Fortran SOURCE

Lahey Computer Systems, Inc.
SOFTWARE SOLUTIONS for Science & Engineering

Volume 19, Issue 2
SUMMER 2003

Specializing in Fortran Language Systems Since 1967

Introducing Lahey/Fujitsu Fortran v7.0

Lahey/Fujitsu Fortran v7.0 integrates the new Fortran for .NET language system and our LF95 Windows Fortran language system into the latest Microsoft® Visual Studio development environment. You can take advantage of all .NET offers by creating .NET applications and services with Fortran, generate native Win32 code for ultimate performance, and work in the legendary Microsoft Visual Studio environment. We're offering LF Fortran v7.0 in four editions: Enterprise, Developer, Professional, and Educational.

LF Fortran v7.0

		Tool Groups			
		Fortran for .NET Language System	Win32 Fortran Language System	Win32 Development Tools	Visual Studio .NET
LF Fortran Edition	Enterprise	X	X	X	X
	Developer	X			X
	Professional		X	X	X
	Educational	X			

The following sections describe the four Tool Groups (Fortran for .NET Language System, Win32 Fortran Language system, Win32 Development Tools, Visual Studio .NET) that make up the four LF Fortran product editions (Enterprise, Developer, Professional, and Educational).

Fortran for .NET - .NOT. your Father's Fortran

Fortran for .NET allows Fortran users to create applications for the Microsoft .NET Framework. Lahey and Fujitsu have combined advanced compiler technology with support for Forms designers and Web Services to enable Fortran organizations to develop .NET applications with Fortran as easily as with other Microsoft .NET languages. Fortran for .NET consists of a Fortran compiler and associated tools designed to help you create applications that run in the .NET Framework.

Specifically it provides:

- A Fortran compiler that can generate verifiable .NET MSIL (Microsoft Intermediate Language) code.
- Object-oriented Fortran 2000 syntax mapped to support .NET interoperability.
- Syntax extensions to support .NET features that are not covered by Fortran 2000 syntax.
- .NET-specific options added to the compiler to configure object-file attributes.
- Support for using Fortran in ASP.NET web forms and web services.
- Support for using Fortran as an ASP.NET scripting language.
- Support for using ADO.NET design tools to create Fortran-powered database applications.
- Integration of Fortran into Visual Studio .NET.
- Support for using Windows Forms and Web Forms designers to create Fortran applications.
- Ability to build Visual Studio .NET solutions and packages incorporating Fortran.
- Full access to Microsoft's .NET architecture and APIs.
- Seamless integration with other .NET languages allows classes created with Fortran and other .NET languages to be used interchangeably.
- Debugging of mixed language applications using the Visual Studio debugger.

(*Continued on page 3*)

Dear Fortran Programmer,

The Fujitsu/Lahey Fortran for .NET release is the most important product Lahey has brought to market since our Linux Fortran, September 1999. We have great expectations for this release, not only for increasing our presence in the marketplace, but also because we believe Fortran for .NET will solve MANY of the problems we have being hearing about from you, our users.

In particular:

- No, none, nada, ... reason to learn (or buy!) another language in order to develop a GUI and in other ways develop interfaces to Windows technology.
- In concert with developing a GUI, users will be able to access and code in almost every language conceived by man, for sure every popular language (COBOL, C, C++, C#, BASIC, J# - go to Microsoft for a complete inventory). This is a BIG one! Fortran for .NET users will have access to the kind of GUI-building features that Visual Basic users have had for years.
- Some Fortran 2000 features are included, especially those that enhance interfacing to OOP technology.
- There is a way to preserve the utmost performance (create a DLL using LF95).

Ok, those are the pluses, what are the minuses?

- .NET Fortran is a proper subset of Fortran 2000 plus popular extensions. We have concentrated on the most-popular/most-efficient/most-useful features we believe exist in code that we expect will be ported to .NET.
- Existing Visual Basic interfaces don't port to .NET without some work.

