

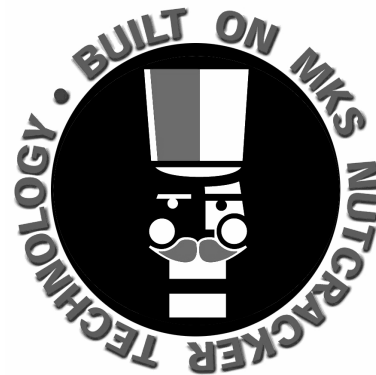


MKS Toolkit Evaluation Guide

for MKS Toolkit Developer Products

MKS Interoperability Products
12450 Fair Lakes Circle, Suite 400
Fairfax VA 22033 USA
Sales: 1-800-637-8034
+1-703-803-3343

<http://www.mkssoftware.com>



January 2002

Contents

Introduction	2
Product Family Introduction	2
Overview	2
MKS Toolkit System Administration Products	2
MKS Toolkit Developer Products	3
What's in the MKS Toolkit Products	4
Installing the MKS Toolkit Evaluation Package	7
Installing the MKS Toolkit Resource Kit	7
MKS Toolkit Basics	8
Evaluating the MKS Toolkit Developer Products	9
Evaluating MKS Toolkit for Developers	9
Developer Features	9
Editing	9
Scheduling Tasks	10
Visual Differencing	11
Build Environment	12
Evaluating MKS Toolkit for Professional Developers	13
Using Curses	13
Using POSIX Threads	13
Building DLLs	14
Building Windows Services (daemons)	14
Evaluating MKS Toolkit for Enterprise Developers	15
Interoperability Features	15
UNIX and Windows Application Interoperability	18
Customer Support	24
Additional MKS Toolkit Resources	24
Features Summary	25
Ordering Information	27

Introduction

This guide will help you evaluate, become familiar with, and choose the appropriate MKS Toolkit® product to meet your needs. While this guide is aimed primarily at software and web developers, it also discusses the general scripting and automation capabilities inherent in all MKS Toolkit products and applicable to almost any use. There is also a separate evaluation guide aimed more at system administrators, available from the same source as this evaluation guide.

This evaluation guide will:

- Give you an overview of the MKS Toolkit product family.
- Help you install the MKS Toolkit evaluation kit.
- Walk you through some of the highlights of each of the MKS Toolkit developer products.
- Tell you where to get more information about these products.
- Help you get customer support, should you need it.

Product Family Introduction

Overview

There are several products in the MKS Toolkit family aimed at different kinds of people, performing different kinds of tasks. All products in the MKS Toolkit family are unified by our goal of making your use of Windows more efficient and more enjoyable. Our products fall into two broad categories – those for system administrators and those for software developers.

MKS Toolkit System Administration Products

MKS Toolkit system administration products are built on a solid foundation of robust tools and engines that have been optimized for building best-of-breed system administration solutions. Whether you are administering a Windows® environment or one that mixes Windows and legacy systems, these tools and engines let you effectively manage that environment.

- **MKS Toolkit for System Administrators** – a powerful administration suite featuring MKS Toolkit for System Administrators also gives you the ability to move data and files between machines, remotely administer systems, and perform backups across multiple platforms. Increase productivity and automate repetitive tasks like password synchronization, adding users and groups, setting up new machines, cloning a system file or a document tree on local or remote systems, and automatically scheduling recurring tasks.
- **AlertCentre™** – a complete solution for monitoring, alerting and job scheduling. AlertCentre can monitor your mission-critical systems and applications 24 hours a day, seven days a week to provide you with the peace of mind of knowing that

your network, applications, and Internet/Intranet-based information systems are running normally.

MKS Toolkit Developer Products

MKS Toolkit development products facilitate a cross-platform, "write once-deploy anywhere" development and deployment strategy across Windows and legacy systems, and preserves existing investments in technology and expertise. The MKS Toolkit development products also allow pure Windows developers to automate routine tasks with scripting and increase productivity with file and text manipulation tools, batch search and replace tools, and scheduling interfaces. MKS Toolkit development products provide solutions in the following areas:

- **User Compatibility** – a variety of command shells and hundreds of command-line utilities help make your software engineers more productive on their Windows machines.
- **Platform Interoperability** – a selection of connectivity utilities provides you with secure interactive access to both your remote Windows servers and workstations and your legacy systems.
- **Application Portability** – more than 2700 APIs let you port all manner of legacy applications to Windows, while maintaining a single source code baseline.

The MKS Toolkit product family is the only solution for developing and deploying enterprise-critical cross-platform applications. It is also the leading product for evolving those applications to incorporate the latest Windows and web technologies, such as COM and Java™. Whether you are a native Windows or cross-platform software, script, or web developer, the MKS Toolkit developer products can help you be more efficient in your job.

- **MKS Toolkit for Developers** – a Windows development product for software, script, and web developers, who can choose from command-line or graphical Visual Studio ® build environments. It is ideal for developing common, cross-platform scripts for UNIX ® , Linux, and Windows, from a single Windows desktop. Using the power of more than 300 command-line utilities, including make, cc, and vi, you can develop native Windows applications, as well as Perl, Korn shell, C shell, Tcl, sed / awk,
- **MKS Toolkit for Interoperability** – a comprehensive interoperability suite, allowing remote access, remote system administration, interconnectivity, file sharing, and full automation and scripting capabilities. It includes an X server, for displaying and running X Windows graphical applications, a telnet server, xterm, remote and secure client/server utilities, for command-line access to remote machines, and a complete command-line environment for Windows. Perform distributed builds, multi-platform scripting, remote administration and increase your productivity with this connectivity solution.
- **MKS Toolkit for Professional Developers** – an ideal solution for cross-platform development, deployment, and migration of legacy commands, utilities, and

scripts to Windows. From a single desktop, using over 1500 APIs, you can extend the power of the MKS Toolkit and migrate custom utilities, non-graphical C, C++, or FORTRAN applications as well as Perl, Korn shell, C shell, Tcl, sed / awk, and CGI scripts.

- **MKS Toolkit for Enterprise Developers** – the most complete solution for enterprise cross-platform development, deployment, interoperability, and migration of legacy applications and scripts to Windows. From a single desktop, using more than 2700 APIs and over 300 utilities, you can develop and migrate Motif, X Windows, 3-D, and OpenGL applications in C, C++, or FORTRAN, as well as non-graphical applications and scripts. Choose from command-line or graphical Visual Studio build environments. Use powerful scripting capabilities to automate your development and administration tasks. Quickly evolve legacy business logic by web-enabling it, or by integrating it with standard Windows desktop productivity tools.

What's in the MKS Toolkit Products

This section details the contents of each of the MKS Toolkit products:

- **MKS Toolkit for System Administrators**—a powerful administration suite that lets you move data and files between machines, remotely administer systems, automate administration tasks, and perform UNIX-compatible backups, across UNIX, Linux, and Windows platforms. It includes:
 - ♦ Over 250 utilities (full POSIX.2 specification), including remote utilities (**rsh**, **rshd**, **rexec**, **rexecd**, **rcp**, **rlogin**) for accessing UNIX systems.
 - ♦ MKS Korn shell (**ksh**) and MKS C shell (**csh**) command environments.
 - ♦ Powerful scripting tools such as **Perl**, **awk**, and **sed** and standard UNIX workhorse tools such as **vi** and **grep**.
 - ♦ SNMP (Simple Network Management Protocol) utilities that let you control and monitor network devices and their functions.
 - ♦ Utilities for setting up users, groups, and permissions on Windows.
 - ♦ Tape and archive commands (**tar**, **pax**, **cpio**, **mt**), for creating UNIX-compatible backups.
 - ♦ Service and registry commands to start and stop local or remote Windows NT services and manipulate the Windows registry.
 - ♦ Commands to manage device drivers, Windows domains, and file associations (**dev**, **domain**, **ftype**).
 - ♦ **NuTCRACKER** Workstation.
- ♦ **MKS AlertCentre** – A point-and-click solution for Availability Monitoring and Alerting. AlertCentre enables System Managers, System Administrators and other IT professionals to monitor Network Connectivity, System Resource Availability and Application Availability for critical resources such as: Web sites, Mail Servers, DBMSs and ERP packages. MKS AlertCentre is available as an add-on to all MKS Toolkit products.

