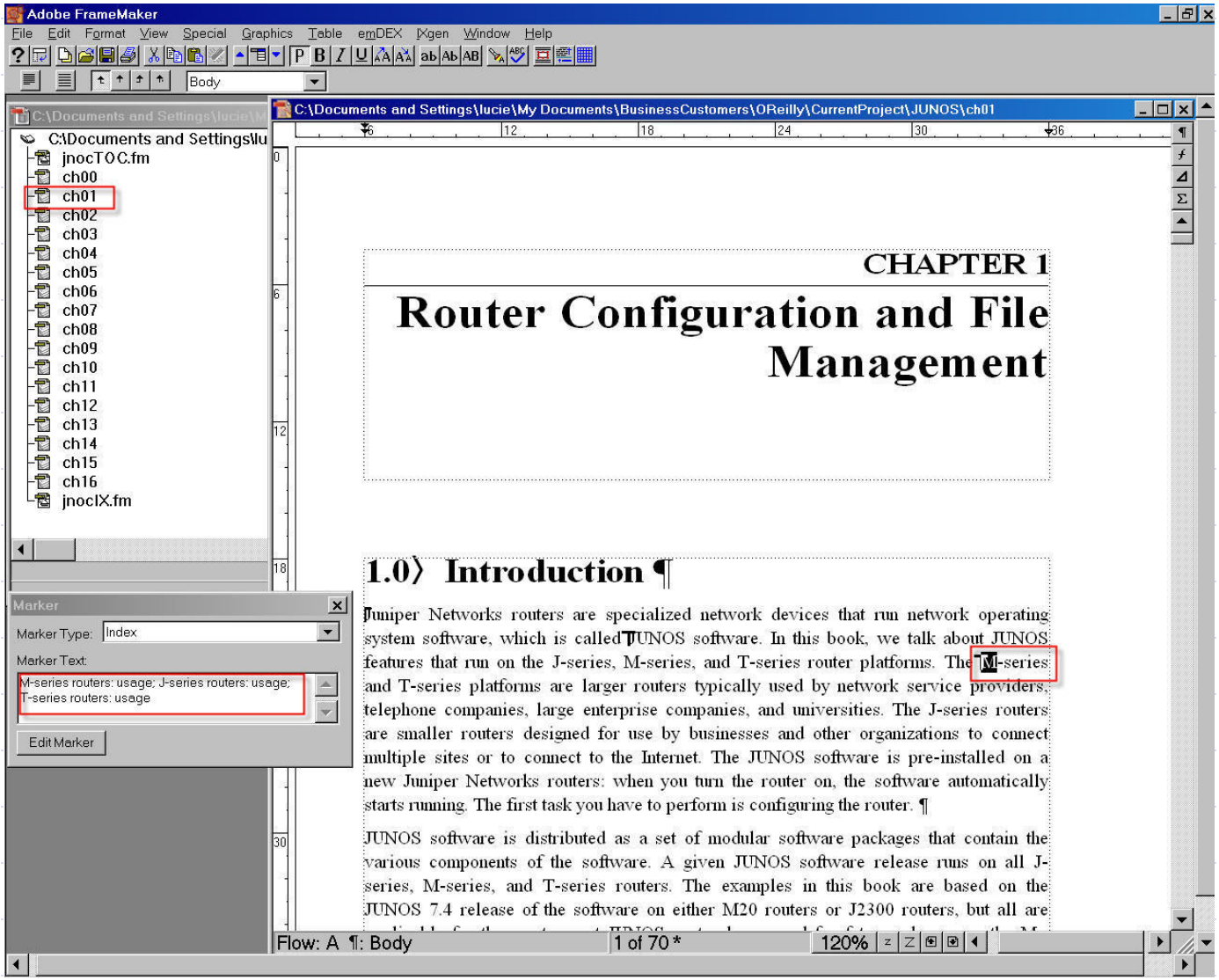
FrameMaker data entry process



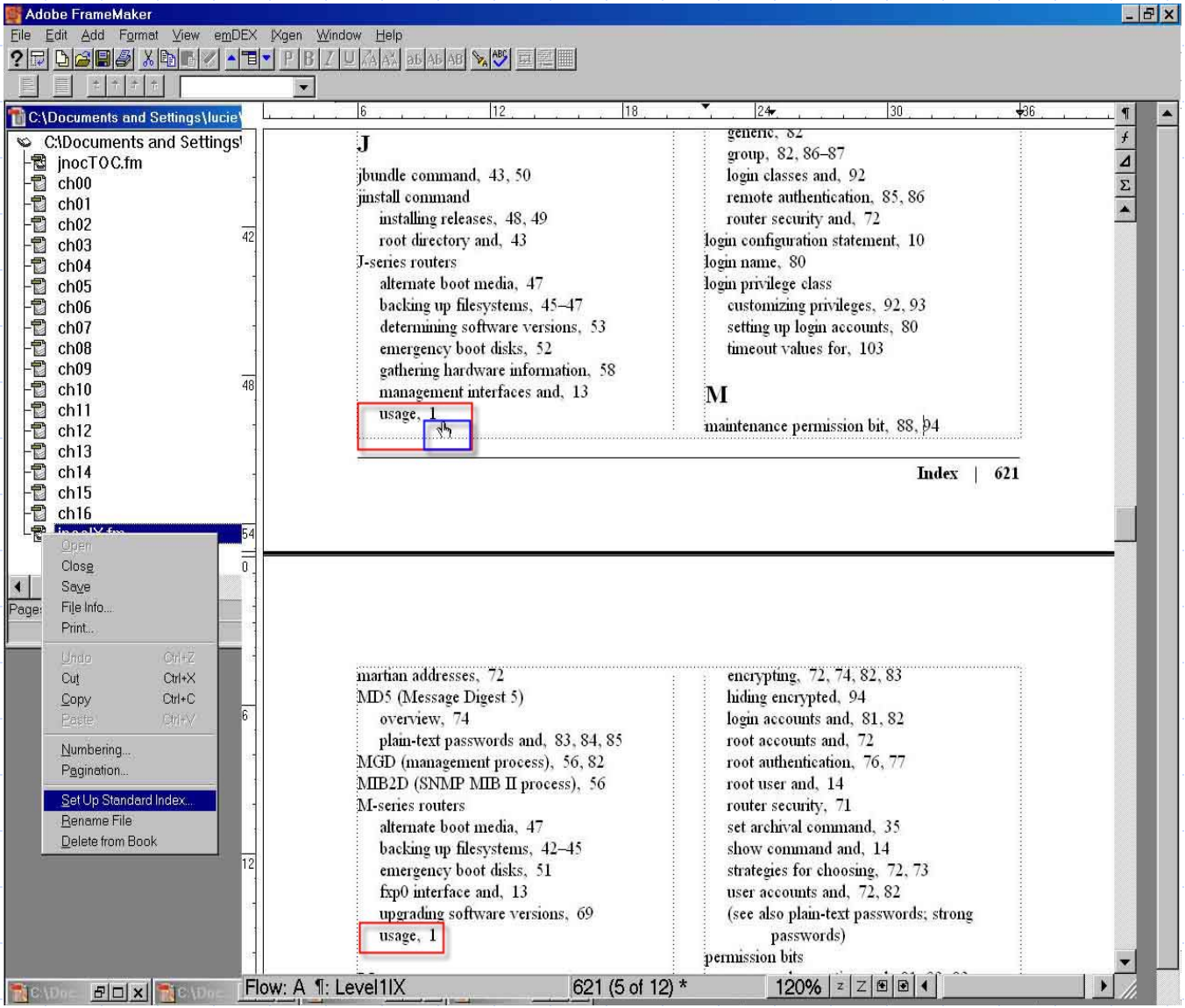
FM Process:

- open book
- open chapters
- index

Creating terms:

- index preview
- autocomplete
- marker box size
- special strings
- autogeneration

FrameMaker editing process



FM Process:

- generate index
- hyperlinks

Editing terms:

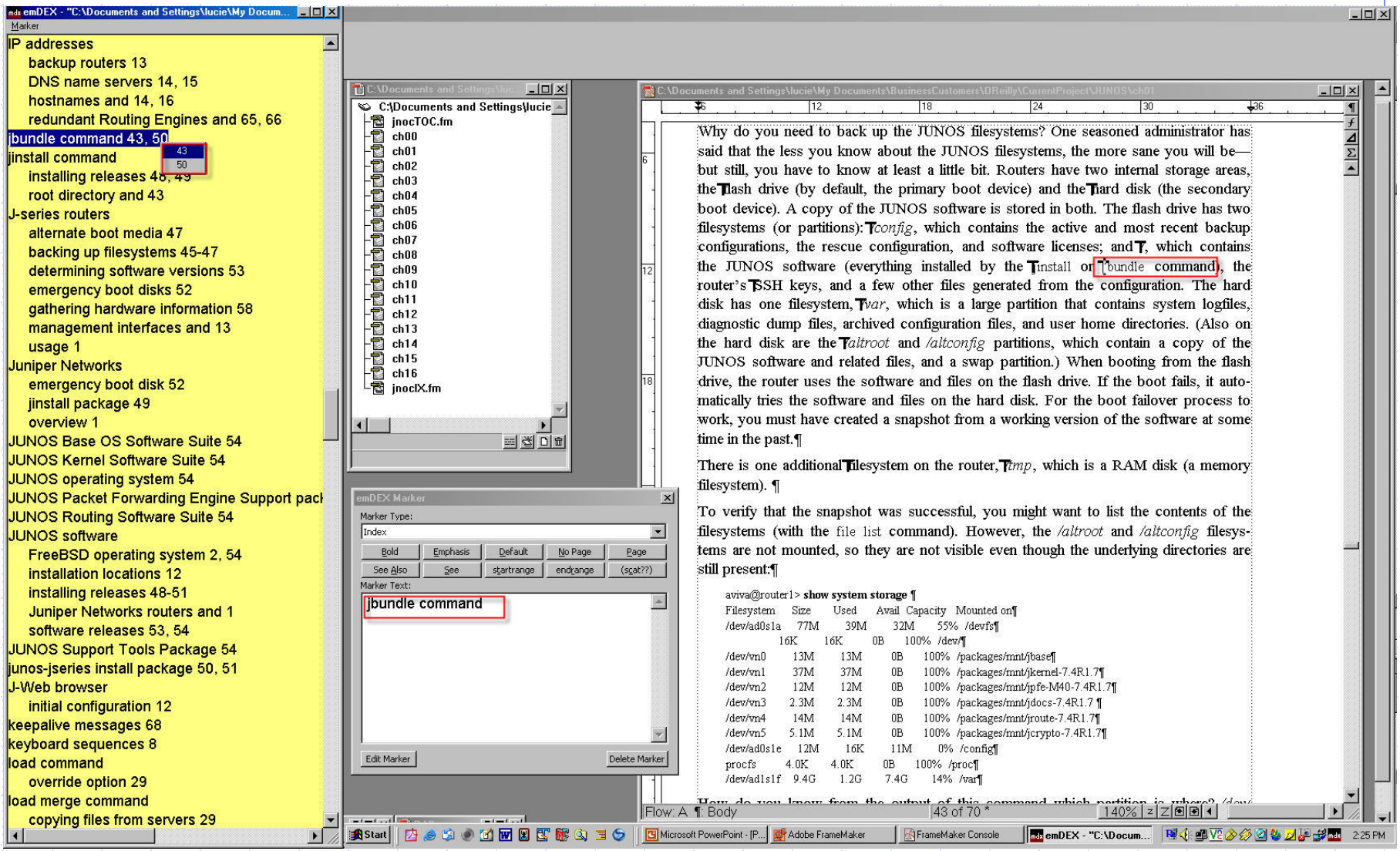
- change propagation
- index preview
- viewing entries

