

Embedded Document Notes

White Paper

on

Managing Document Versions

Table of Contents

Purpose	4
Scenarios	4
1 – Customer Calls Tech Support.....	4
2 – Product Rollback	5
3 – Errors, Omissions, Bug Fixes, and Workarounds.....	5
4 – Document Rollback	5
5 – Many Copies.....	5
6 – Different Documents With Common Information	6
7 – More Than One Writer	6
8 – Parallel Review of Work-in-Progress	6
9 – Differences Between Single-Sourced Outputs	6
10 – Maintaining Single-Sourced Documents.....	7
11 – Traceability	7
12 – Time and Cost Overruns	7
13 – Parallel Hardware and Software Development.....	7
Writer Circumvention	7
Existing Solutions.....	8
Enterprise Document Management Software (EDMS)	8
Advantages.....	8
Disadvantages.....	8
Version Control Software	9
Advantages.....	9
Disadvantages.....	9
Document Numbering System	10
Advantages.....	10
Disadvantages.....	10
Separate Changes Document.....	10
Advantages.....	10
Disadvantages.....	10
Disciplined Save and Record.....	11
Advantages.....	11
Disadvantages.....	11

Overview of the Document Notes Solution	12
What Are Document Notes?	12
Why Embed?	12
Possible Techniques.....	13
A. Conditional Text	13
B. Markers (Comment or Generic).....	13
C. Markers (Hypertext)	13
Overcoming Marker Limitations and Automation.....	14
255 characters	14
Conditional text	14
Time consuming selection	14
Complex prefixes/suffixes	14
Sample Implementations.....	15
Acrobat Notes	15
Comment Marker and IOM	15
Conclusion	16
Review	16
Keys to Creating a Solution	16
Followup.....	16
How To FAQ	17
How Do I Add a Document Note?	17
What Should I Write In a Document Note?	17
How Do I Generate a Report?	18
Do I Need To Configure TimeSavers?	18
Do I Need To Write a FrameScript?.....	19
Comment Document Note Script.....	19
Acrobat Document Note Script	20
Load Script	20
Can I share my source files with other writers?	20

Purpose

We need to keep track of all of the versions of our documents and quickly identify the following:

- Up-to-date or out-of-date
- Changes made since the previous version
- Which product and version
- Location of the source file
- Rollback information

Scenarios

1 – Customer Calls Tech Support

A customer calls Technical Support. This happens all the time. In some cases the customer did not bother to read the manual or check online help, in other cases, the customer tried but it didn't help.

In order for Tech Support personnel to quickly solve problem, they need prepared solutions to most problems. If the answer is in a manual or help file that the customer didn't read, he can just email it or read it to the customer.

What happens when the customer already checked the documentation? Should your first line of Tech Support try and troubleshoot, walking the customer through a long battery of tests? Should they escalate the problem, tying up valuable resources and leaving the customer without an immediate solution? In many cases, with good document management, neither of these alternatives is necessary and you can quickly provide a prepared solution. Moreover, when you do need to troubleshoot or escalate, you are in a better position to update the documentation for the next time this problem occurs.

In many cases, the customer's documentation is not the most recent version and changes have since been made that solve the problem. However, knowing that a document is out of date is not enough, you also need to know whether the relevant information was changed or added. Otherwise, you run the risk of increasing your customer's frustration and channeling it from your product to your company as a whole.

You need to quickly identify if the information exists in the documentation and is not found in the version that is in the customer's hands. Sending the correct documentation may not solve every problem, but it will solve many. If the correct documentation is not clear, is incorrect, or does not cover a previously unknown problem, then this information along with the results of whatever troubleshooting and/or escalation needs to be communicated to the technical writer(s) responsible in order to improve future documentation. Likewise, if the more detailed documentation used by Technical Support is lacking or incorrect, this also needs to be communicated to the technical writer.

2 – Product Rollback

When this happens the obvious solution is to return to the documentation for the previous version. But it is not always so simple.

What happens when improvements were made to the documentation that are relevant to the previous version as well? Do you just throw those changes out the window? What if they were corrections to erroneous information? This shouldn't happen, but it does.

