Developer Relations Group

Hybrid Application Cookbooks

# Connected CDs

Intel Corporation
Developer Relations Group
*www.intel.com/drg*

**Disclaimer**

Information in this document is provided in connection with Intel products. no license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document. Except as provided in Intel's Terms and Conditions of Sale for such products, Intel assumes no liability whatsoever, and Intel disclaims any express or implied warranty, relating to sale and/or use of Intel products including liability or warranties relating to fitness for a particular purpose, merchantability, or infringement of any patent, copyright or other intellectual property right. Intel products are not intended for use in medical, life saving, or life sustaining applications. Intel may make changes to specifications and product descriptions at any time, without notice.

Copyright © Intel Corporation (1997). Third-party brands and names are the property of their respective owners.

# TABLE OF CONTENTS

# 1. Introduction

The explosive growth of the Internet and the World Wide Web has introduced the world to new ways of working, playing, and communicating using computers. Applications that combine the resources of the Internet with those of a local PC are known as hybrid applications. Hybrid Connected CD applications take this approach further, and combine the storage capabilities of CD / DVD-ROM drives with the Internet's resources and two-way communication abilities to deliver rich multimedia content and interactivity within applications.

This cookbook provides the application and content developer with an overview of hybrid Connected CD applications, provides guidelines for creating Connected CD applications, and discusses some of the issues associated with this exciting new application model.

The Connected CD is more that a melding of two different applications types. It is also a melding of two different development and business methodologies. Most readers of this cookbook will find that they will fall into one of two different camps. Either they develop applications for the CD / DVD-ROM or they develop applications for the World Wide Web on the Internet. For each camp there are distinct differences in approach to presenting content and interactivity to the end-user.

The typical Web developer will most likely rely upon the user's web browser and will create content in HTML format. The issues of distribution center around how to make the public aware of the website and how to entice them to return frequently. The typical CD-ROM developer will most likely create a custom application or use a CD-ROM authoring tool. Distribution issues center around how to get space in the retail channel.

The Connected CD combines these two different methodologies. For a given application, a well-designed Connected CD draws upon both the CD-ROM and the Internet technology and business models by combining the most appropriate attributes from both. This creates a new product category that provides the end-user with a greater experience than could be realized from either CD-ROM or Internet alone. For more information on hybrid applications see the developer relations website at www.intel.com / drg.

## 1.1 Why Should I Create a Connected CD?

The quality of multimedia content available to PC users has increased dramatically as new, faster processors and CD-ROM drives have become available. Today's PC users have come to expect content that offers them a rich, interactive environment, and they expect to see the same level of quality on all types of applications, even over the Internet. Realistically, however, the Internet by itself can not provide the bandwidth to deliver high-impact multimedia. For this reason, the application developer needs to be aware of the advantages and limitations of both the PC and the Internet and design his or her application accordingly. By understanding the capabilities of the local PC and Internet communication, the developer can deliver applications that set his or her company apart from the rest of the market.

# 2. CD-ROM + Internet = Great Hybrid Application

It is interesting to compare and contrast the advantages and disadvantages of the CD-ROM and the Internet to show how these two technologies are very compatible. The Internet provides a very interactive and timely experience, yet it lacks the ability to deliver rich multimedia. The CD-ROM can deliver a rich multimedia experience, yet it lacks the ability to deliver timely data. Let’s take a closer look at the attributes of the content delivery for both mechanisms.

## 2.1 Speed

Consumer application developers must face the reality that the data pipe to the home will remain thin for some time to come. There is a good news/bad news story here: The good news is that developers today can rely upon the fact that almost every PC shipped today includes a 33.6 Kbs modem. The bad new is that Analog modems are the common denominator, and will remain so until higher-speed infrastructures become common. The bottom line is that the Internet’s capability to deliver multimedia is significantly diminished by the low bandwidth provided by the modem.

More good news: For delivering multimedia, there is no comparison to the CD-ROM. PCs today are capable of displaying rich, compelling multimedia applications. High-quality multimedia requires the use of high bandwidth delivery mechanisms, and this demand has been met with CD and DVD-ROM drives. A 4X CD-ROM can deliver 4800 Kbs. The baseline speed for DVD-ROMs start at the equivalent of a 10X CD-ROM and will go up from there. CD-ROMs are clearly the preferred medium for delivering high bandwidth data.

## 2.2 Bulk and Size

Both CD/DVD-ROM and the Internet offer attractive features for the delivery of large sets of data. With the Internet it is possible to offer end users access to an immense data set, limited to the number of hard drives and servers on the Internet. The CD at 600MB and the DVD starting at 4.7GB are impressively large, yet dwarfed by the size of the Internet.

However the Internet only provides *access* to the large ocean of Internet data. The user is still constrained to drinking it from a modem-sized straw. The CD/DVD-ROM offers the ability to deliver a considerable bulk of data all at one time. With a 33.6 modem, it would take a user over two days to download a CD-ROMs worth of data.

## 2.3 Interactivity

The Internet can provide a whole new interactive dimension to the CD-ROM experience. Social experiences such as text and audio chat are simple examples. New environments are emerging on the Internet with persistent characteristics allowing users to create, build, and share with others. This new Internet community can be used to enhance the experience of the CD/DVD-ROM.

## 2.4 Timeliness

The process of publishing CD-ROMs is cumbersome in comparison with Internet publishing. With the Internet one can publish data dynamically, updating it continuously and/or automatically allowing the ability to deliver timely data. Most will agree that it is wholly impractical to create and distribute CD-ROM content on the same time scale as the Internet.

## 2.5 Persistence

From the user's perspective, one of the problems with the Internet is that one cannot rely upon the ability to access the same data day after day. The Internet changes continuously, new content appears, and other content disappears. Users appreciate the ability to be able to access the some types of content multiple times. CD / DVD-ROM provides a good mechanism for data that is repetitively viewed.

In summary, the benefit of the CD is its ability to persistently deliver large multimedia assets and bulk data. The Internet provides timeliness, interactivity and access to immensely large sets of data.

### 2.6 The Best of Both Worlds: The Connected CD

For different types of data and for different uses of data, the CD and Internet are very compatible delivery mechanisms, each compensating for the other's weaknesses. By combining the capabilities of the local PC, the cheap mass storage of the CD / DVD-ROM, and an Internet connection, it is possible deliver an awesome multimedia application with extended interactivity that offers the users an incredible Internet experience.

The production of an application that achieves this can give the developer that differential edge in the marketplace. As for marketability, both technologies gain by the advantages of the other. Hybrid Connected CD-ROMs can draw more users online and websites can refer to and enhance CD-ROMs.

As the methods of extending the scope of an application’s content across the Internet become better understood by independent software vendors (ISVs) and end users alike, the superiority of such titles will become more and more apparent. Because users will first demand, and then expect, these qualities in content-based applications, ISVs that master these hybrid techniques will first derive a premium return and count themselves among the leaders in this highly competitive environment.

There is no limit to what can be done. Very aggressive rich connected media can be delivered. A great example is the Infinite CD project using the connected CD methodology, combined with push technology and Intel’s Intercast ™ technology. For more detailed information on the Infinite CD project see Appendix C.

## 3. Connected CD Key Features

There are several key features to a successful Connected CD. This chapter describes these features. Chapter 4 discusses these features in greater detail and gives examples of how to implement them.

### 3.1 Content

By far the most critical part of application development is the creation of compelling content. Interesting and exciting content is what makes a top-selling title. Rich multimedia technologies such as video, 3D modeling, soft MIDI, software signal processing all demand bandwidth usually unavailable via a 28.2 kps modem, but are easily handled by a client’s high-powered CPU and graphics subsystem. The Connected CD directly addresses this issue by using the CD-ROM to deliver the multimedia content. Content on the CD can consist of stereo audio tracks, full-motion video, high resolution 3D textures…basically any media file that is undeliverable by the Internet.

### 3.2 ICTWTDICF: I Can’t Tell Where The Data Is Coming From

The function of the application is to present data (content) to the user independent of its source or retrieval method. It is important to realize that on a Connected CD, content may be retrieved from CD / DVD-ROM, the local hard drive, the Internet, or some combination of these. The retrieval of this content should be seamless, that is, without jarring the user with different UIs for CD content and remote Internet content.

If content is located on the Internet, it may be helpful to notify the CD-based application's user interface when an attempt is made to access remote data. This is especially useful to users who are paying for ISP services by the minute. The developer may wish to enable certain properties to be user-defined, such as the setting precedence of searching resource locations or distinguishing the source of content by color-coded links. Likewise, if the application is Web-based and needs to access a CD, the user may be alerted ahead of time to insert (or purchase) the CD before selecting a particular link.

## 3.3 Seamless Connectivity

The success of a Connected CD will depend on how easily and quickly the application connects with little effort on the part of the user. The application should be intelligent enough to determine if the user has an installed connection to the Internet, or if the user needs to set up an ISP account. Also, the application should have the ability to invoke the dial-up method when an Internet resource is requested, and to disconnect when the application has retrieved the necessary data.

## 3.4 Prefetching and Caching

Citing the lack of sufficient bandwidth as the primary limitation of today’s Internet applications is often an oversimplification when the issue is in fact *instantaneous* bandwidth. It may turn out that the existing Internet connection is capable of transferring content to the client, over the duration that the application is active, in sufficient volume to fully occupy the users attention. The techniques of prefetching, caching, and the web surfer’s equivalent of branch prediction can be combined to minimize otherwise wasted connection idle time, and apparent transfer latency. Clever content organization can create opportunities for prefetching and caching data.

## 3.5 Content Protection

The ISV’s business model may require that either, or both, local media-rich content and Internet-based data be protected from unauthorized access. In such a model, a Web-based application could unlock assets on a local CD or DVD-ROM, possibly at a predetermined time concurrent with the release of related material, such as a concert or album release. Likewise, an application running off a CD or DVD-ROM could present a password to unlock a website protected from web surfers who do not posses the application.