While I was enjoying a latte at my local B&N, I lucked into The Post-OOP Paradigm, an article in the current issue of American Scientist by Brian Hayes that I urge you read because it includes an introduction to OOPs that is readable and educational. If you appreciate the acronym OOP, you'll be happy to know that a possible successor is PloP, Pattern Languages of Programming. The article is available at: http://www.americanscientist.org/Issues/Comsci03/03-03Hayes.html.

Last two points:

1) Depending on demand, Fujitsu/Lahey will continue to develop features until we have completed Fortran 95. Please let us know what feature(s) we should implement to make your port easier (sales@lahey.com).

2) Fortran for .NET might be the final reason you need to port that code that was written decades ago.

Stay healthy,

Tom

Tom

Lahey Computer Systems, Inc.
SOFTWARE SOLUTIONS for Science & Engineering

Lahey Computer Systems, Inc.
775-831-2500

Fax:
775-831-8123

Technical Support:
775-831-2500
support@lahey.com

Sales:
800-548-4778
775-831-2500
sales@lahey.com

Web:
www.lahey.com

Fortran SOURCE

Contributing Editor:
Bill Lassaline

Fortran for .NET - .NOT. your Father's Fortran

(Continued from page 1)

What can you do with the Fortran for .NET language?	Why would you want to?
Mix Fortran and other .NET languages in the same application.	Libraries written in other languages exist. Use the right language for the right task.
Create Windows user interfaces by dragging and dropping buttons, data entry fields, check boxes, and more.	Reduce development costs. You want more than a command-line interface.
Create web forms using the same technique.	You want to make your application and its professional user interface available to the world over the Internet.
Create reusable custom controls, with Fortran, that anyone can use in their Visual Studio environment.	You know Fortran, you've created custom controls for one application, why not reuse them? Reduce development costs.
Create procedures that access or collect data that exist somewhere over the Internet. (XML Web service.)	Somebody else already did the work of collecting the information. Make it available, accessible. Offer a service.
Create applications that call procedures that access or collect that data.	Use the data that's available. You can reduce the time to obtain answers.
Create COM objects with Fortran.	You know Fortran. Use your Fortran procedures from other, non-.NET language programs.
Call, from Fortran, COM objects written in other languages.	You have access to any COM object that's ever been written.
Create applications that run wherever .NET has been implemented.	Microsoft's .NET Framework is based on open standards and will be implemented on other platforms.
Create libraries of useful functionality w/ Fortran (i.e., class libraries) that can be used and extended by any .NET language.	Don't limit your users to only those who know Fortran.
Easily call Windows APIs from Fortran.	You know Fortran and want to use the building blocks of Windows.
Code functionality into your web pages using Fortran.	You know Fortran. It's a way to add smarts to a web page.
Easily read from and write to databases. (ADO.NET - drag and drop database access.)	It's where you keep your data.

(Continued on page 4)

News Briefs

GinoGraphics.NET

Bradly Associates Ltd (www.gino-graphics.com) is pleased to announce the first release of GinoGraphics.NET, comprising their 2D and 3D graphics libraries, GINO, GINOGRAF, and GINOSURF. The new implementation will be available as part of the new 6.0 release of GINO and will encompass a collection of .NET assemblies under the GinoGraphics namespace, with exactly the same functionality as the Win32 libraries/DLLs.

The .NET implementation has the advantage of being accessible from all .NET languages, including Fortran, VB, C#, J#, etc. using object-orientated constructs available in each language/compiler. The .NET binding exposes each function/ routine as a class method accepting standard argument types such as integer, double, .NET strings and .NET arrays. Where Fortran derived types or C structures were used in the Win32 implementations, .NET objects and arrays of objects are used.

You can easily integrate the GinoGraphics assemblies into a .NET application using Visual Studio .NET. Graphics can then be directed to independent GINO windows or for an integrated .NET GUI application, .NET picture boxes can be used with the graphics manipulated by other .NET controls as required.

