI DPC Manager	OAT		2000 May 25


## Document Status Sheet

<b>1. Document Title:</b> An LFI data processor to create a Quick-Look Source Catalogue			
2. Issue	3. Revision	4. Date	5. Reason for change
N	0	2000-May-25	Initial version

	<b>A Tool for the Automated Conversion of F77 code Into F90 Code</b>	Document No. Issue/Rev. No. Date Page	PL-LFI-OAT-TN-012 0.0 23 May 2000 3 of 12
---	--	--	--

## Table of Contents

Document Approval .....	2
Document Status Sheet.....	2
Table of Contents .....	3
1. Introduction.....	4
2. The Interface .....	4
3. Verbosity.....	4
4. Operations.....	5
5. Side Effects .....	6
6. Known Bugs.....	6
7. Efficiency.....	6
8. History .....	6
Appendix: The PERL Script.....	7

	<b>A Tool for the Automated Conversion of F77 code Into F90 Code</b>	Document No. Issue/Rev. No. Date Page	PL-LFI-OAT-TN-012 0.0 23 May 2000 4 of 12
---	--	--	--

## 1. Introduction

In order to perform conversion of already written F77 code into F90, which rigorously follows the *Fortran 90 Programming Guidelines* (Maris (1999)) a lot of straightforward editing work is required. For this reason, it would be helpful to have tools which allows a reduction of the repetitive work. Such operation requires extensive but simple parsing, text analysis and patter matching of the code to be converted while speed does not represent a problem. More over, since not all the existing F77/F90 code is written rigorously, the conversion tool must be simple to be adapted to handle specific, frequent errors in the source code. For this reason it is more convenient to use a script language instead of a compiled one like C + Yac + Lex. For this project PERL is the language of choice since it is especially designed to perform textual analysis and handle lists of strings.

## 2. The Interface

In the present version the tool is based on a PERL script named: `f772f90.pl`, which may is used by the command line. The script requires in input the name of the input file and the name of the output files. File names must be different. Optionally it is possible to change the Verbosity of the tool, which by default is settled to its maximum level.

The code is started by the command line:

```
perl f772f90.pl <<InputFile>> <<OutputFile>> [<<Verbosity>>]
```

by example:

```
perl f772f90.pl fft.f fft.f90
```

## 3. Verbosity

During operation the tool will produce a message for each operation it will perform on the source code. The amount of messages is indicated by the `Verbosity` parameter according to the following description.

	<b>A Tool for the Automated Conversion of F77 code Into F90 Code</b>	Document No. Issue/Rev. No. Date Page	PL-LFI-OAT-TN-012 0.0 23 May 2000 5 of 12
---	--	--	--

**Verbosity = 0:** No messages are produced

**Verbosity = 1:** Only those input lines, which are modified, are displayed.

**Verbosity = 2:** Both input and output lines, which are modified, are displayed.

default verbosity is 2.

## 4. Operations

At the beginning of the input code a short comment block is added stating the operation performed.

The comment sounds similar to the following example:

```

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!
! This code has been formatted in F90 + Fortran 90 Guidelines style using
! f772f90.pl - 0.0 - By M. Maris - 23 May 2000 -
!
! Original Name : planck_HEALPIX_lfft_all_new_cp.f
! Final      Name : fs.f90
! Date        : Wed May 24 08:15:31 2000
! Operated by : M. Maris
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

```

Then line by line the following operations are applied to the input code:

1. **Object:** Comment lines

**Pattern:** First column character \* c or C

**Action:** First column character replaced by !

2. **Object:** Continuation characters.

**Pattern:** Individuated by \*, +, &, 0, 1, ..., 9; in column 6<sup>th</sup> or 7<sup>th</sup> preceded by 5 or 6 blanks.

**Action:** The continuation character is replaced by &.

At the first line, which is not a comment line, before the current one an & is added.

The program tries to take in account:

	<b>A Tool for the Automated Conversion of F77 code Into F90 Code</b>	Document No. Issue/Rev. No. Date Page	PL-LFI-OAT-TN-012 0.0 23 May 2000 6 of 12
---	--	--	--

+ Double quotes “ in order to avoid to consider the content of strings.

+ In order to avoid to queue & to a comment, the & is inserted between the last valid character and the “!”.

