Escale User's Guide

For Escale 1.0.1

by Nicolas Zinovieff

May 3, 2005

As long as there are ill-defined goals, bizarre bugs, and unrealistic schedules,
there will be Real Programmers willing to jump in and solve the problem,
saving the documentation for later.

Contents

1 What is Escale?

Escale is a MacOS X native Newton connection software.

Unlike traditional MacOS connection utilities and applications based on Apple's Desktop Integration Libraries, Escale does not rely on the Classic environment at all.

Unlike other MacOS X native connection utilities, Escale uses the NewtonOS built-in dock protocol and therefore makes a true bridge between MacOS X technologies and NewtonOS technologies. For example, all text is transfered as Unicode since both systems support it natively.

Escale could be considered as a replacement for Newton Connection Utility. However, Escale makes intensive use of MacOS X technologies which are not available in NCU, while still retaining technologies in NCU.

Key features

- **Both legacy and modern network interfaces available**

  The Newton originally had 3 ways to communicate with the world: IrDA, Serial, and AppleTalk. Escale retained the Serial and AppleTalk interfaces and added a stunning new TCP/IP with Rendezvous way to communicate with the Desktop. (At this point, IrDA is not available but should be in the near future.)

- **AppleScript aware**

  Escale can be used by an external AppleScript to automate tasks, provide a new graphical interface, or synchronize with applications.

- **Graphical enhancements provided by the Quartz graphical engine**

  Escale has used the Cocoa framework to get stunning and fluid graphics for its user interface. Say hello to animated buttons, sheets and Aqua!

- **AddressBook synchronization with meta-data**

  Escale not only synchronizes with the AddressBook, it also transfers all the Newton meta-data. This means that any data on your Newton will actually be saved on the Mac and could be retrieved later. You can use the Macintosh as a Newton Card File synchronization hub.

And how much does that cost? Nothing. Escale is free. Actually, Escale is more than free... It is Open-Source, under the KRL license (cf last section of the manual).

2 What is the DCL?

Escale is a sample code based on the Desktop Connection Library, DCL for short. This library is a framework that replaces Apple's Desktop Integration Libraries. Like the DILs, the DCL is aimed to be targetted on several platforms, although it currently only builds for MacOS X since this is what we use.

Unlike the DILs, the DCL uses modern technologies built into MacOS X, especially the Unix foundations. The DCL is completely threaded using POSIX threads (there is minimal support for MacOS legacy threads as well) and uses as little CPU as possible.

For more information on the DCL, please refer to the forthcoming DCL programmer reference.

3 Hardware and Software Requirements

3.1 Newton Requirements

To use Escale, you need a NewtonOS 2.0 or a NewtonOS 2.1 device and a Macintosh running MacOS X 10.2 or higher. Indeed, Escale only supports the connection protocol built into NewtonOS 2.0 and NewtonOS 2.1 devices. Support of the NewtonOS 1.x protocol could come later although since we do not have NewtonOS 1.x units, we leave it to other developers.

- **NewtonOS 2.0 devices include:**
  - Newton MessagePad 120 with NewtonOS 2.0
  - Newton MessagePad 130
  - various NewtonOS 2.0 equipped clones.
- **NewtonOS 2.1 devices include:**
  - Newton MessagePad 2000 and 2100
  - eMate 300
  - various NewtonOS 2.0 equipped clones such as Schlumberger's Dr Watson.

For more information, please consult the Newton community FAQ at: http://www.splorp.com/newton/faq/

3.2 Macintosh Requirements

Escale uses and relies on technologies only available in MacOS X 10.2 or higher.

For a list of MacOS X 10.2 compatible computers and MacOS X resellers, please consult Apple's website at: http://www.apple.com/macosx/

3.3 Serial Connection requirements

Since Escale is a connection software, you need a way to connect your Newton to your Macintosh.

For a serial connection, you need a Macintosh equipped with a serial port or a MacOS X compatible USB/Serial adapter. We use a Keyspan adapter, but any MacOS X compatible adapter should work (it should register itself within the I/O Kit for Escale to use it). You can tell that your adapter is compatible with MacOS X if the serial ports appear in Escale preferences or in MacOS X network preferences. Please consult your adapter's documentation for details.

You also need a cable to plug the Newton to your adapter or Macintosh. If you have a Newton MessagePad 2000 or 2100, you might need the Interconnect dongle or a SER-001 Board.

3.4 TCP/IP and AppleTalk Connection requirements

TCP/IP and AppleTalk connection can be performed over Ethernet or Airport. Please note that both TCP/IP and AppleTalk require a Newton OS 2.1 device. (Escale cannot do LocalTalk connections, but since it can do fast serial connections, it is not really an issue).

Ethernet and Airport connections are exactly like networking several Macintoshes. For an Ethernet connection, you need an Ethernet-equipped Macintosh (all MacOS X compatible Macintoshes are equipped with Ethernet) and for an Airport connection, you need an Airport-equipped Macintosh. You can also use a network where the Macintosh or the Newton is connected to Airport via an Airport base. Please consult Apple's documentation on Airport and Networking Macintoshes for details.

Please note that Newtons do not take advantage of Airport Extreme high speed as we write this. Use of an Airport Extreme Base card or station works (there are some issues with ADSP, cf the Troubleshooting section), but you need to configure them to perform 802.11b only or 802.11b compatible networking. Please consult the Airport Extreme documentation for details.

On the Newton side, you need either a Newton-compatible Ethernet card or Airport-compatible card. You also need the driver for your card. This includes the Newton Devices package. Airport connections require Hiroshi Noguchi's WaveLan driver and a compatible card.

Please consult Victor Rehorst's Newton Ethernet Page for details: `http://www.chuma.org/Newton/ethernet/`

TCP/IP requires a TCP/IP dock module package such as the one we provide and Apple Newton Internet Enabler 2.0 with Ethernet Support.

4 The bootstrapping problem

The bootstrapping problem refers to the installation of packages required to connect your Newton and therefore to install packages. These packages include Ethernet drivers and TCP/IP module.

Since Escale uses NewtonOS built-in docking protocol, the bootstrapping problem is limited, but it still exists.

There are several ways to solve the problem.

4.1 The Serial Connection

Since Escale can connect to your Newton via the serial line, you just need hardware to perform the first connection and then be able to install either Fast serial connection packages or Ethernet/Airport drivers.

4.2 The AppleTalk Connection

Another way to solve the bootstrapping problem is to use a working AppleTalk connection. If your Newton already has an Ethernet card with its driver and Newton Devices, it can perform AppleTalk connections and therefore you can connect it to Escale to install the TCP/IP dock module and Newton Internet Enabler packages.

If you have a LocalTalk to EtherTalk bridge (such as Farallon's or a Macintosh running MacOS 8.x running Apple's Localtalk bridge), you can also use the AppleTalk connection to install Ethernet drivers on your Newton.