## 3.6 Scalability

To provide the best experience for users while supporting the largest installed base, it is necessary to scale content to take full advantage of the PC’s processing and connection speeds. Scalable content can take the form of media compressed at multiple levels of data and frame rates. Another method of creating a scalable application is to use different capabilities of the processor technology.

### 3.7 Flexibility

Judicious allocation and duplication of content between local storage and the Internet can enable an application to function satisfactorily in the absence of either the local mass storage medium or an Internet connection. The objective should not only be to minimize user disruption and frustration, but to genuinely offer satisfactory, albeit a lesser level, of gratification for their efforts. For example, a Web-based application that accesses a CD-ROM to play a video clip might offer a slide show if the CD-ROM is not present. Likewise, a CD-ROM-based title should still offer local content and functionality to a reduced degree in the absence of the timely data offered by the Internet connection.

## 4. Building the Connected CD

This chapter explores in detail each of the key Connected CD features described in Chapter 3, and provides some examples of how to implement these features.

### 4.1 Where Do We Start?

A Connected CD application can be classified with either of two basic categories: the CD-based application or the Web-based application. How you create a Connected CD depends in part upon which of these categories you're application belongs to. We'll start by defining these categories and giving an example of each.

A CD-based application is one in which the user interface exists on the client's local hard drive or CD/DVD-ROM. The application can be used to a limited degree without connecting to the Internet, but a much richer, timely experience is achieved when a connection is established. An Infinite CD is a Connected CD that integrates other hybrid application technologies such as Push and Intercast$^{TM}$ Technology. Intel, with the help from Macromedia , Intel, Marimba and the Public Broadcasting System (PBS), are currently working together on an Infinite CD pilot (as was previously announced at the recent Web Developer's Conference in San Francisco). As a pilot, it's anticipated the project will enable PBS to deliver an extremely rich Internet-connected application, targeted to their viewer base. Marimba's Castanet technology is used to create specific channels for the delivery and receipt of multimedia content to the client's hard drive.

A Web-based application is one in which the user interface exists on the application's website. Some content can be accessed via the Web, but certain areas make calls directly to a complementary music CD or CD/DVD-ROM on the client's system, accessing audio tracks or video clips that complement the website. Sony Music's CD EXTRA is an example of a Web-based application, which merges Sony Music with Macromedia Shockwave technology to deliver an interactive audio experience from the Sony website[i].

The following sections describe the basic components which can be used for building either CD-based or Web-based Connected CD applications. When necessary, specific requirements for either type of application are given.

### 4.2 Content

As stated earlier, the most critical work in application development is the creation of compelling content. Interesting content is what makes a top-selling app. This section doesn't go into content creation, but rather discusses the user interface of a Connected CD and how to partition content on CD and the Web to best take advantage of the benefits of each.

### 4.2.1 User Interface

#### *4.2.1.1 Browser Look*

Applications that access websites need to understand and / or accept HTML-coded files. The simplest way to do this is to create the application GUI itself as an HTML document in the form of a web page. An example of an application that has done just that is Books That Work's *Home Improvement Encyclopedia** (HIE). With an interface that runs in the Netscape* Browser, HIE has the advantage of both guaranteed compatibility, and ease of use for users who are familiar with surfing the Internet.

#### *4.2.1.2 Customized Look*

Developers who want a customized look may choose instead to create their own browser that has more the appearance of a stand-alone application. There are several possibilities for customization, from simply changing the appearance of a Netscape or Microsoft Internet Explorer* (IE) browser window, to completely replacing the browser look and feel with a standalone look.

For the first option, Microsoft and other companies provide tools for customizing the browser window. For example, Microsoft's Internet Explorer Administration Kit (IEAK) is a free software package that allows customization of the IE web browser. It allows the developer to insert a company logo, to create a single, one-click installation package of all the IE components, and to preconfigure and control options.

Completely replacing the browser requires a bit more thought and some programming. Below are three simple examples of how to create a basic browser window to act as a basis for a hybrid application.

## Example 1: Using Microsoft Visual Basic 4.0 or CCE 5.0

When you install Internet Explorer 3.0 you get two ActiveX controls that allow access to the Web. These are called the "Microsoft Internet and Shell Controls," and they give you the Web Browser control and the Shell Item control. Using the Web Browser control, you can create a simple browser in Visual Basic with one line of code:

1) Create a new form in Visual Basic.
2) Add the Web Browser (shdocvw.dll or shdocvw.ocxii) control to the toolbox. This file is usually located in the Windows\System directory.
3) Add the Web Browser to your form.
4) Add a Combo Box to your form.
5) Add a Command Button to your form.
6) Edit the “Click” procedure code for your Command Button by adding the following line: WebBrowser1.Navigate (Combo1.Text)
7) Run the app by typing a URL in the Combo Box and then clicking on the Command1 button.

You can even build your Visual Basic project as an ActiveX control and then call it from an HTML file, creating a browser within a browser, if you will.

## Example 2: Using Microsoft Visual C++ 4.2

1) Create a new project workspace.
2) Choose OLE control. If the choice to display the control in a dialog box is picked, the control will automatically be registered. This enables the control to be inserted into the test container for debugging purposes.
3) Add Properties, Methods, and Events to control. Add property pages for access to control properties during use in browser.
4) Create CAB and associated INF file. Authenticate code if it is to be released publicly.

Refer to Appendix A for examples.

## Example 3: Using Macromedia Director 5.0

Several companies today provide plugin tools for creating customized browsers with Macromedia Director 5.0. One such company, Human Code[iii], has created a technology they call XtraNet, which allows Director and Shockwave interactive movies to communicate with each other via Netscape or Internet Explorer. XtraNet tools include Content Linker, which allows posting to and from websites and real time updates to hybrid applications created with Director 5.0. The Content Linker connects the application to a Web server for downloading and formats new content to fit into existing title interfaces. Palladium Interactive used Human Code to design its Connected CD children's adventure/educational title, Wishbone.

WebXtra, by Tabuleiro da Baiana Multimedia[iv], is another tool which uses the Microsoft Internet Explorer engine to display HTML pages directly in the Director stage window.

Allegiant Technologies, Inc. created Marionet[v], the first complete, script-level interface to the Internet for Macromedia. It gives developers the ability to build custom Internet solutions using popular authoring tools--including Allegiant SuperCard, Macromedia Director, and Apple HyperCard, as well as popular business applications that support AppleScript or Frontier. Simple, one-line commands are all that is required to get HTML documents, navigate FTP sites, send or retrieve e-mail and so on. Marionet takes care of all the complexity of communicating with the Internet.

### 4.2.2 HTML Authoring

*4.2.2.1 Simple Authoring Tools*

For basic text and image pages, you can get away with some of the easy to use, “What You See is What You Get” (WYSIWYG) HTML Authoring tools. Three popular examples are listed below:

1. Netscape Communicator* or Netscape Navigator Gold* provides a nice WYSIWYG environment for HTML authoring.
2. Microsoft FrontPage* is another easy to use HTML authoring application for simple text, and image-based pages.
3. Adobe PageMill*: Just released for the Windows-based PC, Adobe PageMill is yet another popular WYSIWYG web page authoring tool.

Most likely, an HTML-based application will go beyond the basic requirement of static text and images, and more advanced HTML authoring will be necessary. Some popular examples are discussed in the next section.

*4.2.2.2 Advanced Authoring Tools*

To extend the capabilities of an HTML page beyond the basic text and image capabilities will require the use of web extensions. Web extensions fall into three basic categories: helper-apps, plugins, and applets. Some tools used to create web extensions are listed below.

*4.2.2.2.1 Macromedia Director*

Macromedia Shockwave can be used to convert Director movies into Web-friendly form. Shockwave files are viewed with the Shockwave plugin.

Macromedia Shockwave can be used to incorporate interactive Director movies into web pages to enrich the user's experience on the Internet. Sony Music presents a new format CD EXTRA, an enhanced CD that merges Sony Music with technology such as Macromedia Shockwave. CD EXTRA offers the consumer a full audio album when played in your audio CD player and an interactive experience on a computer's CD-ROM drive. See the CD Extra section in Appendix B for details.

In order to run a Macromedia application from the Web, the following client and server requirements must be met (assuming that we are running Netscape).

1. Shockwave plugins must be installed for the client's browser. The plugins can be obtained from Macromedia. (http: / / www.macromedia.com)
2. The server side must set up a MIME type for the Shockwave plugins.

When the client tries to access the Shockwave file on the server, the server will recognize the file as a Shockwave application through its mime type. The server will then request the client to launch the Shockwave plugins. Once the plugin is launched, the Shockwave file will be downloaded and executed.

*4.2.2.2.2 Microsoft ActiveX*

ActiveX is Microsoft's answer to integrating the Windows operating environment with the Internet. ActiveX is an Object Linking and Embedding(OLE) wrapper around the Win32 Application Programmer Interface (API) and the Component Object Model (COM). ActiveX enables standard Win32 applications to communicate over the Internet without having to develop everything from scratch. ActiveX components can be developed with Visual Basic 5.0 Control Creation Edition or Microsoft's Visual C++ 4.2, and are embeddable in Internet Explorer 3.0 or any OLE Container.

Using Microsoft Visual C++ 4.0 or greater, or Microsoft Visual Basic 4.0 or higher, you can create ActiveX controls to run as separate downloaded applets that use the Windows OLE messaging system to run on client systems with Windows-based operating systems.

*Microsoft ActiveX Control Pad*

If you are using several ActiveX controls, you may want to organize them with Microsoft's ActiveX Control Pad*. The ActiveX Control Pad lets you create an HTML layout in a visual environment similar to Visual Basic for your web page. This interactive tool makes it easy to use controls and or Java applets to build a web page, using either VBScript* or JavaScript*. The layout is saved as an .alx file, which is a simple ASCII file that contains the ActiveX control information for a page. It is interpreted by the HTML Layout control, an ActiveX control itself.

