

Introduction to NPL

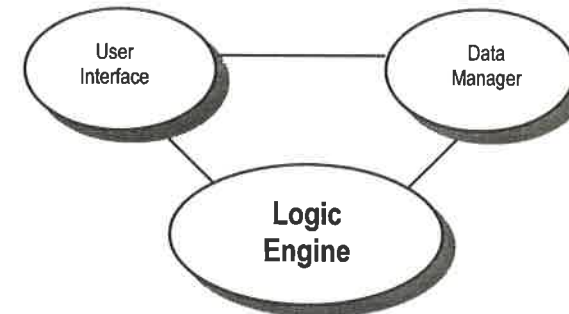
The Niakwa Programming Language (NPL) is a robust, powerful computer language designed for developers who need to quickly deploy mission-critical applications. This is accomplished by providing these essential components of the NPL Logic Engine:

- Portability
- Ease of learning and maintenance
- Programmer productivity

NPL provides portability for professional software developers to greatly expand their target markets without spending time developing and supporting different versions of source code for each environment. NPL supports source and object code compatibility across most major operating environments. This allows programs developed in one set of code to operate on a variety of hardware and operating system environments.

Because of its ease of learning and maintenance, developers (especially C and BASIC programmers) find the structured format of NPL very easy to learn and maintain. The use of Long Identifier Names (LINs) and program constructs provide programmers greater flexibility to logically name and identify variables, which allows for improved program maintenance in a minimal amount of time.

NPL significantly improves programmer productivity by providing modular technology, LINs and structured program constructs along with many other convenient time saving features.



The three components of a language; logic engine is where the processing occurs.

The NPL Logic Engine provides the essential components of the NPL Programming Language. Niakwa also offers tools for developing graphical user interfaces and managing data in industry standard formats. These are covered in separate data sheets.

This data sheet provides an orientation to the core of NPL, the Logic Engine, and a detailed description of the language.

NPL Logic Engine



**Processing Engine
for Mission Critical
Applications**

How NPL Code is Created and Executed

NPL consists of three physical software products which create and/or execute NPL pseudo-code (p-code): the interpreter, compiler and RunTime program.

The **NPL Interpreter** (an incremental compiler) creates and executes p-code and allows you to efficiently develop and maintain NPL programs. Using "Immediate Mode," the interpreter supports real-time source-code editing, syntax checking, variable inspection, variable modification, program halting, single instruction stepping and trapping, instruction logic tracing, etc.

In addition, the interpreter's relatively small code size and high performance allow its development capabilities to be used for end user support without disturbing application performance.

The **NPL Compiler** creates p-code on a batch basis, providing assistance to the interpreter for groups of programs.

The **NPL RunTime Program** executes p-code generated by the interpreter or compiler at end user sites. As an execution-only engine, the RunTime program does not allow for the program inspection or modification provided by the interpreter.

What NPL Offers to Developers

NPL's logic engine offers a feature-rich development and execution environment. Its components are:

- Language and programming features
- Source code facilities
- Library support

Language

NPL provides a **robust 3GL language** suitable for the development of sophisticated business or engineering applications. (This power is projected into the 4GL world by Visual Basic for building Windows user interfaces.)

NPL's **structured language** components provide a set of powerful features for developing and using modular technology, detailed in the following points:

- **FUNCTIONS** and **PROCEDURES** provide for structured subroutine development and access.
- **Program modules** allow development of true "black box" routines (application-independent libraries.)
- NPL offers a collection of highly useful **program constructs** including:

WHILE / WEND SWITCH / CASE
REPEAT / UNTIL FOR / BEGIN / NEXT
IF / ELSE / END IF

- Using **RECORDS** and **FIELDS**, programmers may define logical record layouts. When used with modules these permit a very high level of abstraction within database applications.