Please note that there is a bug between Airport Extreme Base Stations and MacOS X that affects AppleTalk. With MacOS X 10.2.x, you will probably not see your Macintosh running Escale in the Dock's browser. With MacOS X 10.3.x, you will see it but connecting will fail. We've been working with Apple on this problem.

4.3 The Other Newton

Another method is to install the required packages on a card and then insert the card in your Newton to activate those packages. This means you need someone else to install the packages on the card. You can also beam the packages between two Newtons. Please note that Newton Internet Enabler packages cannot be beamed without EEUnProtect. Please consult the Newton Community FAQ for details.

5 Installation of Escale

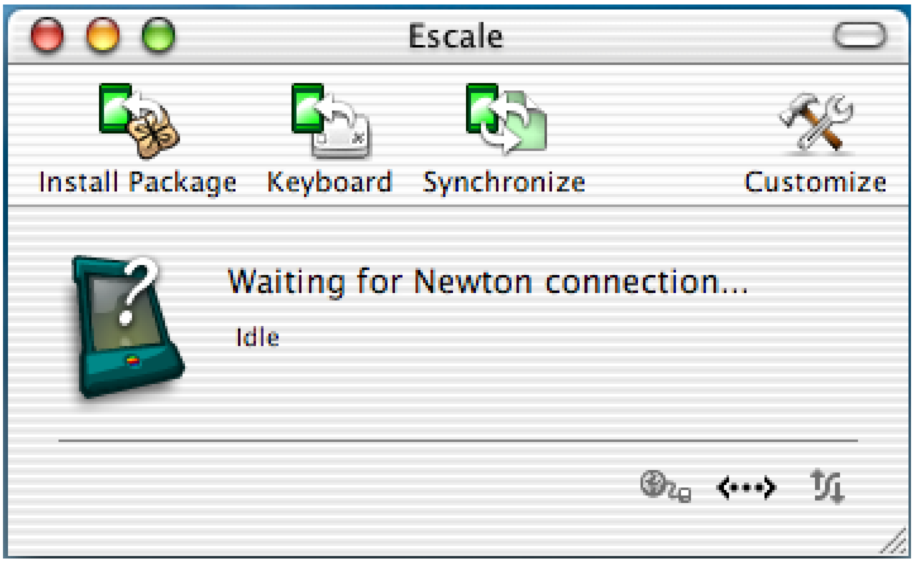
Ecale can be downloaded as a disk image from http://www.kallisys.com/newton/dcl/. It comes with source code and this documentation.

Installation of Escale is limited to just dropping the icon from the CD or disk image to any place on your hard drive. You can even run Escale from the CD or disk image.

Please note that Apple suggests that applications are installed into the Application folder (or an Application folder in your Home folder). Although Escale would work elsewhere, some MacOS X services might expect it to be there.

Once everything is installed, you will be presented with Escale's main window. There, you can find the interfaces status (mark the blinking Quartz exclusive effect), the Newton status, and the toolbar items. We hope the interface is straightforward enough to avoid documenting it.

On the disk image or CD-ROM, you will find dock modules that can be installed on your Newton to enhance the connection capabilities. These includes our Dock TCP/IP module which is RendezVous enabled and Serial modules for various serial speeds by Richard C. L. Li.



Waiting for Newton connection...

Idle

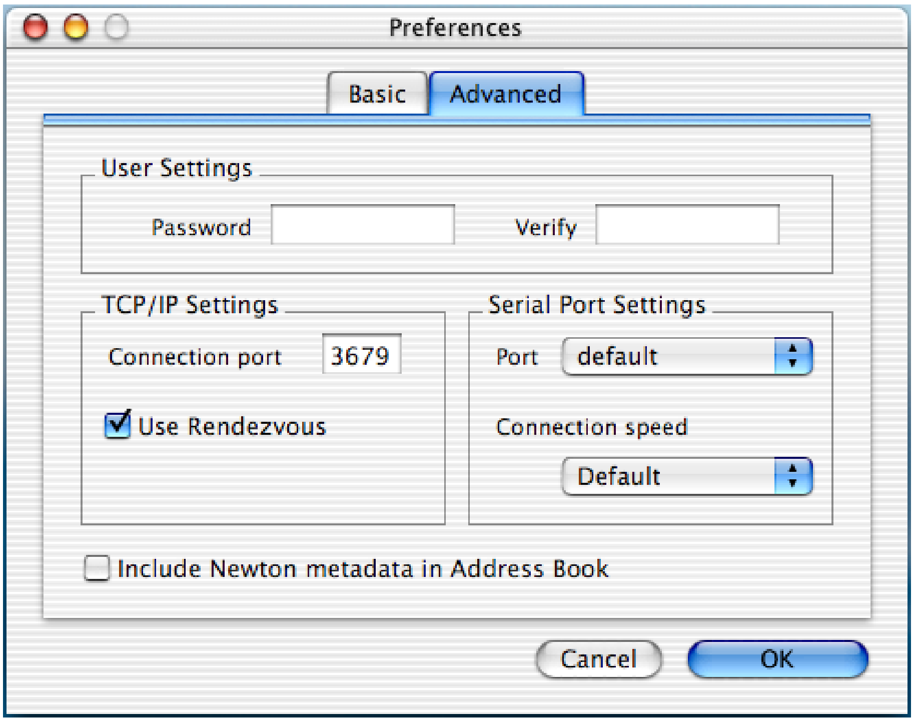
Escale main window

6 Configuration of Escale

You can configure Escale to determine on which interfaces it should wait for a Newton connection. By default, Escale listens on AppleTalk if available, TCP/IP if available with RendezVous and the first available serial port. It uses the IANA assigned port for TCP/IP (3679) and the Newton's default speed for serial connections (38400 bauds).

You have two sets of configuration variables. In the basic one, you will find the interfaces to enable.

In the advanced one, more in depth options are set, such as port number (if you would like to use another one instead of the IANA allocated one), the serial interface to use, the password (if you want to set one), etc...



User Settings

Password:
Verify:

TCP/IP Settings

Connection port: 3679

☑ Use Rendezvous

Serial Port Settings

Port: default

Connection speed: Default

□ Include Newton metadata in Address Book

Advanced section

We tried to keep the preferences as simple as possible.

7 Installing Packages

Unless you have installed it earlier with Delivery, the first package you might want to install is our Dock TCP/IP module. Please note that unlike other TCP/IP modules, our module provides RendezVous support and lets you automatically detect Escale running on your Macintosh. It also resolves local names.

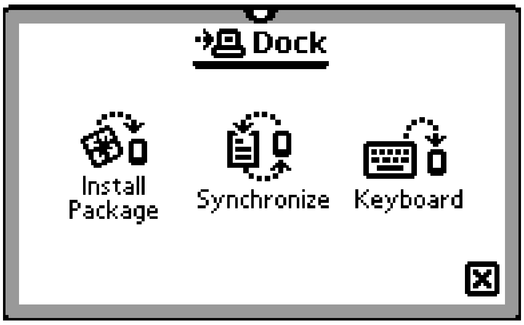
Make sure you have the Dock TCP/IP Module version 1.1 or later. Version 1.0 had a bad bug that affected the Zeroconf feature.

