

EXHIBIT B  
(Part 1 of 3)



US008280932B2

(12) **United States Patent  
Horn**

(10) **Patent No.:** US 8,280,932 B2  
(45) **Date of Patent:** \*Oct. 2, 2012

(54) **COMPUTER SYSTEM FOR AUTOMATIC ORGANIZATION, INDEXING AND VIEWING MULTIPLE OBJECTS FROM MULTIPLE SOURCES**

5,899,997 A	5/1999	Ellacott	
6,094,649 A	7/2000	Bowen et al.	
6,243,724 B1	6/2001	Mander et al.	
6,401,097 B1	6/2002	McCotter et al.	
6,457,017 B2 *	9/2002	Watkins et al.	707/696
6,523,022 B1	2/2003	Hobbs	
6,691,282 B1	2/2004	Rochford et al.	
2003/0088593 A1 *	5/2003	Stickler	707/205
2003/0105771 A1 *	6/2003	Tiefenbrun et al.	707/103 R
2003/0126115 A1 *	7/2003	Solomon	707/3

(76) Inventor: **Bruce L. Horn**, San Anselmo, CA (US)

(\* ) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **13/272,028**

(22) Filed: **Oct. 12, 2011**

(65) **Prior Publication Data**

US 2012/0030264 A1 Feb. 2, 2012

**Related U.S. Application Data**

(63) Continuation of application No. 12/856,428, filed on Aug. 13, 2010, now Pat. No. 8,055,692, which is a continuation of application No. 11/903,304, filed on Sep. 21, 2007, now Pat. No. 7,840,619, which is a continuation of application No. 10/621,689, filed on Jul. 16, 2003, now Pat. No. 7,275,063.

(60) Provisional application No. 60/396,439, filed on Jul. 16, 2002.

(51) **Int. Cl.**  
**G06F 17/30** (2006.01)

(52) **U.S. Cl.** ..... 707/829; 707/830

(58) **Field of Classification Search** ..... 707/829,  
707/830

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,425,139 A	6/1995	Williams et al.	
5,544,360 A *	8/1996	Lewak et al.	1/1

**20 Claims, 45 Drawing Sheets**

OTHER PUBLICATIONS

Practical File System Design with the Be File System, Giampaolo, D., Morgan Kaufmann Publishers, Inc., 1999, text, 247 pp.

(Continued)

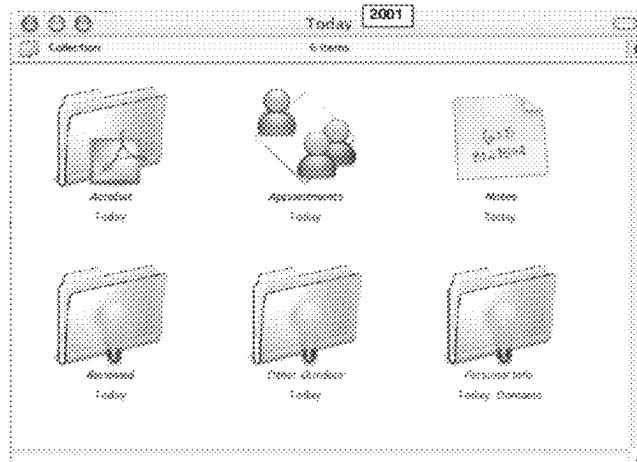
*Primary Examiner* — John E Breene

*Assistant Examiner* — Dennis Myint

(74) *Attorney, Agent, or Firm* — Jacques M. Dulin, Esq.;  
Innovation Law Group, Ltd.

(57) **ABSTRACT**

A computer data processing system including a central processing unit configured with a novel integrated computer control software system for the management of data objects including dynamic and automatic organization, linking, finding, cross-referencing, viewing and retrieval of multiple objects regardless of nature or source. The inventive system provides underlying component architecture having an object-oriented database structure and a metadata database structure which is unique in storing only one instance of each object while linking the object to multiple collections and domains by unique metadata links for the grouping into and retrieval from any of the collections. The system employs configurable, extensible attribute/properties of data objects in metadata format, and a truly user-friendly configurable interface that facilitates faster, more unified, comprehensive, useful and meaningful information management. Additional features include a sticky path object hierarchy viewing system, key phrase linking, viewing by reference, and drag-and-drop relationship link creation.



OTHER PUBLICATIONS

Semantic File Systems, Gifford, D., Jouvelot, P., Sheldon, M., and O'Toole, J., presented at the 13th ACM Symposium on Operating Systems Principles, 1991, and appeared in its proceedings, ACM Operating Systems Review, Oct. 1991, pp. 16-25.

Dourish, Edwards, Lamarca and Salisbury, "Using Properties for Uniform Interaction in the Presto Document System," UIST 1999, Computer Science Laboratory, Xerox Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, CA 94304.