Third party utilities for FrameMaker

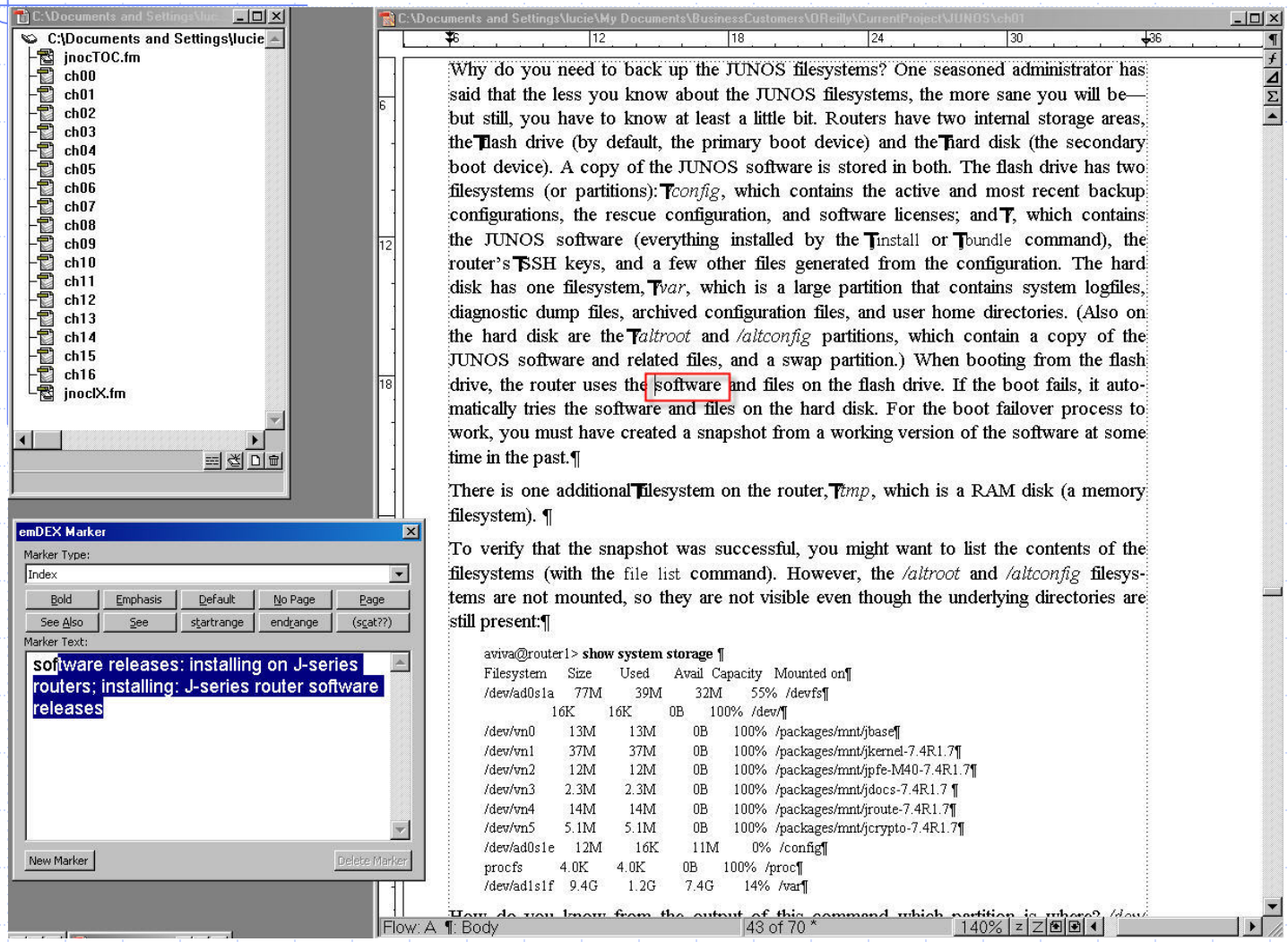
emDEX
http://www.emdex.ca

IXgen
http://www.fsatools.com

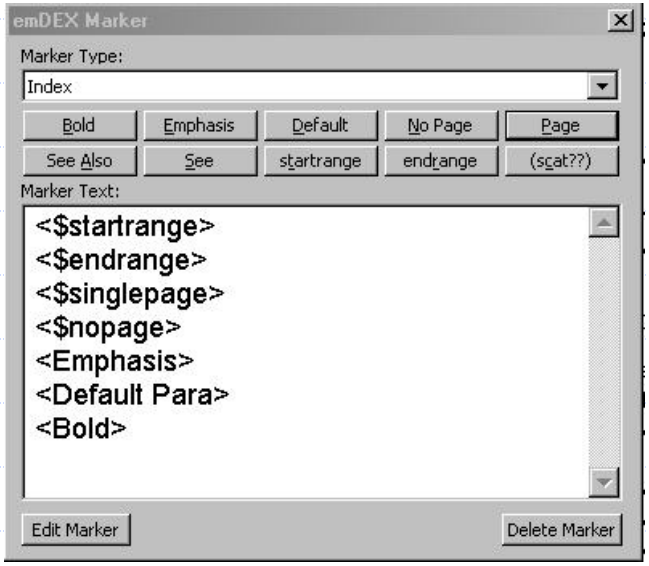
Benefits/functionality of third party utilities – index preview (emDEX)

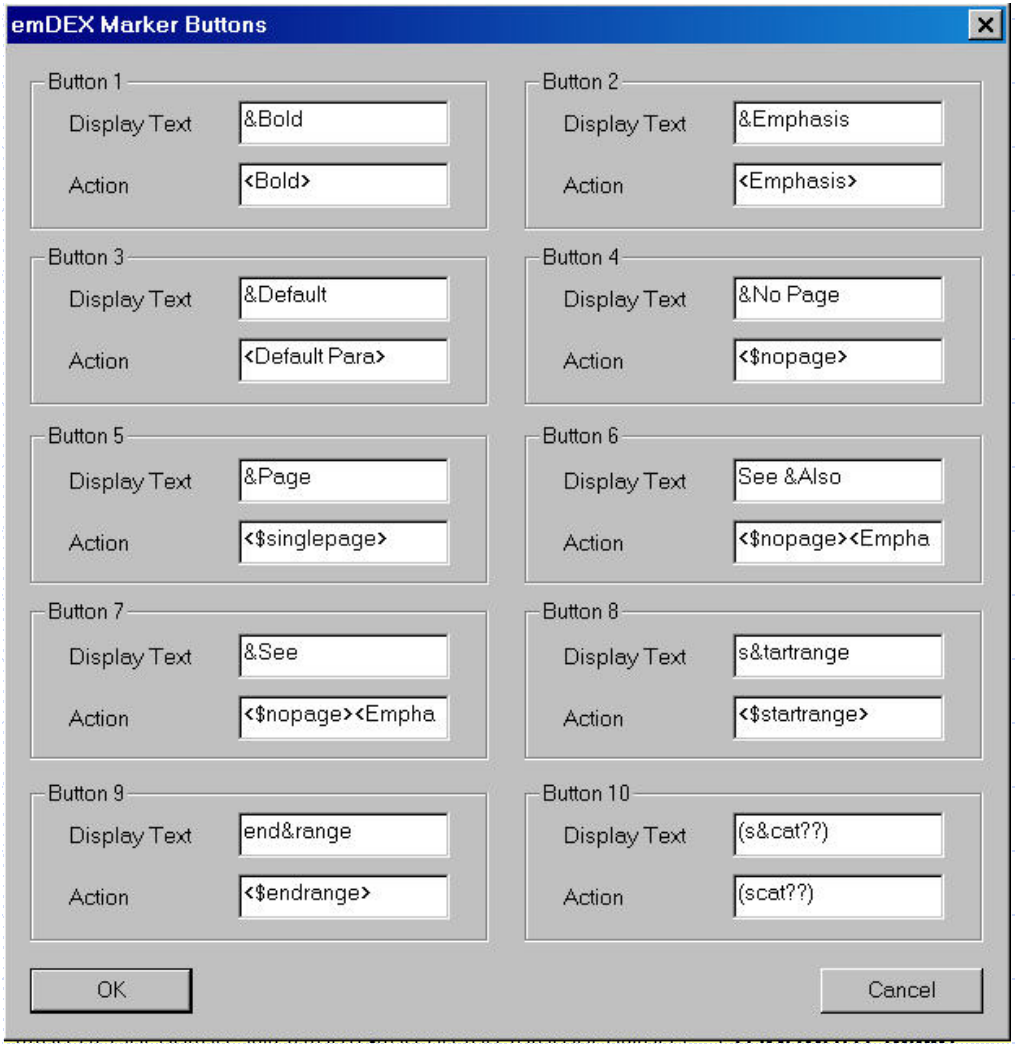


Benefits/functionality of third party utilities – autocomplete for entries (emDEX)



Benefits/functionality of third party utilities
– customizable marker box and special strings (emDEX)





Benefits/functionality of third party utilities – autogeneration of entries (IXgen)

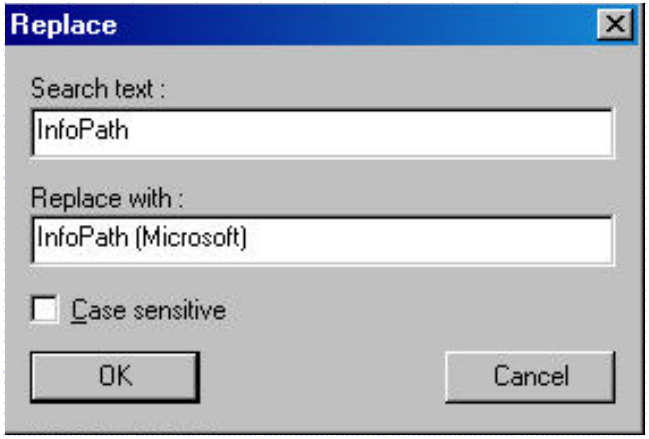
- from Keywords
- from Paragraph Tags
- from Character Tags
- permute (rotate) marker text

quick brown fox can be rotated to these two additional entries:

brown, fox quick

fox, quick brown

Benefits/functionality of third party utilities – change propagation (emDEX)