To install a package, first run Escale, wait until all interfaces are ready (AppleTalk interface may take a while to fire up) and then initiate the connection from the Newton using the Connection (NewtonOS 2.0) or the Dock (NewtonOS 2.1) built-in application (you will find it in the Extras drawer).

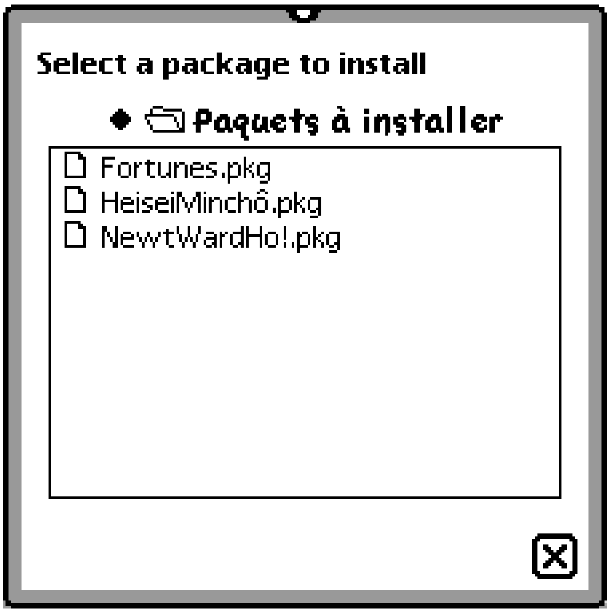
Once the connection is established, both Escale and the Newton should display the Dock menu since operations can be initiated from both computers. At that point, you can install the package either from the Macintosh or from the Newton.

7.1 Installing from the Newton

Once the connection is established, the Newton should display the Dock menu:



If you tap Install Package, the file browser will show up and let you browse your Macintosh hard drive. The desktop includes both mounted volumes and files and folders on your desktop. You normally only see there files that Escale considers as packages.

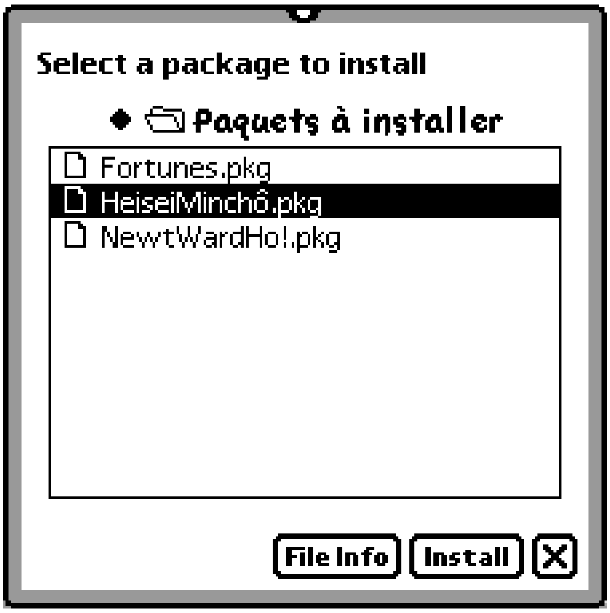


Select a package to install

◆ Paquets à installer

Fortunes.pkg
HeiseiMinchô.pkg
NewtWardHo!.pkg

Select a package to install, two buttons appear. You can then tap the Install button. File info displays a small information slip about the file.



Select a package to install

◆ Paquets à installer

Fortunes.pkg
HeiseiMinchô.pkg
NewtWardHo!.pkg

After installation, the Newton will go back to the Dock menu and you can either close the dialog to disconnect or install another package.

7.2 Installing from the Macintosh

To install a package from the Macintosh, you can:

- Drop the package on Escale's window
- Tap the install button in Escale's toolbar and choose a package from there

- Choose install from the Newton menu

Once the package is installed, Escale will go back to idle, letting you disconnect or install another package.

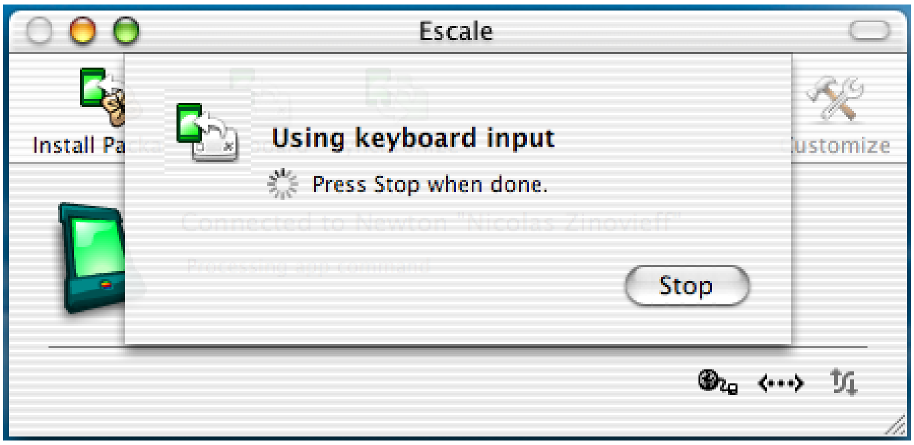
8 Using keyboard

Escale brings back the "keyboard" ability for the Newton. When you are connected to the desktop computer, you can use its keyboard to type on the Newton.

To call the keyboard window, use the keyboard menu item or the toolbar item in Escale or the keyboard icon on the Newton.

The keys are sent to the Newton just as you type them

To exit the keyboard mode, either press "Stop" in the Escale window, or end it from the Newton by coming back to the Dock (or Connection) window.



Using keyboard input

Press Stop when done.

Connected to Newton "Nicolas Zinovieff"

Processing app command

The keyboard window at work

9 Synchronizing with the AddressBook

You can start synchronizing by stapping Synchronize from the Newton's dock Menu or choose Synchronize from Escale menu/toolbar.

10 Troubleshooting

- **I am using a direct connection through {Airport / Ethernet } with my Newton and they never see each other**

  That comes from the way Mac OS X handles connection status. For an Ethernet/Airport connection, it requires the other end of the wire (ie the Newton) to have an UP status before activating its own interface.