Programming Features

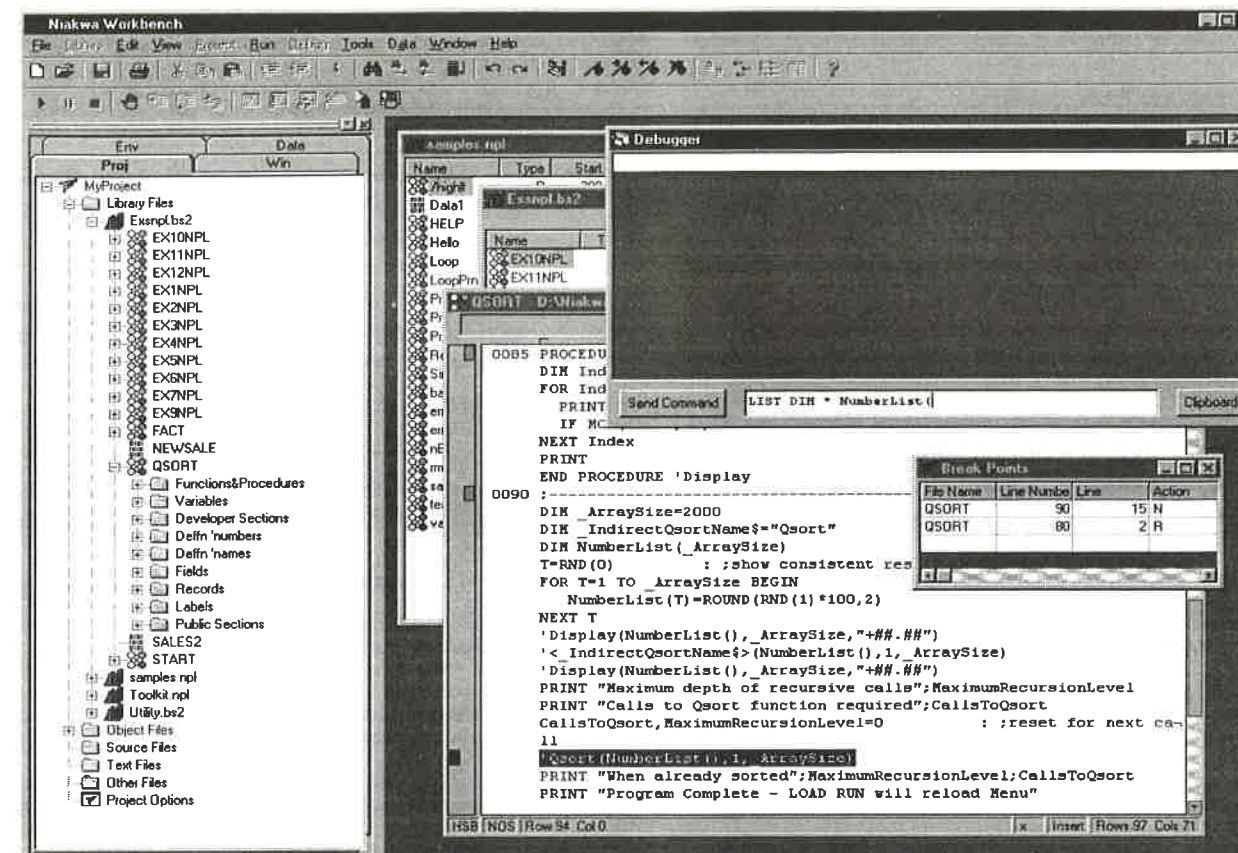
NPL provides intrinsic **advanced math functions** such as trig, log and exponential with math coprocessor support in many environments. NPL also uses an internal BCD numeric format which delivers fast arithmetic operations while yielding accuracy to 13 digits. In addition, NPL provides extremely powerful **MAT** (matrix) math functions, such as MAT multiplication, MAT transposition, MAT identity and MAT inversion. NPL supplies a series of **data conversion statements** including CONVERT, BIN, VAL, \$PACK and \$UNPACK, which efficiently maximize code power and flexibility to convert data from one form to another.

NPL provides sophisticated **string manipulation functions** such as MAT SEARCH, MAT SORT (built-in sort) and MAT MERGE operations. NPL also offers low level operations that allow complete bit-level control of strings. Strings may be dynamically expanded up to 2GB on 32-bit platforms.

NPL handles **multi-user operations** with ease, supporting up to **512 simultaneous users**, while performing numerous critical functions such as file locking and unique terminal, task and user identification.