- **MKS Toolkit for System Administrators with AlertCentre** – all the features of MKS Toolkit for System Administrators and a powerful point and click Availability Monitoring and scheduling solution. A specially packaged product bundle that includes MKS Toolkit for System Administrators and MKS AlertCentre.
- **MKS Toolkit for Developers**—a Windows development product for software, script, and web developers, who can choose from UNIX-style command line or graphical VisualStudio build environments. It includes:
 - ♦ All the features of MKS Toolkit for System Administrators.
 - ♦ Over 275 utilities, including command line build utilities (`make`, `cc`, `ar`, `ld`, `vi`, `grep`), for migrating and unifying your builds across UNIX and Windows.
 - ♦ Tools and utilities for manipulating HTML and other web content as well as pulling and pushing content from local or remote servers (`web`, `htdiff`, `htsplit`, `url`, `mkurl`, `PScript™`, `mkscgi`).
 - ♦ NuTCRACKER Workstation.
- **MKS Toolkit for Interoperability**—a full UNIX and Windows interoperability suite, allowing remote access, remote system administration, interconnectivity, file sharing, and full automation and scripting capabilities. It includes:
 - ♦ All the features of MKS Toolkit for Developers.
 - ♦ A fully network-capable X server, for displaying both local and remote X Windows applications.
 - ♦ A single-connection telnet server, allowing remote access to the Windows machine.
 - ♦ NuTCRACKER Workstation.
- **MKS Toolkit for Professional Developers**—an ideal solution for cross-platform development, deployment, and migration of non-graphical and server-based UNIX/Linux applications and scripts to Windows. It includes:
 - ♦ All the features of MKS Toolkit for Developers.
 - ♦ Over 1500 UNIX APIs for migrating non-graphical UNIX applications to Windows.
 - ♦ Process management: `fork()`, `exec()`, `signals`.
 - ♦ POSIX, DCE, and Solaris threads.
 - ♦ File system handling: `mount()`, `umount()`, `symlink()`.
 - ♦ IPC and Networking: sockets (including UNIX-domain sockets), shared memory, message queues, semaphores, and FIFOs.
 - ♦ UNIX to Win32 file and user security mapping.
 - ♦ UNIX to Win32 file path and file system mapping, including devices such as `/dev/null` and `/dev/lp`.
 - ♦ Curses and terminal handling.

- ♦ A complete build environment including `ar`, `cc`, `cxx`, `ld`, `make`, `gmake`, `tcl`, `bison`, and `flex`.
- ♦ A Deployment Wizard, a simple, graphical tool for preparing applications for distribution.
- ♦ The *MKS Toolkit Cross-Platform Developer's Guide*.
- ♦ NuTCRACKER Workstation.
- **MKS Toolkit for Enterprise Developers**—a complete solution for integrating UNIX and Windows software development on a single developer system, solving both interoperability and cross-platform development needs; for migrating complex, graphical UNIX and Linux applications and scripts to Windows; and for evolving and modernizing those applications to take advantage of Windows, to integrate with Windows applications, and to become web-enabled. It includes:
 - ♦ All the features of MKS Toolkit for Professional Developers and all the features of MKS Toolkit for Interoperability.
 - ♦ Over 2700 UNIX APIs including X11R6, Motif 1.2.5, OpenGL, and XView.
 - ♦ The Wintif option, for runtime selection of Windows or Motif look-and-feel.
 - ♦ A selection of X servers, for providing display from local and remote X Window System clients.
 - ♦ NuTCRACKER Workstation.

Summary

Here is a summary of the features available in each of the MKS Toolkit development products to help you choose the right product:

	MKS Toolkit for System Administrators	MKS Toolkit for Developers	MKS Toolkit for Interoperability	MKS Toolkit for Professional Developers	MKS Toolkit for Enterprise Developers
Command shells and scripting utilities	◆	◆	◆	◆	◆
System administration utilities	◆	◆	◆	◆	◆
Graphical and command-line scheduling interfaces	◆	◆	◆	◆	◆
Tape backup and archiving utilities	◆	◆	◆	◆	◆
Client/server remote utilities	◆	◆	◆	◆	◆
Client/server secure utilities	◆	◆	◆	◆	◆
Telnet server	◆	◆	◆	◆	◆
Software development utilities		◆	◆	◆	◆
Web development utilities		◆	◆	◆	◆
Tcl shell and built-in commands		◆	◆	◆	◆
X server			◆		◆
Advanced development utilities				◆	◆
Non-Graphical APIs				◆	◆
Tcl development APIs				◆	◆
X, Motif, OpenGL APIs					◆
Wintif					◆

Installing the MKS Toolkit Evaluation Package

All MKS Toolkit products install the Evaluation Guide and companion scripts by default. Custom installations have an option to disable the install of the Evaluation Guide, so if your installation is missing the Evaluation Guide, please rerun the installer and add the Evaluation Guide. In all other cases, the start menu MKS Toolkit→Evaluation Guide will contain the evaluation guides and links to the companion scripts and demonstrations.

Installing the MKS Toolkit Resource Kit

You must have an MKS Toolkit product installed on your machine before the MKS Toolkit Resource Kit can be installed.

- Visit <http://www.mkssoftware.com/reskit> and download and run The MKS Toolkit Resource Kit self-extracting installer.
- Follow the on-screen instructions.

MKS Toolkit Basics

There are a few basics you should know before embarking on your evaluation of MKS Toolkit.

