



CyberSoft White Papers

SmartScan

Copyright © 2001 by CyberSoft, Incorporated.

Permission is granted to any individual or institution to use, copy, or redistribute this documentation and code so long as it is not sold for profit, and provided that it is reproduced whole and this copyright notice is retained.

- Reading *SmartScan* Input
- Writing *SmartScan* Output
- Download SmartScan source code for VFind Version 162 and earlier
- Download the newest version of SmartScan for VFind Version 163 and above
- Modified SmartScan to allow output to a named file instead of stdout. Also, modification to send a quoted "EOF" after each file which helps VFind note the end of a file for the -p,--per-file option.

By Bryan Berkowitz, Dolphin Technology, Inc., bryan@dolphtech.com, 9/24/03

Introduction

The CyberSoft *SmartScan* format is a method of encapsulating a group of files, file names, and types, into a single stream. This format is utilized by the CyberSoft products such as *UAD* and *VFind*. This release of *SmartScan* documentation and source code is being made available to the public as a service to the community to promote interoperability. Using the source code provided here, you can develop your own programs to read and write *SmartScan* format, for example to provide a custom front-end to *VFind*.

Note that SmartScan format uses native integer representations, so it is not portable between machines with different endianness or word size.

SmartScan API

SmartScan output is created by invoking *smartscan_output()* with an open input file pointer and/or buffer of input data, together with the file name and type information:

```
void smartscan_output( FILE *f, char *buf, int bufn,
                      char *name, ftype type, char *top_name);
```

If the buffer count *bufn* is greater than 0, *smartscan_output()* first sends *bufn* bytes from *buf* to *stdout*. Then, if the input file pointer *f* is not NULL, it sends the file contents. This allows you to open a file and read part of the contents into a buffer in order to determine the file type before invoking *smartscan_output()*. *top_name* is the main file name, e.g. it could be the name of a .zip or .tar.Z archive file, and *name* is the sub-file name, e.g. a file entry from an archive. *type* is a pointer to a file type list, defined as follows:

```
typedef struct ftypel_s
{
    char *t;
    struct ftypel_s *next;
} *ftypel;

typedef struct ftype_s
{
    char *fst; /* file-system type info */
    ftypel t; /* data-derived type info */
} *ftype;
```

SmartScan input is read by invoking *smartscan_read()*, which returns non-zero on success:

```
int smartscan_read( FILE *fp, struct smartscan_read_client *client);
```

fp is the input file pointer to read from, e.g. *stdin*, and *client* is a pointer to a structure of call-back functions which you must provide:

```
struct smartscan_read_client
{
    void (*file_top_name)(char *s);
    void (*file_name)(char *s);
    void (*file_fstype)(char *s);
    void (*file_type)(char *s);
    void (*file_type_end)(void);
    void (*file_data)(char *s, size_t n);
    void (*file_end)(void);
    void (*unexpected_eof)(void);
    void (*quote)(char *s);
};
```

As the **SmartScan** input stream is read, call-backs are made to your functions to handle the file names, types, and data, etc.

Reading SmartScan Input

smartscanrx.c is a sample program for reading **SmartScan** input. It displays the file names, types,

etc. and creates output files from the input stream.

Sample Run:

```
% tar tvf test.tar
-rw-r--r-- 103/11      8743 Nov 14 16:46 2000 test/REPORT.006
-rw-r--r-- 103/11     58450 Jun  2 09:31 1999 test/ZR8E_001.jpg
-rw-r--r-- 103/11     81038 Jun  2 09:31 1999 test/ZR8E_002.jpg
-rw-r--r-- 103/11     79911 Jun  2 09:31 1999 test/ZR8E_003.jpg
-rw-r--r-- 103/11      1667 Jun  2 10:11 1999 test/all.html
-rw-r--r-- 103/11       369 Nov 16 08:42 2000 test/encrypted.dat

% uad -ssw test.tar | smartscanrx

File top level name: test.tar
File name: test/REPORT.006
Sending data to smartscanrx.7465.0
File type: text (no enclosures found)

File top level name: test.tar
File name: test/ZR8E_001.jpg
Sending data to smartscanrx.7465.1
File type: JPEG/JFIF compressed image file OR JPEG picture

File top level name: test.tar
File name: test/ZR8E_002.jpg
Sending data to smartscanrx.7465.2
File type: JPEG/JFIF compressed image file OR JPEG picture

File top level name: test.tar
File name: test/ZR8E_003.jpg
Sending data to smartscanrx.7465.3
File type: JPEG/JFIF compressed image file OR JPEG picture

File top level name: test.tar
File name: test/all.html
Sending data to smartscanrx.7465.4
File type: HTML text

File top level name: test.tar
File name: test/encrypted.dat
Sending data to smartscanrx.7465.5
File type: unknown
```

Writing SmartScan Output

smartscanwx.c is a sample program for writing *SmartScan* output. For each command-line file name argument, it invokes *smartscan_output()*.

Sample Run:

```
% ls test/
REPORT.006 ZR8E_001.jpg ZR8E_002.jpg ZR8E_003.jpg all.html encrypted.dat

% smartscanwx test/* | vfind -ssr -sst
##==>    Vfind Version: 11, Release: 1. Patchlevel: 0 (July 2001)
##==>> Initiating SmartScan processing of standard input
##==> Checking file: "test/REPORT.006" -> "REPORT.006"
```

```
##==>> SmartScan file type: `unknown'  
##==>> SmartScan file type: `binary'  
##==> Checking file: "test/ZR8E_001.jpg" -> "ZR8E_001.jpg"  
##==>> SmartScan file type: `unknown'  
##==>> SmartScan file type: `binary'  
##==> Checking file: "test/ZR8E_002.jpg" -> "ZR8E_002.jpg"  
##==>> SmartScan file type: `unknown'  
##==>> SmartScan file type: `binary'  
##==> Checking file: "test/ZR8E_003.jpg" -> "ZR8E_003.jpg"  
##==>> SmartScan file type: `unknown'  
##==>> SmartScan file type: `binary'  
##==> Checking file: "test/all.html" -> "all.html"  
##==>> SmartScan file type: `unknown'  
##==>> SmartScan file type: `binary'  
##==> Checking file: "test/encrypted.dat" -> "encrypted.dat"  
##==>> SmartScan file type: `unknown'  
##==>> SmartScan file type: `binary'  
##==> Number of files read: 6  
##==> No apparent virus infections found.  
##==> Vfind program termination
```

View source code:

- [makefile](#)
- [ftype](#)
- [smartscan-internal.h](#)
- [smartscan.c](#)
- [smartscan.h](#)
- [smartscanrx.c](#)
- [smartscanwx.c](#)

White Paper Index

Next





This site certified 508 Compliant