\* cited by examiner

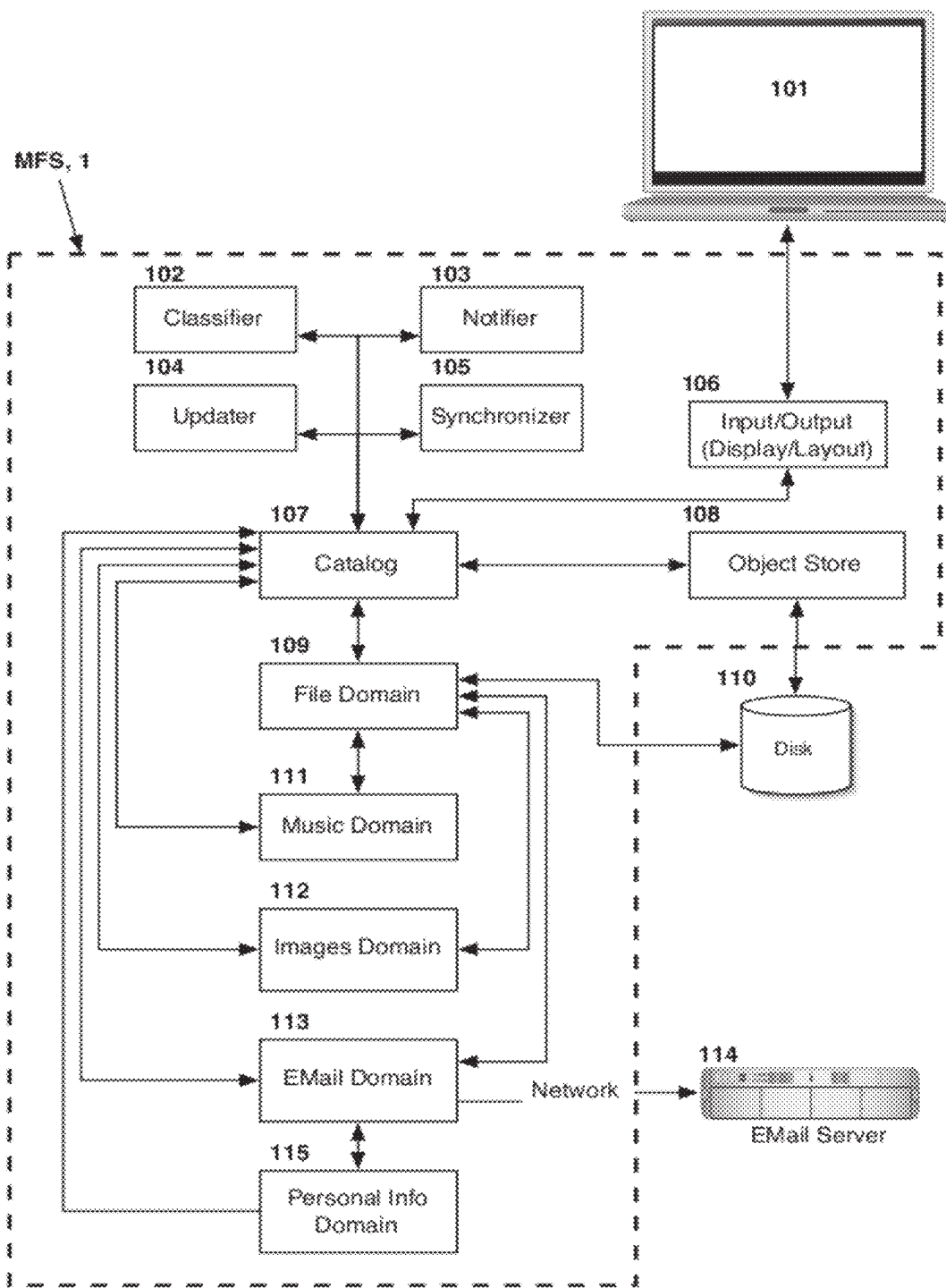


Figure 1

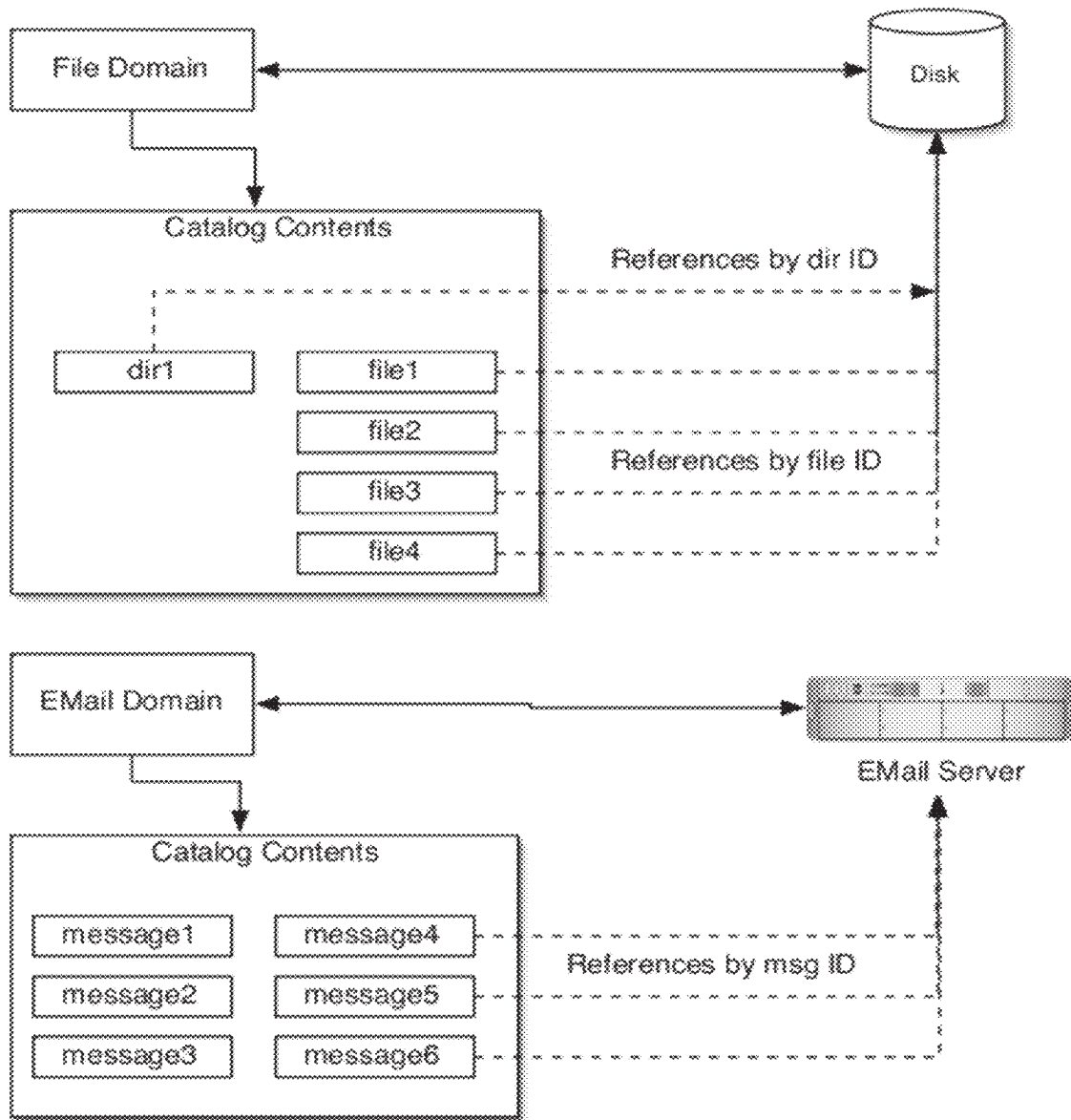


Figure 2

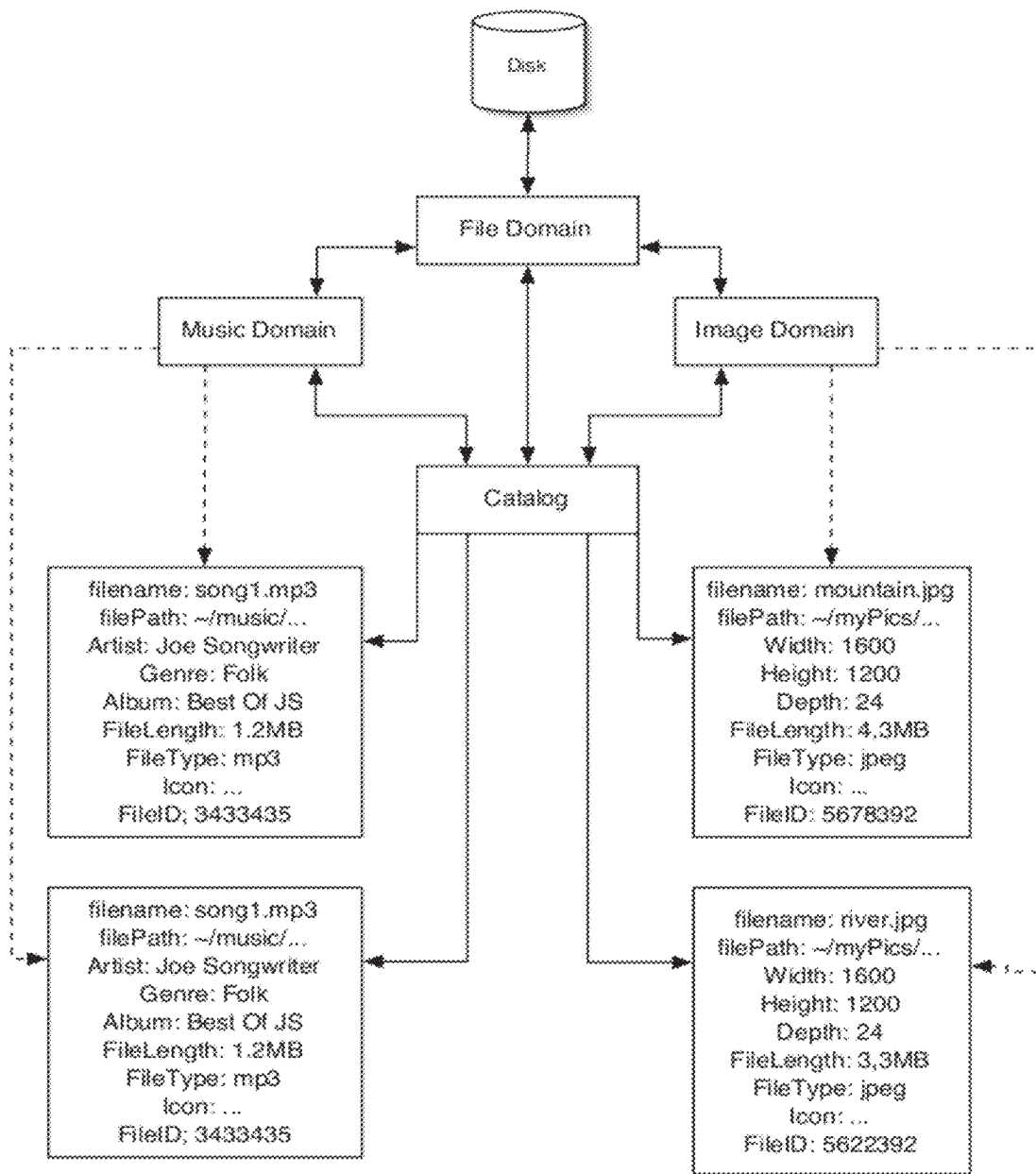


Figure 3

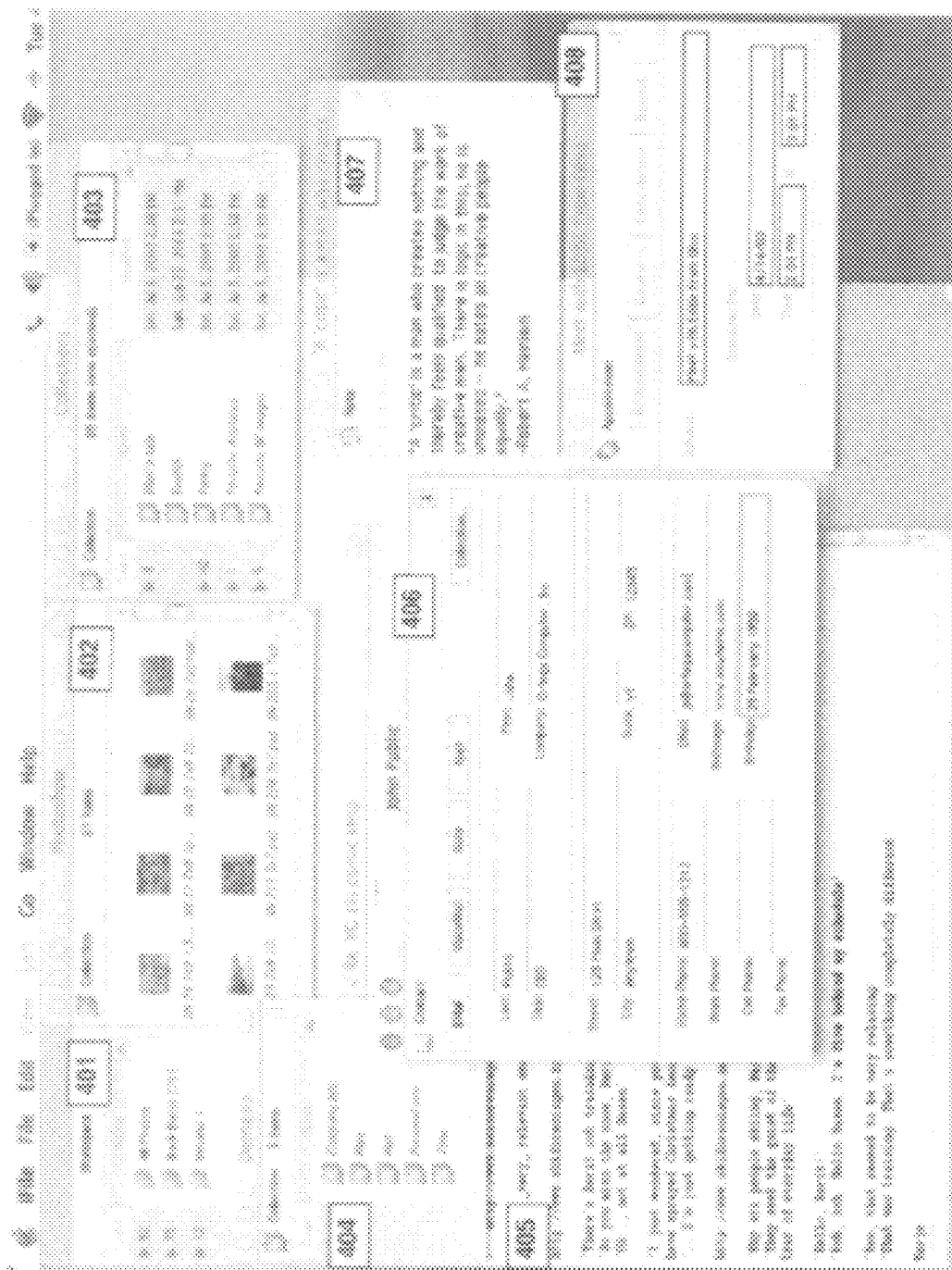


Figure 4

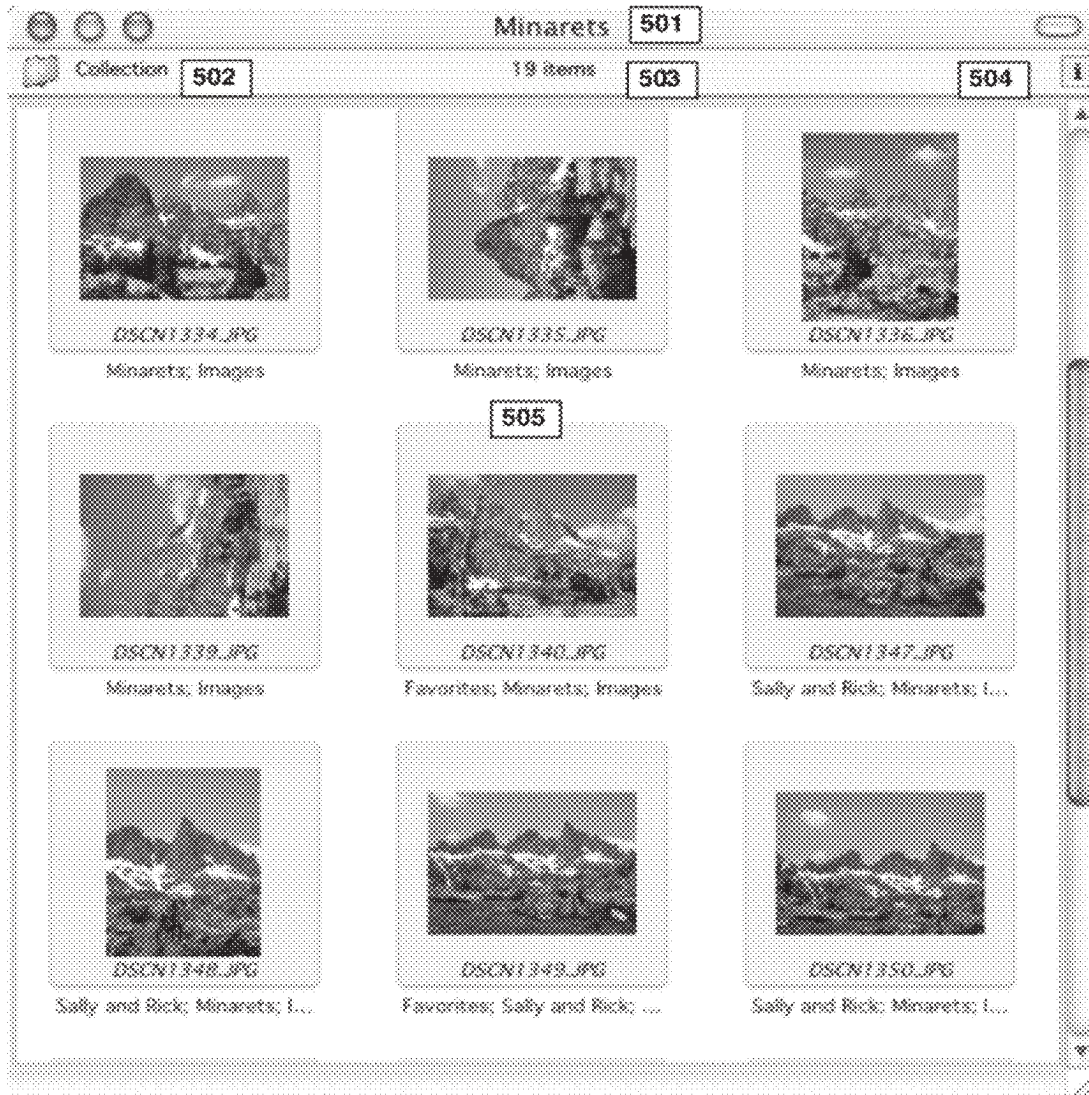


Figure 5





Figure 6



Figure 7a

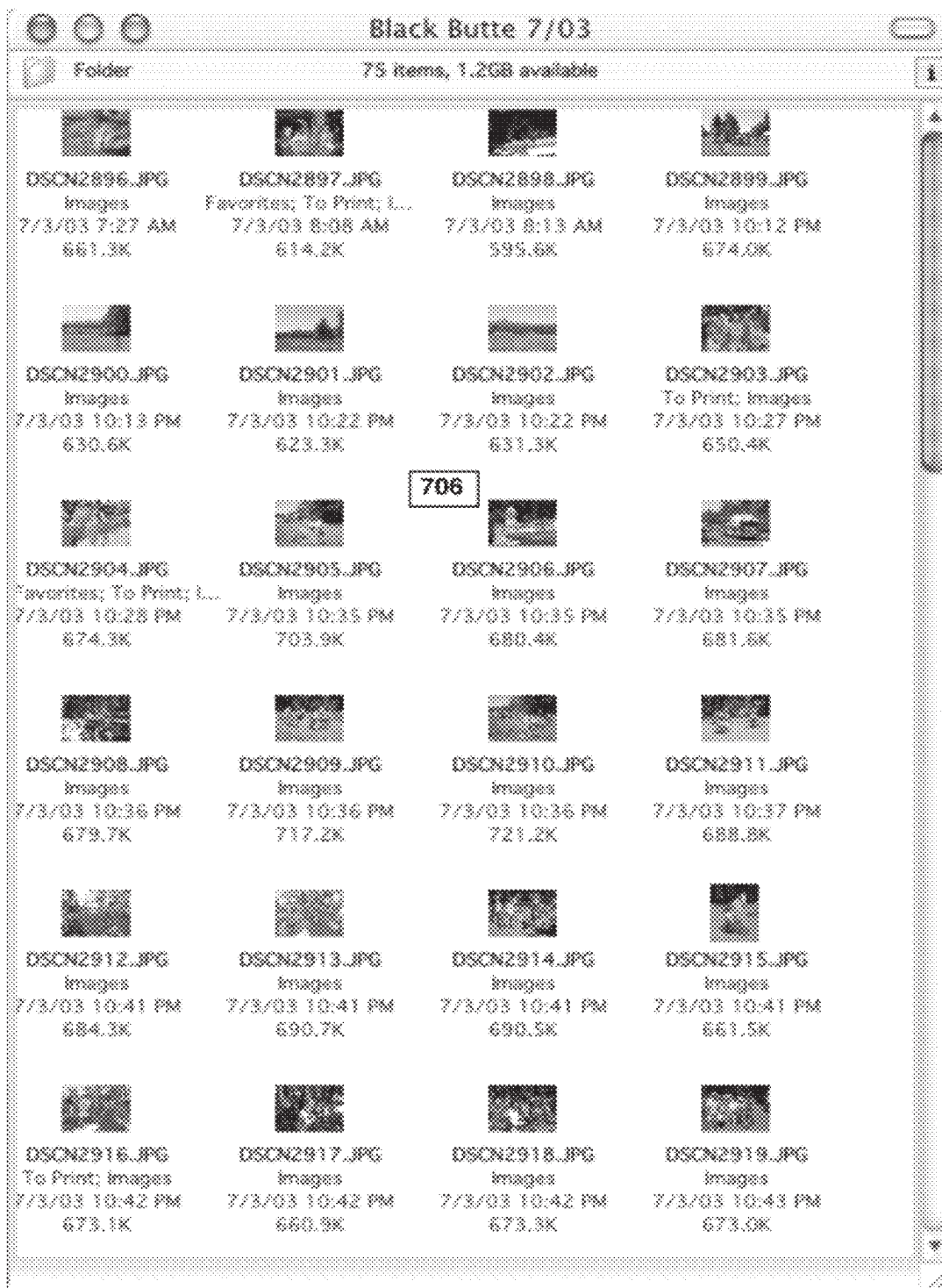


Figure 7b

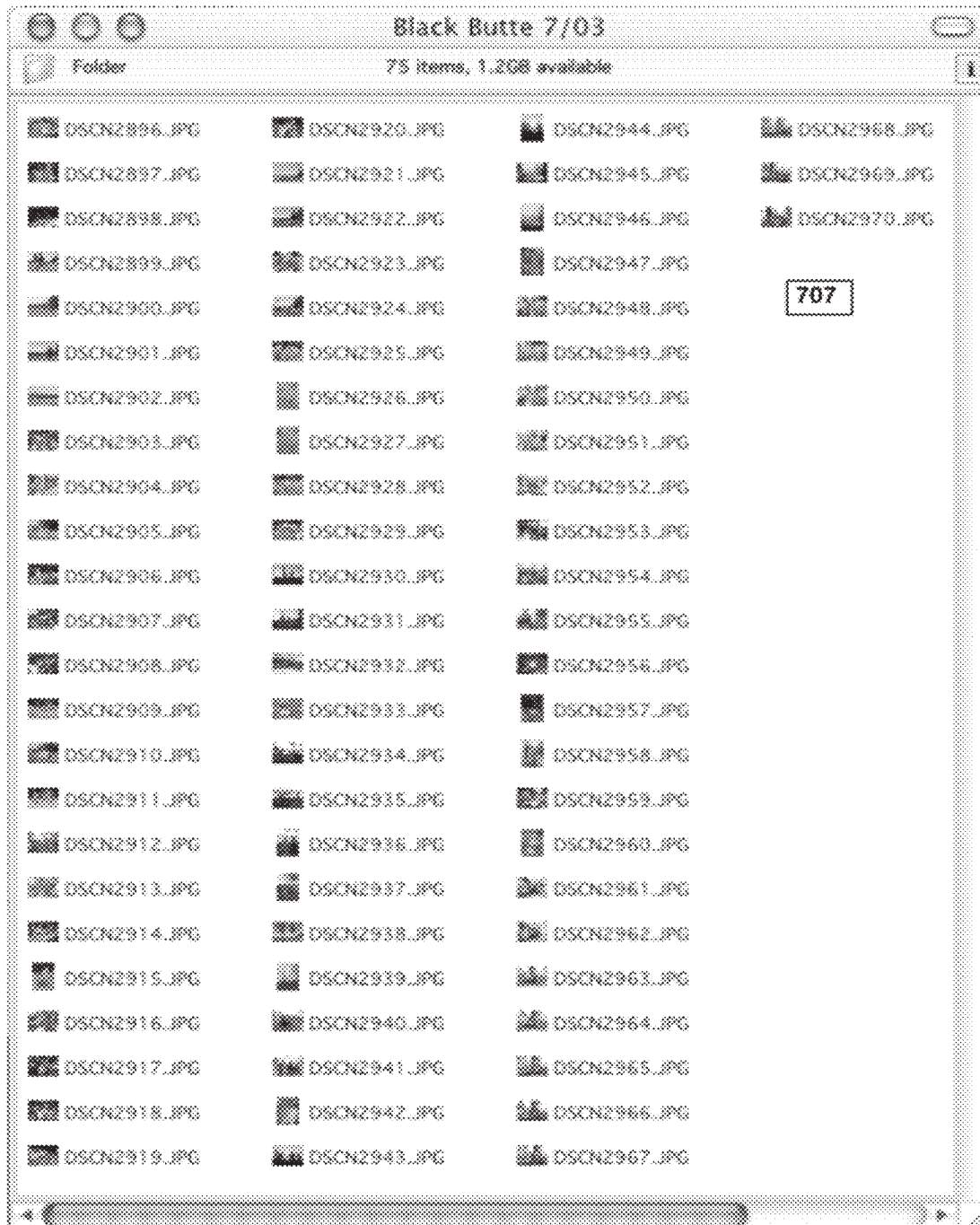
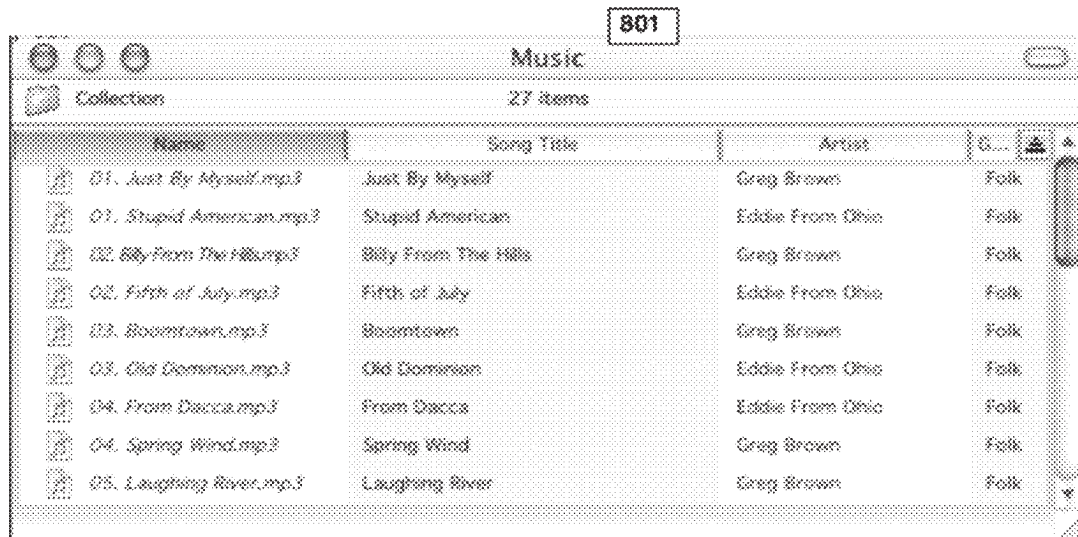


Figure 7c

**801**



Name	Song Title	Artist	Genre
01. Just By Myself.mp3	Just By Myself	Greg Brown	Folk
01. Stupid American.mp3	Stupid American	Eddie From Ohio	Folk
02. Billy From The Hills.mp3	Billy From The Hills	Greg Brown	Folk
02. Fifth of July.mp3	Fifth of July	Eddie From Ohio	Folk
03. Boomtown.mp3	Boomtown	Greg Brown	Folk
03. Old Dominion.mp3	Old Dominion	Eddie From Ohio	Folk
04. From Dacca.mp3	From Dacca	Eddie From Ohio	Folk
04. Spring Wind.mp3	Spring Wind	Greg Brown	Folk
05. Laughing River.mp3	Laughing River	Greg Brown	Folk

**802**



Name	Song Title	Artist
01. Just By Myself.mp3	Just By Myself	Greg Brown
01. Stupid American.mp3	Stupid American	Eddie From Ohio
02. Billy From The Hills.mp3	Billy From The Hills	Greg Brown
