Adobe Acrobat PDF Files

Adobe® Portable Document Format (PDF) is a universal file format that preserves all of the fonts, formatting, colours and graphics of any source document, regardless of the application and platform used to create it.

Adobe PDF is an ideal format for electronic document distribution as it overcomes the problems commonly encountered with electronic file sharing.

- ***Anyone, anywhere*** can open a PDF file. All you need is the free Adobe Acrobat Reader. Recipients of other file formats sometimes can't open files because they don't have the applications used to create the documents.

- PDF files ***always print correctly*** on any printing device.

- PDF files always display ***exactly*** as created, regardless of fonts, software, and operating systems. Fonts, and graphics are not lost due to platform, software, and version incompatibilities.

- The free Acrobat Reader is easy to download and can be freely distributed by anyone.

- Compact PDF files are smaller than their source files and download a page at a time for fast display on the Web.

Create PDF files quickly and easily!

The pdf995 suite of products - Pdf995, PdfEdit995, and Signature995 - is a complete solution for your document publishing needs. It provides ease of use, flexibility in format, and industry-standard security- and all at no cost to you.

Pdf995 makes it easy and affordable to create professional-quality documents in the popular PDF file format. Its easy-to-use interface helps you to create PDF files by simply selecting the "print" command from any application, creating documents which can be viewed on any computer with a PDF viewer. Pdf995 supports network file saving, fast user switching on XP, Citrix/Terminal Server, custom page sizes and large format printing. Pdf995 is a printer driver that works with any Postscript to PDF converter. The pdf995 printer driver and a free Converter are available for easy download.

PdfEdit995 offers a wealth of additional functionality, such as: combining documents into a single PDF; automatic link insertion; hierarchical bookmark insertion; PDF conversion to HTML or DOC (text only); integration with Word toolbar with automatic table of contents and link generation; autoattach to email; stationery and stamping.

Signature995 offers state-of-the-art security and encryption to protect your documents and add digital signatures.

The Pdf995 Suite offers the following features, all at no cost:

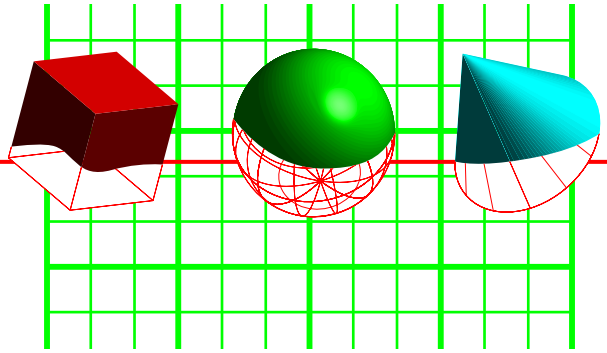
- Automatic insertion of embedded links
- Hierarchical Bookmarks
- Support for Digital Signatures
- Support for Triple DES encryption
- Append and Delete PDF Pages
- Batch Print from Microsoft Office
- Asian and Cyrillic fonts
- Integration with Microsoft Word toolbar
- PDF Stationery
- Combining multiple PDF's into a single PDF
- Three auto-name options to bypass Save As dialog
- Imposition of Draft/Confidential stamps
- Support for large format architectural printing
- Convert PDF to JPEG, TIFF, BMP, PCX formats
- Convert PDF to HTML and Word DOC conversion
- Convert PDF to text
- Automatic Table of Contents generation
- Support for XP Fast User Switching and multiple user sessions
- Standard PDF Encryption (restricted printing, modifying, copying text and images)
- Support for Optimized PDF
- Support for custom page sizes
- Option to attach PDFs to email after creation
- Automatic text summarization of PDF documents
- Easy integration with document management and Workflow systems
- n-Up printing
- Automatic page numbering
- Simple Programmers Interface
- Option to automatically display PDFs after creation
- Custom resizing of PDF output
- Configurable Font embedding
- Support for Citrix/Terminal Server
- Support for Windows 2003 Server
- Easy PS to PDF processing
- Specify PDF document properties
- Control PDF opening mode
- Can be configured to add functionality to Acrobat Distiller
- Free: Creates PDFs without annoying watermarks
- Free: Fully functional, not a trial and does not expire
- Over 5 million satisfied customers
- Over 1000 Enterprise Customers worldwide

Please visit us at www.pdf995.com to learn more.