GinoGraphics.NET provides a powerful resource for any .NET developer creating end-user applications or re-usable controls containing 2D and 3D graphics.

www.laheyforum.com

A new discussion forum is available at www.laheyforum.com. It offers a web-based interface to threaded discussions and the ability to search both current discussions and an archive of postings from the older (but still operational) fortran@lahey.com e-mail forum. (Note that Lahey has no plans to discontinue the fortran@lahey.com e-mail forum.)

(Continued on page 4)

(Continued from page 1)

Win32 Fortran Language System - Ultimate Performance

Lahey/Fujitsu Fortran 95 (LF95) for Windows is included in the LF Fortran Enterprise and Professional editions. In addition to doing everything you've always done with LF95, you can now develop applications in the Visual Studio environment, and when you need the fastest speed in your .NET applications, put your computational code in a Win32 DLL and call your procedures from the Fortran for .NET program.

Win32 Development Tools - The Professional Collection

The W*interacter* Starter Kit (W*i*SK) for creating Windows GUIs and displaying graphics, Polyhedron's Automake utility, Fujitsu's WinFDB Windows debugger, the Coverage Analysis Tool that detects unexecuted code and performs range-of-operation checking, and the Sampler Tool, an execution profiler that helps tune program performance, are all included in LF Fortran Enterprise and Professional.

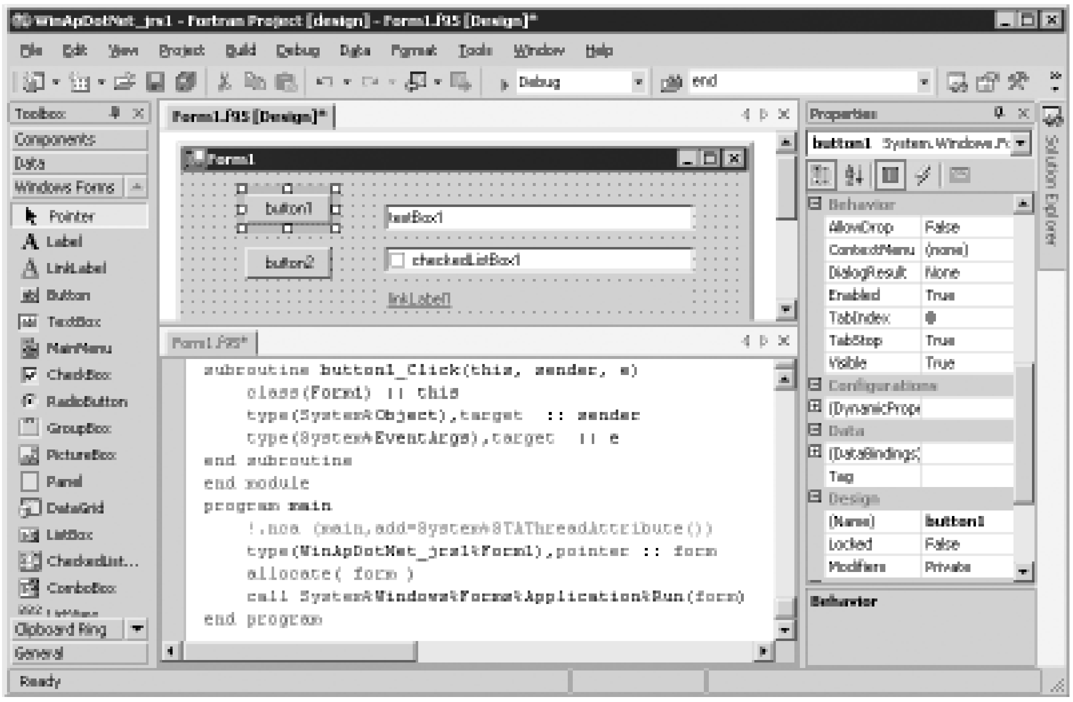
Microsoft Visual Studio - Putting it all together