## 5. Side Effects

The only side effect up to now produced by the PERL script on the generated file is the insertion, time by time, of an extra space character at the end of the translated code line.

## 6. Known Bugs

In the present version an exclamation mark in 6<sup>th</sup> column “!” is understood as the beginning of a comment line. Since an exclamation mark in column 6<sup>th</sup> is rarely used as a continuation, presently the bug is not fixed, but the original code has to be cleaned by such continuation symbols before to proceed.

## 7. Efficiency

The speed is quite high. The conversion of a 7189 lines file with 1595 Comment Replacements, 440 Continuation Characters Replacements and 440 Continuation Characters Chaining plus I/O disk operations on a SUN Solaris machines requires 2 seconds. The error rate in the Comment Replacement zero as in performing Continuation Characters Replacements and Continuation Characters Chaining. However, especially for the second operation, it cannot be excluded a bit smaller efficiency in some exceptional situations for which the rules previously defined does not hold.

## 8. History

**March 23, 2000:** First Release of f772f90.pl and related documentation.

	<b>A Tool for the Automated Conversion of F77 code Into F90 Code</b>	Document No. Issue/Rev. No. Date Page	PL-LFI-OAT-TN-012 0.0 23 May 2000 7 of 12
---	--	--	--

## Appendix: The PERL Script

This appendix is the PERL script version 0.0 – 23 May 2000.

```

#
# f772f90.pl - 0.0 - By M. Maris - 23 May 2000 -
#
# usage f772f90.pl <<InputFile>> <<OutputFile>> [<<Verbosity>>]
#

# Starting header
$Verbosity = 2;

$Today = gmtime(time);

$Operator = '';

$Version = "f772f90.pl - 0.0 - By M. Maris - 23 May 2000 -";

print << "EOH";
$Version

usage f772f90.pl <<InputFile>> <<OutputFile>> [<<Verbosity>>]

Verbosity : 0 None
           : 1 Only input displayed
           : 2 Input/Output displayed
           default = $Verbosity

EOH

# Parameters consistency check
$InputFile = $ARGV[0]; if (length($InputFile) == 0) { die "Missing input name" };
$OutputFile = $ARGV[1]; if (length($OutputFile) == 0) { die "Missing output name" };

if ($InputFile eq $OutputFile) { die "Input file can not be equal to output file" };

if ($#ARGV == 2) {
    $Verbosity = $ARGV[2];
};

# Source scan
open FID, '<' . $InputFile;

```

	<b>A Tool for the Automated Conversion of F77 code Into F90 Code</b>	Document No. Issue/Rev. No. Date Page	PL-LFI-OAT-TN-012 0.0 23 May 2000 8 of 12
---	--	--	--

```

@Listing = <FID>;
close(FID);

# resets statistical counters
$NLineInput = $#Listing+1;
$NCommentRep = 0;
$NContinuationRep = 0;
$NContiunationChain = 0;

#
# Begin of Operations
#


# Scans for c, C or * in the first column and replaces with !
$NLine = 0;
for $NLine (0..$#Listing) {
    $key = $Listing[$NLine];
    if ($key =~ m/^[cC*]/) {
        if ($Verbosity > 0) {
            print "COMMENT REPLACED (C -> !) at column 1, Line ", $NLine+1, "\n";
            print ">>", $key;
        };
        substr($key,0,1) = '!';
        $Listing[$NLine] = $key;
        if ($Verbosity > 1) {
            print "<<", $Listing[$NLine];
        };
        $NCommentRep = $NCommentRep+1;
    };
};

# Scans for any character in the 6th column, if required replace with
# & and if absent at the end of the previous line, adds it

if ($Listing[0] =~ /\s*!/) {
    $LastNotCommentLine = 0;
} else {
    $LastNotCommentLine = 1;
};

for $l (1..$#Listing) {
    $key = $Listing[$l];