This document illustrates several features of the Pdf995 Suite of Products.

VRML

VIRTUAL REALITY MODELING LANGUAGE



Introduction

The **Virtual Reality Modeling Language** (VRML) is a language for describing multi-participant interactive simulations -- virtual worlds networked via the global Internet and hyperlinked with the World Wide Web. All aspects of virtual world display, interaction and internetworking can be specified using VRML. It is the intention of its designers that VRML become the standard language for interactive simulation within the World Wide Web.

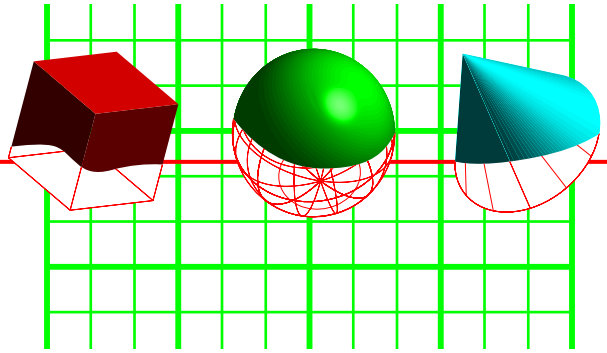
The first version of VRML allows for the creation of virtual worlds with limited interactive behavior. These worlds can contain objects which have hyperlinks to other worlds, HTML documents or other valid MIME types. When the user selects an object with a hyperlink, the appropriate MIME viewer is launched. When the user selects a link to a VRML document from within a correctly configured WWW browser, a VRML viewer is launched. Thus VRML viewers are the perfect companion applications to standard WWW browsers for navigating and visualizing the Web. Future versions of VRML will allow for richer behaviors, including animations, motion physics and real-time multi-user interaction.

This document specifies the features and syntax of Version 1.0 of VRML.

VRML Mission Statement

The history of the development of the Internet has had three distinct phases; first, the development of the TCP/IP infrastructure which allowed documents and data to be stored in a proximally independent way; that is, Internet provided a layer of abstraction between data sets and the hosts which manipulated them. While this abstraction was useful, it was also confusing; without any clear sense of "what went where", access to Internet was restricted to the class of sysops/net surfers who could maintain internal cognitive maps of the data space.

Next, Tim Berners-Lee's work at CERN, where he developed the hypermedia system known as **World Wide Web**, added another layer of abstraction to the existing structure. This abstraction provided an "addressing" scheme, a unique identifier (the Universal Resource Locator), which could tell anyone "where to go and how to get there" for any piece of data within the Web. While useful, it lacked dimensionality; there's no *there* there within the web, and the only type of navigation permissible (other than surfing) is by direct reference. In other words, I can only tell you how to get to the VRML Forum home page by saying, "http://www.wired.com/", which is not human-centered data. In



fact, I need to make an effort to remember it at all. So, while the World Wide Web provides a retrieval mechanism to complement the existing storage mechanism, it leaves a lot to be desired, particularly for human beings.

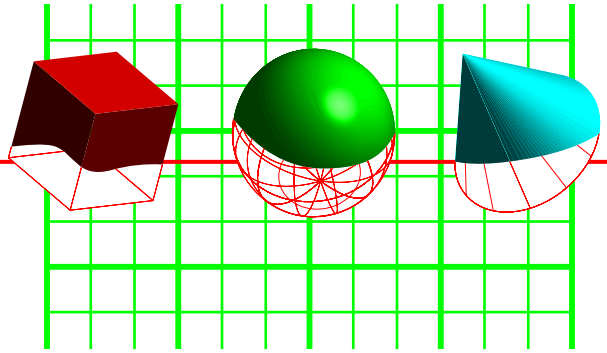
Finally, we move to "perceptualized" Internetworks, where the data has been sensualized, that is, rendered sensually. If something is represented sensually, it is possible to make sense of it. VRML is an attempt (how successful, only time and effort will tell) to place humans at the center of the Internet, ordering its universe to our whims. In order to do that, the most important single element is a standard that defines the particularities of perception. Virtual Reality Modeling Language is that standard, designed to be a *universal description language for multi-participant simulations*.