Microsoft Visual Studio .NET is the environment you use to develop your applications. Note that even though ".NET" is part of the name of the new Visual Studio, it doesn't restrict you to developing only .NET applications. VS .NET includes the Windows and ASP.NET Web Forms designers, a project management system, .NET and Win32 project and code templates, and online integrated Fortran for .NET and Fortran 95 help. Menu options to control building Fortran for .NET and Win32 Fortran applications are integrated into VS .NET. The inset picture shows the Windows Form designer in VS .NET:

1 Drag & drop components and controls

2 Generate Fortran code automatically

3 Add some glue code and Go!



```
subroutine button1_Click(this, sender, e)
    class(Form1) :: this
    type(System%Object),target  :: sender
    type(System%EventArgs),target  :: e
end subroutine
end module
program main
    !.nca (main,add=System%STAThreadAttribute())
    type(WinApDotNet_jrs1%Form1),pointer :: form
    allocate( form )
    call System%Windows%Forms%Application%Run(form)
end program
```

button1 System.Windows.Fo

Behavior	
AllowDrop	False
ContextMenu	(none)
DialogResult	None
Enabled	True
TabIndex	0
TabStop	True
Visible	True
Configurations	
(DynamicProp	
Data	
(DataBindings)	
Tag	
Design	
(Name)	button1
Locked	False
Modifiers	Private

News Briefs

(Continued from page 3)

SciComm, SciPlot, and SciSnet

MicroGlyph Systems (www.microglyph.com) has three products compatible with the new LF Fortran for .NET language system. The products provide serial communications (SciComm) for instruments and devices, scientific plotting on printers, screens, or exportable disk files (SciPlot), and high-speed network data communications for real time applications (SciSnet).

Sample Fortran for .NET Code

The following source code demonstrates how an object-oriented class might be coded using some of the new features of the Fortran for .NET compiler. The code is Fortran 2000-proposed-standard-compliant.

```
! The optional module name defines the namespace.
module NamespaceName
     ! ClassName is the class prototype name.
     ! It can extend or be extended by another class.
     type, extensible :: ClassName
          ! FieldName1 is a field (global variable) of the class.
          integer :: FieldName1 = 10
          ! Color is an enumerator field of the class.
          enum :: Color
               enumerator :: Red = 1
               enumerator :: Green
          end enum
     contains    ! The class method prototypes are declared here.
          ! MethodName is declared as an instance (pass) method.
          procedure, pass :: MethodName1
          ! OverloadName is a generic name for multiple static methods.
          generic, nopass :: OverloadName => OverloadFun1,OverloadFun2
     end type ClassName
contains    ! The class methods are implemented here.
     ! Method (subroutine or function procedure)
     subroutine MethodName1(this)
          ! 'this' is used to pass the class object for instance methods.
          class(ClassName) :: this
          ...
     end subroutine MethodName1
     ! Overloaded method with one argument.
     function OverloadFun1(a)
          ...
     end function overloadfun1
     ! Overloaded method with two arguments.
     function OverloadFun2(a, b)
          ...
     end function OverloadFun2
end module NamespaceName
```

More Sample Code

You can find the following examples of Fortran for .NET programming in the trial download available at www.lahey.com:

- .NET Windows Forms - Pythagorean Triplets
- ASP.NET Web Forms - Pythagorean Triplets
- ASP.NET Web Services (Browser UI) - Conversion
- ASP.NET Web Services Client - Conversion (Client)
- ADO.NET SQL Server - Database
- ASP.NET Script Blocks - ASP Script Block
- ASP.NET Render Blocks - ASP Render Block
- ASP.NET Server Controls - ASP Code Generation
- .NET Internet Data Mining - Weather Graphs
- .NET App Calling a Win32 DLL - DotNetWrapper
- .NET Fortran Class Exposed as Unmanaged COM Object - COMobject
- .NET Fortran Calling Unmanaged COM Object - COMObjectCall
- .NET App Calling a Win32 DLL - Snowflakes
- .NET Multi-Language Interop, Multi-Threading - Sorts

Q&A

Q: What's happening to the LF95 Windows v5.7 products, ***Express***, Standard, and PRO?

