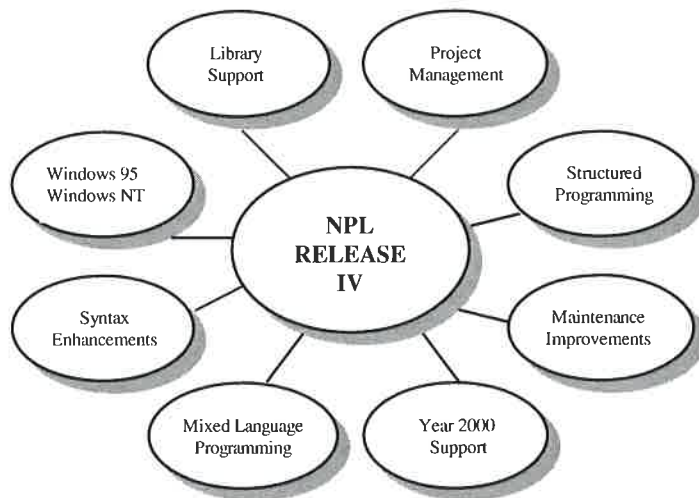


Introduction to NPL Release IV

NPL (Niakwa Programming Language) has long been known for its *source code compatibility* across most major platforms, its *capability* in handling a breadth of application needs ranging from business to agricultural to engineering and beyond, and its *stability*, which is legendary among our worldwide developer network.

The goal of NPL Release IV is to significantly improve programmer productivity and to modernize NPL while maintaining the powerful development environment, portability and high quality NPL is known for. Release IV maintains all of the strengths of prior versions of NPL while adding a collection of state-of-the-art development components to bring NPL to a new level of programming productivity.

Release IV supplies many new features and enhancements including new tools which aid in structured programming, project management, maintenance improvements, Year 2000 support, language syntax enhancements, Windows 95 & NT support and library support for incorporating third party programs.



NPL Release IV provides a wealth of benefits to NPL developers.

The features available in Release IV make NPL one of the most productive and modern languages available to serious application developers. With Release IV, you can implement a modernized application both from a developer's and end-user's viewpoint.

This data sheet provides detailed benefits of NPL Release IV.

NPL Release IV



**Your Key to
State-of-the-Art
Software Development**

Project Management

Improved project management is an area where NPL Release IV increases programmer productivity. The use of program modules and structured design can greatly reduce the difficulties of managing large scale, multi-programmer projects by allowing key functional components to be developed independently without consideration of variable and line number usage. All that needs to be agreed upon is the parameters required to access these independent routines. In addition, developers can incorporate an array of third party library products into their applications.

In addition to the wide variety of C libraries available to the NPL developer, Niakwa has a number of development tools which make extensive use of this external capability. Present product offerings such as the Niakwa Data Manager (NDM), NDM for ODBC, Visual NPL (Vinny) and future offerings such as NDM for TCP/IP are all developed with the goal of allowing developers to modernize and extend existing time tested applications.

Structured Programming

A key objective of Release IV is to provide a means for more structured programming of new applications, along with the ability to better structure existing applications. Two key components which make this improved structure possible are the introduction of **modules** and **function interface** utilities.

Modular Program Development

The module concept allows the programmer to create routines which use a unique address space for line numbers, variable names and marked subroutines. This in affect "compartmentalizes" the operations performed in the NPL work space, thus greatly simplifying integration of new and third party libraries into existing applications.

Function Interface

The function interface utilizes named identifiers as opposed to line numbers to access routines housed in a module program. The function interface extends the programmer's capability in several ways by allowing subroutines to be recursively called, allowing parameters to be passed by value or indirectly by reference and by providing a standard method for return values and error indication.

Maintenance Improvements

Maintenance and debugging are important qualities of any programming language. With NPL Release IV, maintenance and debugging efforts can be greatly reduced in a number of ways.

Long Identifier Names

With Release IV, long identifier names can be incorporated to make NPL code more readable and understandable. Liberal use of long variable names, long subroutine names and statement identifiers dramatically simplifies maintenance of program code by improving the readability and meaning of the code.

Structured Constructs

The use of **structured constructs** alleviates the jobs of maintenance and debugging by utilizing constructs such as:

- IF / ENDIF
- REPEAT / UNTIL
- SWITCH / CASE
- WHILE / WEND

The logical flow of a program becomes easier to follow and maintain.

Editing and Debugging Improvements

In addition to these language features, NPL programmers will further improve their code output when developing in the **NPL Workbench**. The NPL Workbench is a full featured editor/debugger designed to maximize a programmer's development efforts.