These three phases, storage, retrieval, and perceptualization are analogous to the human process of consciousness, as expressed in terms of semantics and cognitive science. Events occur and are recorded (memory); inferences are drawn from memory (associations), and from sets of related events, maps of the universe are created (cognitive perception). What is important to remember is that the map is **not** the territory, and we should avoid becoming trapped in any single representation or world-view. Although we need to *design to avoid disorientation*, we should always push the envelope in the kinds of experience we can bring into manifestation!

This document is the living proof of the success of a process that was committed to being open and flexible, responsive to the needs of a growing Web community. Rather than re-invent the wheel, we have adapted an existing specification (Open Inventor) as the basis from which our own work can grow, saving years of design work and perhaps many mistakes. Now our real work can begin; that of rendering our noospheric space.

History

VRML was conceived in the spring of 1994 at the first annual World Wide Web Conference in Geneva, Switzerland. Tim Berners-Lee and Dave Raggett organized a Birds-of-a-Feather (BOF) session to discuss Virtual Reality interfaces to the World Wide Web. Several BOF attendees described projects already underway to build three dimensional graphical visualization tools which interoperate with the Web. Attendees agreed on the need for these tools to have a common language for specifying 3D scene description and WWW hyperlinks -- an analog of HTML for virtual reality. The term Virtual Reality Markup Language (VRML) was coined, and the group resolved to begin specification work after the conference. The word 'Markup' was later changed to 'Modeling' to reflect the graphical nature of VRML.

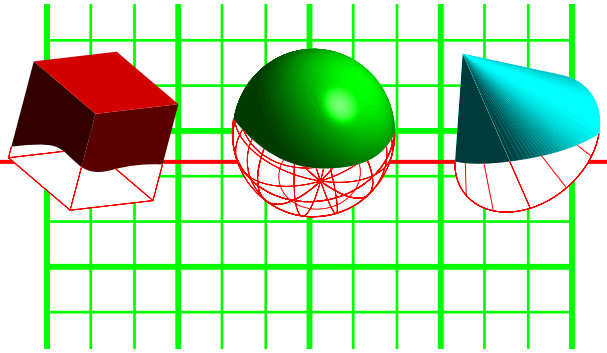


Shortly after the Geneva BOF session, the www-vrml mailing list was created to discuss the development of a specification for the first version of VRML. The response to the list invitation was overwhelming: within a week, there were over a thousand members. After an initial settling-in period, list moderator Mark Pesce of Labyrinth Group announced his intention to have a draft version of the specification ready by the WWW Fall 1994 conference, a mere five months away. There was general agreement on the list that, while this schedule was aggressive, it was achievable provided that the requirements for the first version were not too ambitious and that VRML could be adapted from an existing solution. The list quickly agreed upon a set of requirements for the first version, and began a search for technologies which could be adapted to fit the needs of VRML.

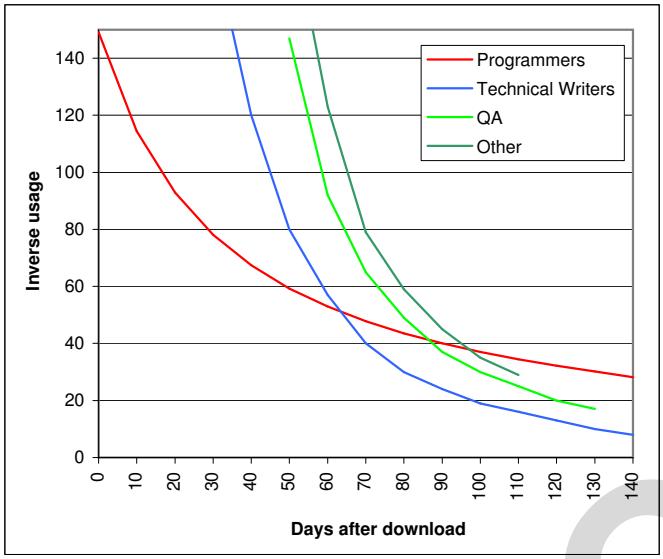
The search for existing technologies turned up a several worthwhile candidates. After much deliberation the list came to a consensus: the Open Inventor ASCII File Format from Silicon Graphics, Inc. The Inventor File Format supports complete descriptions of 3D scenes with polygonally rendered objects, lighting, materials, ambient properties and realism effects. A subset of the Inventor File Format, with extensions to support networking, forms the basis of VRML. Gavin Bell of Silicon Graphics has adapted the Inventor File Format for VRML, with design input from the mailing list. SGI has publicly stated that the file format is available for use in the open market, and have contributed a file format parser into the public domain to bootstrap VRML viewer development.