A: We will continue to offer LF95 ***Express*** v5.7 for Windows. LF95 Standard v5.7 for Windows has been discontinued. LF95 PRO v5.7 for Windows is still available for those with Windows 95, 98, Me, and NT because LF Fortran v7.0 requires Windows 2000 or XP.

Q: What language features does Fortran for .NET support?

A: Most everything in Fortran 77/90/95, and more. A list of supported and unsupported language features and restrictions is included in the Fortran for .NET documentation available at www.lahey.com.

Q: There are a lot of concepts associated with .NET. Where can I go for more information?

A: Here are a few useful links and a book we've found helpful: www.microsoft.com/net, www.gotdotnet.com, msdn.microsoft.com, and *Introducing Microsoft .NET*, by David S. Platt.

Q: What hardware and OS do I need to use LF Fortran v7.0?

A: We recommend the following:

- 600-MHz Pentium III-class processor or faster
- Windows Server 2003 with 160 megabytes (MB) of RAM, Windows XP Professional with 160 MB of RAM, Windows XP Home Edition with 96 MB of RAM, Windows 2000 Professional with 96 MB of RAM, or Windows 2000 Server with 192 MB of RAM
- 900 MB of available space required on system drive, 3.3 gigabytes (GB) of available space required on installation drive
- Additional 1.9 GB of available space required for optional MSDN Library documentation
- CD-ROM or DVD-ROM drive
- Super VGA (1024 x 768) or higher-resolution display with 256 colors
- Mouse or compatible pointing device

Q: Is the .NET Framework SDK included in LF Fortran 7.0?

A: Yes.

Q: Does Fortran for .NET have COM support?

A: Fortran for .NET contains substantial COM support and allows you to easily use existing COM objects. Fortran for .NET provides a simple means of creating Fortran COM objects that can be used by any language with COM support, such as Visual Basic and Visual C++.

Q: I already have Fortran code, where do I start when changing to Fortran for .NET?

A: First, try compiling the code with the Fortran for .NET compiler, using the `-ncs` option. The `-ncs` option indicates that names are not case sensitive. It is likely that the code will compile without problems if it previously compiled without problems. Once the code compiles, evaluate how you want to use the code and to what extent you want to integrate it into the .NET scheme. The possibilities range from keeping the code as close to original as possible and calling it from wrappers, to converting it into Fortran 2000-style, object oriented class libraries. Either way, the code will be available for use as code behind Windows or Web forms, COM objects, etc.

Q: Many example Fortran for .NET programs contain a variable named "this." What is "this?"

A: The variable "this" is an artifact of object-oriented programming, and it is called the passed object. When used in an example method, the variable "this" is always of the class that the method belongs to, and it always refers to the specific instance of the class that called the method. For example, consider the following class implementation:

```
module m
  type p
    integer :: i
  contains
    procedure, pass :: proc
  end type
contains
  subroutine proc(this,j)
    type(p) :: this
    integer :: j
    this%i = j
  end subroutine
end module
```

Notice that in the type declaration statement, the method proc has the pass attribute, which means that a passed object must be declared. Procedure proc is a member of class p, and the variable this in method proc is the passed object. Now suppose that we use the class in the following program:

```
program myprog
  use m
  type(p) :: c1,c2
  call c1%proc(1)
  call c2%proc(2)
  write(*,*) c1%i, c2%i
end program
```

Notice that in the call statements, only one actual argument is passed, even though there are two dummy arguments in the subroutine itself. The passed object is automatically

(Continued on page 7)

Fortran for .NET Joins the Army

The US Army Corps of Engineers, Engineer Research and Development Center (USACE-ERDC), Airfields & Pavements Branch (APB) has incorporated over 50 years of pavements research into an integrated software suite for the structural design and evaluation of road and airfield pavements. The software package is the result of a successful effort between research civil engineers and computer scientists. The most recent version of this product was written using Visual Basic 6.0 for the graphical user interface and Fortran 77 legacy code for the computational routines. The product was delivered in January of 2003 to a tri-service committee consisting of pavements engineers from the US Army, Air Force, and Navy. We're now working on the next version.