The combined features offered in each functional area of long identifier names, structured constructs and the NPL Workbench, bring together a rich development environment providing developers with the necessary edge required to stay competitive in today's application software market.

Year 2000 Support

Niakwa has provided a Century Date function in Release IV to assist developers in adapting their applications to handle turn of the century issues.

An NPL system library is simple to implement into new or existing routines to distinguish Year 2000 dates.

Release IV Code Example

```
0010 ;FACT - Module "FACT"
0020 FUNCTION 'Factorial (Value)/PUBLIC      ;;Declare 'Factorial as Public
0030 DIM Answer                             ;;This is a recursive variable
0040 IF Value<0                              ;;test for valid parameter
      :RETURN (1)
      :END IF
0050 IF Value>1                               ;;This ends the recursion
      :RETURN (1)
      :END IF
0060 Answer=Value* 'Factorial (Value- 1)    ;;Recursively calling itself
0070 RETURN (Answer)
0080 END FUNCTION
```

Simple NPL Module Program

Release IV Application Code using the above module

```
0010 DIM X
0020 INCLUDE T "FACT"
0030 INPUT X
0040 PRINT 'FACTORIAL (X)
```

NPL Root Module Program

Mixed Language Programming

NPL has supported external subroutine development using third party languages since Release III. NPL Release IV expands this feature in the NPL Windows environments by adding a callback facility, allowing full access to the Windows development environment.

This provides a two-way communication channel allowing Windows applications developed in other languages (such as Visual Basic or C), to fully utilize NPL routines and vice versa.

Syntax Enhancements

NPL Release IV includes the following language syntax enhancements:

- Long identifier names (up to 255 characters)
- Structured constructs
- Variables larger than 64K supported on selected platforms

- New variable types
- Mouse support for MS-DOS, Windows and 386/DOS-Extender versions
- HELP system enhancements
- Enhanced external call interface
- Century date function for Year 2000 issues
- RECORDS / FIELDS
- Statement labels
- TRUE / FALSE constants
- Library functions for use with \$SOURCE/\$OBJECT
- NDM RECORDS / FIELD specifications
- New LIST statements for functions and modules
- "Debug" pop-up window support
- Mixed network file sharing
- "Pass through" printer support for Windows
- Mixed network support

Windows 95 & NT Support

Release IV provides support for Windows 95 and Windows NT, which join the long list of fully compatible platforms NPL and your applications can run on, with only *one set of source code to maintain*.

Library Support

A Library is a collection of standard programs, routines and subroutines designed to perform a repetitive set of functions. Once complete, a single library is easily implemented into existing applications.

Library support provides the ability to create code which is reusable within the same program and can be shared among other Release IV developers using NPL's modules. This allows fast integration of new NPL products to increase functionality in less time than writing code from ground zero.

Release IV Development Tools

Current Offerings:

- **Visual NPL** - allows NPL applications to leverage the GUI screen management of Visual Basic
- **Niakwa Data Manager** - offers access to industry standard ISAMs (Btrieve, C-ISAM)
- **Scientific & Communications Drivers**
- **NPL Gateway to ODBC**
- **Third Party Libraries** (Quasar, DDE & others)
- **Shareware Libraries** - available from Niakwa's Web Site

Planned Development Tools:

- NPL Workbench (Graphical Editor/Debugger)
- Niakwa Data Manager for ODBC
- Niakwa Data Manager for TCP/IP
- Visual NDM

Upward Compatibility

Applications developed using Release III (or earlier) versions of NPL are fully upward compatible to Release IV and will execute with no changes or compilation required.

Downward Compatibility

Applications developed or modified within Release IV will operate with Release III RunTimes as long as no Release IV specific syntax is utilized.

Availability

Release IV was first made available in September 1993. Niakwa continues to make enhancements to Release IV which are released periodically in updated versions.

NPL Operating Environments

AIX (RS6000)	Novell NetWare	Super DOS
AT & T UNIX	OS/400 (AS400)	*VMS (DEC VAX)
*HP-UX (HP9000)	Personal NetWare	Windows 3.x
Interactive UNIX	SCO UNIX	Windows 95
LANtastic	SCO XENIX	Windows NT
MS-DOS	*Sun Solaris (SPARC)	Windows for Workgroups 3.11
NetBIOS	* Currently at Release III	



Niakwa Inc.
23600 North Milwaukee Avenue
Vernon Hills, Illinois 60061
(847) 634-8700. Fax (847) 634-8718

Copyright 1998 by Niakwa Inc. All rights reserved. Niakwa is a registered trademark of Niakwa. Other product names, company names and/or logos may be trademarks of their respective companies.