*Registering ActiveX Controls*

Before a user can run an ActiveX control from their local system, the control must be registered on their system. For examples of registering ActiveX controls, refer to Appendix A.

*ActiveX Support for Netscape*

While Netscape 3.0 does not support ActiveX controls, Ncompass Labs[vi] has created plug-in support for many of the most popular ActiveX controls, and it is expected that a future release of Netscape will provide native support for ActiveX.

*4.2.2.2.3 Java*

Java-based applets run on a temporary Java Virtual Machine, which is loaded on the client system while the applet is running and unloaded when it is finished. This makes Java applets platform-independent. The security features of Java however restrict access to specific information about the client's system.

*Intel Media for Java*

Intel Media for Java* is a set of tools which Java developers can use to create high-performance, engaging, media-rich applications for the world's largest group of Web clients. The tools can be used to create Java-based applications and applets for Web- or CD-based Connected CDs. Intel Media for Java provides cross-platform APIs optimized for MMX™ technology and are scaleable to processor and bandwidth requirements. To build the application or applet, a Java developers kit, such as Microsoft Visual J++*, Sun's JDK or Symantec Café* will be needed. The beta version of the Intel Media for Java package, which is currently available for free download from the Intel Media for Java Home Page[vii], (URL: http://www.intel.com/ial/jmedia/index.htm) consists of:

> *Java Media Framework:* Part of the Java Media API from JavaSoft[viii], the Java Media Framework provides the necessary support to incorporate rich video and audio into your Java Applets and Applications.
> *Intel Animation for Java:* Allows for the development of high performing, high quality and extremely flexible 2.5D animations.
> *Intel Spatial Audio for Java:* Introduces Java developers to 3-dimensional sound.

### 4.2.3 Data Partitioning

As we discussed earlier, there are advantages to both the CD and the Web as vehicles for delivering application content. CDs are better equipped to handle large, media-rich content and the Web is superior in its timeliness and communication aspects. Keeping these advantages in mind, you can build a hybrid application that is both rich and timely by partitioning your content accordingly.

*4.2.3.1 CD or Web?*

Careful consideration needs to go into what would be better located on the CD or local hard drive, what could go on the Web, and what would be the best means of accessing the web data. For a start, consider some cold hard facts. Single-speed CD-ROM drives are practically non-existent today, but we'll look at them as a worst case example. A single-speed CD-ROM can deliver 150 kilobytes of data per second, or 150,000 kps. Now look at the typical modem available today, which gives you 28.8 kps. This means the slowest CD-ROM you'll ever encounter can deliver data about 41 times as fast as anything you'll be able to pipe through a phone line.

#### *4.2.3.2 Rich Multimedia Content on the CD/DVD-ROM*

PCs today are designed to handle a wide range of rich multimedia content. High speed CD / DVD-ROM drives, combined with the processing power of Intel Pentium® processors with MMX™ technology and supporting high speed graphics hardware gives the user the best possible experience. Your application can shine with full-motion video, CD-quality audio, and high resolution 2D and 3D graphics played directly from the CD.

Connected CDs that are Web-based can also take advantage of local storage and processing muscle by accessing the CD / DVD-ROM when rich media is desired. The content on the CD may be “locked” until a certain date or until the user has accessed a particular location on your website.

#### *4.2.3.3 Timely Content on the Web*

By the very nature of the Internet, content on websites can be updated, increased, and modified at any time, and the changes will be immediately visible to the entire audience. Current events, live performances and application updates can be provided to everyone who owns a copy of your application in real time and at a cost much lower than it would cost to produce a CD upgrade. There are, however, limitations to what you will be able to deliver. Considering the fact that a large percentage of your customers will be connecting to the Web via 14.4kps and 28.8kps modems, you’ll be limited to smaller files and less rich content. You won’t be able to hold your client’s attention (or connection, most likely) while you download a short full-screen, full motion video clip that’s 100MB in size. Instead of employing high-bit rate audio and video, technologies such as streaming audio and video can be employed. Progressive Networks RealVideo offers high-quality live and on-demand streaming audio and video. VDO and Xing Streamworks also have tools for creating and distributing streaming video.

In addition to timeliness, the Web also provides new levels of interactivity for your customers, both with you, the developer and with other users of your products. By monitoring your website for page hits, you can identify which content your customers find most interesting and continue to tailor your application to customer interests long after they have purchased the product. You may also choose to include on your website questionnaires, contest forms and support requests, thereby reducing the need for phone calls and direct mailings.

Some personal productivity applications, such as Broderbund’s Family Tree Maker* allow the user to contribute to the hybrid application, increasing its value by creating and posting web pages with the user’s personalized information.

Applications can exploit the growing number of companies using e-commerce technologies by including hyperlinks to supporting websites. Books That Work* has included on-line purchasing capabilities into their *Visual Home** hybrid CD-ROM. By selecting a refrigerator within the application’s 3D world, a user can be connected to General Electric Co.’s website and make a purchase from within the application. This provides advertising for GE and Books That Work benefits by selling manufacturers digital catalog space on their CD.

### ***4.2.3.4 Additional Considerations***

#### *4.2.3.4.1 CD-Based Applications*

If you are creating a CD-based application, you’ll need to include the application interface and any required connection software on the CD. Providing a quick, simple way for Web beginners to establish an ISP account on your CD will add value to your application for the users and will help ensure that they experience the full potential of your Connected CD application. In some cases it might be important to provide basic level ISP support for users and / or “1-800” access to your Internet-site only

*4.2.3.4.2 Web-Based Application*

If you are creating a Web-based application, the user interface will sit on the web server side, along with the controls for accessing the hidden data on the CD. In Appendix A, you’ll find two examples of such controls. The first is an ActiveX control for accessing and playing an audio track from a CD on the client’s system. The second is an ActiveX control for accessing and playing a video from a CD / DVD-ROM on the client’s system.

## 4.3 ICTWTDICF: I Can’t Tell Where The Data Is Coming From

The primary function of the application is to present data (content) to the user. This presentation need not be segregated by the source nature of the content, i.e. whether it comes from the Internet, a CD-ROM or from a cache on the user’s hard drive. The organization of content as it is presented to the user should be driven only by its relevance, not its source. For example, an encyclopedia title may choose integrate the results of the most recent World Series (Internet data) with the static information on baseball that resides on the CD-ROM.

The first step toward this goal is to use a common user interface for viewing data from all sources. This user interface must me capable of identifying which content assets reside on the local system. In general, content on the local system takes priority over content from the Internet because of the much greater access performance. On the local system, content that has been “pushed” onto the hard drive takes priority over content on the CD-ROM because it is more current. The process in the client that determines which content assets are display is called Link Resolution.

### 4.3.1 Link Resolution

There are two methods for applying link resolution process to a client application: client-sided and server-sided resolution. Client-sided resolution does not involve a server, except for serving standard HTTP requests. The client determines the locality of the resources and dynamically changes the location of these resources before displaying the content to the user. Server-sided link resolution entails determining what resources the client has available locally, then, depending upon that information, dynamically creating a web page with the resources either pointing to the local or remote sites.

On the client-side, implementation of link resolution can take many forms: an ActiveX control, a privileged Java application, VBScript, or Java script. All or any combinations are applicable methods of communicating to the server. On the server-side, implementation can be done using CGI scripts, Active Server Pages, or HTTP requests. All or any combination is an applicable method of communicating to the client.

### 4.3.2 An example of Client-sided link resolution:

Link resolution can be implemented entirely by the client application without server intervention. This can be accomplished by intercepting the user request for an Internet resource, then searching for the content asset in local storage. If found, the access to the Internet is redirected to the local system. The local resources can either be a CD-ROM that is distributed by the ISVs, or resources on the hard drive that are created by push technology such as

BackWeb*[ix], PointCast*[x] or Marimba* (See the *Information Delivery* cookbook for more information on push technology).

After the application intercepts the user's request for resources, it must search for these resources locally. Many models could be applied at this point, but, for a working example, assume that a stack model is used in which the hard drive cache is searched first, then the CD, then finally content is requested from the server. (Refer to Figure 1) This follows the idea that the hard drive contains the most current resources, the CD has static high bandwidth material, and the Internet provides the slowest, but possibly the latest, content.