- **Using MKS Toolkit Features.** While MKS Toolkit has several graphical utilities for doing useful things such as compressing archives of files, the majority of the utilities in MKS Toolkit are non-graphical in nature. These non-graphical utilities are designed to be used from inside of a command processor, which we call a shell. UNIX users will be familiar with shells, which are more extensive than, but similar to the Windows command processor, `cmd.exe`. You will find the graphical utilities on and be able to launch them from the Start menu (Start→Programs→MKS Toolkit→...). On the other hand, to launch the non-graphical utilities, you will need to be in a shell or command processor, preferably one of the MKS shells.
- **Launching a Shell.** All versions of MKS Toolkit come with both a Korn shell (`sh`) and a C shell (`csh`). The easiest way to launch these is from the Start menu, Start→Programs→MKS Toolkit→Korn Shell or Start→Programs→MKS Toolkit→C Shell. There are some basic differences in these shells, but they are conceptually very similar. If you have no experience with shells or have no preference, you should probably start with the Korn shell. This evaluation guide assumes that you are running the Korn shell. The MKS Toolkit products contain extensive documentation on these shells that you can read at your leisure to understand the differences in shells.
- **Getting Help.** There are two main ways to get help in MKS Toolkit, the Windows way and the UNIX way, whichever you prefer. Traditionally, in UNIX, in a shell you type `man` and then the name of a utility, and this displays what is known as the manual page for that utility. Try typing `man sh`, for example, for help on the Korn shell. On Windows, however, help tends to be more graphical in nature. You will find graphical documentation for the MKS Toolkit utilities, plus additional tutorials and other information, on the MKS Toolkit entry in the Start menu, under Documentation (Start→Programs→MKS Toolkit→Documentation). There, you will find the MKS Toolkit Utilities Reference, which contains the same information as the manual pages, only in a browsable, graphical format.
- **Launching a Script.** Launching files on Windows is different from launching files on UNIX. Windows identifies how to launch any file by looking up the program to run by the file extension (the three letter suffix after the dot in the file name, such as `.exe` or `.doc`). Files on UNIX often have no extensions, while extensions are fairly critical on Windows. Windows maintains a list of registered file associations that allows a given file extension to be associated with a given file type which in turn is associated with a given application. This allows you to launch an application and load the specified file by simply clicking on the file name in the Windows Explorer or by simply typing the file name on the command line of a shell or command interpreter. For example, if you have Microsoft Word installed on your system, typing `file.doc` on the command line launches Microsoft Word with `file.doc` open. When you install a member of the MKS Toolkit product family, a number of MKS Toolkit-specific file associations are registered. The

following table shows the registered file extensions, their associated file types, and the MKS Toolkit utilities used to run files with those extensions:

Extension	File Type	Associated MKS Toolkit Utility
.awk	mks_awk	MKS AWK (awk)
.cpi	mks_vpax	Visual Pax (vpax)
.cpio	mks_vpax	Visual Pax (pax)
.csh	mks_csh	MKS C Shell (csh)
.gz	mks_gunzip	gunzip
.ksh	mks_shell_sh	MKS KornShell (sh)
.mk	mks_make	MKS Make (make)

Evaluating the MKS Toolkit Developer Products

When evaluating the MKS Toolkit developer products, keep in mind that each product in the product line is a superset of the preceding product. For example, if you are evaluating MKS Toolkit for Professional Developers, in addition to its own features, it also includes everything in MKS Toolkit for Developers.

Evaluating MKS Toolkit for Developers

MKS Toolkit for Developers is a Windows development product for software, script, and web developers, who can choose from UNIX-style command line or graphical VisualStudio build environments.

Developer Features

MKS Toolkit for Developers has many utilities that developers will find simplify their daily development tasks:

- ♦ Command line build utilities (**make**, **cc**, **ar**, **ld**), for migrating and unifying your builds across UNIX and Windows.
- ♦ Standard editing and scripting tools for editing source code, make files, and for building complex development environments (**vi**, **grep**, **find**, **awk**, **sed**, **Perl**). MKS Toolkit also includes a graphical version of **vi**, Vi for Windows, (**viw**) that responds to the standard **vi** commands, but supports standard Windows features such as font control and printer integration – the best of both worlds.
- ♦ Source differencing tools for determining the scope of changes in your source code (**diff**, **diffh**, **bdiff**), including a side-by-side visual differencing tool (**vdiff32**).
- ♦ Utilities for manipulating HTML and other web content as well as pulling and pushing content from local or remote servers (**web**, **htdiff**, **htsplit**, **url**, **mkurl**, **PScript™**, **mkscgi**).
- ♦ A scheduler for easily and automatically starting repetitive tasks, such as nightly builds, backups, and automated test runs.

Editing

The **vi** editor is the editor that is common to every version of UNIX. MKS Toolkit for Developers includes two versions of this editor for your convenience. The first is a

standard UNIX version, which you can exercise by typing `vi` from any command or shell prompt, or by selecting Start→Programs→MKS Toolkit→Evaluation Guide→Vi. If you are unfamiliar with `vi`, there is a tutorial in `$ROOTDIR/samples/vi/tutor.vi`.

Try this: `vi` is integrated with the Windows clipboard making it easy to cut and paste between Windows and `vi`. At the `vi` command prompt, type `set clipboard=x`, setting the Windows clipboard to buffer `x`. (You may want to add this to your profile file `ex.rc` in your home directory, `$HOME`.) Yank a couple of lines into the clipboard, `"x2Y`. Move your cursor into a Windows application, such as Notepad or Word, and use control-V to paste this text. Now, highlight some text in the Windows application and copy it using control-C. Move into `vi` and paste it: `"xp`.

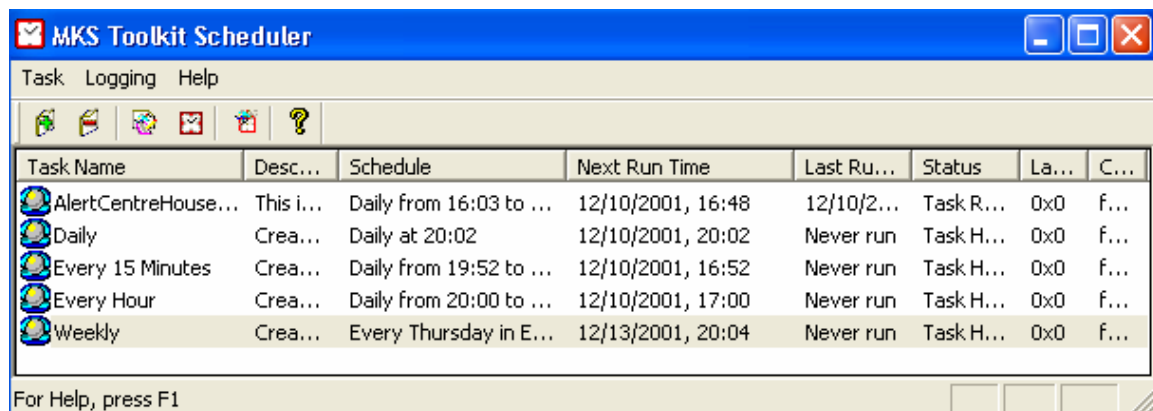
The second version of `vi` in MKS Toolkit for Developers is a graphical version, `vi` for Windows, that responds to the standard `vi` commands, but supports standard Windows features such as font control and printer integration – the best of both worlds. You can try it by typing `viw` from any command or shell prompt, or by selecting Start→Programs→MKS Toolkit→Evaluation Guide→Vi for Windows.

Try this: UNIX and Windows use different line termination characters: UNIX uses a linefeed, while Windows uses the combination of a carriage control and a linefeed. You can control the style to use by selecting Options→PC Specific from the `viw` menus. While you're at it, go to a command or shell prompt and type `man flip` to find out how to automatically convert whole files.

Scheduling Tasks

MKS Toolkit for Developers includes a scheduler that lets you define the time and frequency of execution of any program, including MKS Toolkit utilities and scripts. For developers, this is a good way to automatically start nightly and weekly builds and to run automated test suites.

Invoke the scheduler at Start→Programs→MKS Toolkit→Evaluation Guide→For Developers→Scheduler.



MKS Toolkit features a number of command-line utilities and the graphical MKS Toolkit Scheduler that serve as interfaces to the Windows Task Scheduler. Using these interfaces provides greater feedback and flexibility than is available with the Windows Task Scheduler.

- The `at` command-line utility schedules a task to run at a specified time.