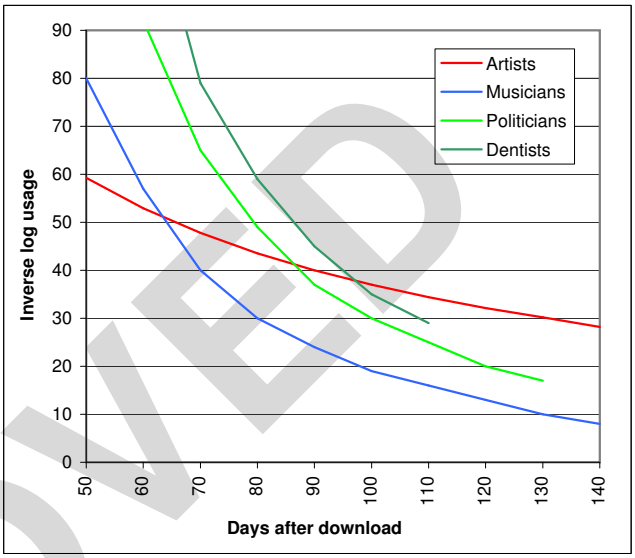
VRML

VIRTUAL REALITY MODELING LANGUAGE



A Graphical Representation of Inverse VRML Uptake





Change the number in red below to adjust for download rate and/or bandwidth.

1 The number 1 represents an engineer with an "average" cube *

CF	Min	fsw	Air	EANx 32%	EANx 36%
80.0	149.12	**0**			
61.4	114.43	**10**			
49.8	92.846	**20**			
41.9	78.102	**30**	**180**		
36.2	67.402	**40**	**120**		
31.8	59.275	**50**	**80.0**	**147.0**	**192.0**
28.4	52.9	**60**	**57.0**	**92.0**	**123.0**
25.6	47.774	**70**	**40.0**	**65.0**	**79.0**
23.4	43.543	**80**	**30.0**	**49.0**	**59.0**
21.5	40.001	**90**	**24.0**	**37.0**	**45.0**
19.9	37	**100**	**19.0**	**30.0**	**35.0**
18.5	34.409	**110**	**16.0**	**25.0**	**29.0**
17.3	32.154	**120**	**13.0**	**20.0**	**n/a**
16.2	30.178	**130**	**10.0**	**17.0**	**n/a**
15.1	28.202	**140**	**8.0**	**n/a**	**n/a**

Sample PDF Document

Robert Maron
Grzegorz Grudziński

February 20, 1999

Contents

Chapter 1

Template

1.1 How to compile a `.tex` file to a `.pdf` file

1.1.1 Tools

To process the files you (may) need:

- `pdflatex` (for example from `tetex` package $\geq$ `0.9-6`, which you can get from Red Hat 5.2);
- `acroread` (a PDF viewer, available from http://www.adobe.com/);
- `ghostscript` $\geq$ `5.10` (for example from Red Hat Contrib) and `ghostview` or `gv` (from RedHat Linux);
- `efax` package could be useful, if you plan to fax documents.

1.1.2 How to use the tools

Follow these steps:

1. put all source `.tex` files in one directory, then chdir to the directory (or put some of them in the LaTeXsearch path — if you know how to do this);
2. run "`pdflatex file.tex`" on the main file of the document three times (three — to prepare valid table of contents);
3. to see or print the result use `acroread` (unfortunately some versions of `acroread` may produce PostScript which is too complex), or

4. run `ghostscript`: "`gv file.pdf`" to display or:
   "gs -dNOPAUSE -sDEVICE=pswrite -q -dBATCH -sOutputFile=file.ps file.pdf" to produce a PostScript file;

5. run "`fax send phone-number file.ps`" as root to send a fax, or — if you know how to do this — modify the `fax` script to be able to fax `.pdf` files directly (you have to insert "`|%PDF*`" somewhere...).

1.2 How to write a document

1.2.1 The main document

Choose the name of the document, say `document`. Copy `template.tex` to `document.tex`, then edit it, change the title, the authors and set proper include(s) for all the chapters.