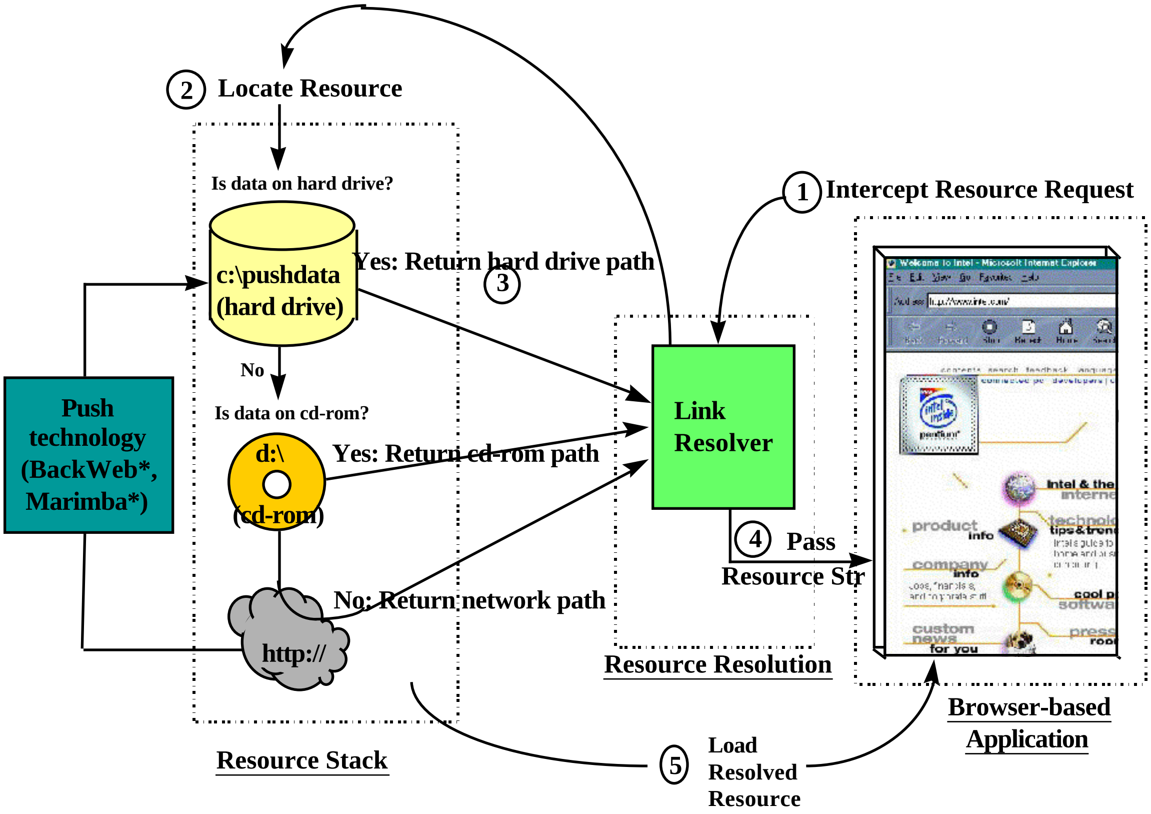


**Figure 1: Client-based Link Resolution**

Browser-based applications can be created using tools described in sections 4.2.1 and 4.2.2.

Steps to implement a browser-based application with link resolution:

1. Intercept User Resource Request
2. Determine Available Stack Of Paths.
   - Hard Drive Cache
   - CD-ROM Drive
   - Internet
3. Create Full Resource String for application

   E.G.

   Resource path = CACHE_PATH = “C: \cache” + FILE_PATH

   or

   Resource path = CD_PATH = “D:\” + FILE_PATH

   or

   Resource path = INTERNET_PATH = “http:/ /” + FILE_PATH

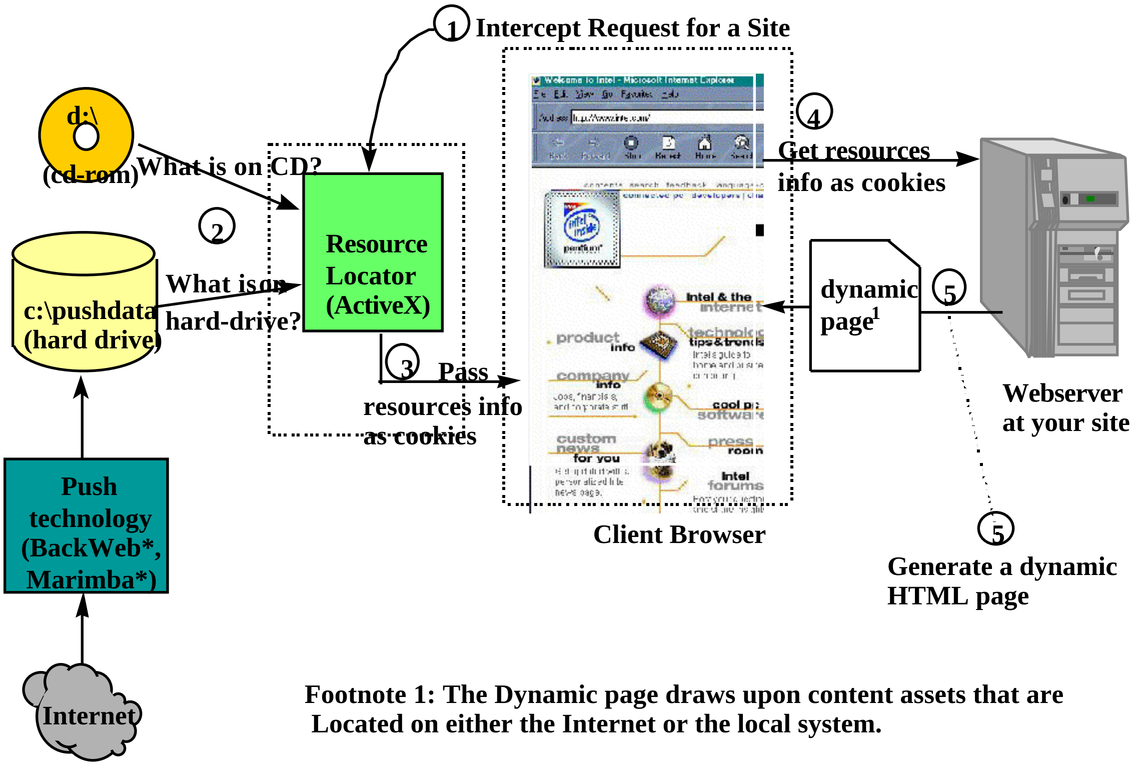
4. Pass Resource String to application for loading
5. The browser-based application loads the resolved resource.

For an MSVC--ActiveX implementation example see the Link Resolution Control in Appendix A.

### 4.3.3 An example of Server-sided link resolution: Link Resolution from the Internet

Link resolution can be incorporated into a website by providing resource resolution used by the browser. (Refer to Figure 2) This can be accomplished by intercepting the users request for an HTML page on your website, then determining which resources exist on the local system, such as: CD-ROM with label “Media01” is in the CD-ROM drive, and/or package XYZ is presented on the hard drive. The information is sent to the server located at your site. The server uses the information to generate a dynamic HTML page and return this to the client browser. The local resources can either be a CD-ROM that is distributed by the ISVs or resources on the hard drive that are pushed there by some push technology such as BackWeb*, PointCast* or Marimba* (See *Information Delivery* cookbook for more information).

After intercepting the user’s request for an HTML page, the application must identify which resources exist on the local system. This can be done by using an ActiveX control or any other mechanism that can access the local file system. After determining which resources exist on the local system, the information is sent to the server by setting cookies or passing them as HTTP request parameters. This can be done using JavaScript or VBScript scripts. At the server, the cookies or parameters are read and a dynamic HTML page is generated and returned to the client’s browser.  Dynamic HTML pages can be generated using Active Server Pages or CGI scripts.



Footnote 1: The Dynamic page draws upon content assets that are Located on either the Internet or the local system.

**Figure 2:  Server-based Link Resolution**

Steps to implement website with link resolution:

1. Intercept User Request for your site
2. Determine what resources are presented on your local system.
3. Pass resource information to the server as cookies or HTTP requests parameters.
4. On the server, get resource information from parameters or cookies.

5. Generate a dynamic HTML page and returns to the client application.

The major difference between client-sided resolution and server-sided resolution is based on information on the local system; the server decides what content the client will see. The server might decide to ignore all content on the local system and provide content from its website over the Internet.

## 4.4 Seamless Connectivity

Considering the fact that a hybrid application, especially a Web-based hybrid application, requires an Internet connection, the ease and efficiency with which a user can establish a connection is probably the single most crucial factor in hybrid application design. An app that leaves it all up to the user will most likely reduce the frequency and/or appeal to users of the Internet content-side of your Connected CD. To make a hybrid application successful, simple and seamless connectivity is a must.

### 4.4.1 How to Get Internet Connectivity

Getting an Internet connection is much trickier than picking up a phone to make a call. Many of today's computer users have avoided getting connected to the Internet because of the complexities involved. Questions about what service is needed, what software is needed, PPP or SLIP, etc. are all issues that many people simply don't have the time or inclination to explore.

*4.4.1.1 Connections and services*

If you're planning to connect to the Web, make sure the user has an ISP that is full-featured. Some ISP packages still offer only shell or Telnet access, which is text-based only. For fast, seamless connectivity, the connection type should be either SLIP or PPP.

*4.4.1.2 Browser Setup Software*

Whether your user interface design is based on a browser technology or a custom built application, an easy way to provide Internet connectivity for your customers is to include an Internet browser on your CD-ROM. Today's most popular browsers such as Microsoft Internet Explorer or Netscape Navigator, come with their own easy to use connection setup software. Including one of these browsers on your application CD gives users a simple, standardized setup program and provides them with a list of ISPs available in their area code. The drawbacks to this approach are that you have to obtain a license agreement from the company that produces the browser and that you have to give up space on your CD to accommodate a complete browser program, most of which you may not need. The advantages are that you don't have to create your own connectivity program and that you are potentially adding value to the CD for users who have previously had no Internet access at all.

*4.4.1.3 ISPs*

According to one ISP directory service, there are over 4,445 ways to connect to the Internet. If you are creating your own connectivity setup program and you plan on listing suggested ISPs for your connection, it is important to do some research first. Some may provide only SLP connections and some may require their own proprietary browser. To help you get started, c|net has compiled a list of the top ISPs, based on their own criteria[xi]. See URL: http://www.cnet.com/Content/Reviews/Compare/ISP/national.html c|net Top ISP Listing.

### 4.4.2 How to Seamlessly Connect to the Web

To maintain seamlessness in a Connected CD application, you will want to automatically connect the user to the Web when needed without any intervention on their part. The next sections give some examples of how this might be done.

*4.4.2.1 Windows 95 Dialup Networking*

Windows 95 includes a Winsock dialer in their Dialup Networking application. If your hybrid application is CD-based and uses standard Winsock calls, when the user clicks on hyperlink text that points to a location on the Web (i.e., the URL begins with http:// or ftp://), the Windows dialer application will automatically launch and attempt to connect via the user's chosen default method.

*4.4.2.2 Login Scripts*

If you expect the user to be working with a particular ISP and you know that there is a series of configuration steps required to complete a connection, you may choose to include with your application a customizable script that will issue the commands necessary to log into the ISP. In the case of a PPP connection, many providers do not require manual log in steps--a user name and password in the Connect To dialog box of the dialer will be sufficient, and no scripting will be needed. However, all SLIP/CSLIP connections, some PPP connections, and any other connection that requires menu selections, request advanced input that will require a script. For some examples of scripts for different ISPs refer to the Windows95.com website: http://windows95.com/connect/dscript.html .