Following the selection of Visual Studio.NET as the development platform for the next version, the team began investigating options for the legacy Fortran code that had been passed down, updated, and modified over the last 30 years. There are over 15,000 lines of Fortran code that produce a range of required engineering solutions. It was not considered cost effective to re-write that code. By reviewing articles in magazines, information at Microsoft's web site, and sessions at the VBITS conference, it became clear where the future was headed: .NET. Four features were of particular interest: 1) an end to "DLL hell," 2) true object-oriented programming, 3) zero-touch software deployment, and 4) cross-language usage of objects and code modules (without wrappers).

We used the beta version of Visual Basic.NET to rewrite a couple of small modules that were originally written in VB6. Several of the sample projects that demonstrated features of interest were evaluated. The .NET Framework exposed methods and properties previously available only to Visual C++ programmers (unless you liked playing around in the Win32 API). .NET was found to have a lot of functionality available in just a few lines of code. The big test was accessing the Fortran code from VB.NET. The complexity of existing Fortran structures (structures within structures and arrays of structures) caused a significant problem – passing the data to the Fortran code was no longer simple. The only solution found was to write the structure to a byte array and then use the System.Runtime.InteropServices to pass a pointer to the byte array to the legacy code. At this point, it became clear that we needed a .NET version of Fortran.

After exploring websites, industry news, and a visit to the Lahey booth at the 2002 VSLive conference, we decided to test Lahey's beta release of LF Fortran for .NET. We tested from two different perspectives. The first was a Fortran programmer with a thorough knowledge of Fortran, but no knowledge of objects. The second was a Visual Basic programmer with a good knowledge of objects, but little understanding of Fortran. The experienced Fortran programmer was able to quickly grasp the concept of objects and classes and the way it was implemented in LF Fortran for .NET. In less than two days he was successfully creating classes, adding properties, and overloading constructors. His only concern was execution speed. The Visual Basic programmer took the opposite approach and delved into the .NET Framework. Creating objects and placing them in collections and working with other operations were easy to learn. Learning to use ALLOCATE and the pointer attributes took a little time and effort. Some initial tech support was necessary to use System.IO. All said and done, the Lahey Fortran for .NET compiler was thoroughly tested. Features that did not work were reported and many are being corrected.

We found that with .NET there is a trade-off between speed and language and skill set compatibility. LF Fortran for .NET allows programmers to use Fortran in a method consistent with the skills that exist in programmers using object-oriented languages. Fortran for .NET offers users access to the .NET Framework objects, access to a visual interface that is easy to use, integration of Fortran code and objects with other .NET language code and objects, and an incentive to initiate the update of legacy Fortran 77 code. So, we decided on LF Fortran for .NET and development has begun. The Fortran code is being modified such that it can be compiled as a Win32 DLL or as a class library. We created an "engineering" class that accepts our objects as parameters to its methods. They are then translated to Win32 types and the updated code is compiled as a .NET library. We changed COMMON statements in order to use data blocks. Using LF Fortran for .NET was the right decision. This new Fortran may even help alleviate the attitude that Fortran is prehistoric.

Contributed by
John T. Lott, Computer Scientist, Southern Division, Applied Research Associates, Vicksburg, MS 39180-5160

Carlos R. Gonzalez, Research Civil Engineer, Geotechnical and Structures Laboratory, U.S. Army Research and Development Center, Vicksburg, MS 39180-6199

Robert S. Walker, Computer Scientist, Geotechnical and Structures Laboratory, U.S. Army Research and Development Center, Vicksburg, MS 39180-6199

Q&A

(Continued from page 6)

added by the compiler. When the first call statement is executed, the passed object this in method proc refers to the instance of class p named c1. When the second call statement is executed, the passed object this in method proc refers to the instance of class p named c2.