Developers may integrate a **HELP subsystem** into their applications to improve end user support.

NPL offers **error control** at the program, line, function and command levels as well as easy design of generic error traps and "user-proof" applications.



NPL features one of the strongest visual development environments on the market.

NPL also offers convenient, cost-efficient capabilities for the use of demonstration software, including:

- limited-execution count **demo diskette** RunTimes which allow for on-site application demonstrations
- **self-running demonstrations** and automatic user-training tutorials generated with the \$DEMO function

NPL developers have a well-deserved reputation for success in the international marketplace due to NPL's support for extensive **non-English development**.

- NPL supports standard **International character sets**
- NPL lends itself to **local application development** in a variety of languages
- NPL provides **easy-to-read documentation**

NPL developers enjoy the flexibility of running their applications on a variety of hardware platforms while maintaining a consistent look and feel, because NPL's **virtual machine environment** provides many operating system-like services (workstation, disk, printer and communication controls) that are built directly into the language.

Source Code Facilities

NPL's source code facilities allow rapid development of integrated applications.

Integrated Editor/debugger

NPL's integrated editor/debugger affords fast, easy program editing and flexible, powerful debugging with commands such as LIST, TRACE, PRINT and STEP. NPL is editable in all supported character environments with this integrated editor, and under Microsoft Windows with the new graphical Niakwa Workbench. For complete details on this powerful language tool, see the Workbench Data Sheet.

Incremental Compiler

NPL's incremental compiler enables you to enter source code lines which are immediately compiled by the interpreter into executable p-code. The original source line is then discarded, and later regenerated on demand for listing, editing, and debugging by a built-in decompiler.

Utilities

NPL provides a comprehensive set of utilities for convenient program development, including:

- File backup, copy, deletion and recovery
- Library creation, listing and sizing
- Device property setting

Program Encryption

NPL's "scramble-protected" program encryption prevents unauthorized inspection of sensitive code. Gold Key software security protects all NPL applications from piracy.

Library Support

NPL's modular architecture lets developers create and use true application-independent library programs. Developers may create libraries of routines for their applications with NPL modules, which can be used with other NPL programs and external (to NPL) routines written in C, Pascal and Assembler.

NPL developers can easily integrate added functionality into their applications without the overhead of code development by using libraries from two sources:

- **Niakwa Libraries** are available to all NPL developers. They are typically written by Niakwa and always written in NPL.
- **Third-party Libraries** provide developers with a variety of add-in-functions. These are written in NPL, C and DLL (Microsoft Windows Dynamic Link Library) form.

Summary

NPL Benefits to Developers

- Use of NPL's efficient development features **increases programmer productivity**.
- NPL's interactive environment **speeds up development** by eliminating the need for recompilation while editing programs.
- NPL's incremental compiler technology makes it **easy to learn, code and debug** because of its many C-like features and its roots in BASIC.

NPL's **fast prototyping capabilities** allow for quick development and testing of application routines.

NPL Benefits to Users

- NPL's **consistent interface** between the virtual machine and the native environment adds to the portability and longevity of NPL applications.
- NPL delivers **superior performance** using high-level functions that quickly perform entire operations, where other languages require significantly more code and time.
- NPL source code is **fully compatible** across all NPL supported platforms, giving applications consistency regardless of the operating environment.
- NPL's **unparalleled portability** ensures NPL applications may operate on practically any platform the user chooses.
- NPL supports **device independence** via sophisticated control of all native device types (terminals, printers, storage devices, serial I/O devices, etc.) provided across all NPL supported operating environments.

NPL Developer Support

Niakwa provides one of the most prompt and comprehensive support systems in the industry today. We accept inquiries by phone, fax and e-mail.

NPL Operating Environments

AIX (RS6000)	Novell NetWare	VMS (DEC VAX)
HP-UX (HP9000)	Personal NetWare	Windows 3.x
LANtastic	SCO UNIX	Windows 95
MS-DOS	Sun Solaris (SPARC)	Windows 98
NetBIOS	Super DOS	Windows NT



Niakwa Inc.
1850 W. Winchester Road
Libertyville, IL 60048
(847) 816-7400, Fax (847) 816-7420

