mindfire solutions
exceptional software delivery™

Porting and Migration Services

desktop:

- Windows (Win 3.x, Win 95/98, Win NT, Win Me, Win 2000, Win XP)
- MacOS (Mac 7+, Mac 8.6, Mac 9, Mac OS X)
- Linux (RedHat, Corel)
- BeOS
- DOS

database:

- Oracle, MS SQL Server, IBM DB2
- Informix, Sybase, Ingres, Unify
- Foxpro, Access, FileMaker, dBase, Paradox, Btrieve
- mySQL, mSQL, PostgreSQL

device:

- PalmOS
- WinCE, PocketPC, NT Embedded
- Embedded Linux
- Symbian/EPOC
- RTOS, vxWorks

team:

- Extensive porting experience
- Knowledge of multiple platforms/technologies
- On-demand expertise availability

Mindfire Solutions has a well-developed Porting and Migration team, with supporting infrastructure and techniques. Porting offers great scope for efficiency and productivity. Mindfire uses its proprietary methodology to generate ports upto 40% faster, and with 100% compliance to the original.

principles of porting

Mindfire has developed a set of principles to be followed for effective porting. Effective porting is not just making software for the target platform, it is also making sure ongoing releases are easier.

- The target user *expects* behavior consistent with her platform, so don't mimic
- Go through and understand relationships between source files and functionality
- Use re-engineering tools or simple diagrams to *visualize* source code
- Keep the same code-base, or at least common code
- Implement *wrappers* as abstraction layers, both for code and data-types
- If required, modify the source code itself to get it ready for porting
- **++/-- Strategy**: The ++ approach takes a new project and you keep adding functionality as you code, while the -- approach takes a full project and you comment/delete code you don't want
- **Parts/Relationships Strategy**: You should also decide which is more important to focus on: parts of the source code (classes/modules/files) or their inter-relationship. In a C++ program, for example, you may keep the header files same, thereby forcing relationships to remain unchanged. You may then emulate these headers inside the classes to generate equivalent functionality
- Get the dirty work out of the way: Port the resources, help-files etc. early
- Do not deter from using target-specific technology, when it makes sense
- You *need* cross-platform Configuration and Bug Management Systems
- As always: Test, test, test!

different ports

Mindfire provides porting services across different platforms and technologies.

- Desktop ports
- Development tool/language ports
- Database data and program migration
- Web platform migration
- Universal platform strategy (dos-windows-web-pda-wireless)

quality assurance

- Normal life-cycle Quality Assurance (same as in any development)
- Target platform certification (does it look and behave like it should, on the target?)
- Equivalence validation (does it do everything the original did?)