Lahey® Computer Systems, Inc.

SOFTWARE SOLUTIONS
for Science & Engineering

865 Tahoe Blvd.
P.O. Box 6091
Incline Village, NV 89450
U.S.A.

PRSRT STD
U.S. POSTAGE
PAID
PERMIT NO. 2310
SACRAMENTO, CA

RETURN SERVICE REQUESTED

Lahey's ***Fortran* SOURCE** newsletter–
Your source for the latest news from Lahey.

Lahey Product Information Comprehensive information on all products is available at www.lahey.com.

Language Systems
- LF Fortran Enterprise v7.0 US$1,095
- LF Fortran Enterprise v7.0 + IMSL US$1,395
- LF Fortran Developer v7.0 US$795
- LF Fortran Professional v7.0 US$795
- LF Fortran Professional v7.0 + IMSL US$1,095
- LF Fortran Educational v7.0 US$75 (Requires MSDN Academic Alliance membership)
- LF95 Windows ***Express*** US$249
- LF95 Windows ***Express*** + IMSL US$549
- LF95 Linux PRO US$695
- LF95 Linux PRO + IMSL US$995
- LF95 Linux ***Express*** US$249
- LF95 Linux ***Express*** + IMSL US$549
- LF90 US$895
- LF90 + IMSL US$1,195
- Essential LF90 US$195
- Essential LF90 + IMSL US$495

Productivity Tools
- *Winteracter* US$675
- *plus*FORT Full Kit US$995
- *plus*FORT Starter Kit US$595
- IMSL US$495
- GINO-F Super Bundle US$2,995
- GINO-F Bundle US$2,100
- GINO-F Bundle Linux US$1,495
- GINO-F US$1,425
- CALGINO US$300
- CALGINO Linux US$150
- GINOMENU US$450
- GINOMENU Studio US$450
- GINOMENU Bundle US$750
- GINO .NET US$600
- GINO .NET Bundle US$995
- VAST Fortran 77 to 90 Translator from US$450
- Essential Fortran 90 & 95 Text Book US$45
- *f90*SQL US$249
- *f90*VB US$299

Updates/Upgrades
- LF Fortran Enterprise v7.0 Update
  from LF95 Windows PRO and Standard v5.7 US$495
  from other Lahey Fortran language systems US$895
- LF Fortran Enterprise v7.0 Competitive Upgrade
  from any Fortran language system US$895
- LF Fortran Developer v7.0 Competitive Upgrade
  from any Fortran language system US$595
- LF Fortran Professional v7.0 Competitive Upgrade
  from any Fortran language system US$595
- LF95 Windows ***Express*** v5.7 Update
  from LF95 Windows ***Express*** v5.x US$149
- LF95 Linux PRO v6.2 Update
  from LF95 PRO v6.1 US$195
  from LF95 ***Express*** v6.1 US$595
- LF95 Linux ***Express*** v6.2 Update
  from LF95 ***Express*** v6.1 US$149
- Essential LF90 v4.0 Update
  from any previous version of Essential LF90 US$59

Please call for prices on other updates.

Lahey offers competitive, floating, and site-license pricing.
Educational discounts are available.
Prices do not include shipping and handling.

Call today with your order and questions.
800-548-4778 or 775-831-2500 / Fax: 775-831-8123 / sales@lahey.com

Buy Online at www.lahey.com!
Customer Satisfaction Guaranteed
Same-day shipping on most orders. 30-day money-back guarantee.

All prices in U.S. dollars.

Help us end waste. Let us know if your change your address, receive duplicate copies of Fortran SOURCE, or would like to be removed from our mailing list. Call us toll-free at 800-548-4778 or send us e-mail at sales@lahey.com. Thank you.