02. Fifth of July.mp3	Fifth of July	Eddie From Ohio
03. Boomtown.mp3	Boomtown	Greg Brown
03. Old Dominion.mp3	Old Dominion	Eddie From Ohio

Figure 8



Figure 9a

```

906
Fullpath.c
CodeWarrior source file 907 8.9K 908 909 i
pascal OSErr FSPLocationFromFullPath(short fullPathLength,
    const void *fullPath,
    FSSpec *spec)
{
    AliasHandle alias;
    OSErr result;
    Boolean wasChanged;
    Str255 nullString;
    910

    /* Create a minimal alias from the full pathname */
    nullString[0] = 0; /* null string to indicate no zone or server name */
    result = NewAliasMinimalFromFullPath(fullPathLength, fullPath, nullString,
    nullString, &alias);
    if ( result == noErr )
    {
        /* Let the Alias Manager resolve the alias. */
        result = ResolveAlias(NULL, alias, spec, &wasChanged);

        /* work around Alias Mgr sloppy volume matching bug */
        if ( spec->RefNum == 0 )
        {
            /* invalidate wrong FSSpec */
            spec->parID = 0;
            spec->name[0] = 0;
            result = noErr;
        }
        DisposeHandle((Handle)alias); /* Free up memory used */
    }
    return ( result );
}