What happens when the rollback is to an earlier build that was never released, and therefore does not have its own version of the documentation? Do you rollback to the previous released documentation and start to rewrite from there? Do you look for the most relevant backup or intermediate version of your document? Once you choose your starting point, how do you proceed at a time when all of the SMEs are rushing to meet a deadline that by definition is yesterday if not earlier.

Unless you already have a version of the documentation ready for the rolled back product that contains all improvements that relevant, You will need to compare documents, which is a tedious process that often requires you to consult with SMEs at the worst possible time.

The way that you manage your document versions greatly affects how quickly you can rollback, and at what cost to other company resources. Furthermore, if you can respond quickly with accurate documentation, you actually help your colleagues through their hour of need.

3 – Errors, Omissions, Bug Fixes, and Workarounds

These happen all the time. Some may be your fault and many more may not. In any case, you need to find the source file and change the document.

4 – Document Rollback

After taking the initiative and making improvements to the documentation, you get complaints that some of the improvements are a source of complaints or increased the number of calls to Tech Support. How do you rollback and still retain the changes that were useful. Being able to quickly identify each change rather than comparing documents and analyzing each change, can save you a lot of time.

5 – Many Copies

Whenever you have more than one source files you run the risk of losing some of the corrections. Worse yet, each version is likely to be partially out of date. Tracking down these errors can be extremely time-consuming and doesn't always bring 100% results.

You therefore need to make sure that all edits are made to a single master file. When circumstances dictate that different people work on the files at the same time, all changes made to copies need to be reviewed and integrated into the master document. It is important that the master be identified as such.

6 – Different Documents With Common Information

When you update one document, you need to make sure to update all other documents containing the same information.

One solution to this problem is single-sourcing. However, single-sourcing is not always used; in some cases the drawbacks and added complexity of single-sourcing does not justify its use.

Another solution is to keep track of which documents share information and a record of what edits are made.

7 – More Than One Writer

When more than one person works on a document, you must ensure that everyone works on the same source file **AND** enable each writer to know what changes have already been made? Otherwise the Dr. Seuss like scenario could really happen.

Just because Ted Techwriter made a change based on information from Paula Programmer, doesn't mean that Courtney Coworker is aware of the change or the reason why. Courtney may very well be getting conflicting information from Quinton Q. Assurance. As a result, Courtney Coworker quickly corrected the text to meet Quinton's quabbles. Meanwhile, Mark Eting, who read Ted's draft is off touting the wonders of Paula's programmed features without the slightest clue to Quinton's quabbles or Courtney's corrections.

8 – Parallel Review of Work-in-Progress

There isn't always enough time to wait for everyone's feedback before continuing to write or edit a document. When this is the case, there needs to be a way to identify where feedback from a previous version fits into the latest version and whether the issue has already been dealt with.

This issue is even more critical when more than one writer is involved, because the relevant information may have moved to a chapter handled by another writer or may affect other chapters. It is important that the team leader be able to spot changes that affect other writers in the team.

9 – Differences Between Single-Sourced Outputs

Single-sourced outputs often are quite similar and the differences are not always apparent to customers, resellers, or Tech Support personnel.

If we return to scenario 1, we could easily find a situation where the customer has the documentation for the wrong product. The customer may very well have purchased or licensed two similar products and referred to the wrong document, or maybe downloaded the document for the wrong version. Identifying that customer has the wrong document **AND** that the difference is relevant is important to providing fast, efficient and inexpensive Technical Support.

10 – Maintaining Single-Sourced Documents

Changes that affect some versions of a single-sourced document, but not all, can quickly lead to complications. The types of problems that can occur depend largely on the single-source techniques used.

Whatever techniques are used, single-sourcing increases the need for effective document management.

11 – Traceability

When a change is made, it is important to know who was responsible for the change, even if the change is correct. This information can save valuable time when problems occur or more information is needed to solve a customer's problem. The time saved in finding a solution to a previously undocumented problem can give your company the time it needs to initiate a solution before other customers become aware of the problem.