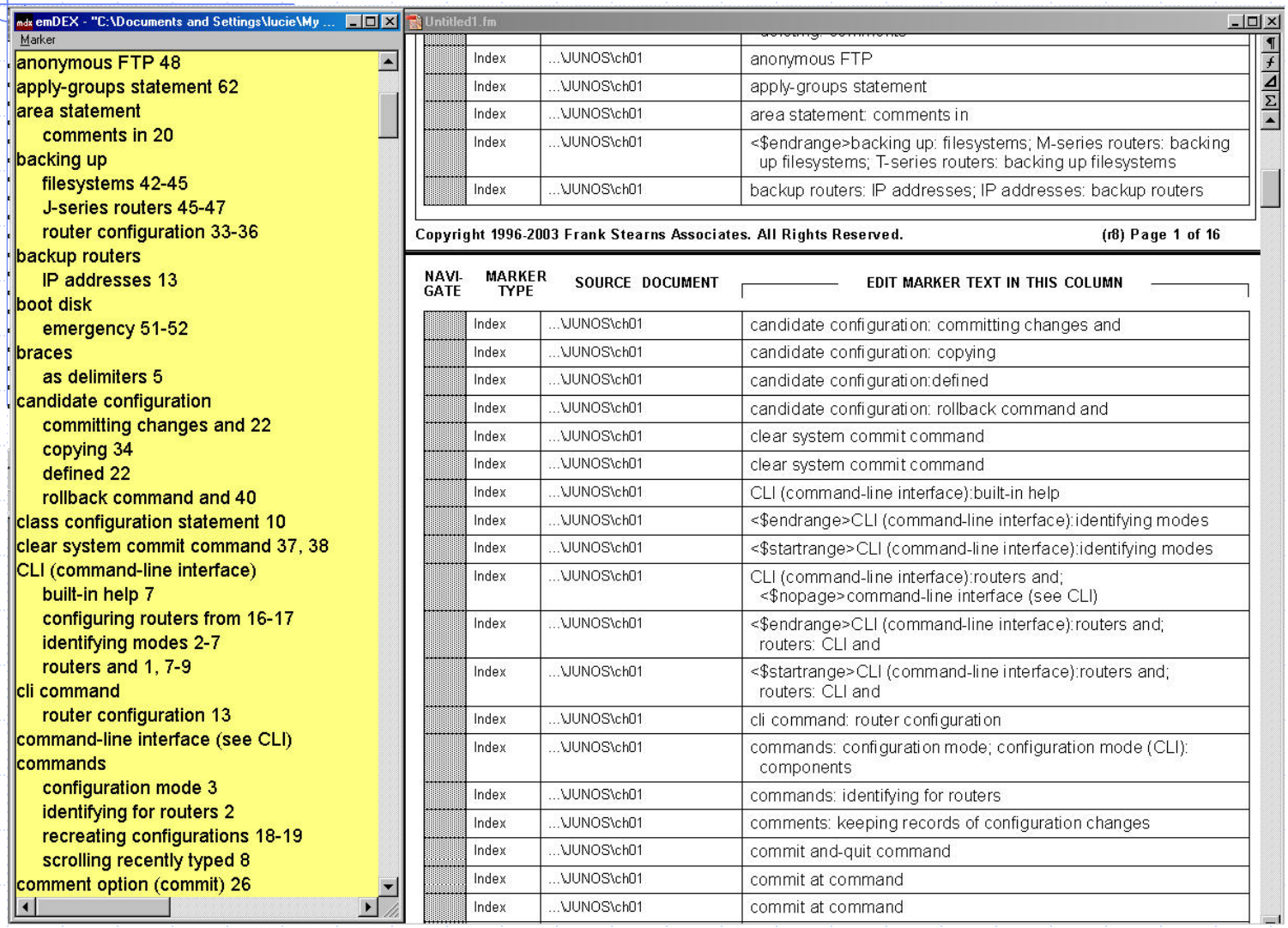
OLD VERSION:

InfoPath
defined 215
Excel and 225-227
generating HTML output 261-263
linking forms 230-236, 245-247
populating controls 236-239
scripts and 243-245, 247-255
sharing data 227-230

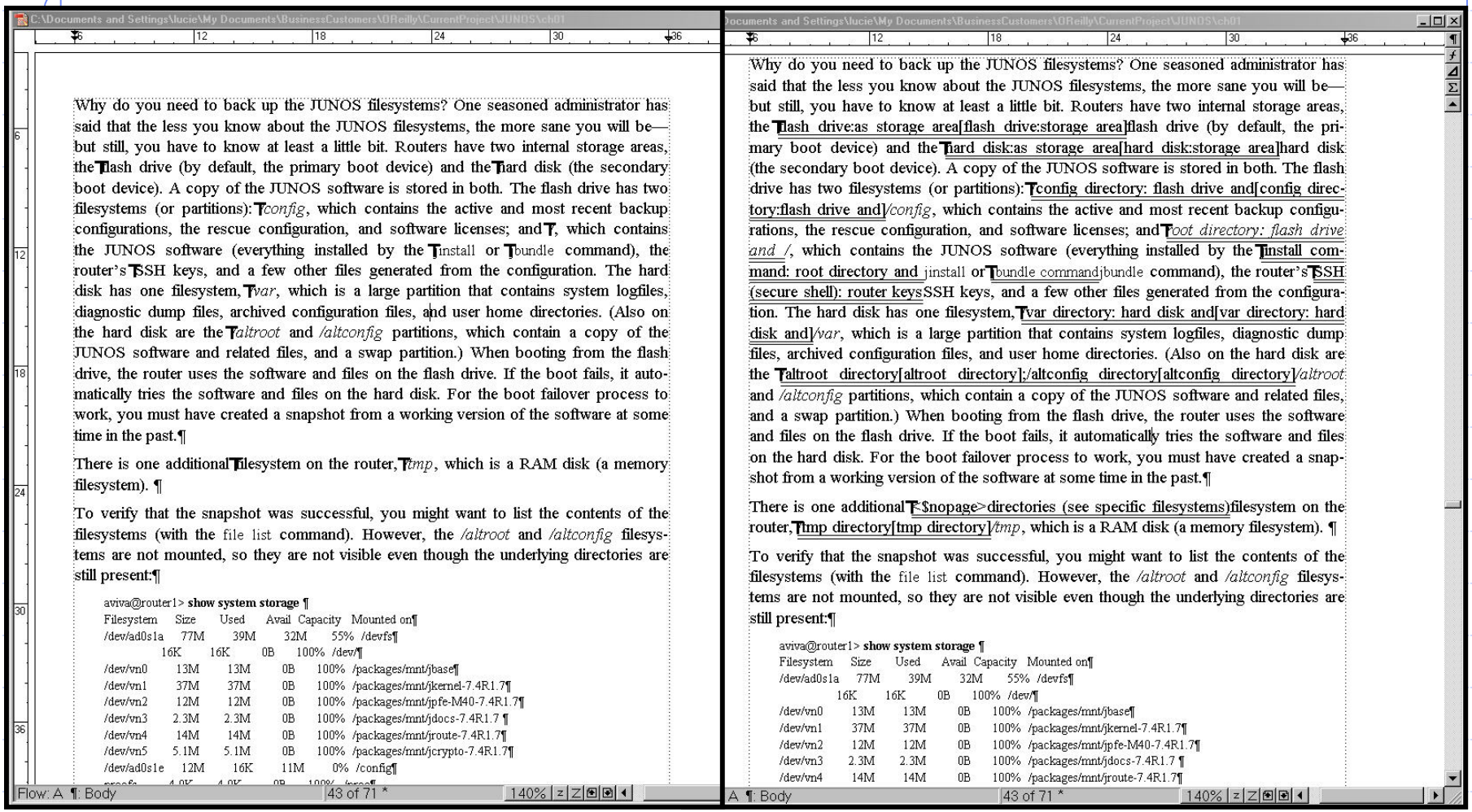
NEW VERSION:

InfoPath **(Microsoft)**
defined 215
Excel and 225-227
generating HTML output 261-263
linking forms 230-236, 245-247
populating controls 236-239
scripts and 243-245, 247-255
sharing data 227-230

Benefits of third party utilities for FrameMaker – change propagation (IXgen)



Benefits/functionality of third party utilities – viewing index entries in document (IXgen)



C:\Documents and Settings\lucie\My Documents\BusinessCustomers\OReilly\CurrentProject\JUNOS\ch01