*4.4.2.3 Connection Via WININET*

The WININET API provides a means of communicating across the Internet without ever having to code directly to a Windows socket. It provides standard protocol support such as FTP and HTTP. The API extends read and write capabilities across the Internet. Complete API details are located at http://www.microsoft.com/intdev/sdk/docs/wininet. A sample of forcing a connection to a particular website is provided in Appendix A.

## 4.5 Prefetching and Caching

The timeliness of the information available from the World Wide Web is what makes a Connected CD application so compelling. In the case of a Web-based application, additional information can be provided simply by creating web pages that unlock previously unaccessed content from the CD. Since the CD is local, opening and playing "new" files requires a simple command from the server.

In CD-based applications, new content is retrieved from the Web and must be downloaded to the client's system for viewing. Ideally, the download should appear seamless to the client, taking no longer than the amount of time it takes to access a file from the local CD/DVD-ROM drive. Since updates may involve the download of larger files, such as video and audio clips, certain technologies should be employed to facilitate updates from the Web. Some examples, such as push technologies and prefetching, are described below.

### 4.5.1 Push Technologies

Push technologies were developed, in part, to allow users to avoid the chore of constantly revisiting sites to obtain that site's latest content. Adding to the existing Web protocols, push technology delivers data to a client computer without the client doing anything more then subscribing or requesting the delivery. Push delivery can be used to deliver news, software updates, corporate announcements, or the joke of the week. The applications use local caching, prefetching and filtering techniques to make information available to the user--without the wait.

The table below lists some of the information push applications currently available.

| Product | URL |
|---|---|
| 1. AirMedia/Global Village Communications - NewsCatcher | http://www.airmedia.com |
| 2. Autonomy - Autonomy Agentware | http://www.agentware.com |
| 3. BackWeb Technologies - BackWeb | http://www.backweb.com |
| 4. Berkeley Systems AfterDark Online 1.07 | http://www.afterdark.com |
| 5. Caravelle - Transceive | http://www.caravelle.com |
| 6. First Floor - Smart Bookmarks | http://www.firstfloor.com |
| 7. inCommon - Downtown | http://www.incommon.com |
| 8. IBM - NewsTicker | http://www.infomarket.ibm.com |
| 9. Individual - FreeLoader | http://www.individual.com |
| 10. Intermind Corp. - Intermind Communicator | http://www.intermind.com |
| 11. Lanacom -- HeadLiner | http://www.lanacom.com |
| 12. Marimba - Castanet Tuner | http://www.marimba.com |
| 13. MSNBC - Personal News Page | http://www.msnbc.com |
| 14. Net Controls Corp/Yahoo! - My Yahoo! News Ticker | http://www.netcontrols.com |
| 15. NETdelivery - NETdelivery | http://www.netdelivery.com |
| 16. PointCast - The PointCast Network | http://www.pointcast.com |
| 17. Tierra Communications - Highlights2 | http://www.tierra.com |
| 18. US Interactive - Digital Bindery | http://www.bindery.com |

### 4.5.2 Prefetch Technologies

To improve latency in user-initiated events, such as clicking on hyperlinks or URL-calling objects within the application GUI, the application can make use of prefetching and local caching. The following sections describe many different types of prefetching, one or more of which may be effectively used by a CD-based hybrid application.

#### *4.5.2.1 Server-Initiated vs. Client-Initiated Prefetching*

Prefetching of web content may be initiated by the client or by the server. When the application requests a page from the Web, the server may anticipate what hyperlinks are likely to be selected, and preload the corresponding web pages to the cache on the client's system. This speeds up response time; when the user selects a hyperlink, the corresponding Web pages have already been transferred or may be in the process of being transferred.

For server-initiated prefetching to work, the client has to be prefetching-aware so it can deal with preloaded web pages correctly. This would require extensions to the current HTTP protocol and modifications to both the client and server software.

A client can also initiate prefetching web pages based on a user's configuration, or a user agent can monitor the patterns of past accesses for particular web pages, and prefetch web pages on the behalf on the user. Because client-initiated prefetching can be done by individual clients in a way transparent to the servers, the implementation is much simpler.

#### *4.5.2.2 Client-Initiated Prefetch Options*

Depending on how you design your hybrid application, you may be able to take advantage of any or all of the following client-initiated prefetch options.

##### *4.5.2.2.1 Batch or Off-hours Prefetching*

If you plan to update some or all of your website content weekly or less often, consider using batch prefetching. In this scenario, contents can be prefetched while the user's system is idle or engaged in minimal activity. Ideally, prefetched downloads could be scheduled to occur during the less busy periods of the day, both to speed up downloads and to reduce cost to the user . This type of prefetching can be done through cache proxies so that all requests are naturally aggregated at the proxies. Batch prefetching could also be scheduled to coincide with multicasting from the server when there is a need to distribute some large popular files, such as prerecorded concerts or news broadcasts, or to distribute software purchased online.

##### *4.5.2.2.2 Start-Up Prefetching*

Prefetching can also be carried out when a hybrid application is first launched. A set of pages that the users will need to look at that day may be prefetched in the background for later viewing as the application is loaded.

##### *4.5.2.2.3 Idle-Time Active Prefetching*

Since a user will usually spend some time (seconds or minutes) on a page, we can potentially pipeline the operation by fetching the next page while the user is looking at the current one. This is particularly useful for large documents, which can be divided up into smaller, quicker to download pages which are expected to be viewed sequentially.

##### *4.5.2.2.4 Adding Prefetch Functionality*

Prefetch commands can be as simple as an HTML <META> tag in the <HEAD> tag to instruct the browser to load another document after a specified amount of time:

> <HEAD><META HTTP-EQUIV=REFRESH CONTENT="*secs*;
> URL=*http://url of next document*"></HEAD>

where *secs* is the number of seconds to wait before pulling in the next page and *http://url of next document* is the URL of the page to be fetched. This method is limited, since once the number of specified seconds has passed, the next page will be displayed regardless of whether or not the client is ready to view it.[xii]

Shockwave movies can be made to link to web pages via URLs by using the Lingo* programming language. Commands such as gotoNetPage, gotoNetMovie and preloadNetThing offer a way to prefetch the next pages while the user is browsing the current pages.

The ActiveX Preloader control, one of the Microsoft ActiveX stock controls, can be used in an application to asynchronously download content to the client's cache. The control can be called using the following HTML code:

```
<OBJECT
      id=Preload1
       classid="clsid:16E349E0-702C-11CF-A3A9-00A0C9034920"
      width=1
       height=1
  >
  <PARAM NAME="URL" value="http://www.intel.com">
  <PARAM NAME="enable" value="1">
  </OBJECT></BODY>
```

In the above example, "URL" refers to the page to be preloaded, and "enable" determines whether or not to activate the control (value=1 to activate, value=0 to inactivate). Refer to Microsoft's website (http://www.microsoft.com) for more details on the Preloader control.

### 4.5.3 Keeping Track of Updated Content

#### *4.5.3.1 Index to Client*

How does the client side know when its content is outdated? One option would be to maintain a list of files stored locally in the system cache and have the server send a list of new content to the client. The hybrid application could then compare the two lists and request a client pull for the new files, or the server could do the compare and initiate a push of the new content.

#### *4.5.3.2 Cookie to Server*

Websites can track and control visitors via the cookie system. When a client accesses a website that uses cookies, their browser software is given a unique identifier which is stored on the client's hard disk. The browser presents this cookie to the server upon further access to the site. The server can use cookie information to deliver customized content to the client. Netscape also provides a temporary cookie which expires after a specified period of time.

One downside to this approach to identification is that a cookie can only identify a web browser, not the person behind it. This means if a user transfers to a different system, a new cookie will have to be created.

## 4.6 Content Protection

Usually, an ISV's content is what separates it from its competition. Content is a very valuable asset, therefore ISVs want to protect it from unauthorized use or copying. Also, an ISV's business model may require that either, or both, local media-rich content and Internet-based content be protected from unauthorized access. In such a model, a website could unlock assets on local CD or DVD-ROM, possibly at a predetermined time concurrent with the release of related material. Likewise, an application running off a CD or DVD-ROM could present a password to unlock a website protected from web surfers that do not posses the application.

There are many schemes used for content protection, from complex encryption using public or private keys to simple bit manipulation of content. This section is not intended to discuss encryption schemes or implementation details for these algorithms. The purpose of this section is to give ISVs an overview of some of the off-the-shelf technology, that they can incorporate into their application in a short period of time.

### 4.6.1 Digital Watermarking using SureSign Technology

SureSign[xiii] is a encryption technology that protects content by embedding an invisible and highly secure identifier or 'fingerprint' within a digital image. The fingerprint is designed to remain in place even after the file has been opened, edited or copied. The fingerprint is 'smart' and varies in intensity according to the information content of the object being fingerprinted, for example, a heavier fingerprint will be incorporated into areas where more detail exists compared to those areas containing less detail.

SureSign's sophisticated design ensures that applying a fingerprint to an image does not increase the image file size and would not degrade the reproducible quality of an image, allowing it to be printed or displayed as normally as a regular unfingerprinted image. It is also designed to withstand format conversions, compression, resizing, flipping and file transfers between computer systems.

Each SureSign fingerprint is made up of two alphanumeric IDs:

Fingerprint ID - a six digit unique code identifies the copyright holder of an image.

Image ID - a seven digit code (which is not necessarily unique) identifies a particular image.

