

DIL 2.0 Overview

The Desktop Integration Libraries (DILs) 2.0 are a suite of libraries that desktop applications (for both Windows and Mac OS) can use for data interchange with Newton™ OS devices. While the basic responsibilities of the various DILs (i.e., CDIL, FDIL, and PDIL) remain the same as their 1.0 counterparts, we have focused on streamlining and enhancing their functionality and interfaces, simplifying the APIs, and of course, strengthening the robustness that will allow you to more easily create more powerful desktop connectivity products.

IMPORTANT NOTE:

In an effort to provide you with DILs that are clean and efficient from the ground up, we were forced to give up some backward compatibility. We understand this will require you to rewrite parts of your existing applications, but we encourage you to read the documentation and explore these new libraries. We hope you'll agree that the changes are worth the effort.

We will be providing conversion guides so that you can modify your programs to use the new APIs and we are investigating the effort required to develop a compatibility layer.

These libraries are for you, and we'd like to make sure we're doing the right things. If you have any comments, suggestions, or requests, please let us know at:

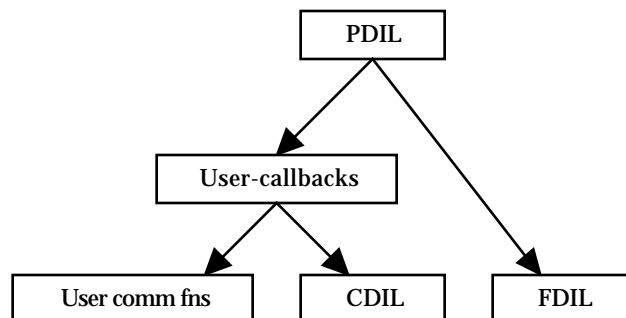
<<mailto:DILs@newton.apple.com>>.

The current set of libraries includes:

- CDIL - Communications DIL
- FDIL - Frames DIL
- PDIL - Protocol DIL

The libraries are lightly coupled, and the application can use any number of them as it requires.

Though one of the goals of the DILs is that the developer can pick and choose which ones to use, there is some interdependence among the DILs. Also, there are some DILs which developers will pretty much always use, even though their use is optional.



The above diagram shows which DILs know about and use other DILs. The CDIL and FDIL are completely stand-alone; neither requires the use of any other DIL. The PDIL makes extensive use of the FDIL and require its presence. The PDIL can use the CDIL if the application developer desires, but it does so indirectly through the use of application-defined callback functions.

CDIL

The Communications DIL is the library of code handling communications with the Newton device. It presents a unified server-oriented interface that allows the application to listen for connection requests. Once such a request is received, the application can accept the request, after which it can exchange data with the Newton device via a two-way data pipe. When communications is complete, the application can close the pipe.

The biggest change between the 1.0 and 2.0 CDIL is that some functionality has been removed. Hopefully, we've only removed the stuff that got in your way, and not the stuff that you actually used. The result is that the API is much simpler to understand and use.

The CDIL supports communications over a variety of services, includes MNP serial, ADSP, TCP, and Communications Toolbox tools. Future versions of the CDIL may also support PPP and IrDA.

FDIL

The Frames DIL is a desktop implementation of the Newton frames data storage and access model. With it, you can read objects sent from a Newton OS device, manipulate the objects (including creating new ones), and send objects to a Newton OS device.

FDIL 2.0 brings vastly increased functionality over the library it replaces — the HLFDIL 1.0. Specific APIs have been added to manipulate most of the well-known object types (integers, strings, frames, arrays, reals, generic binaries, etc.). Additionally, extensive support for Virtual Binary Objects has been added, allowing access to Newton Works data structures previously not possible with the 1.0 FDIL.

Some of your least-loved 1.0 features (like `FDbindSlot` and the unbound slot list) are gone, replaced with an easier to use set of APIs. Also gone is the FDIL's dependence on the CDIL when streaming; the streaming interface has been genericized so that you can stream objects to and from anywhere you choose.

PDIL

The Protocol DIL is the library which uses CDIL & FDIL to communicate with the Dock application in the Newton ROM. It knows about the set of commands (the protocol) used to send information to and receive information from the Dock application. These operations include: connecting to the Dock application, getting a list of stores and soups, reading the entries stored in the soups, and adding new and deleting existing soups and soup entries.

As with the 2.0 FDIL, the 2.0 PDIL is no longer dependent on the CDIL for communications. Instead, the developer is free to specify any form of communications through the installation of callback functions. These callback functions can, in actuality, just turn around and call CDIL functions, or they can implement any other kind of communications protocol of your choosing, in case the CDIL is not sufficient or appropriate.