Why do you need to back up the JUNOS filesystems? One seasoned administrator has said that the less you know about the JUNOS filesystems, the more sane you will be—but still, you have to know at least a little bit. Routers have two internal storage areas, the flash drive (by default, the primary boot device) and the hard disk (the secondary boot device). A copy of the JUNOS software is stored in both. The flash drive has two filesystems (or partitions): */config*, which contains the active and most recent backup configurations, the rescue configuration, and software licenses; and /, which contains the JUNOS software (everything installed by the install or bundle command), the router's SSH keys, and a few other files generated from the configuration. The hard disk has one filesystem, */var*, which is a large partition that contains system logfiles, diagnostic dump files, archived configuration files, and user home directories. (Also on the hard disk are the */altroot* and */altconfig* partitions, which contain a copy of the JUNOS software and related files, and a swap partition.) When booting from the flash drive, the router uses the software and files on the flash drive. If the boot fails, it automatically tries the software and files on the hard disk. For the boot failover process to work, you must have created a snapshot from a working version of the software at some time in the past.¶

There is one additional filesystem on the router, */tmp*, which is a RAM disk (a memory filesystem). ¶

To verify that the snapshot was successful, you might want to list the contents of the filesystems (with the file list command). However, the */altroot* and */altconfig* filesystems are not mounted, so they are not visible even though the underlying directories are still present:¶

```
aviva@router1> show system storage ¶
Filesystem     Size       Used      Avail  Capacity   Mounted on¶
/dev/ad0s1a     77M       39M       32M       55%  /devfs¶
                16K       16K        0B      100%  /dev/¶
/dev/vn0        13M       13M        0B      100%  /packages/mnt/jbase¶
/dev/vn1        37M       37M        0B      100%  /packages/mnt/jkernel-7.4R1.7¶
/dev/vn2        12M       12M        0B      100%  /packages/mnt/jpfe-M40-7.4R1.7¶
/dev/vn3       2.3M      2.3M        0B      100%  /packages/mnt/jdocs-7.4R1.7 ¶
/dev/vn4        14M       14M        0B      100%  /packages/mnt/jroute-7.4R1.7¶
/dev/vn5       5.1M      5.1M        0B      100%  /packages/mnt/jcrypto-7.4R1.7¶
/dev/ad0s1e     12M       16K       11M        0%  /config¶
```

Flow: A ¶: Body 43 of 71 * 140%

Documents and Settings\lucie\My Documents\BusinessCustomers\OReilly\CurrentProject\JUNOS\ch01

Why do you need to back up the JUNOS filesystems? One seasoned administrator has said that the less you know about the JUNOS filesystems, the more sane you will be—but still, you have to know at least a little bit. Routers have two internal storage areas, the flash drive:as storage area[flash drive:storage area]flash drive (by default, the primary boot device) and the hard disk:as storage area[hard disk:storage area]hard disk (the secondary boot device). A copy of the JUNOS software is stored in both. The flash drive has two filesystems (or partitions): config directory: flash drive and[config directory:flash drive and]*/config*, which contains the active and most recent backup configurations, the rescue configuration, and software licenses; and *root directory: flash drive and /*, which contains the JUNOS software (everything installed by the install command: root directory and jinstall or bundle commandjbundle command), the router's SSH (secure shell): router keysSSH keys, and a few other files generated from the configuration. The hard disk has one filesystem, var directory: hard disk and[var directory: hard disk and]*/var*, which is a large partition that contains system logfiles, diagnostic dump files, archived configuration files, and user home directories. (Also on the hard disk are the altroot directory[altroot directory];/altconfig directory[altconfig directory]*/altroot* and */altconfig* partitions, which contain a copy of the JUNOS software and related files, and a swap partition.) When booting from the flash drive, the router uses the software and files on the flash drive. If the boot fails, it automatically tries the software and files on the hard disk. For the boot failover process to work, you must have created a snapshot from a working version of the software at some time in the past.¶

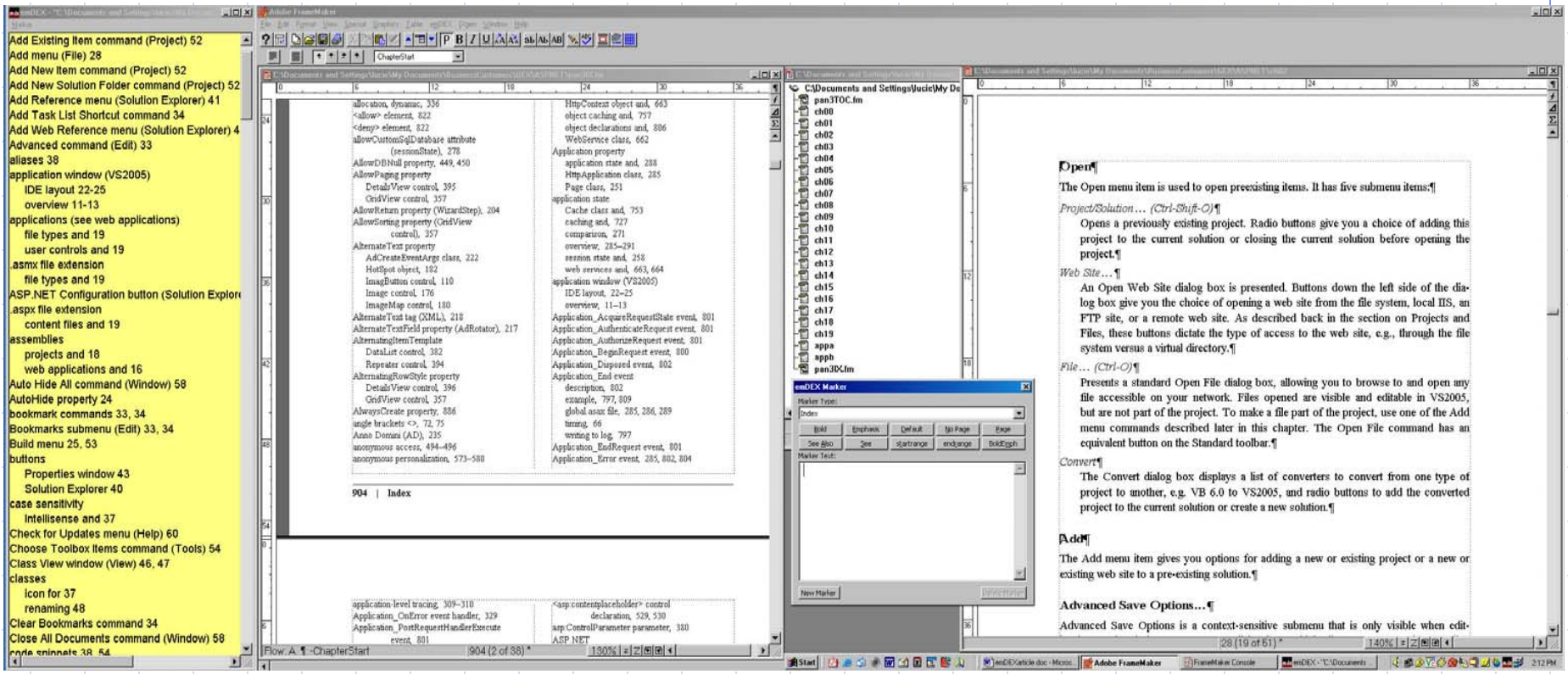
There is one additional <$nopage>directories (see specific filesystems)filesystem on the router, tmp directory[tmp directory]*/tmp*, which is a RAM disk (a memory filesystem). ¶

To verify that the snapshot was successful, you might want to list the contents of the filesystems (with the file list command). However, the */altroot* and */altconfig* filesystems are not mounted, so they are not visible even though the underlying directories are still present:¶