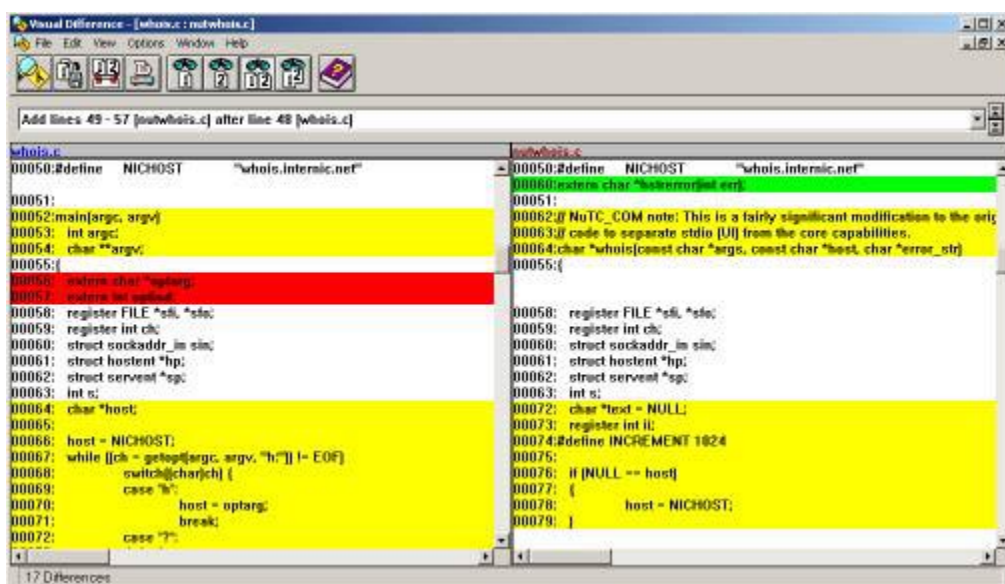
- The **batch** command-line utility schedules a task to run immediately.
- The **crontab** command-line utility lets you create, edit, and view a text file that contains the scheduling information for one or more tasks.
- The **taskrun** command-line utility schedules a task to run at a specified time and sends the output and errors produced by the task to a specified location. This command is primarily intended for use by the **at**, **batch**, and **crontab** utilities although it can also be used directly.
- The **wts** command-line utility is similar to the **at** utility, but allows you to schedule a task to run on a remote machine and to remove or edit an existing task.
- The MKS Toolkit Scheduler graphical utility (**tkshed**) allows you to schedule tasks using a graphical interface that features greater scheduling feedback and flexibility than the Windows Task Scheduler.
- AlertCentre also features scheduling capabilities allowing you to schedule tasks (monitors and jobs) to run at regular intervals over a given period of time.

Visual Differencing

In addition to standard command-line differencing tools (**diff**, **diffh**, **bdiff**), MKS Toolkit for Developers also includes a graphical side-by-side differencing and merging tool, which you can exercise by typing **vdiff32** from any command or shell prompt, or by selecting Start→Programs→MKS Toolkit→Evaluation Guide→For Developers→Visual Diff.

Differences are color-coded to make your work easier. Added text is green, deleted text is red, and changed text is yellow. You can easily change these colors. Visual Diff supports automated merging, side-by-side viewing, merged viewing, printing of difference reports, and summaries of changes.

For more information about Visual Diff, see the comprehensive on-line help in the tool itself.



Build Environment

One of the primary strengths of MKS Toolkit for Developers is its command line build environment, including utilities (**make**, **cc**, **ar**, **ld**), for migrating and unifying your builds across UNIX and Windows.

The native development environment on Windows is Visual Studio, which is extremely comprehensive, and consequently fairly difficult to learn. Although Visual Studio works very well for certain tasks, it is not all that well suited for developers who have an existing UNIX build environment that they are migrating to Windows. These UNIX build environments, in which you have invested time and effort into making work the way you want them to work, and which are now proven technology, can now be reused with minimal effort with MKS Toolkit for Developers.

The build environment demonstration, Start→Programs→MKS Toolkit→Evaluation Guide→For Developers→Build Environment, shows MKS Toolkit for Developers' POSIX-compliant **make** compiling a source file with **cc**, archiving it with **ar**, and running the resulting executable. Note: you must have a compiler installed and in your path before running this demonstration.

This MKS Toolkit for Developers software build environment is designed for those developers who want to do native development on Windows from a command line environment, who want to harness the batch build capabilities of **make**, who want to deploy a common build environment across UNIX and Windows, and who want to migrate scripts from UNIX to Windows with minimal changes. Those developers who want to migrate applications from UNIX to Windows, should evaluate the two migration products, MKS Toolkit for Professional Developers and MKS Toolkit for Enterprise Developers, both of which include the MKS Toolkit UNIX APIs.

KEY POINTS



Improve Developer Productivity

1. You can immediately start using your UNIX skills on Windows with the MKS Toolkit C Shell, Korn Shell, and over 250 commands and utilities (**vi**, **grep**, **find**, **awk**, **Perl**, etc.) that behave just like their UNIX counterparts.
2. Multi-platform scripting capabilities enable existing UNIX build environments to work on Windows, and builds can be automated and replicated in a cross-platform environment. If you have a compiler installed, you will have the full capabilities of **cc**, **ld**, and **ar**.
3. The **vi** and **viw** editors support the Windows clipboard and support both UNIX and Windows file formats.
4. Review changes to your source code with your team using MKS Visual Diff. Create code review packages with scripts built around **diff**.
5. Schedule nightly or weekly builds using the MKS Toolkit Scheduler.

Evaluating MKS Toolkit for Professional Developers

MKS Toolkit for Professional Developers is an ideal solution for cross-platform development, deployment, and migration of non-graphical and server-based UNIX/Linux applications and scripts to Windows. MKS Toolkit for Professional Developers contains the flexible development environment of MKS Toolkit for Developers, extended with additional development utilities such as `gmake`, `tcl`, `flex`, and `bison`; more than 1500 UNIX APIs; a Deployment Wizard, a simple, graphical tool for preparing applications for distribution; and a copy of *The MKS Toolkit Cross-Platform Developer's Guide*, encapsulating our years of experience in the migration and cross-platform development business.

The following demonstrations show just a few of the multitude of ways that you can use MKS Toolkit for Professional Developers. These demonstrations show examples of migrating UNIX code to Windows (one using curses character-based graphics and one using POSIX threads) and examples of evolving migrated applications to take advantage of Windows features (one showing how to package UNIX code as a Windows DLL that can be used from other Windows programs and another showing conversion of a UNIX daemon to a Windows service).

Using Curses

MKS Toolkit for Professional Developers contains a full curses library, including full color support and support for both cooked and raw mode programs. To see a demonstration of a character-based clock in color text, in `cbreak` and `noecho` modes, invoke Start→Programs→MKS Toolkit→Evaluation Guide→For Developers→Professional Developers→Curses.

You can view the source code for this example at `$ROOTDIR/samples/curses/gdc` to see that MKS Toolkit for Professional Developers is the ideal product for migrating your curses-based application to Windows. If your application uses X Windows for its graphics rather than curses, you will need to use MKS Toolkit for Enterprise Developers.

Using POSIX Threads

UNIX applications are more and more commonly threaded, and the most common threading package is POSIX threads. MKS Toolkit for Professional Developers contains POSIX threads and portability wrappers for both Solaris and DCE threads. In addition, the *MKS Toolkit Cross-Platform Developer's Guide*, contains an entire chapter on migrating threaded applications to Windows. Without this threading support, any threaded UNIX application would have to be substantially rewritten to the very different Windows threading model.