Often there is a good reason for the change, but not everyone is aware of it. Knowing who made the change can help clear up misunderstanding regarding the product itself. In cases where there is a question of policy and not accuracy, this helps get the right people together to make a clear and binding decision.

12 – Time and Cost Overruns

Accurately estimating the time and resources needed to document products under development is a difficult if not impossible task. As a result, it is good practice to review and evaluate the process once the project is complete. In some cases, you need to justify overruns and explain how the scope of the work changed.

A clear record or audit trail of changes helps explain what really happened.

13 – Parallel Hardware and Software Development

Clear and concise version change documents can keep the SW engineers updated by drawing their attention to HW developments. This can prevent release of new documentation which is immediately obsolete. Shlomit Laqueur of Excalibur Systems has found that changes made in hardware registers and firmware affect software functions and that “software engineers are oblivious of these changes.” Single-sourcing doesn't solve the problem because HW and SW documentation are different. “

Writer Circumvention

You are in a rush, with no time to follow the standard procedure, so you circumvent the system. If you go back the next day and repeat following your standard procedure, there's no harm done. However that rarely happens. When you return to the source file or try to access the information on that version, the changes won't appear.

To be effective, whatever system you use to manage your document versions MUST be easy enough to use that it won't be circumvented.

Existing Solutions

There are a number of solutions or partial solutions commonly in use, each with their own advantages and disadvantages.

Enterprise Document Management Software (EDMS)

This represents the top of the line solution and can cost hundreds of thousands of dollars. Before even considering such an expensive solution, you need to state your document management objectives, then put these objectives into step-by-step procedures that you believe people in your organization will actually be willing to follow without circumventing the system.

Keith Mahoney of Bulls-Eye Direct Publishing, describes why he liked using an EDMS from Documentum to deal with rollbacks:

1. You can track the iterations back to the beginning.
2. Insert the date variable in your footer info/front matter, etc.
3. Insert a part number/doc tracking number.
4. Provide your customer with explicit directions for submitting change, comments, suggestions, etc.
5. Keep a database specifically for customer interactions, and make sure that the folders contain the name of the customer, the date a request was submitted, the nature of the request, who addressed the request, who made the change, and the resolution.”

Advantages

- **Comprehensive.** Information is updated and distributed instantaneously. Access to information is both controlled and easily accessible to those authorized. An audit trail of changes is automatic and fully manageable. Full rollback capabilities.

Disadvantages

- **Price.** These systems can run into the tens or even hundreds of thousands of dollars.
- **Complexity.** These systems often require professional dedicated to maintaining and managing the system.
- **Cost of Circumvention.** The same powerful features that save time and resources when these systems are used magnify the potential harm of circumventing the system. Old and out-dated versions become accessible to so many more people and can easily be integrated into so many more documents than if these systems were not in place.

Version Control Software

Version control software, such as SourceSafe, is common in many organizations. These programs provide library services, such as:

- Check out / Check in
- Lock / Unlock
- Version histories with access to previous versions
- Accessing document properties
- Making documents show up in multiple locations / listings

Tagging the version of the file in a version control system, such as SourceSafe, identifies the source of the document released with the product. This allows for a control document if you need to compare documents.

Despite the fact that in theory, version control software can compare text files, this does not work for most technical documents even if they are stored as MIF files. This is because even simple changes to a FrameMaker file often results in changes throughout a MIF file.

Furthermore, when VSS stores binary formats (Word, FrameMaker, etc.) it stores the entire file, not just the delta. This takes up a lot of disk space.

Therefore you cannot use it to identify changes made to the documentation between product releases or changes not directly related to product changes.

Advantages

- **Availability.** These systems already exist in most R&D environments.
- **Compatible with Product Development.** Versions of the document can be stored with the corresponding versions of the product.