  It is a little bit tricky to connect without a hub or a base station but it's doable. Proceed as follows:

  - Configure your Internet or AppleTalk connection on the Newton
  - Configure your TCP/IP or AppleTalk connection on the Mac. AppleTalk must be active on the interface your Newton will connect to. Also, if you plan a TCP/IP connection, make sure both ends are on the same subnet. The Newton cannot handle LinkLocal yet, so you will need to manually configure it with a private address such as 192.168.x.y and allocate a similar private address on your Mac (192.168.x.z).
  - Start the Dock application on the Newton and choose Other Computer (AppleTalk) or Browse (TCP/IP). This will fire up the interface of the Newton and subsequently the interface on the Mac.
  - Wait until Escale's corresponding interface icon is up. AppleTalk can take up to 18 seconds to fire up.
  - Once you can see your Mac in the Browser on the Newton, quickly close the browser and click Connect (AppleTalk) or just click Connect (TCP/IP). The Newton will not shut down the interface immediatly and therefore will be able to connect. If you were not fast enough, the interface of the Newton will go down and the interface on the Mac will go down as well (Escale's icon will become gray).

  Also, check your SSID (network name) carefully and make sure your WEP settings are correct. Please refer to the WiFi driver documentation for more details.

- **The connection seems to take forever to launch**

  If you are using AppleTalk, this can be normal: a name lookup for connection takes sometimes up to 18 seconds. Also, be aware that if you have firewalls or routers between the desktop and the Newton (eg an airport base station), there might be settings that prevent packets from arriving. Not all routers are compatible with AppleTalk packets.

  If you are connecting via Airport, make sure your SSID and other WiFi settings on the Newton are correct.

- **I can see the Macintosh in the AppleTalk browser but I cannot connect, the Newton tells me no MacOS computer could be found.**

  If you are using an Airport Extreme Base Station, we are aware of the problem. This is very probably a bug in MacOS X. We are working with Apple to get it fixed (unfortunately, it seems to be in MacOS X Airport drivers and/or Airport Extreme Base Station firmware, two piece of software we don't have access to the source of).

- **I get a -4 error on the Newton while browsing/using keyboard over TCP/IP**

  This seems to be a bug in Newton Internet Enabler. We are working on a workaround.

- **I have any other problem/I think I discovered a bug**

  Please contact us. If you think you found a bug, you can check the bugs database (`http://bugs.kallisys.com/`) and see if that bug has already been reported. If it hasn't, please report it.

11 Frequently Asked Questions

- **What can I do to help you guys?**

  Well, as you can see there is a lot to do...

  If you have OS programmatical skills, you can help up with the porting of Escale and the DCL to other platforms such as Windows, Unices, BeOS, etc...

  If you are an AppleScript user, you can develop scripts that link Mac OS applications to Escale

  If you have some great ideas, share them with us, we will be pleased to discuss and implement them

  If you want to donate hardware (especially Newton hardware), test intensively, feel free to do so

  Finally you can send us money so that we do not have to work on other projects ;-)

- **Why did you guys work on an outdated technology?**

  Well, first, we are not so sure it is outdated. Second, we wanted to complete the transition to Mac OS X. This program should help a lot of Newton users to upgrade their system software. Third, it's a very interesting exercise since we programmed this using technologies from the very core to the most higher interfaces. And last but not least, we are not alone in our desire for such a program to exist.

- **Who are you, by the way?**

  Although Paul is a famous developer in the Newton community, here is a little biography of the main contributors:

  Paul Guyot is a PhD student in artificial intelligence. He wrote a lot of software for the Newton, the last one being the extraordinary ATA driver that allows you to mount standard compact flash cards (or MemorySticks, etc...) in the Newton! He wrote the biggest part of this program.

  Mélanie Perroud is a student in Political Science. She speaks Italian, English, Japanese, Chinese and French and provided the Japanese translation of Escale.

  Michael Vacík is a professional designer. He is famous for his hilarious comics and icons. Many programmers owe him a lot. He fixed the original (awful) interface and provided us with ideas and pieces of advice that led Escale to where it is.

  Filip Zawadiak is a Polish {AS/400 / Newton } programmer. He licensed us his implementation of the MNP algorithm to enable serial connection with Escale under the BSD license. The other known free implementation of MNP is Richard C. L. Li's used in unixnpi[1].

  Nicolas Zinovieff is a professional independant developer. He also wrote some freeware and contributes verbally whenever he can. He wrote a bit of the POSIX stuff and of the network interfaces, as well as a part of the AppleScript layer. He hosts the CVS as well.

- **It's not that I am complaining, but why is it OpenSource?**

  Let us stress something: it is more than OpenSource in the way that to we wanted to be sure that any work on Escale would benefit all users for free. Therefore it is copyleft. There is a copy of the license at the end of this manual, and you can find more discussion on this license on Paul's webpage (`http://www.kallisys.org`). Since everyone can contribute to the development, we hope Escale will continue to evolve, and that people who need such a tool will be willing to comment on the code or improve it (see next steps).

- **What are the next steps?**

  Ah. Future. We are not fully sure of the priorities, but there are a few points we would like to see realized:

  We would like to support more platforms. The Newton itself can communicate easily, so what we need is respondents. Although we could do part of the work, there are a few platforms that are not as well known to us as Mac OS. So what we need is either helpers or time to learn.

[1] `http://unixnpi.sourceforge.net`

We would like to push even further the integration within Mac OS X, like synchronization tools that work with stickies, mail clients, iCal, whatever.

There is a server project based on the DCL to have some kind of faceless Escale that can work with several Newtons and users concurrently. We hope to finish this one pretty soon.

As for the unforeseeable future, let's hope we will never run out of ideas.

12 Building Escale

Escale is Open Source. This means that developers can modify it with ease. This section explains how to build Escale with or without modifications. Regular users do not need to build Escale. Just double-click the Escale application.

Source code is in the "source" folder. Escale can be built with Xcode and CodeWarrior.

12.1 Building with Xcode

Xcode is Apple's free development tool for Panther. To build Escale, you need Apple's latest development tools (Xcode 1.1) which you can download from the Internet on Apple's ADC website. Registration is required but it is free (ADC Online). Please consult Apple's website at `http://developer.apple.com`.

Make sure you have installed the cross-SDKs as Escale is built using the MacOS 10.2.8 cross-SDK (to make sure Escale runs on Jaguar). To intall Xcode with the cross-SDKs, choose Customize in the installer and check the appropriate box.

Once you have installed Xcode, just open the Escale project, choose Escale target (debug or final) and click the build button. The project is at: `source/Desktop Connection Library/_Build_/Xcode/Escale/Escale.xcode`

12.2 Building with CodeWarrior

CodeWarrior is a commercial development environment by Metrowerks. You can obtain it from their website (`http://www.metrowerks.com`). We use the academic version, but it does not differ from the regular version. To build Escale, you need CodeWarrior 8.3 for Macintosh.

The CodeWarrior projects refer to Source Trees to define in CodeWarrior's general preferences:

- **Frameworks OS X** /System/Library/Frameworks/

Once CodeWarrior 8.3 is installed and you added these Source Trees, just open Escale's project, choose the Debug or the Final target from the pulldown menu and select Make in the Project menu. The project is at: `source/Desktop Connection Library/_Build_/CW/Escale.mcp`

12.3 Building the AppleScript applets

The AppleScript applets do not need to be built. Just open them with the Script Editor. Please note that there is a bug in Apple's Script Editor Beta, it will lose the description of the script.