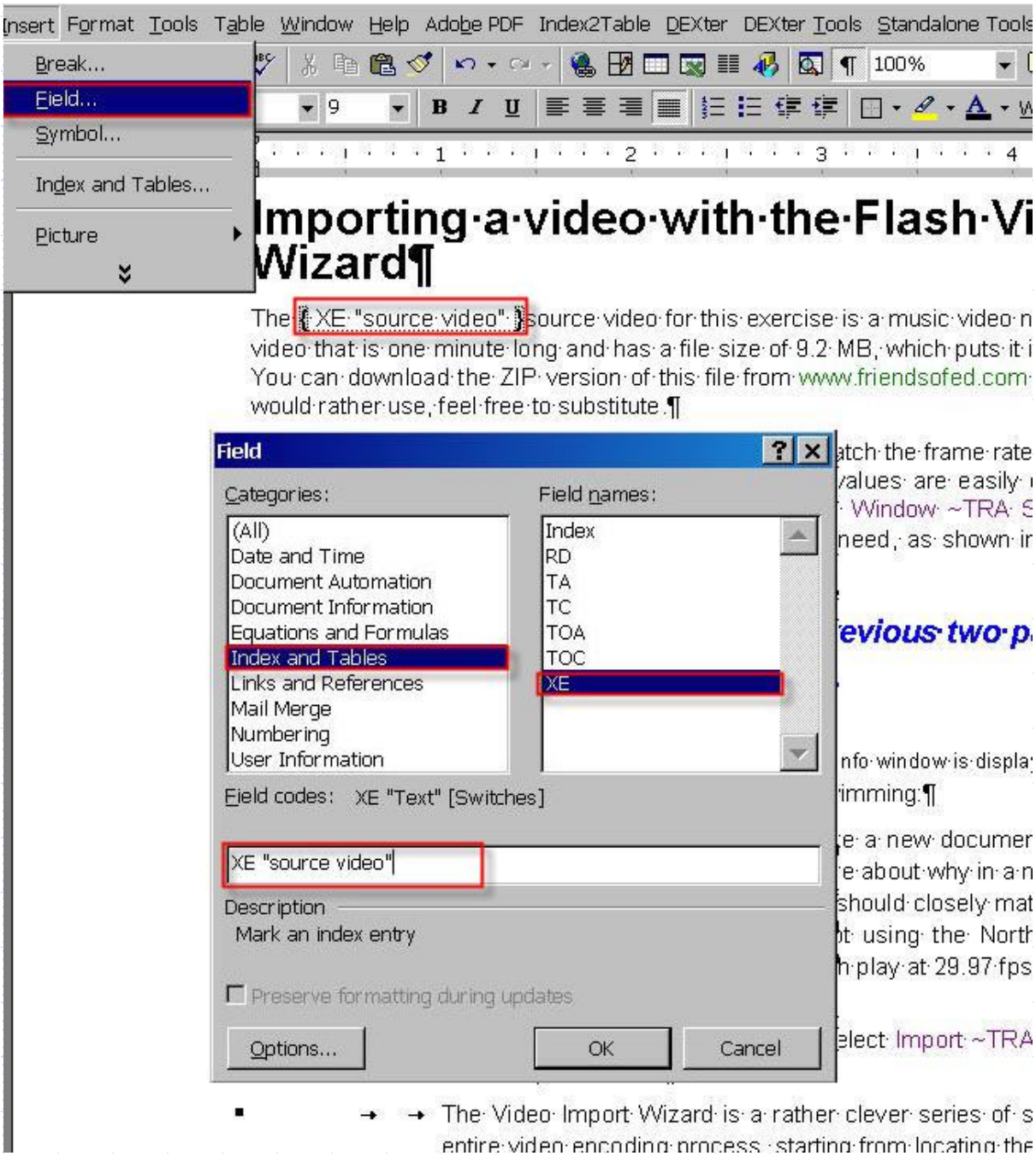
```
aviva@router1> show system storage ¶
Filesystem     Size       Used      Avail  Capacity   Mounted on¶
/dev/ad0s1a     77M       39M       32M       55%  /devfs¶
                16K       16K        0B      100%  /dev/¶
/dev/vn0        13M       13M        0B      100%  /packages/mnt/jbase¶
/dev/vn1        37M       37M        0B      100%  /packages/mnt/jkernel-7.4R1.7¶
/dev/vn2        12M       12M        0B      100%  /packages/mnt/jpfe-M40-7.4R1.7¶
/dev/vn3       2.3M      2.3M        0B      100%  /packages/mnt/jdocs-7.4R1.7 ¶
/dev/vn4        14M       14M        0B      100%  /packages/mnt/jroute-7.4R1.7¶
```

A ¶: Body 43 of 71 * 140%

My embedding process using FrameMaker and utilities



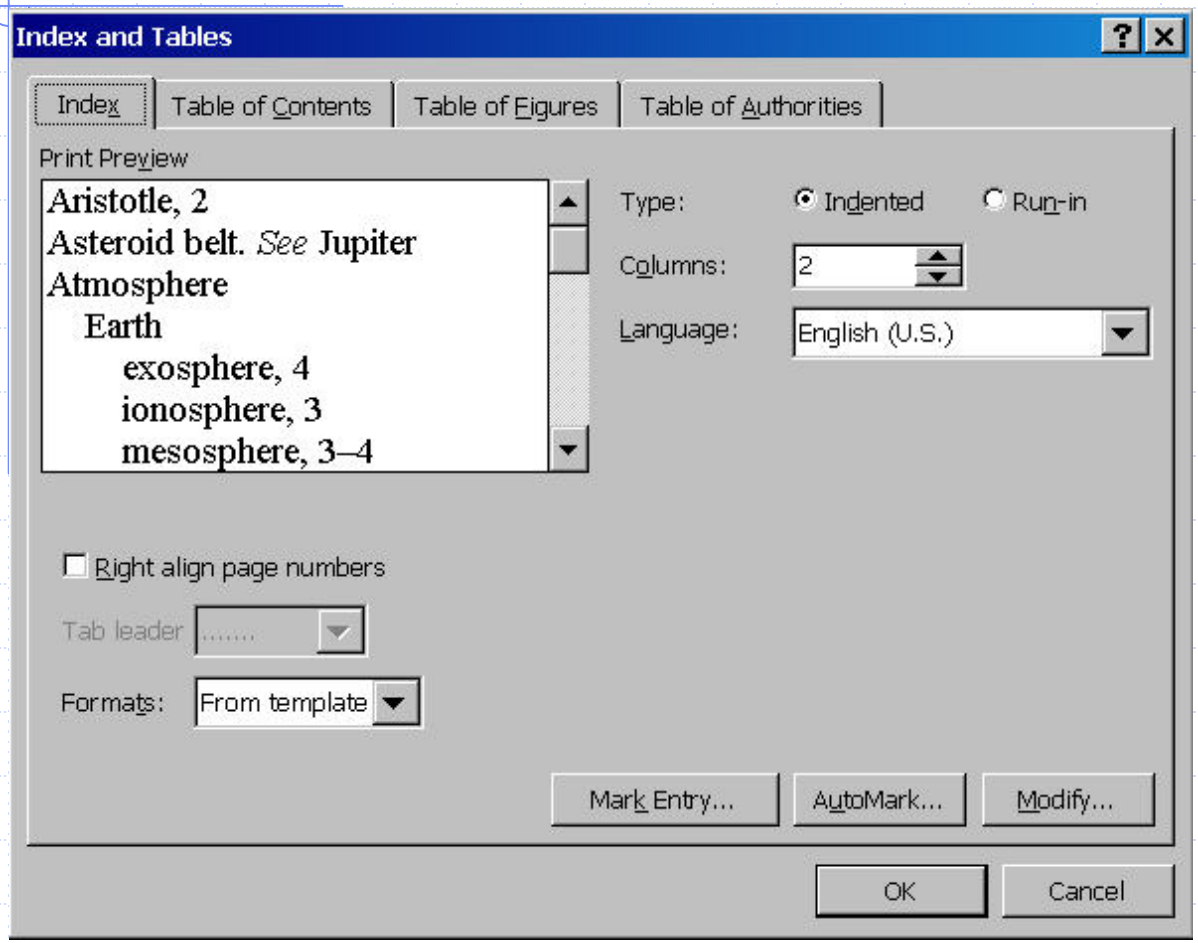
- two-screen layout
- use index as static (previous chapters) guide
- emDEX window as dynamic (current chapter) guide
- edit by hyperlinking from generated index