To see a demonstration of a POSIX-threaded application, invoke Start→Programs→MKS Toolkit→Evaluation Guide→For Developers→Professional Developers→Threaded Grep, a multithreaded program that combines the power of the UNIX `find` and `grep` utilities in a single process.

To see how simple it is to take threaded UNIX code and move it to Windows using MKS Toolkit for Professional Developers, you should examine the threaded grep source code. It will show you how complete our threading support is and how few changes are

necessary to migrate real-world applications. The source code is located in `$ROOTDIR/samples/threads/programs/tgrep`.

Building DLLs

Once you have migrated UNIX code to Windows, it would be nice to be able to reuse this code in new and unexpected ways. With MKS Toolkit for Professional Developers, you can. The simplest way to do this, as this demonstration illustrates, is by converting the code to a DLL—the Windows equivalent of a UNIX shared library. See the *WhoIs: Creating COM Objects from UNIX Code* section in the MKS Toolkit for Enterprise Developers discussion for alternative ways to reuse code.

This demonstration program, at Start→Programs→MKS Toolkit→Evaluation Guide→For Developers→Professional Developers→Using MKS Toolkit DLLs, shows how to build a DLL containing UNIX code and how to use the resulting DLL. You should also note that the DLL uses two native UNIX constructs, alarms and signals, demonstrating that these features may be used even when the driving program has not been migrated with MKS Toolkit.

You may find the source code for this demonstration, located in `$ROOTDIR/samples/tutorial/dll/basic_dll_cc`, illustrative of how simple it is to evolve UNIX code to a new form that can be loaded and run from any Windows program. You will also find an entire chapter on building DLLs in the *MKS Toolkit Cross-Platform Developer's Guide*.

Building Windows Services (daemons)

Not all applications are foreground or interactive applications. Many times control and monitoring programs are set running by the operating system at boot time. These programs are called daemons on UNIX and services on Windows.

On Windows XP, 2000, and NT, early in the boot sequence, a Service Control Manager starts and scans the registry (rather than a static flat file such as found in `/etc/rc`) for service processes to start running. Using MKS Toolkit for Professional Developers, any Unix daemon program can be converted to run as a Windows service, using the MKS Toolkit service framework. With minor source code changes, your daemon will start and stop using the Windows Control Panel or the MKS Toolkit `service` utility. For more information about `service`, useful for controlling services from scripts, from the command line, and from remote telnet or login sessions, type `man service` at a command or shell prompt.

Because services are initiated at boot time, we have not included a demonstration program. However, the source code for the service tutorial, located in `$ROOTDIR/samples/tutorial/service`, shows how to evolve a UNIX daemon into a Windows service. This code and the chapter on building services in the *MKS Toolkit Cross-Platform Developer's Guide* show just how easy it is to convert a daemon to a service using our framework and how few code changes are required.

KEY POINTS



The Power of UNIX on Windows

1. MKS Toolkit for Professional Developers is a complete and robust solution for migrating existing UNIX and Linux applications to Windows.
2. MKS Toolkit for Professional Developers supports advanced UNIX process control constructs such as `fork()`, threads, alarms, and signals.
3. By building DLLs from your legacy UNIX code, you can launch that code from a Windows process and reuse it in novel ways.
4. The more than 1500 UNIX APIs in MKS Toolkit for Professional Developers support curses, terminal handling, file system handling, process control, and file and user security.
5. The *MKS Toolkit Cross-Platform Developer's Guide* is an invaluable resource, based on years of experience and thousands of applications, for migrating and evolving applications.

Evaluating MKS Toolkit for Enterprise Developers

You can use MKS Toolkit for Enterprise Developers to develop single- or multi-threaded C, C++, or FORTRAN applications, shared libraries, and daemons, shell scripts, and make files on Windows XP/2000, Windows NT (including Terminal Server Edition), Windows 9x/ME (Millennium Edition), or both, while maintaining a common source code baseline across all platforms, including UNIX. MKS Toolkit-migrated applications run natively in the Win32 subsystem just like other Windows applications, and can take advantage of all Windows features, including COM, the registry, and Windows help.

Interoperability Features

MKS Toolkit for Enterprise Developers contains several interoperability features to help you harmonize your mixed UNIX and Windows environment, including a high performance X server and a telnet client and server. The following sections discuss these interoperability features.

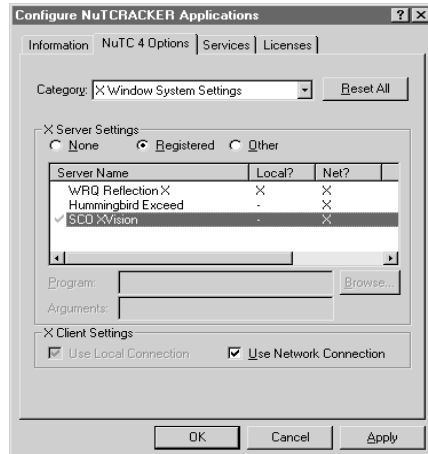
X Server

MKS Toolkit for Enterprise Developers contains SCO's XVision X server. This is a high performance X Server that lets a Windows workstation display native X11/Motif applications running on remote UNIX servers as well as locally running X11/Motif and Open GL applications ported with MKS Toolkit for Enterprise Developers.

Upon installing MKS Toolkit for Enterprise Developers, the default X server should be ready to autostart whenever an X application is run. To confirm that this is the case:

1. Launch the Windows Control Panel. (Start→Settings→Control Panel).

2. Double click on the MKS Toolkit icon.
3. Select the 'NUTC 4 Options' tab and choose 'X Window System Settings' from the Category menu.
4. Ensure that the SCO XVision server is selected and that Registered is toggled.



Next, test that the X server is working properly:

1. Start a command prompt (Start→Programs→MKS Toolkit→Development→Command Prompt for NuTCRACKER Development).
2. Type `xlogo` at the command prompt.
3. After waiting a short period of time (the first time the XVision server starts) you will see the Xlogo application display using the installed X server. (*Note: You may be asked to tune the X server if this is the first time you have run an X application on the machine*).

TROUBLESHOOTING



Problem: An X application will not run.

Solution: If the X server is loaded, you will see a green 'XV' on the task bar. If not, start the server manually (Start→Programs→Vision→XVision Server).

Solution: If the X server is not installed, reinstall MKS Toolkit for Enterprise Developers, and be certain to install the X server.

Solution: The server is installed and loaded, but the application will not run. Ensure that the *TCP/IP Protocol* is installed. From the Network applet in the Control Panel, select the 'Protocols' tab. If *TCP/IP Protocol* is not in the list, click 'Add' to install it.

Solution: If you are running the SCO X Vision server, in the Control Panel, run the Vision Communications applet and select the 'Transports' tab. If *TCP-UNIX* is in the list, but disabled, enable it by clicking Properties. If *TCP-UNIX* is not in the list, add it with the Add button. Restart the X server.

Problem: Colors are strange at startup.

Solution: Click in the window. The colors should correct themselves.

Solution: Configure your display properties to use 65536 colors.

Connectivity Components

MKS Toolkit for Enterprise Developers has several telnet clients (each emulating a different terminal type), a Secure Shell client and server, xterm and a full range of remoting tools such as scp, sftp, rcp and rsh.