12.4 Building the documentation

To build the documentation, you need pdflatex. We got it from Darwinports (http://www.opendarwin.org/). You can also get it from TeX i-Packages [2]. The documentation was edited with TeXShop [3] and BBEdit [4].

The PDF can be generated by invoking pdflatex **twice** in `source/Desktop Connection Library/Documentation/lat`
with: > `pdflatex EscaleUserGuide.tex`

The documentation can be viewed with Acrobat Reader (free, provided with MacOS X) and Preview (built into MacOS X).

[2] `http://www.rna.nl/tex.html`

[3] (freeware, `http://www.uoregon.edu/ koch/texshop/texshop.html`)

[4] (commercial editor, `http://www.barebones.com`)

13 Contact

Got a question, suggestion, comment? You can contact us at the following address:

http://www.kallisys.com/newton/dcl/

14 Kallisys Reflexive License

Licence Réflexive Kallisys | Kallisys Reflexive License
Version: 1.0
Copyright (c) 2001-2003 by Paul Guyot.

Le présent document est un texte bilingue Français et Américain. Ce sont actuellement les deux seules versions officielles.

The present document is a French/American English bilingual text. These are currently the only two official versions.

1. Domaine d'application

   1.1 Cette licence concerne toute œuvre qui présente une mention placée par le ou les détenteurs des droits d'auteur et/ou de la propriété intellectuelle spécifiant que sa distribution et son utilisation sont assujetties aux dispositions de la présente licence.

   1.2 Cette licence s'applique notamment aux programmes informatiques et à leur documentation.

   1.3 La distribution et l'utilisation de cette licence sont assujetties aux termes de cette licence elle-même.

2. Références normatives

   2.1 Les documents suivants contiennent des dispositions qui, par suite de la référence qui en est faite, sont valables pour cette licence. Au moment de la publication, les éditions indiquées étaient en vigueur. Ces documents sont sujets à révisions, et les parties liées par la présente licence sont invitées à rechercher à appliquer les éditions les plus récentes des documents indiqués ci-après:

      ∗ - Convention de Berne pour la protection des œuvres littéraires et artistiques, Acte de Paris du 24 juillet 1971 modifié le 28 septembre 1979 http://www.wipo.org/treaties/ip/berne/berne01-fr.html

      ∗ - The Open Source Definition, Version 1.9 http://www.opensource.org/docs/definition_plain.html

3. Définitions

   3.1 **"ŒUVRE COUVERTE"** signifie l'œuvre dont l'UTILISATION est assujettie aux termes de cette licence.

   3.2 **"AUTEURS"** signifie le ou les détenteurs des droits d'auteur et/ou de la propriété intellectuelle de l'ŒUVRE COUVERTE.

   3.3 **"DISPONIBLE GRATUITEMENT"** se dit d'une chose dont l'obtention ne coûte que les frais facturés au titre de son transfert. Dans le cas d'un programme informatique, ceci signifie également qu'aucun frais n'est facturé pour l'UTILISATION de ce programme.

   3.4 **"LOGICIEL TIERS"** est un programme informatique (par exemple, un exécutable, une bibliothèque ou un cadriciel) dont la mise en œuvre ne requiert pas l'ŒUVRE COUVERTE.

   3.5 **"SYSTÈME D'EXPLOITATION"** signifie le LOGICIEL TIERS requis pour mettre en œuvre le matériel.

   3.6 **"SYSTÈME HÔTE"** signifie le matériel, ou l'ensemble constitué du matériel et du SYSTÈME D'EXPLOITATION, ou l'ensemble constitué d'un SYSTÈME HÔTE auquel on a ajouté un LOGICIEL TIERS dont les spécifications d'interface sont DISPONIBLES GRATUITEMENT et qui est nécessaire pour mettre en œuvre les fonctions prévues de l'ŒUVRE COUVERTE.

3.7 **"EXÉCUTABLE"** signifie un ensemble de données directement utilisables par les SYSTÈMES HÔTES pour accomplir les fonctions prévues de l'ŒUVRE COUVERTE.

3.8 **"SYSTÈME DE DÉVELOPPEMENT"** est un ensemble d'un ou de plusieurs logiciels qui sont nécessaires et suffisants pour créer un EXÉCUTABLE à partir du CODE SOURCE. Dans le cadre de cette licence, et pour chaque EXÉCUTABLE, il doit exister plusieurs SYSTÈMES DE DÉVELOPPEMENT indépendants et fonctionnellement équivalents. Toutefois, il peut n'en exister qu'un seul à condition qu'il soit DISPONIBLE GRATUITEMENT ou inclus dans le SYSTÈME HÔTE.

3.9 **"MODIFICATION"** signifie toute addition, toute suppression, toute traduction et/ou tout changement apporté à l'essence ou à la structure de l'ŒUVRE COUVERTE.

3.10 **"DOCUMENTATION"** signifie une partie de l'ŒUVRE COUVERTE qui explique comment accomplir avec les EXÉCUTABLES les fonctions prévues de l'ŒUVRE COUVERTE. La DOCUMENTATION doit être lisible par un Humain à l'aide du SYSTÈME HÔTE.

3.11 **"CODE SOURCE"** signifie l'ensemble des données utilisées avec les SYSTÈMES DE DÉVELOPPEMENT pour créer tous les EXÉCUTABLES. Cet ensemble de données doit être le plus adapté possible pour faire des MODIFICATIONS.

3.12 **"VOUS"** signifie une personne physique ou morale unique acceptant tous les termes de cette licence.

3.13 **"ŒUVRE DÉRIVÉE DE L'ŒUVRE COUVERTE"** signifie toute œuvre dérivée au sens de la Convention de Berne pour la protection des œuvres littéraires et artistiques.

3.14 **"UTILISATION"** de l'ŒUVRE COUVERTE signifie ou bien:

(i) la distribution ou la copie de l'ŒUVRE COUVERTE

(ii) l'apport de MODIFICATION (produisant ainsi une ŒUVRE DÉRIVÉE DE L'ŒUVRE COUVERTE)

(iii) l'installation ou la mise en œuvre d'une fonction (prévue ou non par la DOCUMENTATION).

(iv) l'UTILISATION d'une ŒUVRE DÉRIVÉE DE L'ŒUVRE COUVERTE

3.15 **"LICENCE RÉFLEXIVE"** signifie le présent document.

4. UTILISATIONS permises de l'ŒUVRE COUVERTE - Conditions et restrictions.

4.1 La LICENCE RÉFLEXIVE est un contrat de licence entre VOUS et les AUTEURS. Si VOUS êtes en désaccord avec les termes de la présente LICENCE RÉFLEXIVE, les AUTEURS ne consentent pas à VOUS concéder une licence sur l'ŒUVRE COUVERTE.

4.2 L'ŒUVRE COUVERTE est protégée par la réglementation et les traités internationaux en matière de droits d'auteur, ainsi que par les autres réglementations et traités internationaux en matière de propriété intellectuelle. L'ŒUVRE COUVERTE n'est pas donnée mais concédée sous licence.