1.2.2 Chapters

Each chapter should be included in the main document as a separate file. You can choose any name for the file, but we suggest adding a suffix to the name of the main file. For our example we use the file name `document_chapter1.tex`.

First, copy `template_chapter.tex` to `document_chapter1.tex` and add the line

```
\include{document_chapter1}
```

in the `document.tex`, then edit `document_chapter1.tex`, change the chapter title and edit the body of the chapter appropriately.

1.2.3 Spell-checking

Do use a spell-checker, please!

You may also want to check grammar, style and so on. Actually you should do it (if you have enough spare time). But you *must* check spelling!

You can use the `ispell` package for this, from within `emacs`, or from the command line:

```
ispell -t document_chapter1.tex
```

1.3 LaTeX and pdfLaTeX capabilities

1.3.1 Overview

First you edit your source `.tex` file. In LaTeX you compile it using the `latex` command to a `.dvi` file (which stands for device-independent). The `.dvi` file can be converted to any device-dependent format you like using an appropriate driver, for example `dvips`.

When producing `.pdf` files you should use `pdflatex`, which produces directly `.pdf` files out of `.tex` sources. Note that in the `.tex` file you may need to use some PDF specific packages.

For viewing `.tex` files use your favourite text editor, for viewing `.dvi` files under X Window System use `xdvi` command, `.ps` files can be viewed with `gv` (or `ghostview`) and `.pdf` files with `acroread`, `gv` or `xpdf`.

1.3.2 LaTeX

A lot of examples can be found in this document.

You should also print

- `doc/latex/general/latex2e.dvi` and
- `doc/latex/general/lshort2e.dvi`

from your `tetex` distribution (usually in

- `/usr/share/texmf` or
- `/usr/lib/texmf/texmf`).

1.3.3 pdfLaTeX

Consult `doc/pdftex/manual.pdf` from your `tetex` distribution for more details. Very useful informations can be found in the `hyperref` and `graphics` package manuals:

- `doc/latex/hyperref/manual.pdf` and
- `doc/latex/graphics/grfguide.dvi`.

1.3.4 Examples

References

MIMUW

Hyperlinks

This is a target.
And this is a link.

Dashes, etc.

There are three kinds of horizontal dash:

- - (use inside words; for example “home-page”, “X-rated”)
- – (use this one between numbers; for example “pages 2–22”)
- — (use this one as a sentence separator — like here)

National characters

- ó, é, í, . . .
- è, à, ì, . . .
- ô, ê, . . .
- õ, ñ, . . .
- ö, ë, . . .
- ż
- ą, ę
- ł, ø, ß

There are other ways to do this, see the documentation for `inputenc` package.

Reserved characters

Some characters have some special meaning, thus cannot be entered in the usual way.

- $ & % # _ { }
- \
- ~ ^

Math

- 1^2, 1^{2n}, ...
- i_1, i_{2n}, ...
- $\frac{1}{2}$, $\frac{2n}{2-3}$, ...
- α, β, γ, Ω, ...
- $\rightarrow$, $\Rightarrow$, $\geq$, $\neq$, $\in$, $\star$, ...
- $\sqrt{2}$, ...
- $\overline{2+2}$, ...

For more examples and symbols see chapter 3 of `lshort2e.dvi`.

Fonts

- Roman
- *Emphasis*
- Medium weight — the default
- **Boldface**
- Upright
- *Slanted*
- Sans serif
- SMALL CAPS
- `Typewriter`
- and sizes:
  - tiny
  - scriptsize
  - footnotesize
  - small
  - normalsize

- large
- Large
- LARGE
- huge
- Huge

A Simple PDF File

This is a small demonstration .pdf file -

just for use in the Virtual Mechanics tutorials. More text. And more text. And more text. And more text. And more text.

And more text. And more text. And more text. And more text. And more text. And more text. Boring, zzzzz. And more text. And more text. And more text. And more text. And more text. And more text. And more text. And more text. And more text.

And more text. And more text. And more text. And more text. And more text. And more text. And more text. Even more. Continued on page 2 ...

Simple PDF File 2

...continued from page 1. Yet more text. And more text. And more text. And more text. And more text. And more text. And more text. Oh, how boring typing this stuff. But not as boring as watching paint dry. And more text. And more text. And more text. And more text. Boring. More, a little more text. The end, and just as well.