```

Figure 9b

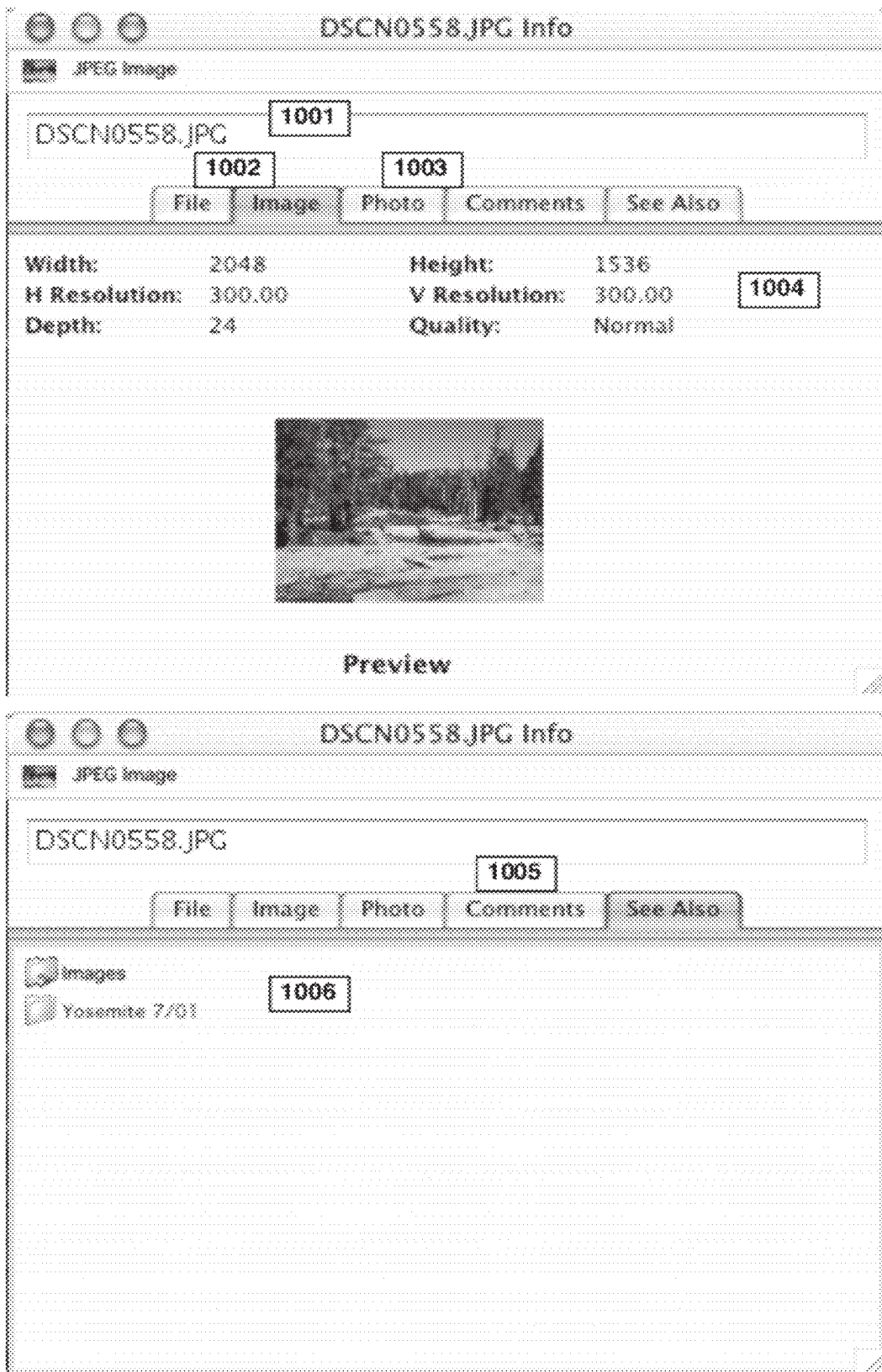


Figure 10a



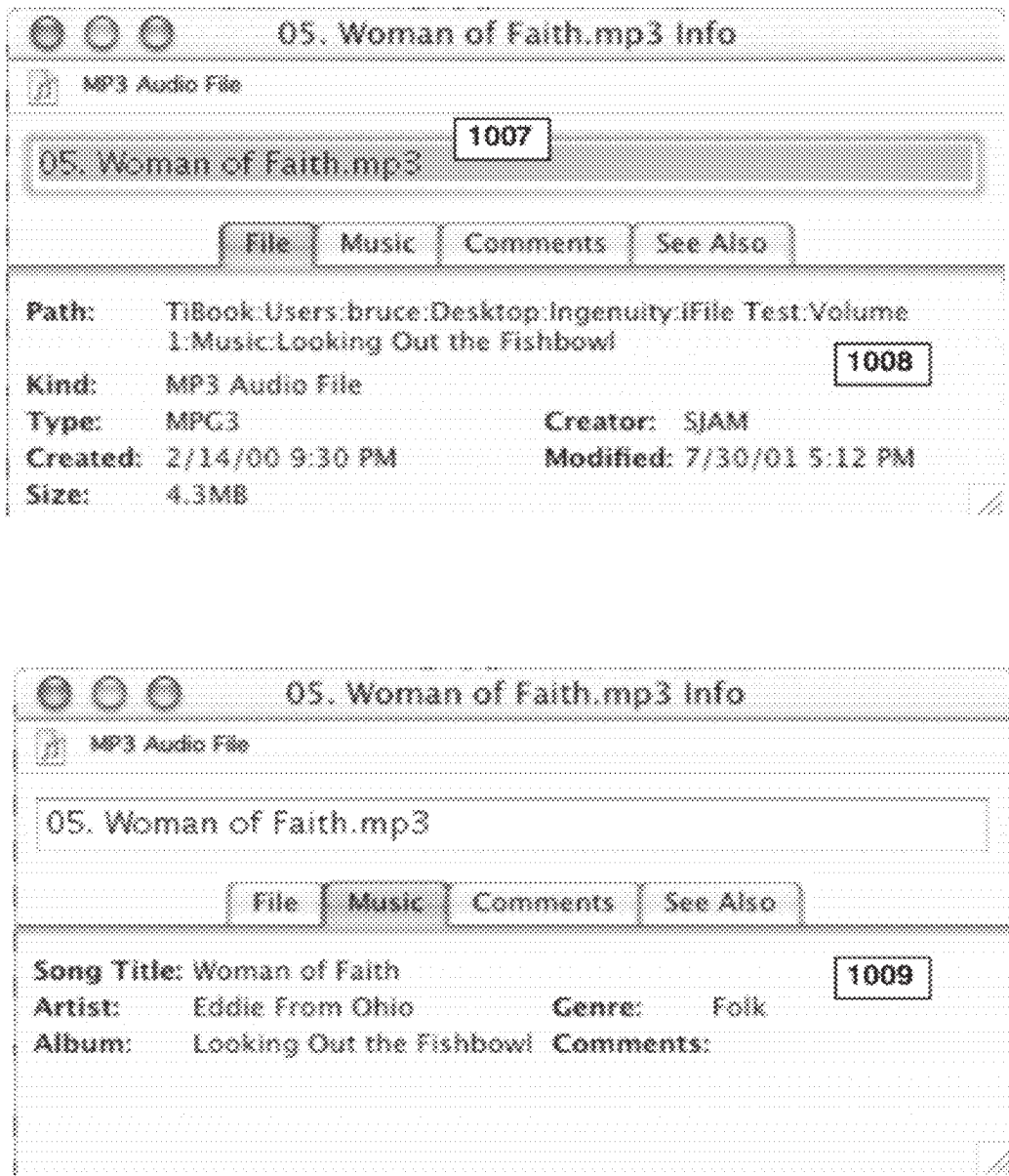


Figure 10b

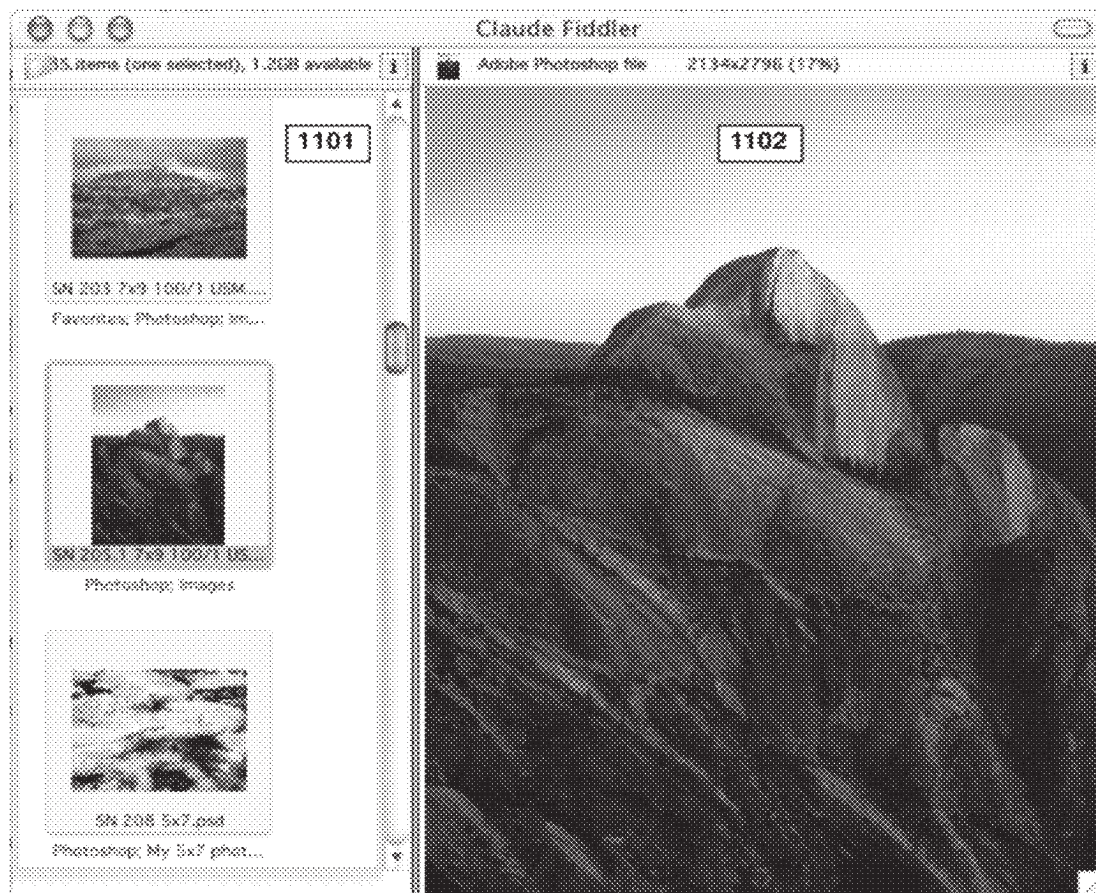


Figure 11a

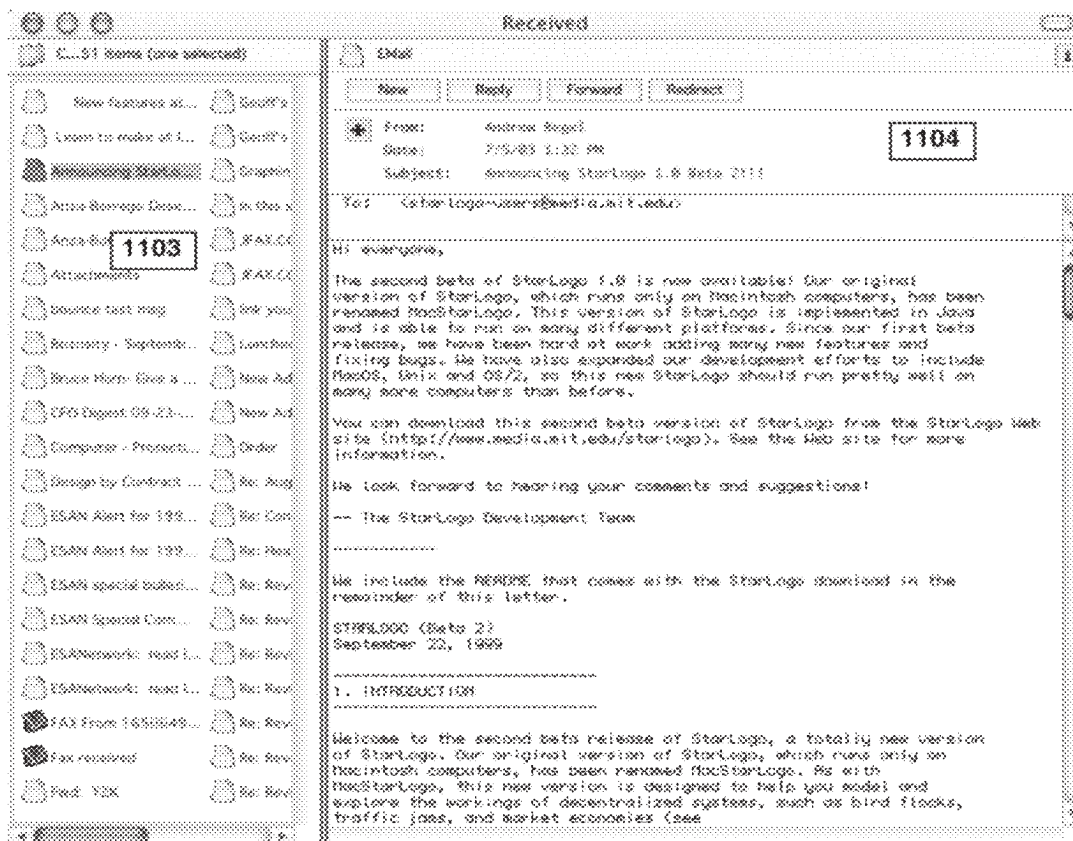


Figure 11b

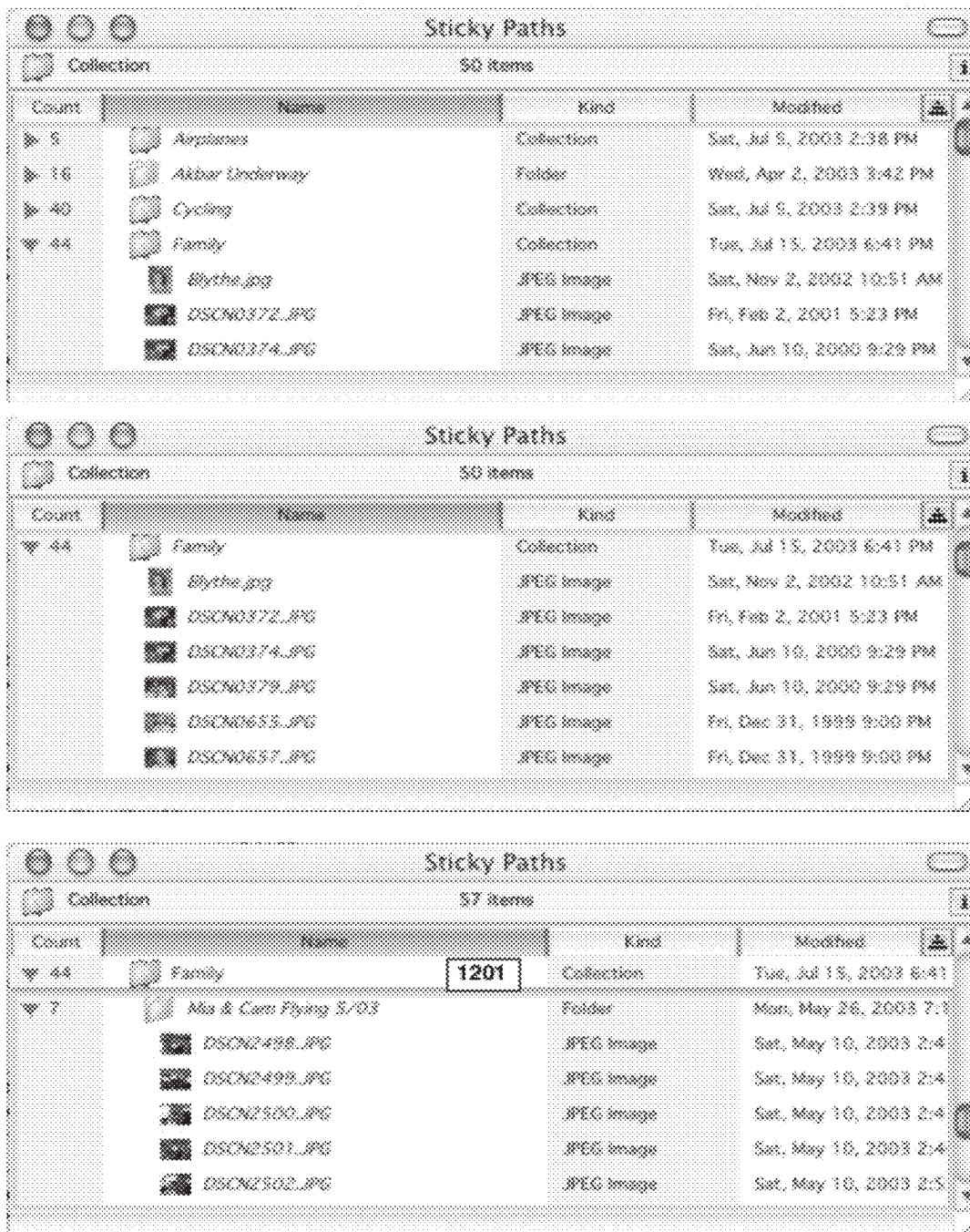


Figure 12a