4.3 VOUS pouvez copier ou distribuer l'ŒUVRE COUVERTE si les conditions suivantes sont respectées:

4.3.1 L'ŒUVRE COUVERTE ne peut être dans une forme archivée ou compressée que si un logiciel de décompression ou d'extraction adapté est DISPONIBLE GRATUITEMENT.

4.3.2 Les EXÉCUTABLES doivent être distribués avec le CODE SOURCE et la DOCUMENTATION.

4.3.3 Le CODE SOURCE doit être distribué avec la DOCUMENTATION et les EXÉCUTABLES pour tous les SYSTÈMES HÔTES prévus.

4.3.4 La DOCUMENTATION doit inclure les éléments suivants:

4.3.4.1 La LICENCE RÉFLEXIVE y compris sa documentation (Annexe B) et l'Annexe A.

4.3.4.2 La liste des fonctionnalités prévues de l'ŒUVRE COUVERTE et des SYSTÈMES HÔTES correspondants.

4.3.4.3 Une notice expliquant comment obtenir les EXÉCUTABLES et le CODE SOURCE.

4.3.4.4 Une notice expliquant comment obtenir le SYSTÈME HÔTE si celui-ci n'est pas le minimum requis pour lire la DOCUMENTATION. En particulier, la documentation doit spécifier les versions des LOGICIELS TIERS, s'il y en a, comment les obtenir et comment obtenir leurs spécifications d'interface qui doivent être DISPONIBLES GRATUITEMENT.

4.3.4.5 Une notice donnant la marche à suivre étape par étape pour installer les EXÉCUTABLES si ceux-ci ont besoin d'être installés pour être mis en œuvre. Cette notice peut faire référence à la documentation des SYSTÈMES HÔTES.

4.3.4.6 Une notice indiquant quelle version du SYSTÈME DE DÉVELOPPEMENT a été utilisée pour générer chacun des EXÉCUTABLES.

4.3.4.7 Pour chacun des EXÉCUTABLES, une notice expliquant comment obtenir le SYSTÈME DE DÉVELOPPEMENT s'il n'y en a qu'un, ou au moins deux s'il y en a plusieurs.

4.3.4.8 Une notice donnant la marche à suivre étape par étape pour générer chacun des EXÉCUTABLES à partir du CODE SOURCE pour chacun des SYSTÈMES DE DÉVELOPPEMENT cités en application des paragraphes 4.3.4.6 et 4.3.4.7.

4.3.5 À l'intérieur du CODE SOURCE, et pour tout élément référençable par le ou les languages dans lequel ou lesquels le CODE SOURCE est écrit, doivent figurer, exprimées dans une langue naturelle, les informations suivantes:

4.3.5.1 La fonction de cet élément et son rôle dans l'ensemble de l'ŒUVRE COUVERTE.

4.3.5.2 La liste, s'il y en a, des éléments de l'ŒUVRE COUVERTE auquel il fait directement référence pour effectuer sa fonction.

4.3.5.3 Pour chacun de ces éléments auxquels il fait directement référence:
- les domaines de valeurs qui lui permettent d'effectuer sa fonction et son rôle ;
- une description facultative du fonctionnement au cas où ces valeurs sortiraient des domaines prévus ;
- et la façon dont il est susceptible de modifier leurs valeurs.

Note: À des fins de lisibilité, il est conseillé de placer ces informations au plus près de la définition des éléments concernés.

4.3.6 Les éléments qui peuvent être référencés incluent souvent, mais ne se limitent pas aux: arguments, champs, classes, constantes, fonctions, méthodes, membres, structures, variables. L'explication de la fonction et du rôle des éléments peut faire référence aux concepts classiques de l'informatique ou à des concepts spécifiques au langage de programmation. Si le langage de programmation le permet, la fonction et le rôle des éléments peut être décrit par le nom même de ces éléments; par exemple, une variable appelée curseurLeNom voire crsrLeNom n'a pas besoin d'une plus longue description de son rôle si, dans ce contexte, le nom pour lequel cette variable est un curseur n'est pas ambigu.

4.4 VOUS pouvez apporter des MODIFICATIONS à l'ŒUVRE COUVERTE produisant ainsi une ŒUVRE DÉRIVÉE DE L'ŒUVRE COUVERTE si les conditions suivantes sont respectées:

4.4.1 L'UTILISATION de l'ŒUVRE DÉRIVÉE DE L'ŒUVRE COUVERTE ainsi produite est sujette à la LICENCE RÉFLEXIVE qui gouverne l'UTILISATION de l'ŒUVRE COUVERTE initiale (y compris les clauses supplémentaires figurant en Annexe A).

4.4.2 L'UTILISATION de l'ŒUVRE DÉRIVÉE DE L'ŒUVRE COUVERTE est aussi sujette à d'autres clauses supplémentaires que VOUS pouvez rajouter à la fin de l'Annexe A conformément aux clauses pré-existantes.

4.5 L'installation ou la mise en œuvre d'une fonction (prévue ou non par la DOCUMENTATION) est sujette à l'acceptation explicite de la présente LICENCE RÉFLEXIVE. Ceci a en particulier pour conséquence que tout système d'installation prévu pour installer l'ŒUVRE COUVERTE et d'autres œuvres doit demander l'acceptation de la présente Licence pour l'ŒUVRE COUVERTE indépendamment des autres contrats de licence éventuels et des autres œuvres.

4.6 Le droit d'UTILISATION de l'ŒUVRE COUVERTE comprend une licence perpétuelle, non-exclusive, sans redevance, et internationale sur tous les brevets qui sont ou seront enregistrés, qui appartiennent ou appartiendront ou qui sont ou seront accordés aux AUTEURS, mais seulement dans les limites requises pour l'UTILISATION de l'ŒUVRE COUVERTE.

5. Limitations de Responsabilité et Garantie

5.1 Dans toute la mesure permise par la réglementation applicable, les AUTEURS ne pourront en aucun cas être tenus pour responsables de tout dommage de quelque nature que ce soit résultant de, ou lié à l'UTILISATION ou à l'impossibilité d'utiliser l'ŒUVRE COUVERTE, même si les AUTEURS ont été prévenus de l'éventualité de tels dommages.

5.2 Dans toute la mesure permise par la réglementation applicable, les AUTEURS fournissent l'ŒUVRE COUVERTE en l'état et sans garantie d'aucune sorte, y compris sans garantie de qualité et d'adéquation à un usage particulier.

5.3 Tout litige concernant la présente LICENCE RÉFLEXIVE avec Kallisys ou Paul Guyot, même en cas de pluralité des défendeurs, sera, à défaut d'accord amiable, porté devant le Tribunal de Grande Instance de Paris.

1. Scope

1.1 This license applies to any work containing a notice placed by the copyright holder(s) or the owner(s) of the intellectual property saying that its distribution and its use are subject to the terms of this license.