MS Word Process:

Create XE markers

Generate index in MS Word



Third party utilities for Microsoft Word

DEXter
http://www.editorium.com/DEXter.htm

WordEmbed
http://www.wordembed.jalamb.com

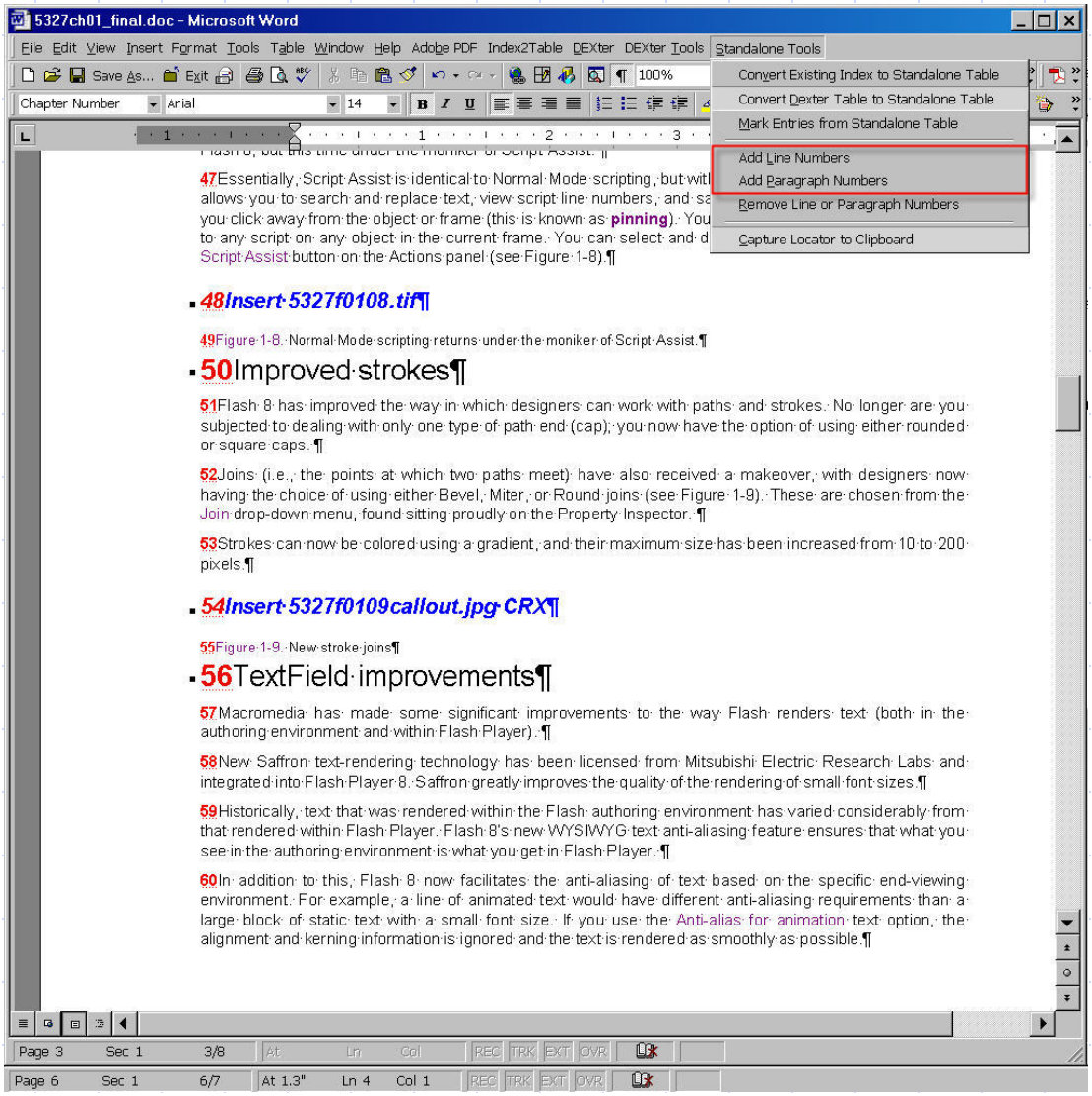
Embedding process in Microsoft Word with utilities

No need to index in Microsoft Word!

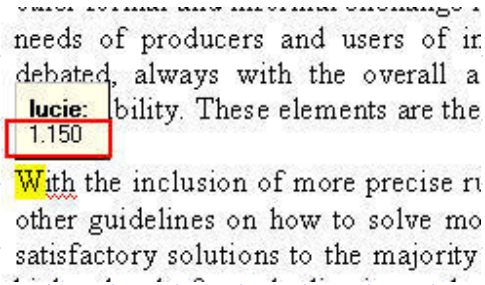
- index in proprietary software using:
  - line/paragraph/page locators (DEXter)
  - bookmarks as locators (WordEmbed)
- save index file as:
  - DAT file (DEXter)
  - RTF or MBK file (WordEmbed)
- import index file into Microsoft Word

Using locators in indexing program

- DEXter



- WordEmbed



Saving index for import into Microsoft Word – DEXter DAT file

```
acceptsURL·method·(DriverManager) → opening·database·connections → 166¶
autocommit·mode → connections·and → 441¶
BatchUpdateException·class → functionality → 393¶
CallableStatement·interface → functionality → 103¶
CallableStatement·objects → creating → 94¶
CallableStatement·objects → creating → 190¶
CallableStatement·objects → creating → 203¶
CallableStatement·objects → creating → 209¶
CallableStatement·objects → ResultSet·objects·and → 105¶
case·sensitivity → Java·and → 19¶
checked·exceptions → defined → 370¶
classes → See·also·utility·classes¶
classes → cores → 82¶
classes → importing → 134¶
classes → initializing → 143¶
CLASSPATH·environment·variable → registering·drivers → 149¶
```