```

	<b>A Tool for the Automated Conversion of F77 code Into F90 Code</b>	Document No. Issue/Rev. No. Date Page	PL-LFI-OAT-TN-012 0.0 23 May 2000 9 of 12
---	--	--	--

```

if ($key =~ /^\s*!/) { # it is a comment line
} else {
  if ($key =~ /^[ ]{5,6}[*!.\&0-9]/) {

    if (substr($key,0,6) eq '      ') {
      $SixthChar = substr($key,6,1);
    } else {
      $SixthChar = substr($key,5,1);
    };
    if ($SixthChar ne ' ') {
      if ($Verbosity > 0) {
        print "CONTINUATION CHARACTER REPLACE & at column 6 Line ", $NLine+1, "\n";
        print ">>", $key;
      };

      substr($key,5,1) = '&';
      $Listing[$l] = $key;

      $NContinuationRep = $NContinuationRep+1;

      if ($Verbosity > 1) {
        print "<<", $Listing[$l];
      };

      $key = $Listing[$LastNotCommentLine];

      $Virgolette = 0;
      $CommentCol = -1;
      $ContinuationCol = -1;
      $SkipFlag = 0;

      substr($key,length($key)-1,1) = " ";
      $key = $key." \n";

      for $i (6..length($key)) { # Hunts for the continuation character
        if ($SkipFlag == 0) { # A termination event occurred
          if ($Virgolette == 0) { # it is not in a string
            if ( substr($key,$i,1) eq "'") {
              $Virgolette = 1;
            } else {
              if ( substr($key,$i,1) eq '!') { # Finds a comment
                $CommentCol = $i;
                $SkipFlag = 1;
              } else {

```

	<b>A Tool for the Automated Conversion of F77 code Into F90 Code</b>	Document No. Issue/Rev. No. Date Page	PL-LFI-OAT-TN-012 0.0 23 May 2000 10 of 12
---	--	--	---

```

if ( substr($key,$i,1) eq '&' ) { # Finds a F90 continuation mark

    $_ = substr($key,$i+1,length($key));

    $mark = 1;
    for $j (0..length($_)) {
        if (substr($_,$j,1) = /[!\s]/) { } else {
            $mark = 0 };
    };
    if ($mark == 0) { # the rest of line or is empty or is a comment
        $ContinuationCol = $i;
        $SkipFlag = 1;
    } else {
        $ContinuationCol = -1;
    };
};

} else { # it is in a string
    if ( substr($key,$i,1) eq '"' ) { # looks for end-of-string
        $Virgolette = 0;
    };
};

};


if ($ContinuationCol == -1) { # No continuation found, add it
    if ($Verbosity > 0) {
        print "CONTINUATION CHARACTER ADDING at Line ", $LastNotCommentLine, "\n";
        print ">>", $key;
    };

    $NContinuationChain = $NContinuationChain+1;

    if ($CommentCol > -1) {
        $left = substr($key,0,$CommentCol-1);
        $right = substr($key,$CommentCol,length($key));
    } else {
        $left = substr($key,0,length($key)-1);
        $right = substr($key,length($key)-1,1);
    };

    $final = $left.' & '.$right;
    $Listing[$LastNotCommentLine] = $final;
    if ($Verbosity > 1) {

```

	<b>A Tool for the Automated Conversion of F77 code Into F90 Code</b>	Document No. Issue/Rev. No. Date Page	PL-LFI-OAT-TN-012 0.0 23 May 2000 11 of 12
---	--	--	---

```

        print "<<", $Listing[$LastNotCommentLine];
    };
};
};
} else {
};

$LastNotCommentLine = $1; # Marks this as the last non comment line
};
};

#
# operations end
#

# writes output
open FID, '>'. $OutputFile;

print FID << "CARTIGLIO"; # Adds a note at the text beginning

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!
! This code has been formatted in F90 + Fortran 90 Guidelines style using
! $Version
!
! Original Name : $InputFile
! Final Name : $OutputFile
! Date : $Today
! Operated by : $Operator
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

CARTIGLIO

print FID @Listing;
close(FID);

print "\n<< Added a note at the beginning of the output file\n\n";

# bye bye message
print << "EOH1";
Conversion: $InputFile -> $OutputFile

```

	<b>A Tool for the Automated Conversion of F77 code Into F90 Code</b>	Document No. Issue/Rev. No. Date Page	PL-LFI-OAT-TN-012 0.0 23 May 2000 12 of 12
---	--	--	---

Relevant Statistics:

Lines : \$NLineInput  
Comment Replacements : \$NCommentRep  
Continuation Characters Replacements : \$NContinuationRep  
Continuation Characters Chained : \$NContinuationChain

EOH1