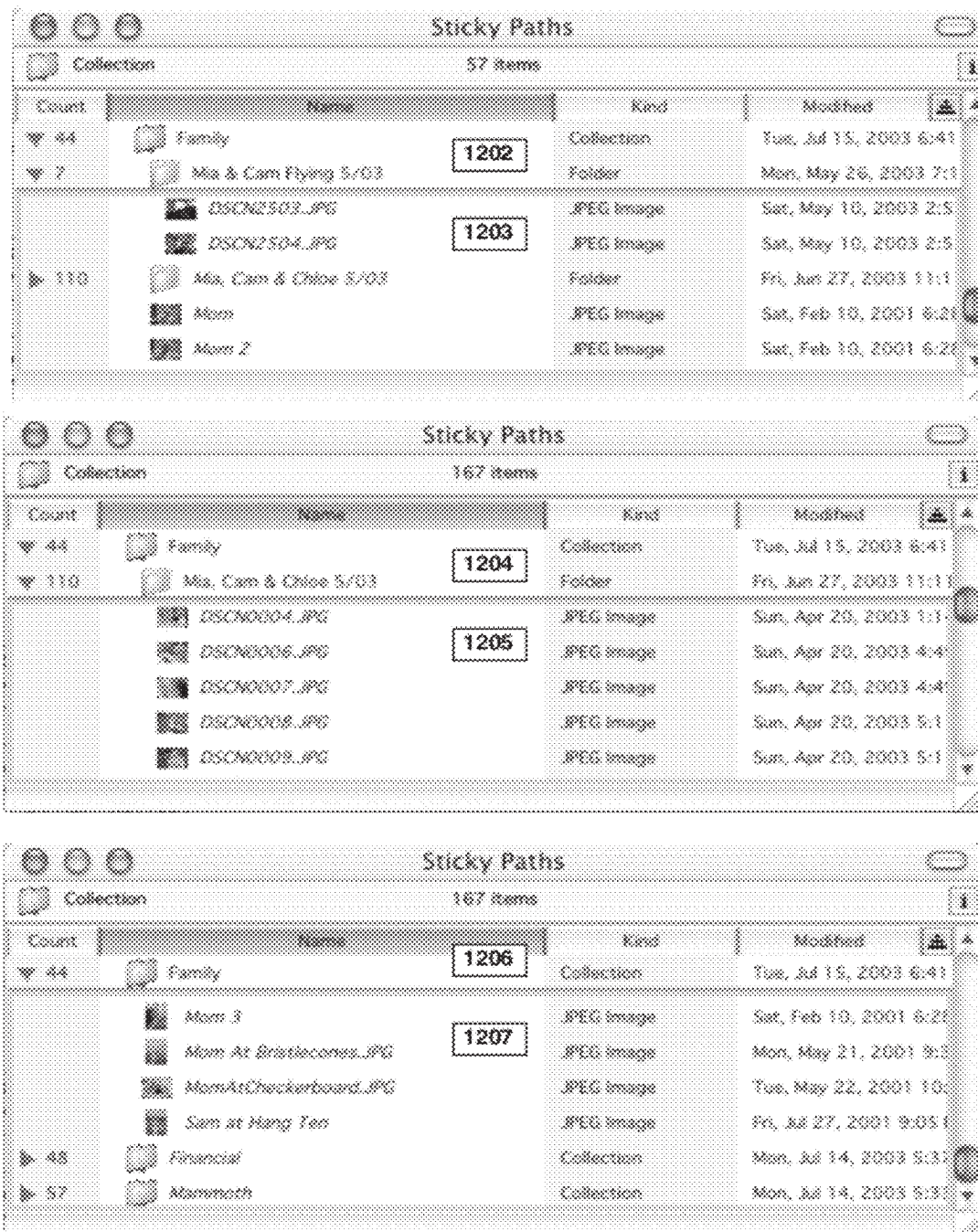


Figure 12b

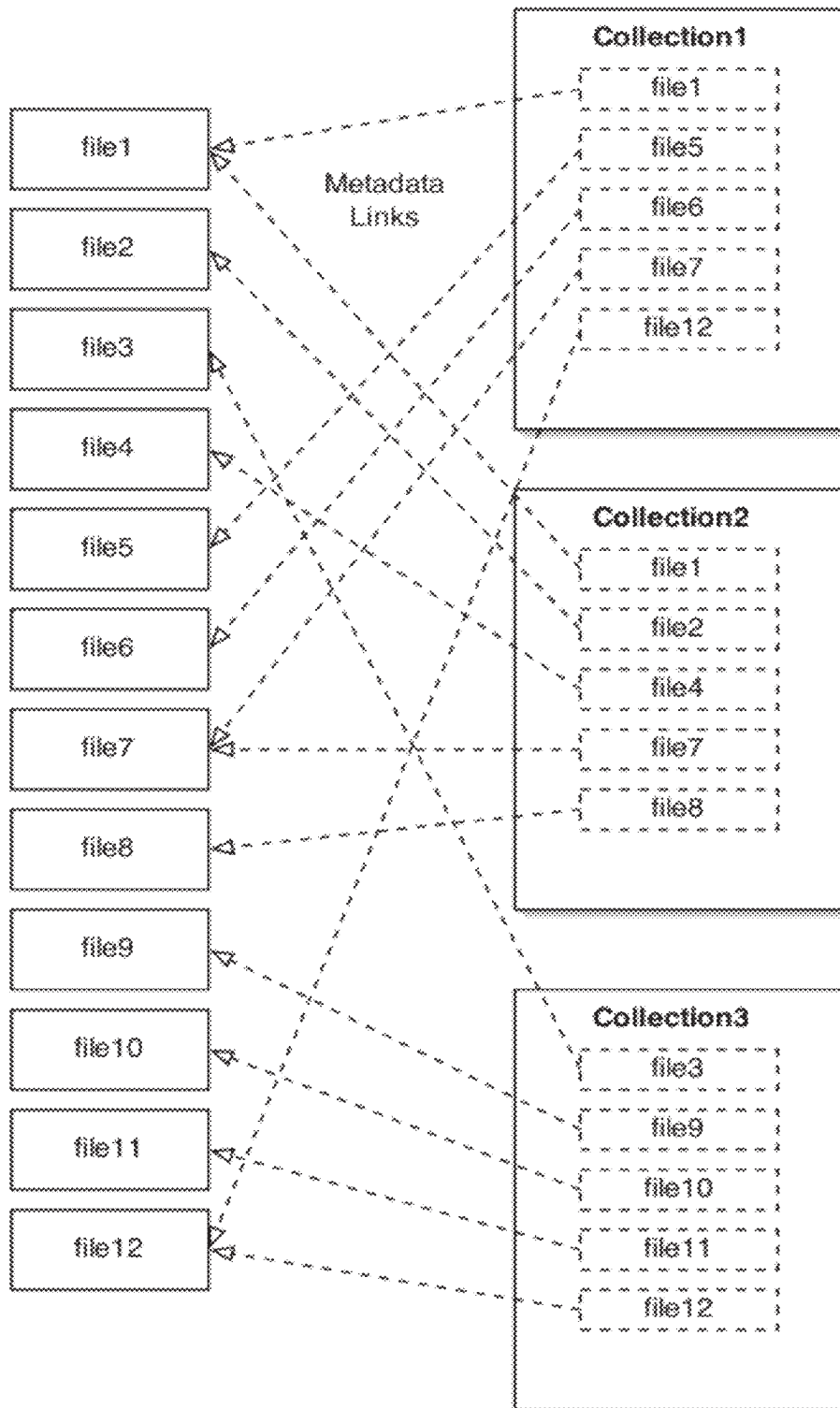


Figure 13

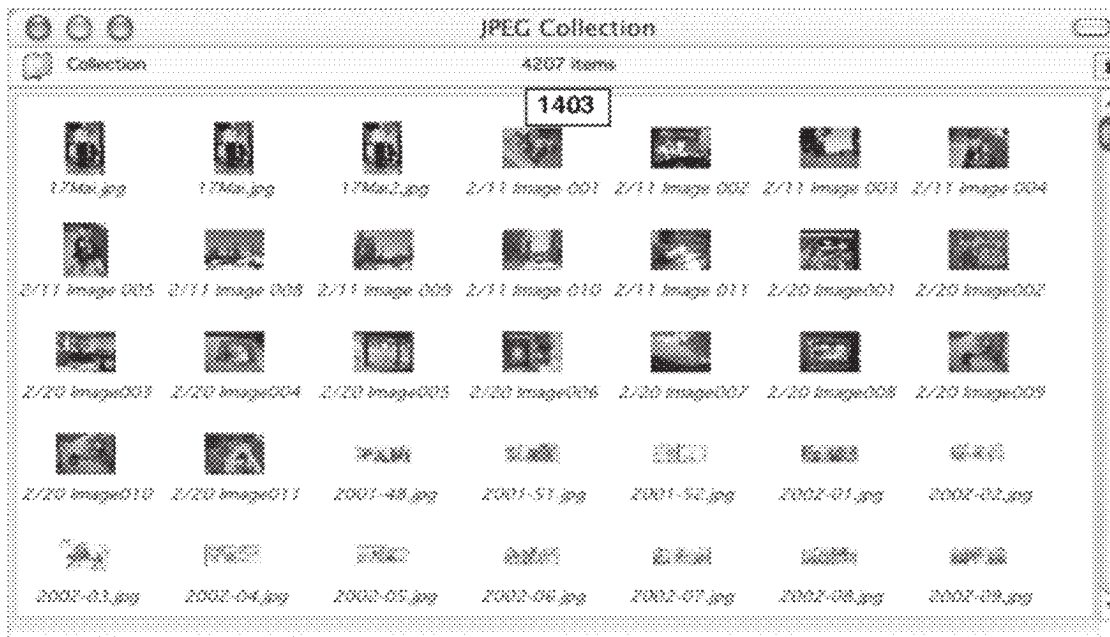
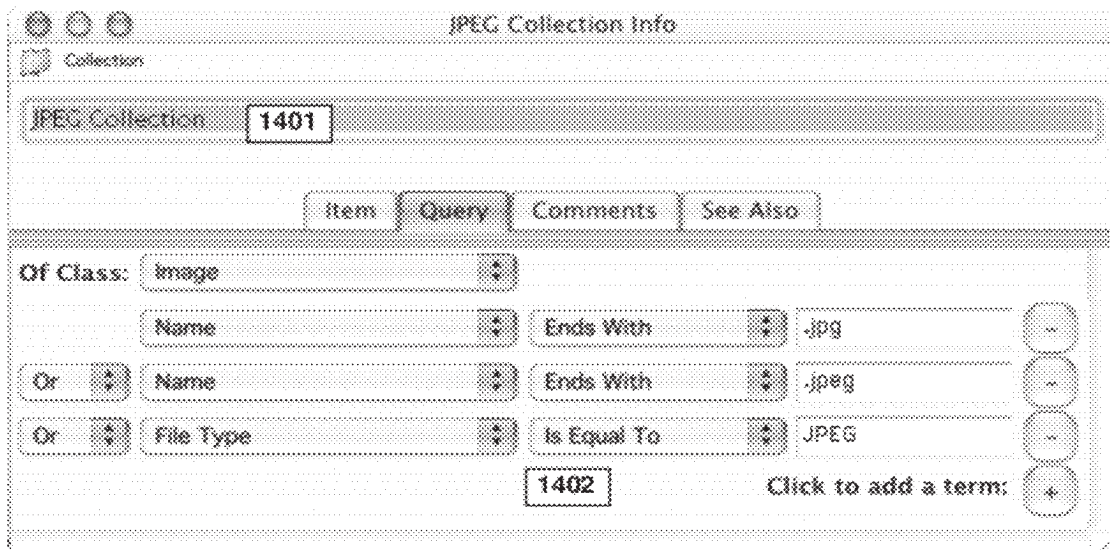


Figure 14

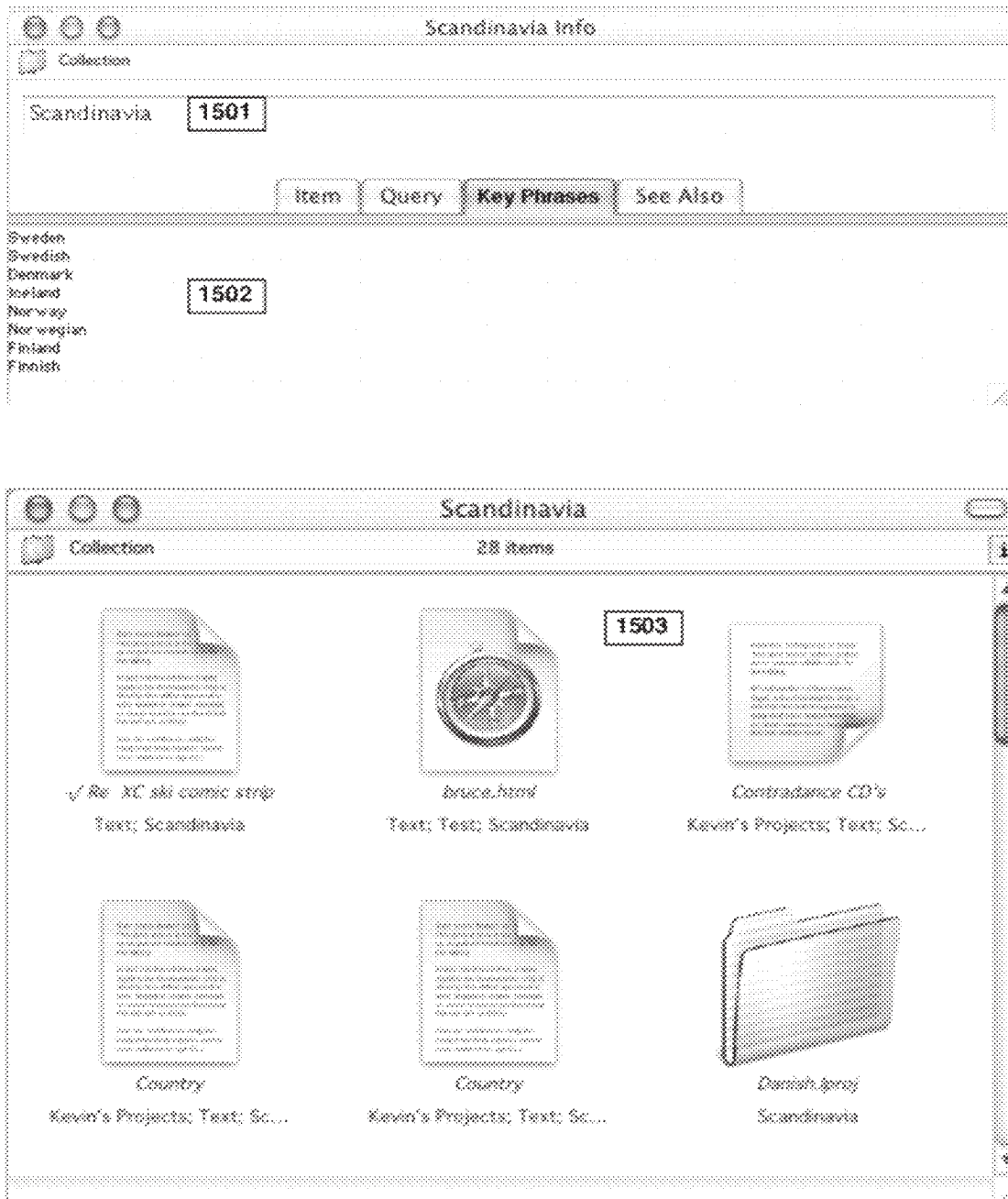
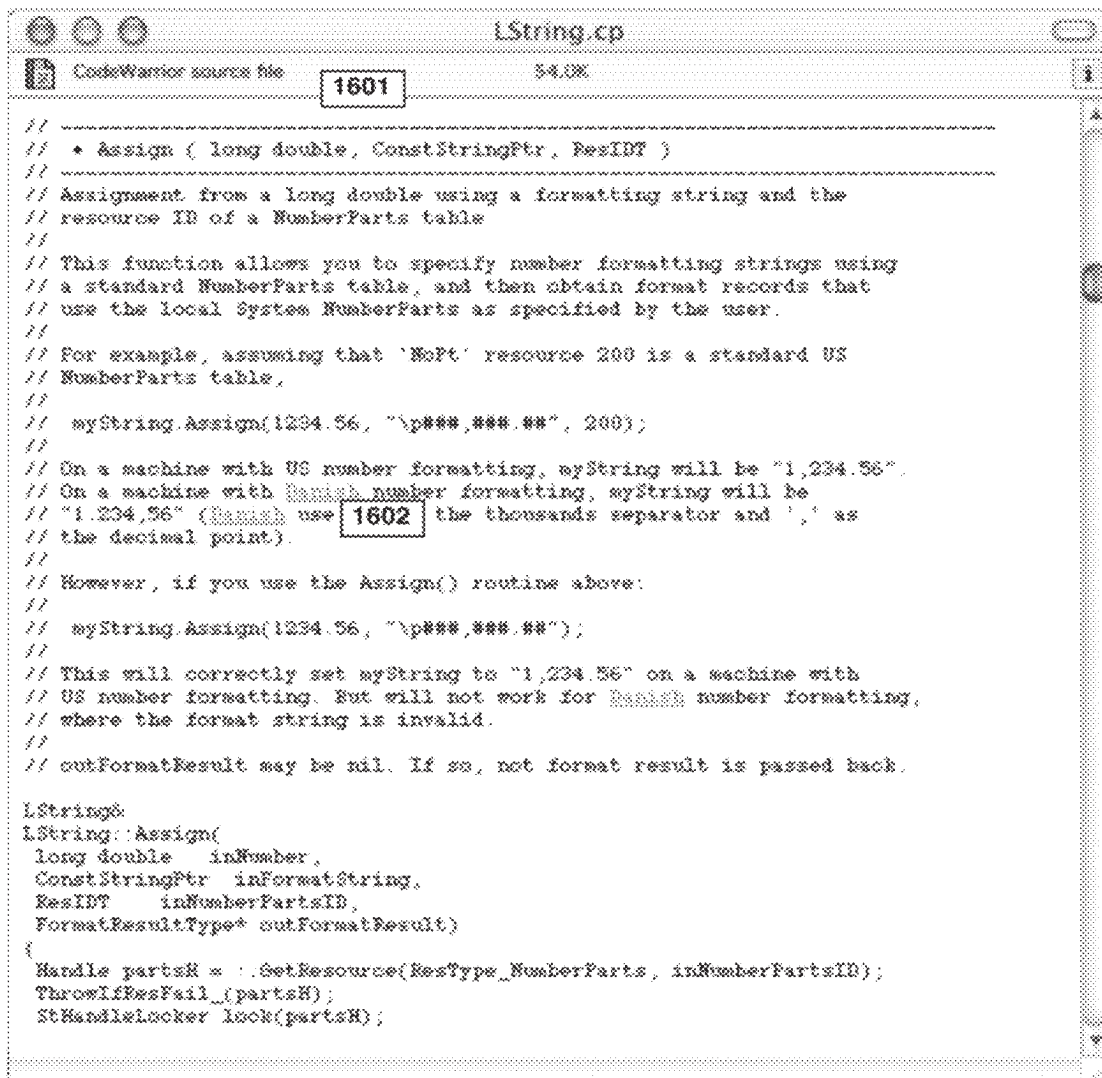