Saving index for import into Microsoft Word – WordEmbed formats

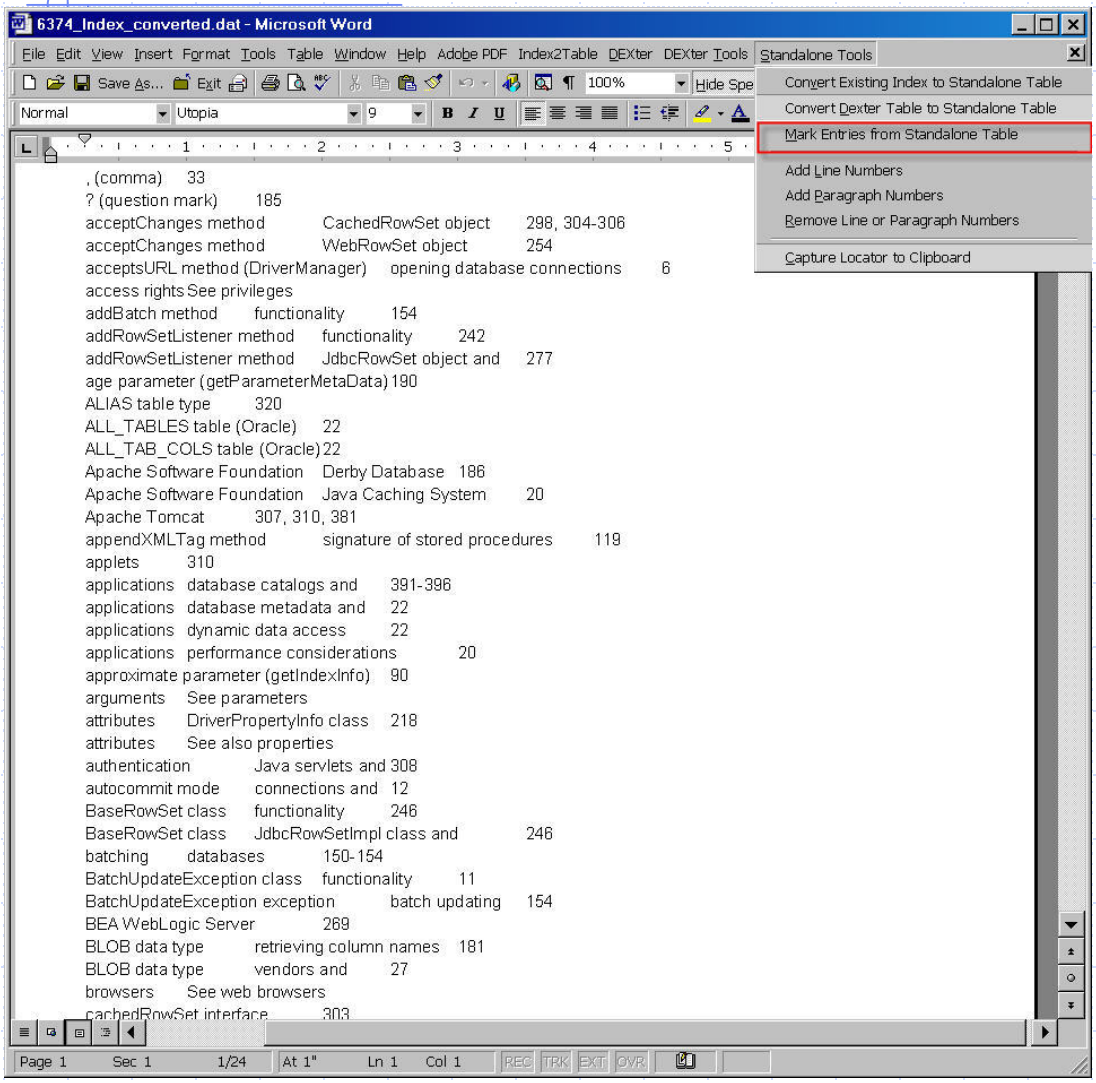
Putnam
    Putnams's Italy, 2.160-2.30
        Naples, 2.30
Scandanavians. See Norwegians
skills: uses of, 2.210

CMS (RTF file)

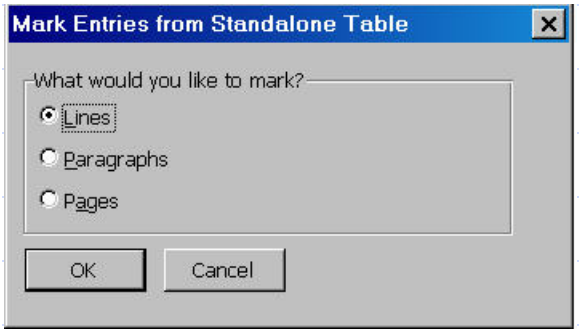
Putnam:Putnams's Italy, 2.160-2.30
Putname:Putnam's Italy:Naples, 2.30
Scandanavians ^see^ Norwegians
skills: uses of, 2.210

MBK (Macrex) style format

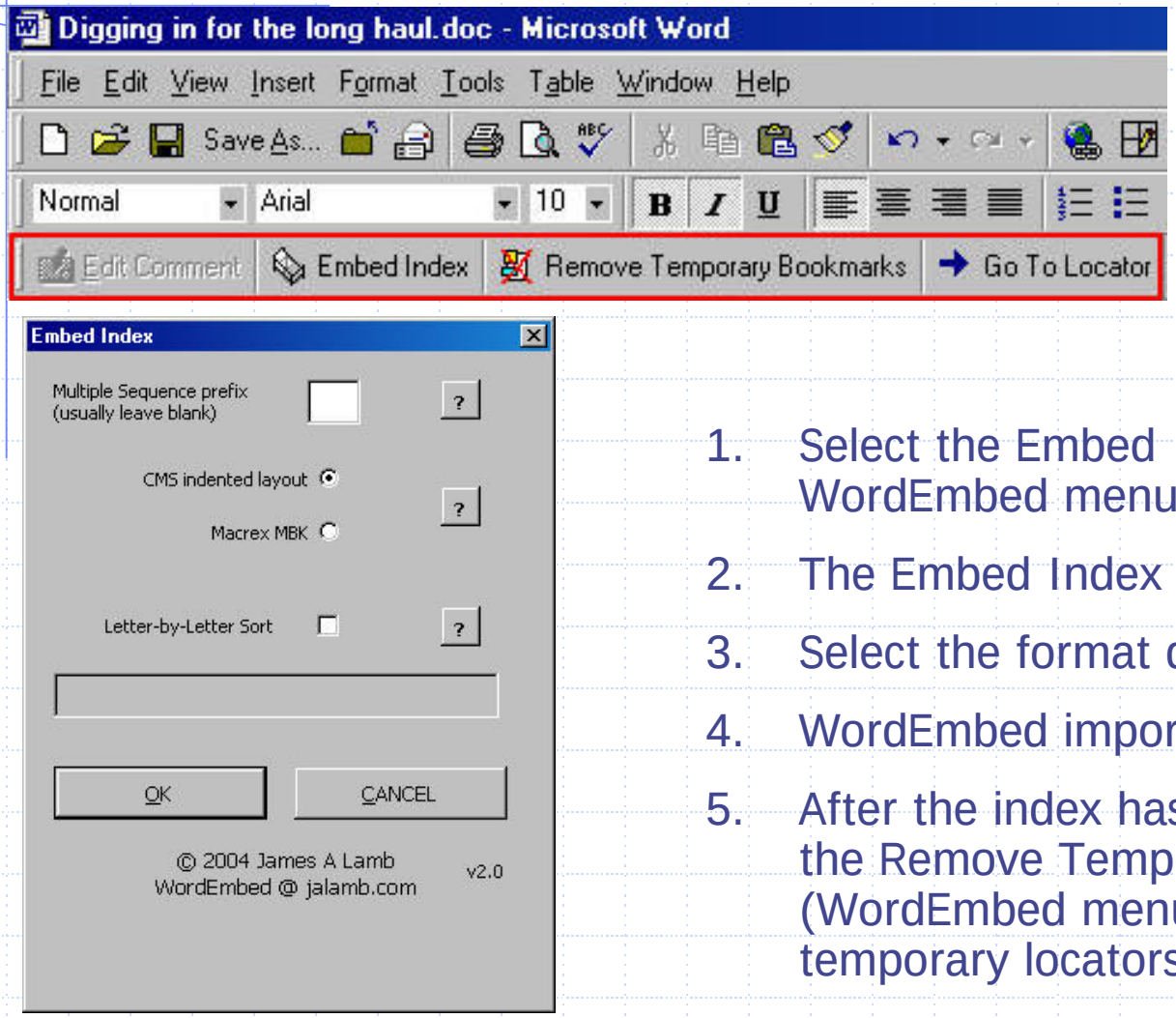
Importing index into Microsoft Word - DEXter



1. Open Word document.
2. Open DAT file.
3. Select Mark Entries from Standalone Table.
4. Select Lines, Paragraphs, Pages from Mark Entries dialog box.
5. Import process completes.



Importing index into Microsoft Word
- WordEmbed



1. Select the Embed Index button from the WordEmbed menu bar.
2. The Embed Index dialog box displays.
3. Select the format desired.
4. WordEmbed import process completes.
5. After the index has been embedded, press the Remove Temporary Bookmarks button (WordEmbed menu bar) to remove the temporary locators created earlier.