To launch a telnet client:

1. Invoke Start→Programs→Vision→ANSI Emulator.
2. Select 'Session/Connect....' In the resulting dialog, type in your Host destination (e.g., brillig.mks.com) and click Ok.
3. Type in your username and password for the chosen host.
4. The telnet connection is now established and you are accessing the remote machine from your local Windows system. Using the XVision telnet client you can have multiple Telnet connections running at the same time.

To launch a secure shell client to connect to your own machine

1. Invoke Start→Programs→Vision→XVision Server if you do not see a green X on your task bar. If you tried the previous X Server demonstration, this will already be running and you may skip this step.
2. Invoke Start→Programs→MKS Toolkit→Evaluation Guide→ Secure Shell
3. You will be for the password for your account. Please enter the correct password and press enter.
4. You should see a standard shell prompt.
5. Please type "xterm &"
6. You will see the shell within the secure shell window give you a process id for the background process it created and a X window should pop up containing a shell prompt.
7. The X11 protocol has been tunneled through the secure shell connection. This is not tremendously useful for the localhost connection established here, but it works to any Secure Shell server that enables X11 tunneling.

KEY POINTS



UNIX-Windows Application Accessibility

1. With the power of MKS Toolkit for Enterprise Developers, application developers can seamlessly coexist within UNIX and Windows environments.
2. Software engineers familiar with the UNIX command line and runtime environment are able to leverage the same utilities on the Windows platform.
3. Using the telnet or secure shell clients, a developer can compile and build source remotely and display the output of the resulting application locally on the Windows workstation via the X Server.

UNIX and Windows Application Interoperability

At the core of MKS Toolkit for Enterprise Developers is a development environment that empowers developers to build Windows software from their existing UNIX source code, facilitating a cross-platform, “write-once, deploy-anywhere” development strategy across UNIX, Linux, Windows XP/2000, Windows NT, and Windows 95/98 systems.

The following migration and portability demonstrations demonstrate the power of the MKS Toolkit for Enterprise Developers. Using MKS Toolkit for Enterprise Developers, developers can compile and run single- or multi-threaded C, C++, or FORTRAN applications, shared libraries, and daemons, on any 32-bit Windows Intel Platform.

The demonstrations also illustrate the ease with which existing UNIX baselines can be integrated with native Microsoft Windows technologies such as Win32 and COM.

Mosaic: Using X Windows and Motif

The Mosaic public domain web browser:

- Comprises over 130,000 lines of X and Motif code written in C.
- Can be migrated to Windows by any developer in less than 30 minutes using MKS Toolkit for Enterprise Developers.
- Requires zero modifications to source code.
- Can have a choice of Motif or Windows look-and-feel via the MKS Toolkit Wintif library.

This demonstration highlights the Wintif add-on, a Motif replacement library that lets you select a Motif or Windows look-and-feel at runtime, without recompiling your program. We have set up two shortcuts to Mosaic: one that demonstrates the Mosaic with the traditional Motif look and a second one that presents Mosaic with a Windows look.

First, run Mosaic with the Motif look:

1. Invoke Start→Programs→MKS Toolkit→Evaluation Guide→ For Developers→Enterprise Developers→MKS Toolkit Mosaic (Motif). There may be a brief delay while the X server autostarts. Once the server is running, X clients start quickly.
2. Notice that the colors, icons, buttons, and sliders are all Motif standard.
3. Notice that the Help menu is to the far right of the menu bar.
4. Select File→Open Local. Notice that you get a standard Motif file selection dialog.

Next, run Mosaic with the Windows look:

1. Invoke Start→Programs→MKS Toolkit→Evaluation Guide→ For Developers→Enterprise Developers→MKS Toolkit Mosaic (Windows).
2. Notice that the colors, icons, buttons, and sliders are all Windows standard.
3. Notice that the Help menu is back to the Windows standard position, the rightmost entry in the menu bar.

4. Select File→Open Local. Notice that you get a standard Windows file finder, complete with drive selector.

All this was accomplished by running the same executable with different command line options. Let the users of your application choose the look-and-feel that they want.

TROUBLESHOOTING



Problem: Windows version still looks like Motif (and you have rebuilt the demonstration source code).

Solution: Reinstall MKS Toolkit for Enterprise Developers and during installation, choose the Wintif option instead of Motif. Rebuild the demonstration.

KEY POINTS



UNIX-Windows Application Portability

1. Large, graphical applications migrate to Windows quickly and with little to no changes. Character mode applications, daemons, and complex server applications migrate just as readily.
2. UNIX developers are productive on Windows immediately, without having to know Microsoft Windows programming.
3. You can maintain a common source code baseline across Windows XP/2000, Windows NT, Windows 95, Windows 98, and UNIX, because you make few changes between the platform ports.
4. You can choose a Motif or Windows look at runtime. Fonts and other UI details can be customized with standard resource files.

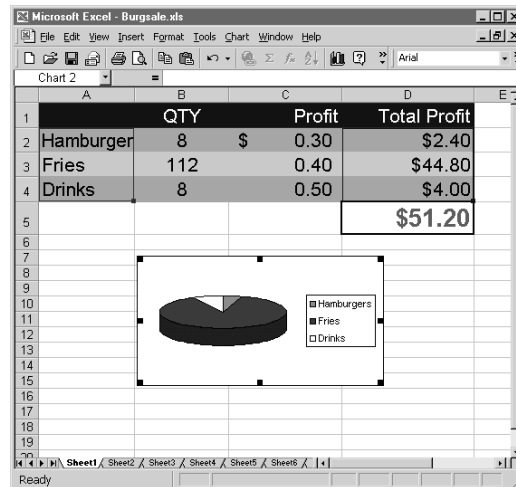
MotifBurger: Using DDE

The MotifBurger application is a simple Motif application that has been modified to include calls to WIN32 DDE functions. *Note: This demonstration requires that you have Excel 95 or later installed.*

To run this demonstration:

1. Invoke Start→Programs→MKS Toolkit→Evaluation Guide→ For Developers→Enterprise Developers→MKS Toolkit Motifburger. This will launch the MotifBurger application and an X Server if one is not already running.
2. From the main menu of the application select 'Order / Create Order Box...'
3. Within the dialog select various characteristics and quantities of each item. When you are finished choose 'Apply'. Then press 'Dismiss'.
4. Return to the main application window and select 'Order / Submit Order'.

5. If Excel is installed, your order data will be transferred to a new Excel spreadsheet via DDE and your profits will be totalled.



X Tetris: Automating an X11 Application with COM Automation

Using MKS Toolkit for Enterprise Developers, legacy UNIX applications can integrate directly with native Microsoft Win32 APIs. This sample illustrates how an X11 application ported to Windows can take advantage of Microsoft's Component Object Model (COM). By exposing a COM Automation Interface from the X11 application, it becomes instantly able to communicate and integrate with other Win32 programs.

To view the X Tetris application in stand alone mode (no automation):

1. Invoke Start→Programs→MKS Toolkit→Evaluation Guide→ For Developers→Enterprise Developers→MKS Toolkit Tetris un-automated demo.
2. To play a game of Tetris, use the arrow keys and the space bar for movement.

To view the X Tetris demo with COM capabilities enabled:

1. Invoke Start→Programs→MKS Toolkit→Evaluation Guide→ For Developers→Enterprise Developers→MKS Toolkit Tetris automated demo.
2. Note that there are now two windows. The first window is the original X application. The second window is actually a native Windows program that is automating the X application via COM automation, essentially remote-controlling the Tetris game.