SureSign provides the following products that either used to apply or detect fingerprints:

| Product(s) | Description |
|---|---|
| SureSign Pro | Apply and detect fingerprints |
| SureSign Writer | Apply fingerprints |
| SureSign Detector, CyberSleuth, Desktop Detective | Detect fingerprints |

### 4.6.2 Digital Watermarking using Digimarc Technology

A Digimarc[xiv] watermark imitates naturally occurring image variations and is placed throughout the image in such a way that it cannot be perceived. It automatically varies the intensity of the watermark in order to remain invisible in both flat and detailed areas of an image.

A Digimarc watermark is used to communicate the name and rights of the copyright holder, so that others know the property is not in the public domain, and the holder intends to retain his or her rights. With this information, the image consumer can determine where the image can be used and how to contact the image artist or distributor to obtain a copy. It can also be used to communicate basic information about the object that maybe useful in metering, scrambling, or tracking.

A Digimarc watermark is durable and survives across file formats and most image transformations, such as copying and editing. Although the watermark is embedded digitally within the image, it remains part of the image even when printed and can be read later simply by scanning the image into the computer.

Digimarc provides the following products that either used to read and write watermarks:

| Product(s) | Description |
|---|---|
| PictureMarc | Writer to embed watermarks and reader to find and read their contents. This bundle is integrated into Photoshop 4.0 |

For more information about the PictureMarc product contact Digimarc at http://www.digimarc.com/

**Note:** Currently, there is no software developer's kit (SDK) or Internet support such as: ActiveX or Netscape’s plug-ins for either Digimarc and SureSign. Also digital watermarking only works on still images.

### 4.6.3 Indeo® Video Key Frame Encryption

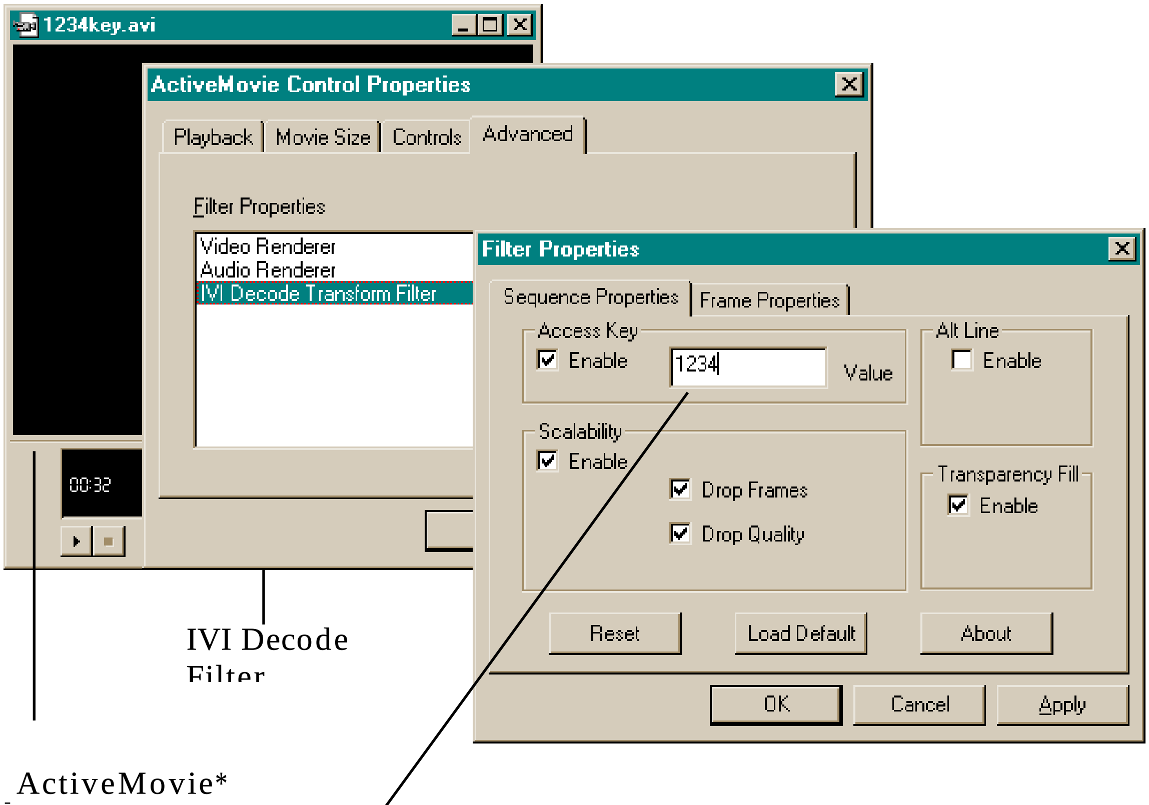
Indeo video interactive is the latest release of Intel’s PC video technology. It allows software-only playback of high-quality video on desktop multimedia PCs. It is Intel’s digital video capture, compression, and decompression codec. (A codec is a software driver used to compress digital video data for storage, and decompress it for playback on a multimedia PC.) Indeo video interactive includes a wide variety of features designed especially for interactive multimedia applications, such as:

real-time video effects,
the ability to decode only a portion of the frame,
and the ability to encode portions of a video image as transparent.

These special features take video beyond simple VCR-like functionality, allowing a new class of interactive applications that showcase the performance of the Pentium processor and audio/visual software such as Microsoft’s ActiveMovie*.

Video access protection is one of the newest features in Indeo Video Interactive. Intended for ISVs who are concerned about illegal distribution of copyrighted material, the Indeo video interactive codec helps prevent misuse of video clips by using access keys: numeric passwords that can be inserted into a video clip during compression. If a clip has been encoded with an access key, no application can play that clip without knowing the key.

<u>For example</u>: How to play back an IVI file that is encrypted with access key using Microsoft AcitiveMovie*.



1234key.avi

ActiveMovie Control Properties

Playback | Movie Size | Controls | Advanced

Filter Properties

Video Renderer
Audio Renderer
IVI Decode Transform Filter

00:32

Filter Properties

Sequence Properties | Frame Properties

Access Key
☑ Enable
Value: 1234

Alt Line
☐ Enable

Scalability
☑ Enable
☑ Drop Frames
☑ Drop Quality

Transparency Fill
☑ Enable

For example: How to programmatically play back an IVI file that is encrypted.

**Figure 3:  Accessing Indeo® Video Key Frame Encryption from Active Movie**

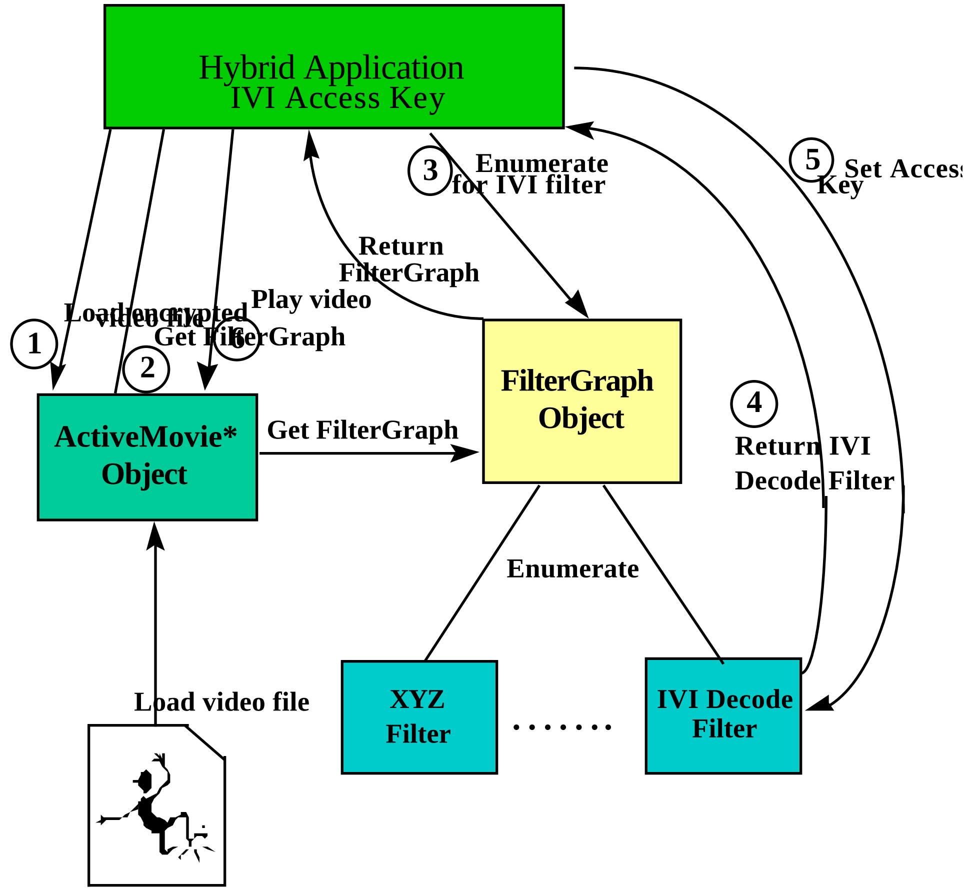


**Figure 4: ActiveMovie for Hybrid Applications**

Company: Intel
Product: Indeo Video Interactive
http://www.intel.com/

## 4.7 Scalability

Scalability is the ability of an application to fully utilize a platform's resources and capabilities to maximize the quality of the user's experience. Scalability can be applied in a number of dimensions. The performance of a hybrid application should at least scale with respect to the processor type and network connection type. Scaling to the processor type requires determining the speed and presence of multimedia enhancing technologies such as, the MMX™ technology. Network connection type implies scaling content based on speed or bandwidth.

### 4.7.1 Determining the Processor Type and Clock Frequency

Software can determine the processor type by using the CPUID instruction. According to the Pentium™ Processor User's Manual (p 25-62):

> The CPUID instruction provides information to software about the vendor, family, model, and stepping of microprocessor on which it is executing.*

Application Note AP-485 explains the use of this instruction in software and BIOS applications and tools. This application note, as well as additional information on CPU identification can be found via the following URL:

http://developer.intel.com/design/perftool/cpuid/appnote.htm

The CPUInfo package contains two DLLs which return CPUID information as well as processor clock frequency. It is available via the following URL:

http://developer.intel.com/design/perftool/cpuid/package.htm

It is possible to use the CPUInfo package to create a control that determines the processor speed of the client computer. This information can then be used to determine which algorithm to use for optimal decode performance and the quality of content to deliver.