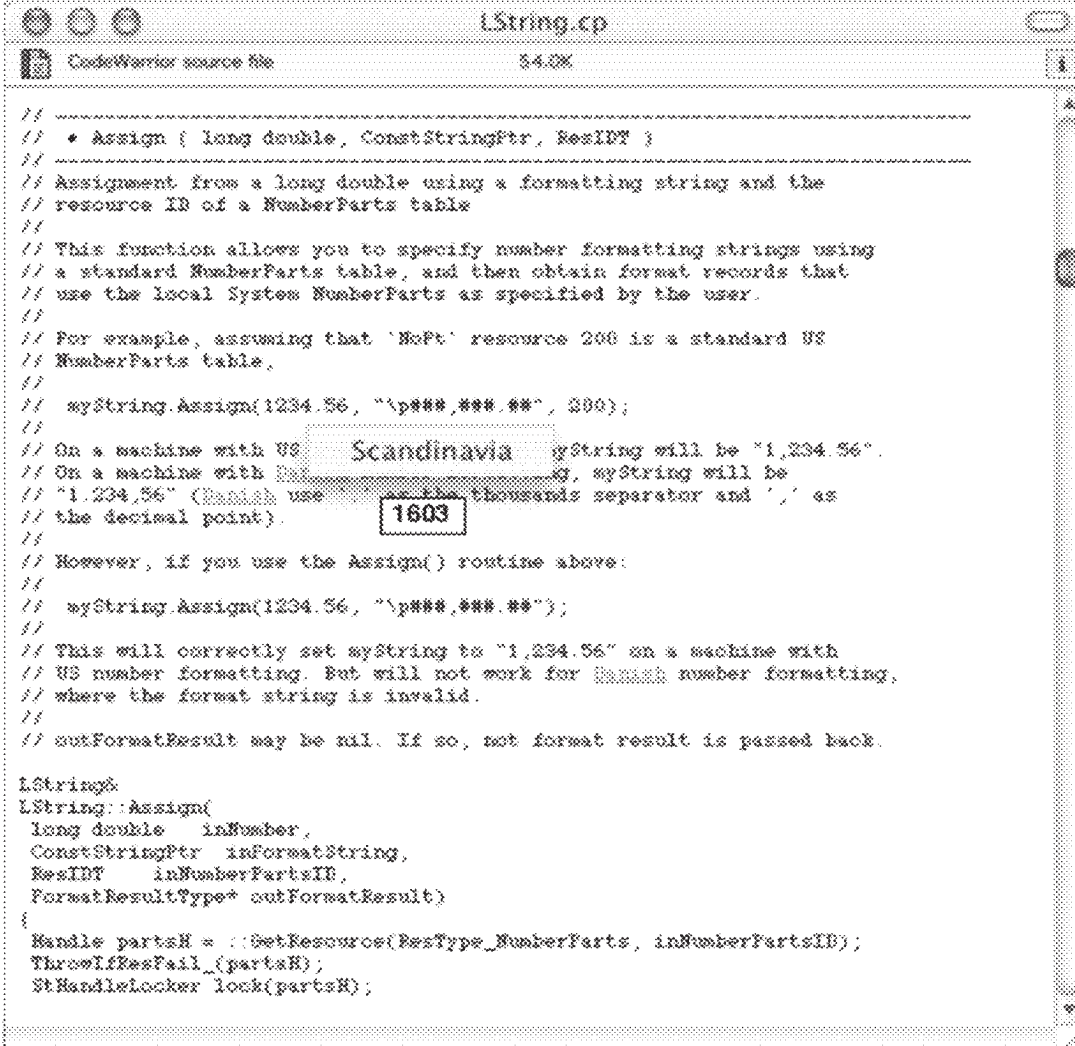
Figure 15





```
CodeWarrior source file 1601 54.0K  
// -----  
// * Assign ( long double, ConstStringPtr, ResIDT )  
// -----  
// Assignment from a long double using a formatting string and the  
// resource ID of a NumberParts table  
//  
// This function allows you to specify number formatting strings using  
// a standard NumberParts table, and then obtain format records that  
// use the local System NumberParts as specified by the user.  
//  
// For example, assuming that 'NoPt' resource 200 is a standard US  
// NumberParts table,  
//  
// myString.Assign(1234.56, "\p###,###.##", 200);  
//  
// On a machine with US number formatting, myString will be "1,234.56".  
// On a machine with Spanish number formatting, myString will be  
// "1.234,56" (Spanish use 1602 the thousands separator and ',' as  
// the decimal point).  
//  
// However, if you use the Assign() routine above:  
//  
// myString.Assign(1234.56, "\p###,###.##");  
//  
// This will correctly set myString to "1,234.56" on a machine with  
// US number formatting. But will not work for Spanish number formatting,  
// where the format string is invalid.  
//  
// outFormatResult may be nil. If so, not format result is passed back.  
  
LString&  
LString::Assign(  
    long double    inNumber,  
    ConstStringPtr inFormatString,  
    ResIDT        inNumberPartsID,  
    FormatResultType* outFormatResult)  
{  
    Handle partsR = :.GetResource(ResType_NumberParts, inNumberPartsID);  
    ThrowIfResFail_(partsR);  
    StHandleLocker lock(partsR);
```

Figure 16a



```

// -----
// * Assign ( long double, ConstStringPtr, ResIDT )
// -----
// Assignment from a long double using a formatting string and the
// resource ID of a NumberParts table
//
// This function allows you to specify number formatting strings using
// a standard NumberParts table, and then obtain format records that
// use the local System NumberParts as specified by the user.
//
// For example, assuming that 'NoPt' resource 200 is a standard US
// NumberParts table,
//
// myString.Assign(1234.56, "\p###,###.##", 200);
//
// On a machine with US NumberParts, myString will be "1,234.56".
// On a machine with Danish NumberParts, myString will be
// "1.234,56" (Danish use the thousands separator and ',' as
// the decimal point).
//
// However, if you use the Assign() routine above:
//
// myString.Assign(1234.56, "\p###,###.##");
//
// This will correctly set myString to "1,234.56" on a machine with
// US number formatting. But will not work for Danish number formatting,
// where the format string is invalid.
//
// outFormatResult may be nil. If so, no format result is passed back.

LString&
LString::Assign(
    long double    inNumber,
    ConstStringPtr inFormatString,
    ResIDT         inNumberPartsID,
    FormatResultType* outFormatResult)
{
    Handle partsH = ::GetResource(ResType_NumberParts, inNumberPartsID);
    ThrowIfResFail_(partsH);
    StHandleLocker lock(partsH);