Disadvantages

- **Comparison of Formatted Documents.** These systems are not usually well-suited to comparing versions of formatted text. Even when stored in native text formats that can be recompiled into the fully formatted versions, the formatting information creates enough “noise” to make comparison difficult at best. This is especially true of FrameMaker documents stored as MIF files, where even the most minor change in FrameMaker can result in change throughout the entire MIF file.
- **Difficult to Account for Multi-Product Single-Source Documents.** In order to retain the benefits of single-sourcing you need to maintain a single version of the source document. However, storing separate versions of the source document with each product is likely to undermine this.
- **Disk Space.** In most cases you will want to store files in binary formats, for which version control cannot identify and store deltas.

Document Numbering System

This can help you identify documents and can be used in conjunction with other solutions.

Chris Gooch of Light Work Design recommends tagging files stored in version control software with a document number that consists of a doc-type code, date code, and serial number.

Advantages

- **No Additional Software.**
- **Provides Tracking Information.**

Disadvantages

- **Limited Information.** This does not provide information on what has changed or the differences between versions. In order to be useful, the document numbers need to be indexed to information on the documents.

Separate Changes Document

Creating a separate document describing each change made, when and why. This document should be updated concurrently with the changes. Another variation is to compare documents after each release and prepare a detailed list of changes.

Advantages

- **No Additional Software.**
- **Informative.**

Disadvantages

- **Time-Consuming.** Updating a changes document concurrently greatly increases the time it takes to implement each edit to a document. Comparing documents is in itself a time-consuming process that is often difficult to justify when there is no immediate demand. However, waiting until there is a need is far more time-consuming and less effective than when the information is fresh in your mind.
- **Discipline Required.** Whether changes are made concurrently or immediately after the version is released, it requires a lot of discipline to maintain this process.

Disciplined Save and Record

Geoff Hart, of the Forest Engineering Research Institute of Canada, creates a separate document consisting only of the changes that he makes. This is what he sends to SMEs for review and only incorporates the changes into the actual document after the changes are approved.

Geoff, who works in Word 97, uses Word's versioning feature, which retains a copy of all previous versions (defined based on the state of the document at the time the file was closed) in a single file. Before making changes to a document, he saves a "checkpoint" copy of the file to enable him to rollback if his live document becomes corrupted. He then opens the Versions dialog, deletes old versions and saves the file before continuing. Whenever there is a major change to the document, he ups the release number of document and saves a copy of the document with its associated "changes" document.

Advantages

- **No Additional Software.**
- **Full Rollback Capabilities.**
- **Informative.**

Disadvantages

- **Time-Consuming.** This process is extremely time-consuming.
- **Disk Space.** The amount of disk space needed to store the intermediate versions is far greater than storage of the delta or separate change documents.

Discipline Required. While this system works well for Geoff, it requires more discipline than most technical writers capable of over time.

Overview of the Document Notes Solution

What Are Document Notes?

Document Notes are notes about changes made to a document. For example, an edit resulting from a change in the product, a change in the installation/operation procedure, a change in the name of a Unix command. Whenever possible, the document note should be dated. If implemented properly, you can use standard FrameMaker features to easily create reports that are more useful and easier to manage than numbering of document versions.

Why Embed?

Embedding answers most if not all of the needs described in the various scenarios. It ensures that the information stays with the document each time a new version is generated. Once a document note is embedded it is always available in every book using that file. If the note is embedded in a text inset, it is available in all documents and books that include that text inset.

At first glance, this appears to be an inconvenient and time-consuming process. However, it is possible to automate the process to the point where embedding is both simple and intuitive.

Possible Techniques

A. Conditional Text

FrameMaker adds the **Comment** condition by default. All you need to do is **Show** or **Hide** the text tagged. When conditional text is not used for anything else, this is the simplest solution to implement.

However, it quickly becomes unwieldy when conditional text is used for other purposes. This is especially true when conditional text is used to single-source different documents because you need to add an additional **Comment** condition for each single source condition.

It should be possible to adapt the Track Changes plugin, which cost \$100 per seat, to simplify the process. For more information, go to <http://www.intech.com/providers/adobe.htm>.

B. Markers (Comment or Generic)

This is a more flexible option. It is compatible with single-sourcing techniques, especially when based on text insets.