### 4.7.2 Scaling Features to Processor Power

Knowing the type and speed of the processor, as well as the presence of the MMX™ technology enables the application developer to anticipate the CPU loading of various features of the application. This information enables the developer to dynamically determine what content or algorithms are used for optimal processor performance. Image size and resolution, frame rates, and audio signal processing functions such as 3D placement and reverberation, are but a few of an application's features that can potentially be scaled for processor power.

Available at http://developer.intel.com/design/contents.htm are the Realistic Sound and Display Mixers (RSX and RDX.) These API's provide multimedia support for the Windows operating environment and have been optimized for Pentium™ platform technologies, including enhanced performance when the MMX™ technology is detected.

### 4.7.3 Determining the Connection Speed

Determining the connection speed of a client logged in to a website requires sending a message to the client and measuring the latency waiting for an acknowledgment packet to be sent back. This will provide the server with the necessary connection information needed to determine what content is deliverable and by what method.

### 4.7.4 Scaling Features To The Connection

The ability to scale to the connection speed has greatest relevance in the area of streaming media, but could conceivably apply to the type of content downloaded. For example, if a browser plugin determined the latency of a connection to a website, it could base what content to deliver on this information. For those who have relatively fast connection rates with low latency, streaming media could be sent. If the connection is slow and latency high then a slide show could be substituted.

## 4.8 Flexibility

Providing both a rich and timely multimedia experience by using a high performance PC with an MMX™ technology-equipped processor to seamlessly combine the resources of the Internet with that of a local CD or DVD-ROM is only half the battle in creating a great hybrid application. What happens when the CD or DVD-ROM is missing? What happens if the computer is unable to make a connection to the user's ISP? In the absence of both of these resources, the user may barely be able to launch the application or view data cached to the local hard drive. However, the application should be designed to deliver a satisfactory, albeit a lesser quality, experience to the user with only one of these key elements present.

### 4.8.1 Performance Degradation with Neither a CD nor a DVD-ROM

The primary utility of CD or DVD-ROM is in the delivery of reusable and/or fat assets. In the absence of a local CD or DVD-ROM, the developer is forced to choose which performance dimension to compromise: time or richness.

Compromising richness:

Full motion video is replaced with slide shows or streaming media
CD quality digital audio is replaced with streaming audio or MIDI
Full screen images are reduced in size and/or resolution

Compromising time:

Downloading multimedia content before playback, possibly scaled to connection speed (see below).

### 4.8.2 Performance Without an Internet Connection

The primary utility of the application's connection to the Internet is the delivery of timely information. Additionally, interactive features may be enabled by access to the Internet community. These elements of the application will be missing without a current Internet connection. Total absence of Internet connectivity tends to defeat the hybrid benefits of the application, however the effects of temporary or periodic absence of such a connection could potentially be mitigated by the use of IOD technologies such as caching, prefetching, or push. Such technologies would enable "offline browsing" of Web content when a connection is unavailable.

## 5. Summary

According to INFOTECH president Julie Schwerin, by the year 2000, "99 to 100 percent" of CD-ROM titles will include links to the Web. The Internet is an unlimited source of media. Unfortunately, bandwidth isn't unlimited. While putting your latest and greatest content on the Web is certainly the desired goal, you are restricted by what the user can receive, what they're willing to wait for, and ultimately, the physical restrictions of the Internet infrastructure itself.

Tomorrow's exciting applications will mix multimedia data ( graphics, animations, audio and video clips delivered both in bulk and in real time) with the interactivity and community of the Web. Hybrid applications incorporating the key features enumerated in this paper take advantage of the complimentary nature of the CD and the Internet to deliver a compelling end-user experience.

# Appendix A  Programming Examples

The following examples are provided for reference only and are not guaranteed to be complete working examples.

## A.1  Registering ActiveX Controls

### A.1.1 Using Visual Basic 5.0 CCE Application Setup Wizard

Before your control can be used on the Internet, it needs to be able to register itself on the client's computer.  If you've used the Visual Basic 5.0 Control Creation Edition* (CCE)  you can easily create an automated setup package for registering your control:

1) Launch the *Application Setup Wizard* from the Visual Basic 5.0 CCE program group.
2) Click *Browse* to locate your control project.
3) Click *Create Internet Download Setup*.
4) Step through the dialog boxes.  When the setup program is complete, the wizard will have generated two sets of files: distribution files and support files.  The distribution files include a .cab file that contains a compressed version of your control and a sample HTML file that illustrates how to insert the ActiveX control into an HTML page.  The support files are created in case you may need to make changes later on.

For more information on preparing Visual Basic controls for the Internet, refer to Microsoft's Website.

### A.1.2 Create your own Cab

Cabinet files, or cab for short, is a file archive used to simplify the compression and installation of  multiple ActiveX controls. It is possible to support multiple platforms with the installation method. Adding an INF file to the cab archive tells the system how to install the control.

Creating a cab file using DIANTZ.EXE
1) create necessary INF file

```
        ;Sample INF file
        [Add.Code]
        ExampleControl.OCX = Example.Control.OCX

        [ExampleControl.OCX]
        File=http://www.example.com/control/ExampleControl.OCX
        CLSID = {Appropriate CLSID}
        FileVersion=1.0.0.0
```

For multi-platform support refer to Microsoft's cabinet development kit.

2) Build Diamond directive file (DDF).

```
        .Option Explicit
        .Set CabinetNameTemplate = Example.CAB
        .Set Cabinet = on
        .Set Compress = on

        ExampleControl.OCX
```

3) run DIAMOND.EXE from the Cabinet SDK.

```
        DIAMOND /f Example.DDF
/f - specifies the a file directive is being used
4) Insert cab into web page
<OBJECT>
        CLASSID="clsid:{Valid CLSID}"
        CODEBASE="controls/ExampleControl.cab#Version=1,0,0,0">
</OBJECT>
```

## A.2 LINK RESOLUTION CONTROL

The Link Resolution control is a customized OLE control that locates locally-available resources. If the resources are not located locally, then the according http address is constructed from the file name, and the URL is pointed there.

The control was written with MSVC++ 4.1. It can be embedded within a web page, a Visual Basic application, or used in a MSVC app. Below is a list of the properties and methods needed to implement the Link Resolution control.

| Properties | Type | Purpose |
|---|---|---|
| **BackColor** | RGB Value | Determines the background color for the control |
| **CDROMLabel** | LPCSTR | Verify CDROMLabel is the label of the CD in the drive |
| **CDROMLinkColor** | RGB Value | Determines the color of the CD linked resources |
| **DisplayCDError** | Boolean | Determines if error messages are displayed or not. |
| **DisplayTest** | Boolean | Controls whether the text property is displayed |
| **LinkName** | LPCSTR | Name of URL to resolve |
| **LocalLinkColor** | RGB Value | Determines the color of the locally linked resources |
| **NameCacheFile** | LPCSTR | Specifies location and name of the cache.tlb file |
| **PushDataDir** | LPCSTR | Specify location where downloaded data is kept |
| **Text** | LPCSTR | Text string displayed if DisplayText is True |
| **UserNameCache** | Boolean | Determines whether or not the default cache filename is used or if a user specified name is used |

| Methods | Function |
|---|---|
| **Resolve** | Resolve the location of the resource. Determine if content is available on the Internet or on the CDROM |
| **AboutBox** | Information about the Link Resolution Control |

**Example Use In HTML**

```
<OBJECT ID="LnkRslv1" WIDTH=93 HEIGHT=27
 CLASSID="CLSID:4E401503-7DCB-11D0-A060-00AA00C0F5D6"
 CODEBASE="/controls/codebase/">
   <PARAM NAME="_Version" VALUE="65536">
   <PARAM NAME="_ExtentX" VALUE="2455">
   <PARAM NAME="_ExtentY" VALUE="709">
   <PARAM NAME="_StockProps" VALUE="21">
   <PARAM NAME="DisplayText" VALUE="1">
   <PARAM NAME="CDROMLabel" VALUE="MyCDROMLabel">
   <PARAM NAME="LinkName" VALUE="http://www.intel.com/video.avi">
   <PARAM NAME="Text" VALUE="Link Resolution">
</OBJECT>
```

### A.3 Password required video playback control

This custom control may be embedded within a Visual Basic application, a Visual C++ application, or in a web browser. The control takes advantage of the Indeo video key frame encryption codec. This enables the Internet server to control when video may be viewed by forcing the user be logged on to the website to get the video access key.

| Properties | Type | Purpose |
|---|---|---|
| **AccessKey** | Unsigned Long | The key needed to decrypt Indeo video 4.1 that was compressed with it. |
| **AutoStart** | Boolean | Activate the video as soon as ActiveMovie loads the file. |
| **CDROMLabel** | LPCSTR | Check for the volume label of the CD that is current in the CDROM drive. |
| **DisplayImage** | Boolean | Display the background image if video can not be rendered. |
| **MovieName** | LPCSTR | The movie file name to be played. |
| **UseLinkResolver** | Boolean | Use link resolution to resolve the location of resources. |

| Methods | Function |
|---|---|
| **AboutBox** | Displays version and copyright information about the control |
| **Run** | Play the video |
| **Stop** | Stop the playback of the video |
| **Pause** | Pause the playback of the video, and keep current position. |

HTML VideoPlay example

```
<OBJECT ID="VideoPlay1" WIDTH=320 HEIGHT=240
 CLASSID="CLSID:7F14A103-83FC-11D0-9FB2-00A0C9008A48"
 CODEBASE="/controls/Jamplay/">
   <PARAM NAME="MovieName" VALUE="d:\avi\1234key.avi">
   <PARAM NAME="AccessKey" VALUE="1234">
   <PARAM NAME="CDROMLabel" VALUE="MyCDROM">
</OBJECT>
```

## A. 4 Forcing a connection using wininet

The following is an example of using WININET to force a connection to the Internet.