1.2 This license may apply in particular but not only to computer programs and their documentation.

1.3 The distribution and the use of this license are subject to the terms of this license itself.

2. Normative references

2.1 The following documents contain provisions which, through reference in this document, constitute provisions of this license. At the time of publication, the editions indicated were valid. These documents are subject to revision, and parties bound by this license are encouraged to investigate the possibility of applying the most recent editions of the documents indicated below:

∗ - Berne Convention for the Protection of Literary and Artistic Works, Paris Act of July 24, 1971, as amended on September 28, 1979 `http://www.wipo.org/treaties/ip/berne/berne01.html`

∗ - The Open Source Definition, Version 1.9, `http://www.opensource.org/docs/definition_plain.html`

3. Definitions

3.1 **"COVERED WORK"** means the work which USE is subject to the terms of this license.

3.2 **"AUTHORS"** means the copyright holder(s) and/or the owner of the intellectual property of the COVERED WORK.

3.3 **"FREELY AVAILABLE"** applies to an item that can be obtained for the mere fees involved in handling the item. For a computer program, it also means that no fee is charged for using it.

3.4 **"THIRD-PARTY SOFTWARE"** is a computer program (for example an executable, a library or a framework) that can be used without the COVERED WORK.

3.5 **"OPERATING SYSTEM"** means the THIRD-PARTY SOFTWARE required to use the hardware.

3.6 **"HOST SYSTEM"** means the hardware, or the hardware and the OPERATING SYSTEM, or a HOST SYSTEM and a THIRD-PARTY SOFTWARE whose interfacing specifications are FREELY AVAILABLE and that is required to perform the proposed functions of the COVERED WORK.

3.7 **"EXECUTABLE"** means data directly usable by the HOST SYSTEM to perform the proposed functions of the COVERED WORK.

3.8 **"DEVELOPMENT SYSTEM"** means a set of one or several software that are required and can adequately create an EXECUTABLE from the SOURCE CODE. In this License, there shall be, for each EXECUTABLE, several independent DEVELOPMENT SYSTEMS that are functionally equivalent. However, there may be only one if it's FREELY AVAILABLE or included in the HOST SYSTEM.

3.9 **"MODIFICATION"** means any addition to, deletion from, translation and/or change to the substance and/or structure of the COVERED WORK.

3.10 **"DOCUMENTATION"** means a part of the COVERED WORK explaining how to perform the proposed functions of the COVERED WORK with the EXECUTABLES. It shall be readable by a Human using the HOST SYSTEM.

3.11 **"SOURCE CODE"** means the series of data used with the DEVELOPMENT SYSTEM to create the EXECUTABLES. This series of data shall be the preferred form to make MODIFICATIONS.

3.12 **"YOU"** means an individual or a legal entity complying with all of the terms of this license.

3.13 **"WORK BASED ON THE COVERED WORK"** means any derivative work in the meaning of the Berne Convention for the Protection of Literary and Artistic Works.

3.14 **"USE"** of the COVERED WORK means either:

(i) distribution or copy of the COVERED WORK

(ii) making any MODIFICATION (thus producing a WORK BASED ON THE COVERED WORK)

(iii) installation or operating any function (proposed or not in the DOCUMENTATION).

(iv) USE of a WORK BASED ON THE COVERED WORKD

3.15 **"REFLEXIVE LICENSE"** means this document.

4. Permitted USES of the COVERED WORK - Conditions & restrictions

4.1 The REFLEXIVE LICENSE is a legal agreement between YOU and the AUTHORS. If YOU do not agree to the terms of this REFLEXIVE LICENSE, the AUTHORS cannot grant YOU a license on the COVERED WORK.

4.2 The COVERED WORK is protected by copyright laws and international copyright treaties, as well as other intellectual property law and treaties. The COVERED WORK is licensed, not given.

4.3 YOU may make copies or distribute copies of the COVERED WORK if the following conditions are met:

4.3.1 COVERED WORK may be in a compressed or archival form, provided that the appropriate decompression or extracting software is FREELY AVAILABLE.

4.3.2 EXECUTABLES shall be distributed with the SOURCE CODE and the DOCUMENTATION.

4.3.3 The SOURCE CODE shall be distributed with the DOCUMENTATION and the EXECUTABLES for every proposed HOST SYSTEMS.

4.3.4 DOCUMENTATION shall include the following elements:

4.3.4.1 The REFLEXIVE LICENSE including its documentation (Annex B) and Annex A.

4.3.4.2 The list of the proposed functions of the COVERED WORK and the corresponding HOST SYSTEMS.

4.3.4.3 A notice explaining how to obtain the EXECUTABLES and the SOURCE CODE.

4.3.4.4 A notice explaining how to obtain the HOST SYSTEM if it is not the minimum required to read the DOCUMENTATION. Specifically, the DOCUMENTATION must specify the versions of the THIRD-PARTY SOFTWARE, if any, how to obtain these and how to obtain their interfacing specifications that must be FREELY AVAILABLE

4.3.4.5 A notice explaining step by step how to install the EXECUTABLES if they need to be installed to be used. This notice can refer to the documentation of the HOST SYSTEMS.

4.3.4.6 A notice indicating which version of the DEVELOPMENT SYSTEM has been used to generate each EXECUTABLE.

4.3.4.7 For each EXECUTABLE, a notice explaining how to obtain the corresponding DEVELOPMENT SYSTEM if there is only one or at least two of them if there are more than one.

4.3.4.8 A notice explaining step by step how to generate each EXECUTABLE from the SOURCE CODE for each DEVELOPMENT SYSTEM pointed to according to clauses 4.3.4.6 and 4.3.4.7.

4.3.5 Inside the SOURCE CODE, for each item which can be referenced by one of the programming languages used to write the SOURCE CODE, the following information, expressed in a natural language, shall be found:

4.3.5.1 The function of the item and its role in the COVERED WORK.

4.3.5.2 The list of the items of the COVERED WORK, if any, that it directly refers to in order to perform its function.

4.3.5.3 For each item it directly refers to:
- the ranges for their values which let it perform its function and its role;
- optionally, a description of the behavior if the values fall outside these ranges;
- and the way it may change their values.
Note: For readability purposes, it is advised to put this information as close as possible to the definition of the concerned items.

4.3.6 Items that can be referenced often include, but are not limited to: arguments, classes, constants, fields, functions, members, methods, structures, variables. The explanation of the function and the role of the items may refer to common concepts of computer science or to specific concepts of the programming language. If the programming language allows it, the function and the role of the items may be described in the name of the item itself; for example, a variable called theNameCursor or even theNameCrsr does not need further description of its role if, in the context, the name this variable is a cursor for is not ambiguous.

4.4 YOU may make any MODIFICATION to the COVERED WORK thus making a WORK BASED ON THE COVERED WORK if the following conditions are met:

4.4.1 Use of the aforementioned WORK BASED ON THE COVERED WORK is subject to the REFLEXIVE LICENSE which governs the USE of the original COVERED WORK (including the additional clauses in Annex A).