- B1. It is normally less convenient for most writers to select a marker other than Index. The notes must appear only in the relevant document.
- B2. Markers are limited to 255 characters.
- B3. In order to sort by date, you need to prefix the marker with the marker in yyyyymmdd format.
- B4. Condition tags cannot be placed in markers.

C. Markers (Hypertext)

This option allows you create Acrobat Notes, which can be summarized in the Comments or Annotations tab. This method is the most complex of when all using standard FrameMaker documentation techniques.

- C1. In addition to the difficulties listed above, this technique requires a more complex prefix and a complex suffix as well.
- C2. In order to make proper use of Acrobat's sorting and reporting, you need to prefix with the date down to the second in yyyyymmddhhMMss format.
- C3. Only a very limited number of characters can be displayed in the Comments/Annotations tab.

Overcoming Marker Limitations and Automation

255 characters

This should not be a problem because document notes should be short. TimeSavers provides a mechanism for overcoming this limitation by allowing you to reference an external text file containing additional text, however the workaround is more complex.

Conditional text

Conditions can be placed on the entire marker. If necessary separate markers can be used when the text differs for each condition.

Time consuming selection

Edit FrameMaker configuration files. The easiest way is to use MicroType's free Express Customization Kit and edit the CustomUI.cfg file.

FrameScript is the best solution, but costs \$150 per seat. There is a steep learning curve and it must be used with care.

Complex prefixes/suffixes

FrameScript is the best solution, but costs \$150 per seat. There is a steep learning curve and it must be used with care.

TimeSavers, which costs \$200, can provide partial solutions, especially if you edit the TS-Shortcuts.ini file. TimeSavers is much easier to use and is worth having for reasons unrelated to document notes, however it does not have the power (or potential risk) of FrameScript.

Sample Implementations

Acrobat Notes

This was originally used to make the information available upon request without interfering with the readability of documents that were normally printed by customers. Technical Support personnel equipped with the full version of Acrobat could use the Comment (Annotations in Acrobat 4) tab for a summary and could generate sorted reports. Customers could view the relevant notes when told where to look.

When TimeSavers was used without FrameScript, shortcuts for each month were added to the TS-Shortcuts.ini file to automatically open preformatted Hypertext markers. This allowed a limited sort, but did not enable full use of Acrobat's reporting capabilities. Files with these markers could only be distilled when TimeSavers was present and included a TS-Shortcuts.ini file with the relevant definitions. Placement of the notes is defined in the shortcut and notes can be effectively hidden from customers with Acrobat Reader, while being available to Technical Support personnel having the full version of Acrobat.

The addition of FrameScript automated the process to the point where a simple hot key sequence opened a special text entry window. Once the window was closed, the script automatically took the time from the OS, converted it to a format that allowed full access to Acrobat's reporting capabilities, and added all prefixes and suffixes. If a more complex script were written that did not use TimeSavers shortcuts, the distilling restrictions could have been removed.

Comment Marker and IOM

This was later implemented when Technical Support requested that the notes be visible to and printable by customers.

A separate script was created to use the Comment marker for document notes. Entries were prefixed with DN and the date (yyyymmdd) taken from the system.

An IOM was added to book files after the Index. A special Reference page was setup to automatically format the Document Notes page(s).

The **DNyyyymmdd** prefix automatically sorted the notes by date. The use of **DN** before the date allowed for future creation of special purpose document notes that could be grouped as a separate sort by type.

Conclusion

Review

We've covered a lot of ground in a short time.

First, we stated the problems that document version management should deal with.

Second, we reviewed the traditional solutions.

Third, we investigated the use of document notes as an alternative solution.

Finally we went into detail on marker based techniques.

Keys to Creating a Solution

1. Management of document versions does not need to be a complex process.
2. The best solution to your needs does not necessarily require expensive software.
3. The two keys to implementing a system that meets your needs are:
 - Analysis of your needs and practical limitations
 - Planning

Followup

For those of you interested in more information, you can contact me at me at:

04-9890532 or svi@ieee.org .

If you need assistance evaluating your needs or setting up a system that meets your needs, I am available for on-site or off site consultation either on an hourly or per project basis.

How To FAQ

How Do I Add a Document Note?