The code sample below:
1) Establishes a connect if none are available, using default ISP connection. This must be enabled in the Internet section of the Control Panel.
2) Connects to the specified server on the Web with the given user name and password.
3) Retrieves the file information of index.htm.
4) Processes the information. In this case, prints it to the debug screen of MSVC.

No error checking is included for brevity.
It is important to note that the wininet.h and the wininet.lib are required.

```
//Internet handles
HINTERNET hOpen = NULL;
HINTERNET hConnect = NULL;
HINTERNET hRequest = NULL;
DWORD dwSize=0, dwCode=0, dwContext=0;
CHAR *lpBuffer, szData[51];
BOOL bRetVal=FALSE;

//Attempt an Internet connection. This will invoke the default dial up
//adapter if the option is enabled in control panel->Internet->Connect.
//If the option is not set, the attempt fails.
InternetAttemptConnect(dwCode);

//Initialize an applications use of wininet functions
hOpen = InternetOpen("My Web Agent",      //Name of Agent Making Calls
      INTERNET_OPEN_TYPE_DIRECT, //Access Type Resolve host name locally
      NULL, 0,0);

//Connect to URL with user name and password.
//This allows for restricted server access.
hConnect = InternetConnect(hOpen,                           //Handle
                           "www.myserver.com",              //Server
                           INTERNET_INVALID_PORT_NUMBER,    //Port
                           "UserName",                      //UserName
                           "UserPassword",                  //UserPasswd
                           INTERNET_SERVICE_HTTP,           //Service
                           0,                               //Flags
                           dwContext);                      //Context

//Open http request handle
//Opens a handle to the file login.htm
```

```
hRequest = HttpOpenRequest(hConnect,            //Session
                           "GET",               //Verb {NULL=”GET”}
                           "login.htm",         //Target Object
                           HTTP_VERSION,        //Address of Version Str
                           "",                  //URL of where last was
                           NULL,                //Accept Types
                           INTERNET_FLAG_RELOAD |  //Access Types
                           INTERNET_FLAG_EXISTING_CONNECT,
                           dwContext);          //Context
//Send request to server
bRetVal = HttpSendRequest(hRequest, NULL, 0, NULL, 0);

//set DWORD size
dwSize = sizeof(DWORD);

HttpQueryInfo (hRequest, HTTP_QUERY_STATUS_CODE |
HTTP_QUERY_FLAG_NUMBER, &dwCode, &dwSize, NULL);

HttpQueryInfo(hRequest, HTTP_QUERY_RAW_HEADERS_CRLF, (LPVOID) NULL,
&dwSize, NULL);

//allocate buffer size based on page size
lpBuffer = new char [dwSize+1];

HttpQueryInfo(hRequest, HTTP_QUERY_RAW_HEADERS_CRLF, (LPVOID) lpBuffer,
                        &dwSize, NULL);

//Terminate the buffer
*(lpBuffer + dwSize) = '\0';



//DO SOMETHING MEANINGFUL!!

      do
      {     //quit if read error
            if(!InternetReadFile(hRequest, (LPVOID)szData, 50, &dwSize))
            {
                  //fail to read
            }

            //if no more data to print break
            if(!dwSize)
                  break;
            else
            {
                  //output the buffer string
                  szData [dwSize] = '\0';
                  OutputDebugString(szData);
            }
      }while(TRUE);

      cout << endl;

      //Close Internet handles
      InternetCloseHandle(hRequest);
      InternetCloseHandle(hConnect);
      InternetCloseHandle(hOpen);
```

```
	//free buffer space
	free(lpBuffer);
```

# Appendix B "Audio" Connected CDs

## B.1 CD EXTRA

What Is CD EXTRA?

CD EXTRA offers the consumer a full audio album that also has an interactive experience when played on a computer CD-ROM drive.

Many traditional music CDs have extra room available at the end of the disc. CD EXTRA takes advantage of this extra space by adding a multimedia, computer portion or second session. CD EXTRA is a multisession disc with the first session containing the audio portion and the second session containing the multimedia portion. CD EXTRA will play on any audio CD player just like a traditional audio CD. CD EXTRA multimedia data will play on CD-ROM drives with multisession compatible firmware and software. (CD EXTRA is not compatible with some older CD-ROM Drives.) Many of Sony's new audio releases will feature a free multimedia portion. The user can access interactive lyrics and animated stories, music videos, exclusive video interviews, and much more.

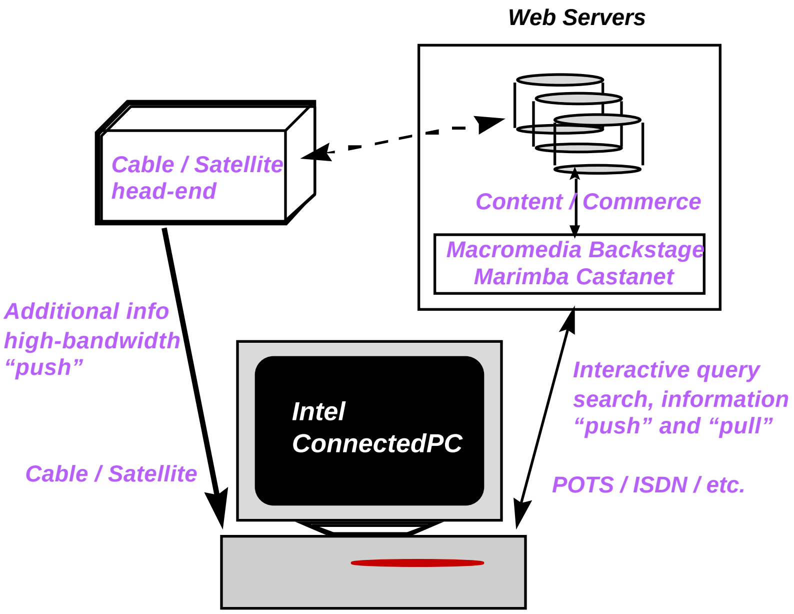
For more information see http://www.music.sony.com/Music/CDEXTRA/.

# Appendix C The InfiniteCD

The InfiniteCD is a direct result of Intel's hybrid application initiative and its industry work with several key software developers and content providers. The InfiniteCD consists of several components: a media-enabled Connected PC with Pentium® processor with MMX™ technology, high-capacity local storage (such as CD-ROM / DVD and HD), connection to the internet through modem, for instance, and a specially prepared CD / DVD-ROM.

The InfiniteCD is intended to bring very rich media - typically the type used in the entertainment industry - to the PC desktop using hybrid methodology. The goal is to provide a rich media experience - such as the consumer is used to having with television, motion pictures and music - and deliver a framework which would allow the content provider to fully immerse the user in their trademark environment and interface.

Physically, the InfiniteCD is a local storage medium (CD or DVD-ROM) containing rich media, executable code and information links which allow it to interact with a connection to a remote internet site and dynamically update and extend information and content.


Web Servers
Cable / Satellite head-end
Content / Commerce
Macromedia Backstage
Marimba Castanet
Additional info high-bandwidth "push"
Intel ConnectedPC
Interactive query search, information "push" and "pull"
Cable / Satellite
POTS / ISDN / etc.

**InfiniteCD Block Diagram**

The architecture of the InfiniteCD encompasses both "pull" technologies which reach out to grab information when it is needed (exemplified by capabilities such as Macromedia's Shockwave) and "push" technologies which have the ability to drive content from the server down to the desktop's hard drive without the user's application intervening (Marimba's Castanet, for instance). In this respect, the InfiniteCD capitalizes on the concept of intelligent pre-caching which attempts to predict what the user will need well in advance of its use. Determining this, the current revision of information on the desktop is checked and an updated version is sent down from the server and linked to the application. Contrary to "pull", which has traditionally implied loading prior to execution or transfer on demand, intelligent push caching would utilize available communications bandwidth whenever it was not being actively used by the consumer. Or another way of looking at it is that all available bandwidth is being used constantly in the background and not simply when it is needed by the application. Consequently, marrying pull and push technologies provides a highly efficient data management framework for the InfiniteCD application.

## Appendix D  References

**Intel Developer Relations Group**
**URL:  http://www.intel.com/drg**

---

[i] Company: Sony Music
Product: CD EXTRA
URL: http://www.music.sony.com

[ii] Company: Microsoft
URL: http://www.microsoft.com

[iii] Company: Human Code
Product: XtraNet
URL: http://www.humancode.com

[iv] Company: Tabuleiro da Baiana Multimedia
Product: WebXtra
URL: http://www.tbaiana.com

[v] Company: Allegiant Technologies, Inc.
Product: Marionet
URL: http://www.allegiant.com

[vi] Company: NCompass Labs
Product: ScriptActive, CaptiveX
URL: http://www.ncompasslabs.com/

[vii] Company: Intel Corporation
Product: Intel Media for Java
URL: http://www.intel.com/ial/jmedia/index.htm

[viii] Company: Sun Microsystems, Inc.
Product: Java
URL: http://java.sun.com/

[ix] Company: BackWeb Technologies
Product: BackWeb
URL: http://www.backweb.com/

[x] Company: PointCast, Inc.
Product: PointCast
URL: www.pointcast.com

[xi] Company: c|net

URL: http://www.cnet.com/Content/Reviews/Compare/ISP/national.html
c|net Top ISP Listing

[xii] URL:
http://www.cs.ucl.ac.uk/staff/zwang/papers/prefetch/Overview.html
Article on Prefetch

[xiii] Company: HighWater
Product: SureSign
http://www.highwaterfbi.com

[xiv] Company: Digimarc
Product: PictureMarc
http://www.digimarc.com/