WhoIs: Creating COM Objects from UNIX Code

The next demonstration program shows how a standard UNIX application (`whois`) can be directly embedded as a COM object in off-the-shelf Microsoft Windows programs, such as Internet Explorer, Word, and Excel, among others.

The `whois` object locates companies and Internet domains. To run the first `whois` demonstration, you need a web browser that can host ActiveX controls, such as Internet Explorer 4.0 or later, and an Internet connection.

To run this demonstration:

1. Invoke Start→Programs→MKS Toolkit→Evaluation Guide→ For Developers→MKS Toolkit Whois COM Demo. This will launch your browser on an HTML page that will show the whois control.
2. In the Query field, type the name of a domain, such as `mkssoftware.com`, and press the Query button.
3. An Abort button replaces the Query button (domain lookups on the Internet can take a long time). The abort function is possible because this application uses POSIX threads.
4. The results appear in the Response field.
5. Highlight a URL in the results, such as `mkssoftware.com`, and right click on it. You can open this URL by selecting the Goto menu item.

The HTML page will direct you to a technical paper, *Building COM Automation Objects with MKS Toolkit*, which describes the general process for converting UNIX applications to COM components, if you want more detail. You may have to scroll down to find the link.

Now you can examine other ways that this COM component (from ported UNIX code) can be used:

1. Start Word and create a new document.
2. Select Insert→Object and from the list, select whois.
3. You will see the control as in the previous example. Enter `mkssoftware.com` and click the Query button and wait for the results.
4. Move your cursor off the control and click it. The control will disappear, but the results will remain.
5. Some of the results text may be clipped. Click in the results text to make resize handles appear. Stretch the bounding box to the size you want.
6. If you double click on the results, the control appears again, ready for another query.

You have just used an embedded control to retrieve information and to insert the results directly into a document. This illustrates what is called *visual editing*, a powerful COM-based integration feature of MKS Toolkit for Enterprise Developers. This integration facility has powerful implications for reuse and ease-of-use of MKS Toolkit-ported UNIX applications.

Consider a mission-critical, DoD battle planning application running on UNIX. Using MKS Toolkit for Enterprise Developers, it can be transformed into a COM component on Windows that can be embedded into a Word document to facilitate editing and distribution of a plan. The same component could also be embedded into a PowerPoint presentation to facilitate a briefing and it could be embedded into an Active Server Page on Microsoft Internet Information Server (IIS) for rapid Intranet distribution purposes.

TROUBLESHOOTING



Problem: The control gets no response when you query a domain.

Solution: Make sure that you have an Internet connection.

Problem: The demo does not run.

Solution: Install Internet Explorer 4.0 or later, then reinstall the demos and then retry the whois demo.

The ATL COM AppWizard for MKS Toolkit

Using the ATL COM AppWizard for MKS Toolkit, generating a COM server is as easy as pushing a few buttons. The wizard, installed with the MKS Toolkit Resource Kit, is a Visual C++ application wizard (VC++ 6.0 only) that lets developers easily generate a skeleton source framework that is MKS Toolkit-aware and that can call MKS Toolkit's UNIX APIs. Note: *The ATL COM AppWizard requires installation of the Resource Kit. See the installation instructions at the beginning of this document.*

To use the wizard:

1. Launch the Visual C++ 6.0 IDE (will not be installed for Visual C++ 5.0 or earlier).
2. From the menu, select File→New..., and then choose the 'Project' tab.
3. Select the 'ATL COM AppWizard for MKS Toolkit' and name your project. Click 'OK' to move on to the next step.
4. Here you may choose the form that your COM server will take. Choose the desired option and push 'Finish'.
5. After selecting 'OK' from the summary screen the wizard will generate the requested source framework and you may then build the application.

KEY POINTS



UNIX-Windows Application Interoperability

1. Build mission-critical Windows components from existing UNIX code.
2. These components can be embedded in off-the-shelf Windows applications, such as Word, PowerPoint, Exchange documents, and Active Server Pages.
3. These components are automatically web-enabled when embedded in HTML pages or Active Server Pages.
4. MKS Toolkit gives direct access to Win32 APIs, libraries, and COM components, enabling a wealth of integration options.
5. With the advent of COM for UNIX, MKS Toolkit-based COM components are portable across Windows and UNIX.
6. POSIX threads are fully operable in Windows applications.

Additional Sample Programs

In addition to the demonstration programs, we have included sample source code for several applications that illustrate other key features. You will find other samples and tutorials in the MKS Toolkit distribution, under `$ROOTDIR/samples`.

To build any of these samples:

1. Start an MKS Toolkit command prompt or Korn shell.
2. Change directory to the desired sample directory.
3. Type **make** (Review the readme file in the respective sample directory file for more details). If the sample uses an Imakefile, you must first run **xmkmf** to generate a make file, before you run **make**.
4. When the build is complete, run the built program.

KEY POINTS



UNIX-Windows Developer Productivity

1. MKS Toolkit for Enterprise Developers easily ports shell scripts and command-line, curses, and X/Motif applications. It supports porting shared libraries, daemons, and threaded applications.
2. MKS Toolkit for Enterprise Developers is the ideal tool for cross-platform development of C and C++ applications across UNIX, Linux, and Windows.
3. MKS Toolkit's development environment is just like that on UNIX, including over 300 UNIX utilities, e.g., **cc**, **ld**, **make**, **vi**, **truss**, **grep**, **find**, **awk**, **Perl**, etc.
4. You continue to use your UNIX make files and build environment on Windows.
5. Many applications just compile and run with MKS Toolkit, with no or very few modifications.
6. Porting to Windows is comparable to porting to another version of UNIX, using MKS Toolkit's over 2700 POSIX and UNIX 98 standard APIs, including POSIX threads.
7. Applications migrated with MKS Toolkit are native Win32 applications that you build with Visual C++ and debug using standard Windows debuggers, such as `msdev`. This lets you enhance your applications with native Microsoft technologies.
8. A common source code baseline dramatically reduces ongoing maintenance across Windows and UNIX.

Customer Support

MKS offers extensive customer support to ensure your success with our products. At any time during your evaluation of our products, please feel free to contact us concerning any issues that may arise.

The evaluation versions of any MKS Toolkit products include free support from the time of installation. In order to continue support beyond the evaluation period you must purchase a fully licensed version of the product along with a Preferred Customer Support (PCS) contract. PCS is renewable annually for a small fee and entitles you to unlimited customer support, patches, bug fixes, and product upgrades. All of our sales channels offer MKS Toolkit products with bundled PCS for your convenience. You may also purchase unbundled PCS contracts by contacting MKS directly

To receive support, you must register. You will have the chance to register with our support organization during installation of your product, or you may do so at any time over the web at <http://www.mkssoftware.com/register>.