```

Figure 16b

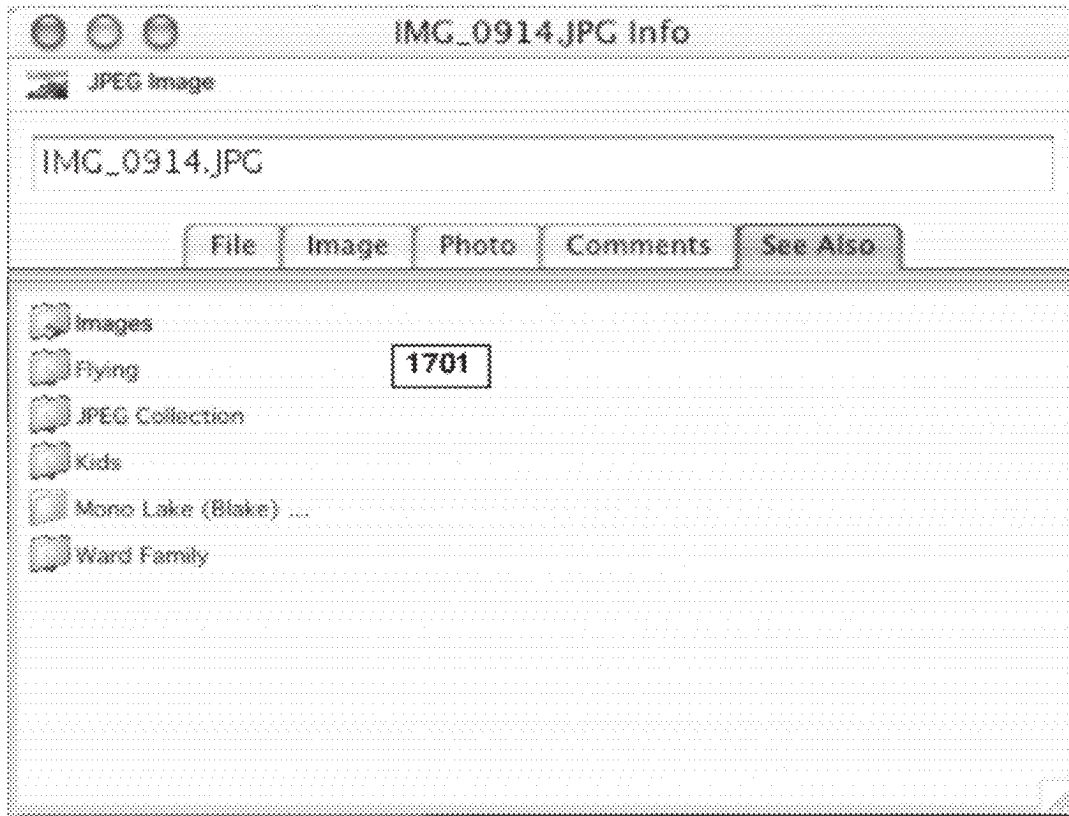


Figure 17

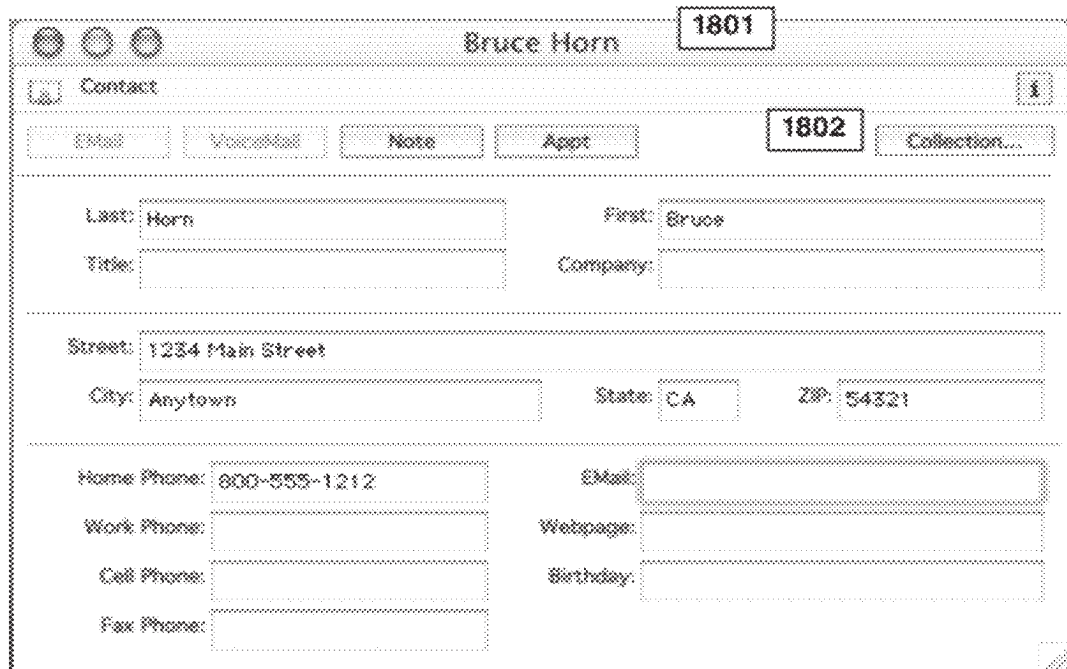


Figure 18

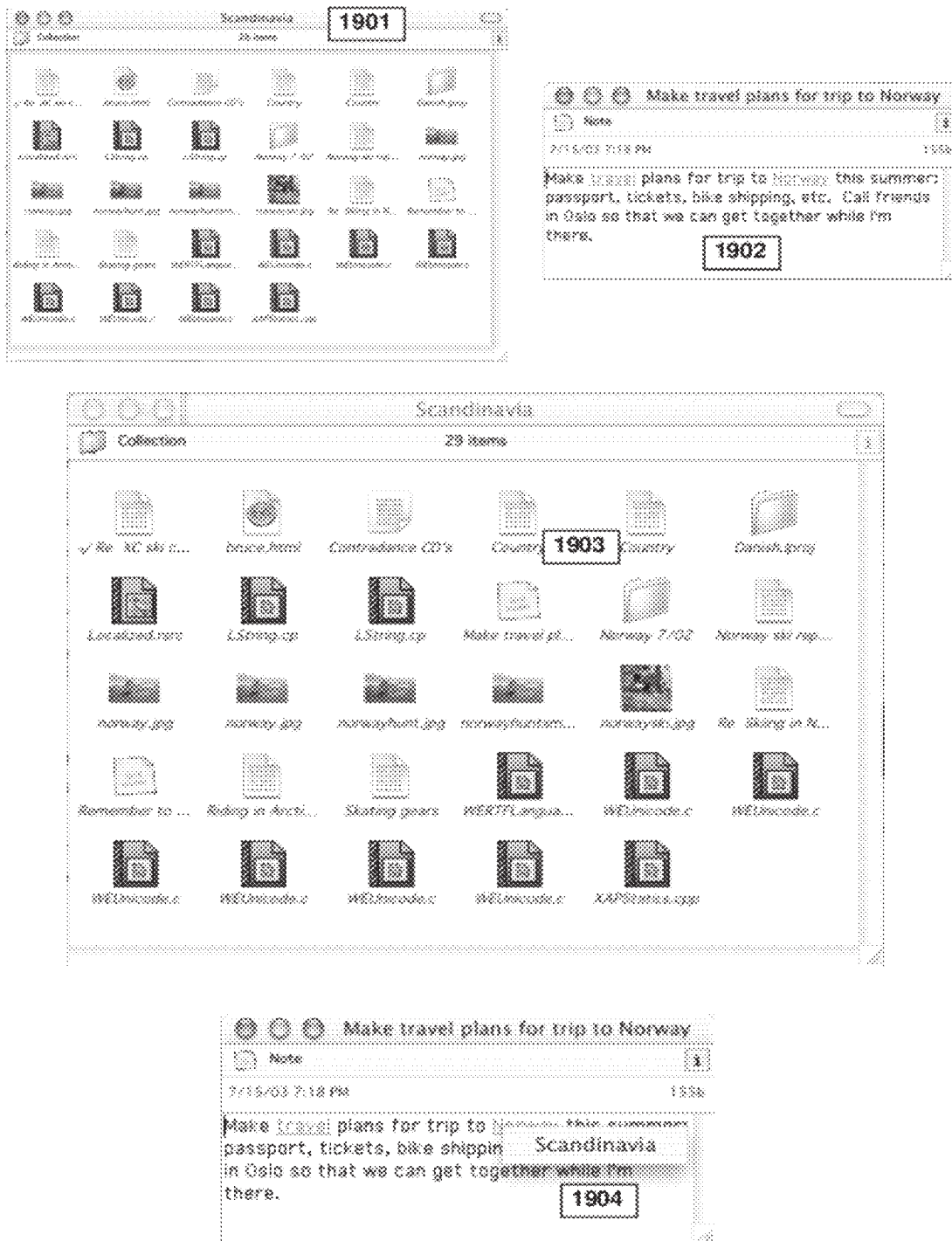


Figure 19

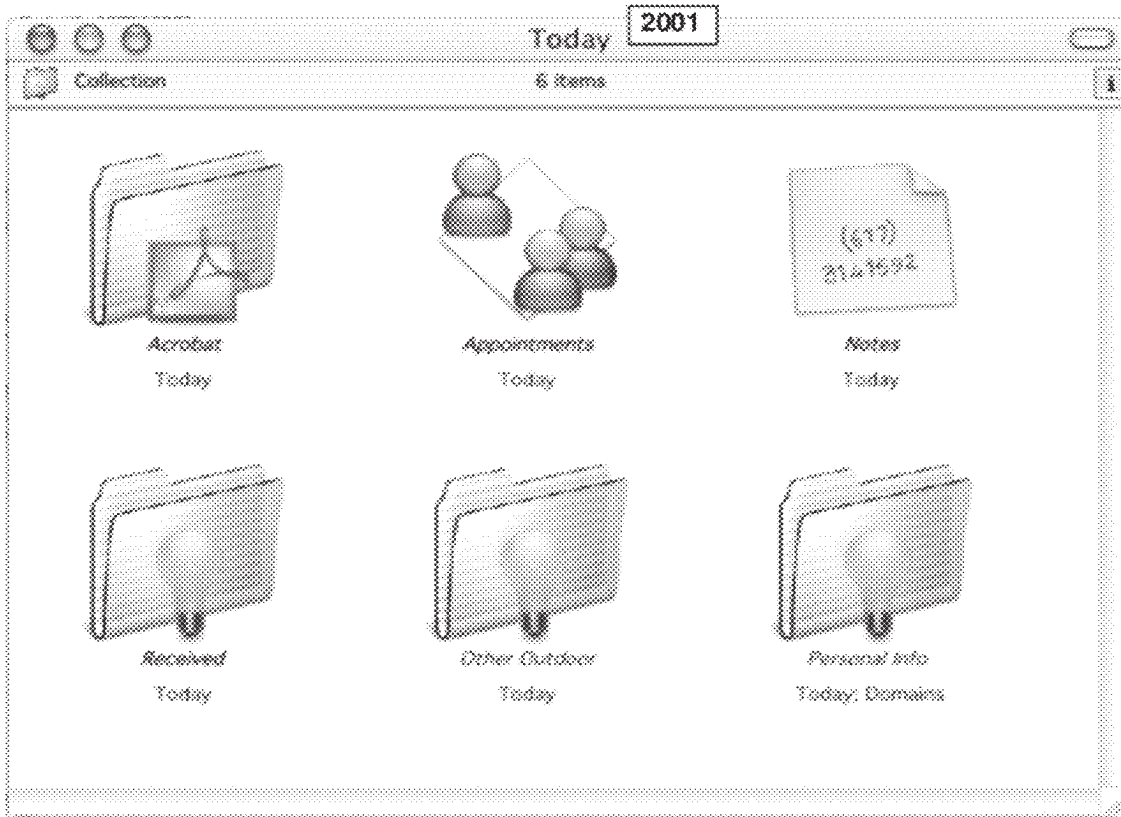


Figure 20

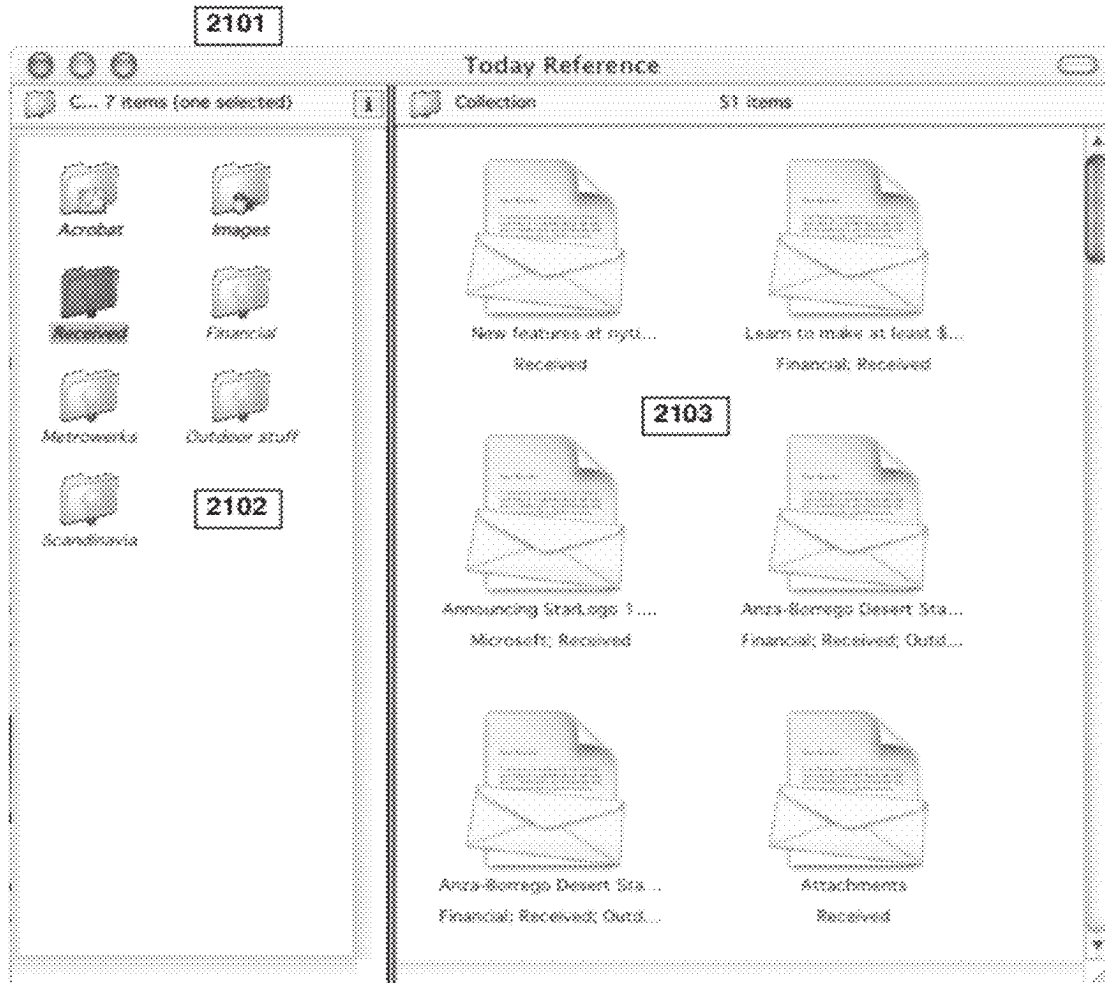


Figure 21

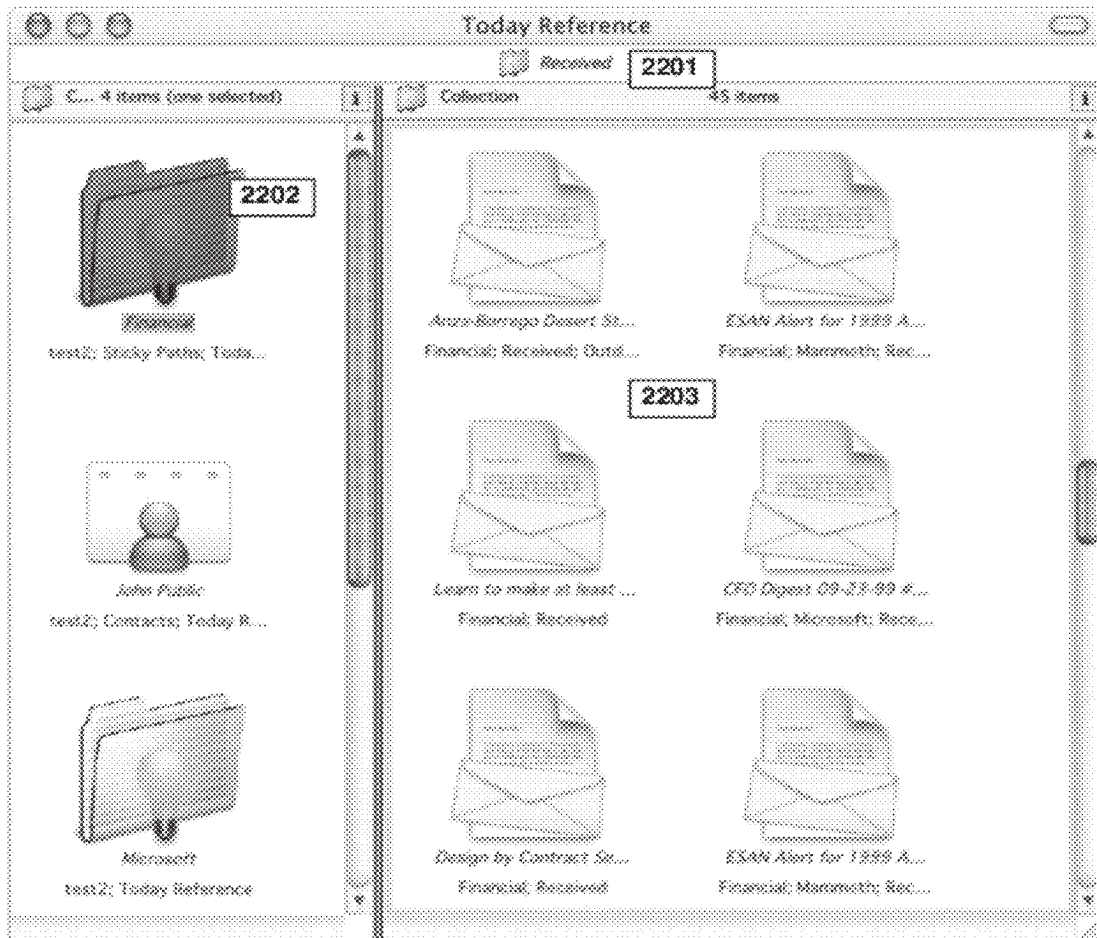


Figure 22



Files Domain	<b>Collections</b> Volumes	<b>Objects</b> Volume Directory File
Music Domain	<b>Collections</b> Artist Album Genre Rating	<b>Objects</b> mp3 file wmp file
Images Domain	<b>Collections</b> JPEG PDF BMP Black and White Nikon Coolpix ...	<b>Objects</b> JPEG file PDF file BMP file GIF file PSD file ...
WebDAV Domain	<b>Collections</b> Remote Volumes Websites	<b>Objects</b> Remote files
Finance Domain	<b>Collections</b> Accounts Payables Receivables	<b>Objects</b> Payees Checks ...