4.4.2 USE of the thus made WORK BASED ON THE COVERED WORK is also subject to additional clauses that YOU may add at the end of Annex A in accordance to the previously existing clauses.

4.5 Installation or operation of any function (whether or not proposed in the DOCUMENTATION) is subject to the explicit agreement of this REFLEXIVE LICENSE. As a consequence, any installer system designed to install the COVERED WORK and other work shall ask the agreement of this REFLEXIVE LICENSE for the COVERED WORK independently from any other license agreement, if any, and other work.

4.6 Grant to USE the COVERED WORK include a perpetual, non-exclusive, royalty-free, worldwide license to any patent that are now or hereafter acquired, owned by or assigned to the AUTHORS but only to the extent necessary to USE the COVERED WORK.

5. Limitations of Liability and Warranty

5.1 To the maximum extent permitted by applicable law, in no event shall the AUTHORS be liable for any damages whatsoever arising out of or in any way related to the use of or inability to use the COVERED WORK, even if the AUTHORS have been advised of the possibility of such damages.

5.2 To the maximum extent permitted by applicable law, the AUTHORS provide the COVERED WORK as is and hereby disclaim all warranties and conditions, including warranty of merchantability and fitness for any particular purpose.

5.3 In respect to any dispute with Kallisys or Paul Guyot that may arise concerning this REFLEXIVE LICENSE, YOU consent to the jurisdiction of the Tribunal de Grande Instance de Paris.

Annexe A Clauses Supplémentaires

- La présente Annexe de la LICENCE RÉFLEXIVE contient les clauses supplémentaires qui régissent l'UTILISATION de l'ŒUVRE COUVERTE.
- La présente Annexe est destinée à être complétée par les AUTEURS, par des clauses exprimées dans l'une au moins des langues de la LICENCE RÉFLEXIVE. Aucune clause figurant dans la présente Annexe ne peut en être retirée ou modifiée. Toute clause supplémentaire doit être compatible avec la LICENCE RÉFLEXIVE et les clauses pré-existantes de la présente Annexe.
- Toute clause supplémentaire ne doit pas avoir pour effet de faire perdre à l'ensemble constitué de la LICENCE RÉFLEXIVE et des clauses supplémentaires leur conformité à "The Open Source Definition".
- Toute clause contraire aux deux paragraphes précédents est nulle et non avenue.
- License for TDCLMNPPipe.*

- License for TMT19937RandomGenerator.cp

A C-program for MT19937, with initialization improved 2002/2/10.

Coded by Takuji Nishimura and Makoto Matsumoto.

This is a faster version by taking Shawn Cokus's optimization,
Matthe Bellew's simplification, Isaku Wada's real version.

Copyright (C) 1997 - 2002, Makoto Matsumoto and Takuji Nishimura,
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The names of its contributors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Annex A Additional Clauses

- This Annex of the REFLEXIVE LICENSE includes additional clauses that govern the USE of the COVERED WORK.
- This Annex is meant to be completed by the AUTHORS with conditions expressed in at least one of the languages of the REFLEXIVE LICENSE. No clause in this Annex shall be modified or removed. Any additional clause shall be compatible with the REFLEXIVE LICENSE and any previously existing clauses of this Annex.
- No additional clause shall make the set of the REFLEXIVE LICENSE and additional clauses lose their conformity to The Open Source Definition.
- Any clause that would be contrary to the two previous paragraphs is null and void.
- License for TDCLMNPPipe.*

Copyright (c) 2001 Filip Zawadiak <philz@vyx.net>
All rights reserved.

– License for TMT19937RandomGenerator.cp

```
   promote products derived from this software without specific
   prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

Annexe B Documentation de La Licence Réflexive

- La LICENCE RÉFLEXIVE est disponible sous forme non compressée. [4.3.1]
- La DOCUMENTATION, le CODE SOURCE et l'EXÉCUTABLE de La LICENCE RÉFLEXIVE sont constitués de la LICENCE RÉFLEXIVE elle-même. La LICENCE RÉFLEXIVE est réflexivement incluse en elle-même. :) [4.3.2, 4.3.3, 4.3.4.1]
- La LICENCE RÉFLEXIVE est un contrat utilisable sur tout SYSTÈME HÔTE qui dispose d'un outil (visualisateur de texte ou butineur) permettant de la visualiser. [4.3.4.2]
- La LICENCE RÉFLEXIVE peut être obtenue sur internet à l'adresse suivante: `http://www.kallisys.org/reflexive/`. [4.3.4.3]
- Tout SYSTÈME HÔTE sur lequel peut être utilisée la LICENCE RÉFLEXIVE doit permettre de la lire et donc de lire sa DOCUMENTATION. [4.3.4.4]
- La LICENCE RÉFLEXIVE n'a pas besoin d'être installée pour être utilisée. [4.3.4.5]
- Le CODE SOURCE de la LICENCE RÉFLEXIVE étant identique à son EXÉCUTABLE, aucun SYSTÈME DE DÉVELOPPEMENT n'est requis. [4.3.4.6, 4.3.4.7, 4.3.4.8]
- Tels qu'ils sont rédigés, chacun des alinéas de la LICENCE RÉFLEXIVE exprime dans une langue naturelle son propre rôle et sa fonction. Le texte des alinéas, s'il fait référence à d'autres éléments de la LICENCE RÉFLEXIVE, en fourni la liste dans une langue naturelle. ;) [4.3.5]

Annex B Documentation of the Reflexive License

- The REFLEXIVE LICENSE is not in compressed form. [4.3.1]
- The DOCUMENTATION, the SOURCE CODE and the EXECUTABLE of the REFLEXIVE LICENSE are constituted by the REFLEXIVE LICENSE itself. The REFLEXIVE LICENSE is reflexively included in itself. :) [4.3.2, 4.3.3, 4.3.4.1]
- The REFLEXIVE LICENSE is a contract which can be used on any HOST SYSTEM which has a tool (a text viewer or a browser) letting you read it. [4.3.4.2]
- The REFLEXIVE LICENSE can be obtained on the Internet at the following address: `http://www.kallisys.org/reflexive/`. [4.3.4.3]
- Any HOST SYSTEM on which the REFLEXIVE LICENSE can be used shall let you read it and therefore let you read its DOCUMENTATION. [4.3.4.4]
- The REFLEXIVE LICENSE doesn't need to be installed to be used. [4.3.4.5]
- The SOURCE CODE of the REFLEXIVE LICENSE being identical to the EXECUTABLE, no DEVELOPMENT SYSTEM is required. [4.3.4.6, 4.3.4.7, 4.3.4.8]
- As they are written, every paragraph of the REFLEXIVE LICENSE expresses in a natural language its role and its function. The text of the paragraphs, if it does refer to other items of the REFLEXIVE LICENSE, provides a list of them in a natural language. ;) [4.3.5]