To request customer support, please contact us by one of the means listed below and in your request, include the name and version number of the product that you are using, your serial number, and the operating system and version/patch level that you are using. Contact MKS customer support at:

Web: <http://www.mkssoftware.com/support>

E-mail: mailto:tk_support@mkssoftware.com

Telephone: +1-703-803-7660 (9:00am to 7:00pm Eastern, Mon-Fri)

Fax: +1-703-803-3344

Additional MKS Toolkit Resources

There are several other sources for additional information about our MKS Toolkit products. We have general product information, including technical specifications, detailed utility listings, and datasheets at:

MKS Toolkit Product Information: <http://www.mkssoftware.com/products>

We offer a resource kit including example scripts, additional utilities, more tutorials, and a wide variety of other useful information at:

MKS Toolkit Resource Kit Page: <http://www.mkssoftware.com/reskit>

The MKS Toolkit product family also offers a number of Add-On components for download from our web site:

MKS Toolkit Add-On Page: http://www.mkssoftware.com/support/add_ons.asp

Through the years, we have accumulated a lot of technical details about the MKS Toolkit products and have put this information in a searchable database at:

MKS Toolkit Knowledge Base: <http://www.mkssoftware.com/support/kb>

Our customers commonly ask certain questions. These questions and their answers are in our Frequently Asked Questions pages at:

MKS Toolkit FAQs: <http://www.mkssoftware.com/support/faqs>

Features Summary

The MKS Toolkit product family is the most comprehensive suite of products for UNIX-Windows interoperability, cross-platform development and system administration, UNIX-Windows application migration, and advanced Windows scripting. Today's power users, developers, and system administrators require powerful utilities that are rock-solid in performance and reliability. MKS Toolkit is the choice of Global 2000 companies worldwide for the management and automation of mission-critical tasks. Used by over 450,000 power users, developers, and system administrators worldwide, MKS Toolkit is the premium brand for addressing your organizations' critical Windows XP/2000/NT and UNIX/Linux interoperability needs.

The primary features of the products are:

- **Advanced Scripting for Windows.** With Korn and C shells, `sed`, `awk`, `Perl`, `dlg`, `filebox`, and over 300 additional utilities, there is no finer scripting environment on Windows.
- **Comprehensive Command Line and Build Environments.** Power users, developers, and system administrators will love the flexibility of our command line environments, for batch processing, for automation of recurring tasks, for remote access, and for general scripting. With rock-solid, proven utilities such as `vi`, `grep`, `find`, `make`, `cc`, and `ld`, our command-line and build environments are unsurpassed.
- **Scriptable System Administration Utilities.** With extensive system administration utilities for managing users, groups, desktops, shortcuts, and permissions; for tape handling, file archiving, and UNIX-compatible backups; for registry manipulation; for service, device driver, domain, and file association management; for queuing actions and scheduling tasks; and for process and system information queries; there are no better system administration products than those in MKS Toolkit.
- **Scriptable Web Development Utilities.** Web developers and maintainers will love the ability to create custom utilities and scripts to manage their sites with our utilities for manipulating HTML, for pushing and pulling content to and from local or remote servers, for Perl scripting in the Microsoft Active Scripting environment, and CGI programming in `Perl` and `mkscgi`.
- **Advanced UNIX-Windows Interoperability Suite.** MKS Toolkit products have everything that you need to interoperate in a mixed UNIX and Windows world, including telnet client and server, remote commands (`rexec`, `rsh`, `rlogin`, `rcp`) and servers (`rexecd`, `rshd`, `rlogind`), and X Windows servers.
- **Most Extensive UNIX-Windows Migration and Cross-Platform Development Facilities.** With over 2700 UNIX APIs and a full command-line development

environment, MKS Toolkit products are the ultimate in migration and cross-platform development environments. No other products can match our full support for C, C++, and FORTRAN; for UNIX process management including `fork()`, signals, alarms, and threads; for file system and security management; and for curses, X Windows, Motif, and OpenGL. And no other product matches MKS Toolkit's access to Win32 APIs for Windows integration and interoperability.

- **Advanced Modernization and Evolution Facilities.** Beyond migration, MKS Toolkit products help you modernize and evolve your legacy applications, by creating objects that can be reused within existing Windows applications, that can be used to build new Windows applications; and that can be embedded in Active Server Pages, web-enabling your application.
- **Most Extensive Suite of Value-Added Utilities for Windows.** No other products on the market address real-world needs for robust utilities like MKS Toolkit products. Following are some of our powerful value-added utilities:

Command	Definition
64decode	Decode a file using base64.
64encode	Encode a file using base64.
appc	Arbitrary precision programmable calculator.
assoc	Set file extension association in the Windows NT registry.
autorun	Specify programs to run on bootup or log in under Win32.
awkc	Compile awk programs into executables.
bindres	Encode resources and insert them into a specified text file; used with dlg.
c	Produce multiple-column output.
chacl	Change the access control list (ACL) on Windows NT objects.
chgrp	Change group attribute of a file on Windows NT.
color	Change foreground and background colors of the shell window
config	Configuration information.
db	Send SQL queries to a database via ODBC.
dde	Perform DDE client operations.
desktop	Simple command-line desktop manipulation.
dev	Display device driver information.
dlg	Load and manage Windows NT dialog boxes; create graphical shell scripts.
domain	Display Windows NT domain information.
filebox	Display Windows NT Open or Save dialog box.
filever	Print file version information.
ftype	Set file type association within the Windows NT registry.
gdf	Graphically display the amount of free space remaining on a disk (a dlg example).
gdir	Graphically display and manage the current directory stack.
ghist	Display and manage command history from a scrollable dialog box.
gps	Display and manage process status in a scrollable dialog box (a dlg example).
groupinfo	Manipulate Windows NT group information.
gset	Graphically define shell settings.
gvar	View or define variables, parameters, functions, and aliases.
halt	Shut down the system.
hist	Display, fix, edit and re-enter previous command.
htdiff	Compare two HTML files and display differences.
htsplit	Split an HTML file into tokens.

Command	Definition
lsacl	List access control lists for Windows NT objects.
manstrip	Strip the unprintable sequences out of online man pages.
mapimail	Send mail on a Win32 system using the MSMApi32 ActiveX COM object.
member	Manipulate Windows NT group membership information.
mkscgi	Run scripts on HTTP server through the Common Gateway Interface.
mkstdiag	Check MKS Toolkit configuration.
mkstinfo	Display MKS Toolkit serial number and other information.
mkstzip	Compress/decompress a file.
msgbox	Display a Windows NT message box.
PScript™	Perl scripting in the Microsoft Active Scripting environment.
registry	Display and modify the Windows NT registry.
security	Find security related information.
service	Manage Windows NT services.
shortcut	Create Windows NT shortcuts from the command line.
sid	Display user's security identifier.
start	Start a new program in another window.
strerror	Display a system error message.
sysinf	Display technical system information.
tb	Modify KornShell Windows toolbar on Windows 95.
tkshed	Launch commands at predefined times.
ugrep	Search for regular expressions from a dialog box (dlg example).
uncname	Return the UNC name for a specified file.
url	Parse Uniform Resource Locators (URLs).
userinfo	Manage Windows NT user information.
VDiff	Compare two text files and show / merge differences.
viw	Display-oriented interactive text editor for Windows.
VPax	(Visual Pax) graphical interface to pax, cpio, and tar archives.
wcopy	Copy from a specified file or standard input to the Windows clipboard.
web	Transfer files to or from a Web server.
windir	Display the name of the Windows directory.
wpaste	Print Windows clipboard text to standard output or a specified file.
ws	Display the name of the current workstation or desktop.
wstart	Start a new program in another window.

No matter what your need in UNIX-Windows interoperability, there is an MKS Toolkit product that is just right for you. Whether you are a power user, developer, or system administrator, MKS Toolkit is the clear choice. After evaluating MKS Toolkit, we hope that you understand, as have our 450,000 existing customers, why no other interoperability product delivers comparable quality, extensive support, and features in one convenient package. If you still have questions or concerns, please contact us at the numbers below.

Ordering Information

MKS Toolkit can be purchased from the [MKS Web Store](#), from [MKS Sales](#), from our [resellers](#), or by calling +1-703-803-3343 or 1-800-637-8034.