Figure 23

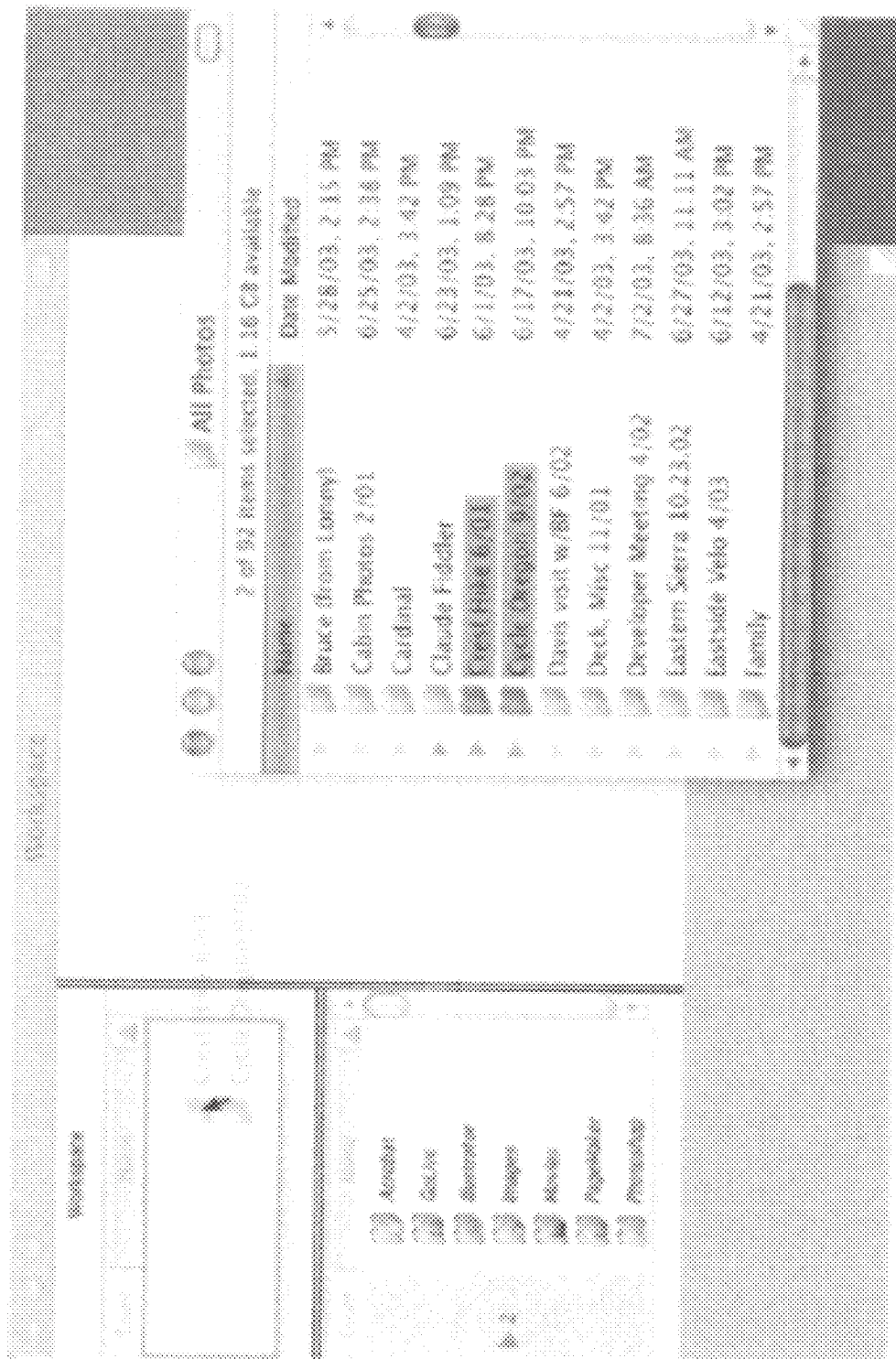


Figure 24

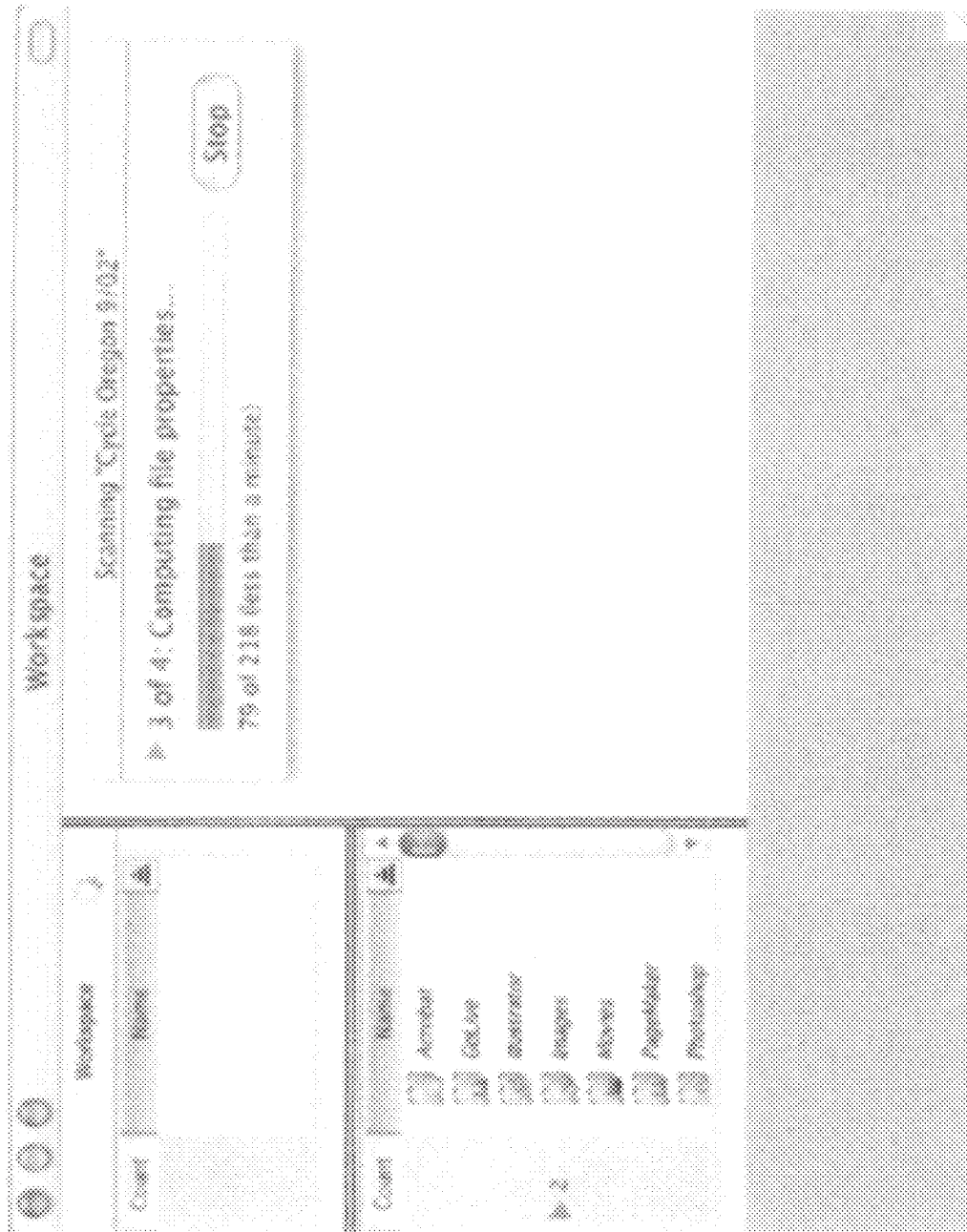


Figure 25a

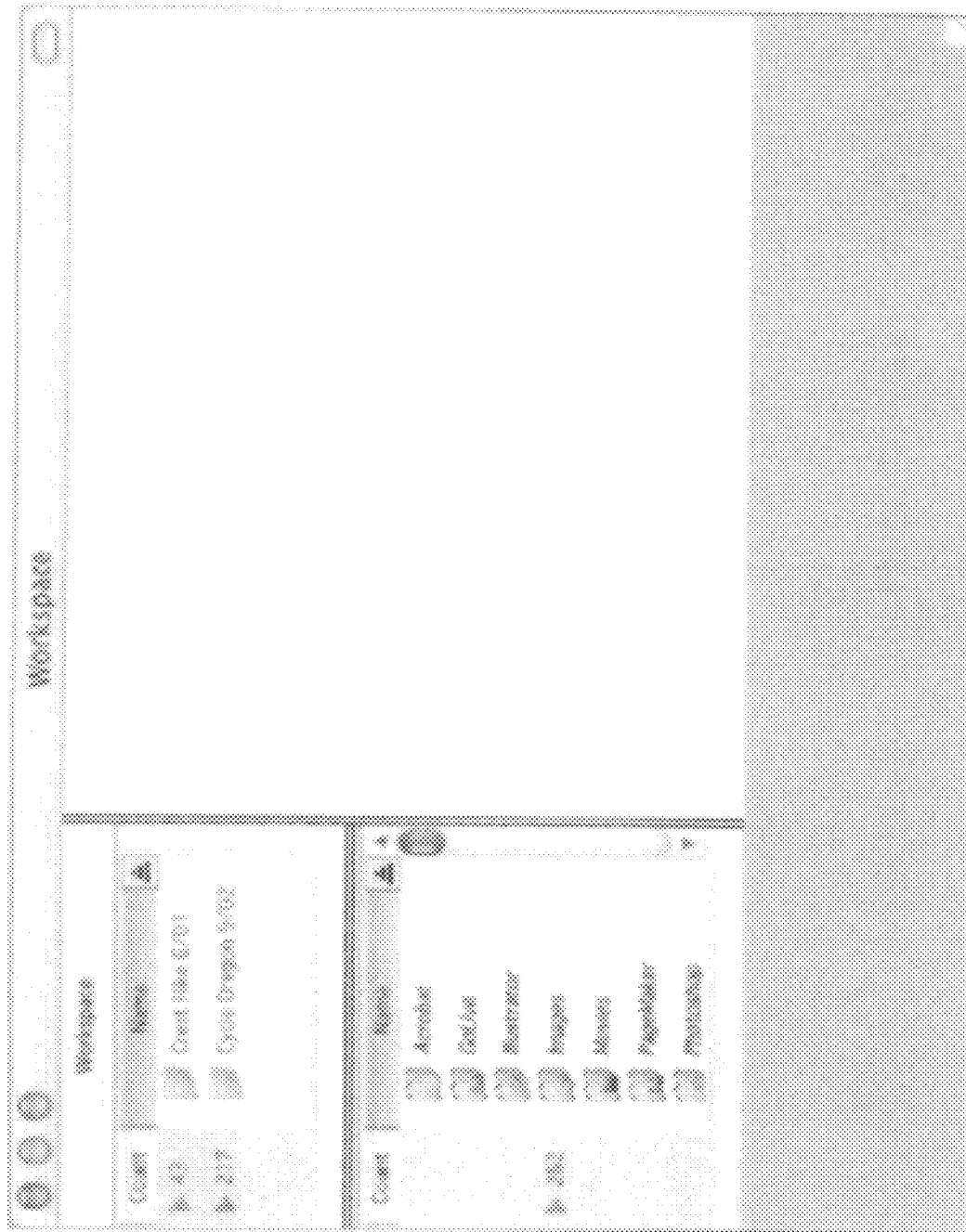


Figure 25b

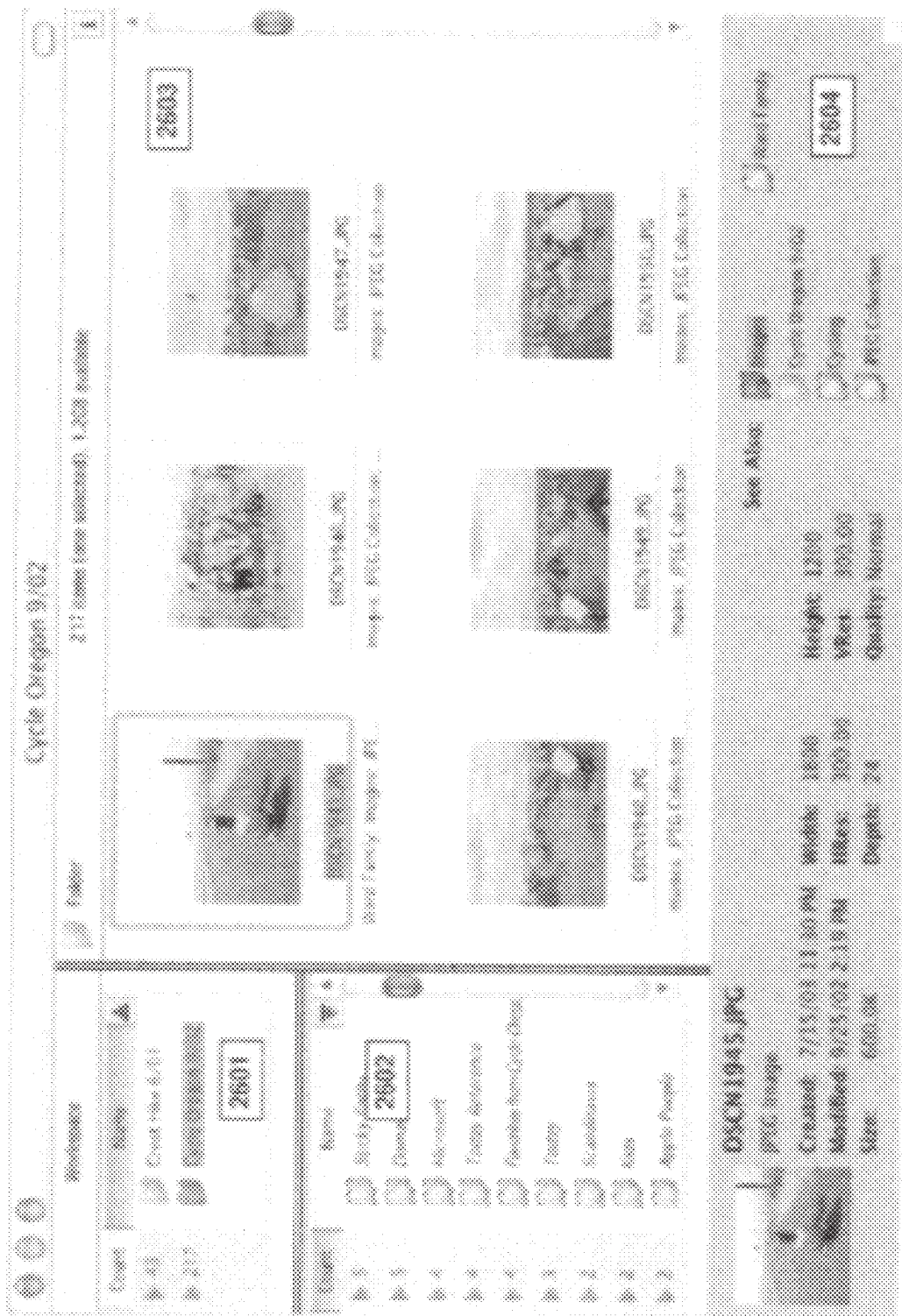


Figure 26