Actually, what you need to add is a FrameMaker marker that you don't use for another purpose, such as the Comment marker. You can add a prefix to make it sortable by document note type and/or date. This can be done automatically. In order to make it sortable by date, the prefix needs to include a prefix format in yyyyymmdd format down to the smallest time unit that you want used for the sort.

This may sound difficult and time-consuming, but the entire process can be automated to the point where all you need to do is hit a hot-key and type in the note. To do this you need at least one of two relatively inexpensive utilities that you may already have for other reasons, TimeSavers and FrameScript.

If you want your document notes to appear as Acrobat notes or to appear in the Comments/Annotation tab, you need to use a specially formatted Hypertext marker. Furthermore, to take full advantage of Acrobat's reporting feature, the prefix should be formatted down to the second. Again, all of this can be automated and the prefix can be taken directly from the system when you embed the document note.

TimeSavers provides marker shortcuts that makes the marker formatting less cryptic, especially if you want to create notes that appear in Acrobat's Comments tab. It has many other uses that justify its purchase, but you can do without it if you are willing to write a more complex FrameScript.

FrameScript lets you automate the entire process. It is a powerful tool that allows you to automate almost any FrameMaker operation, including many that you cannot perform from the FrameMaker interface. However, there is a long learning curve if you choose to write your own scripts and allows you to perform operations that you may regret. However, if you only use it for run existing scripts that you know to be safe, it is easy to use.

What Should I Write In a Document Note?

Any changes that you feel are important. For example, an edit resulting from a change in the product, a change in the installation/operation procedure, a change in the name of a Unix command.

If you want to keep a record of minor changes, such as corrections to spelling or improved working, you should use more than one type of document note.

How Do I Generate a Report?

There are basically two recommended types of document note outputs, Index of Markers (IOM) and Acrobat Notes. The type of output that can be used is determined by the FrameMaker marker and commands used when you embed the document note.

If you use the Index of Markers system, all you need to do is add an IOM for the type of marker used. If you want your customers to be able to view the document notes, simply include the IOM in the book when you print.

If you prefer to create Acrobat notes that appear in the Comments tab (Annotations in Acrobat 4), you simply need to print the file or book with Generate Acrobat Data selected and distill the file. The full version of Acrobat has its own reporting feature that lets you sort by page or date. If you want to create different types of document notes, you can add Author and/or Color attributes to each type of document note.

Do I Need To Configure TimeSavers?

In most cases the answer is yes. Although you can configure everything in FrameScript, it is much easier to edit the TS-short.ini file.

If you use acrobat notes, you can use the ~dnote shortcut. However, I recommend editing this short cut to adjust placement of the note icons to avoid online clutter. It is even possible to completely hide them from view so that they are only available to users with the full version of Acrobat in the Comments/Annotations tab and as reports. The number of characters that can be displayed in the Comments/Annotations tab is limited.

The following text is an example of an alternate short cut added to the TS-short.ini file. This short cut effectively hides the notes from view, but allows them to appear in the Comments/Annotations tab. It is important to comment out the WorldTime adjustment in order for this to work.

```

%% outnote - note %%%%%%%%%%
  % use: alert ~outnote (20020430125959) (X was replaced with Z)

/~outnote
{ /Rect[ -72000 YLL -72000 2 in add YLL 2 in add ]
  /Title (Comment)
  /Contents }
{ exch (D:) exch concatenate
  % (000000-00'00') % WorldTime adjustment
  % concatenate
  /ModDate exch
  /Name/Comment
  /Color[0 0 1]
  /Border [0 0 1] /Open false /ANN }
TS-Shortcut

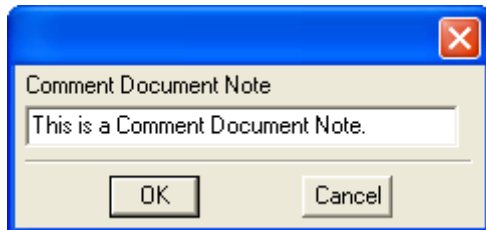
```

Do I Need To Write a FrameScript?

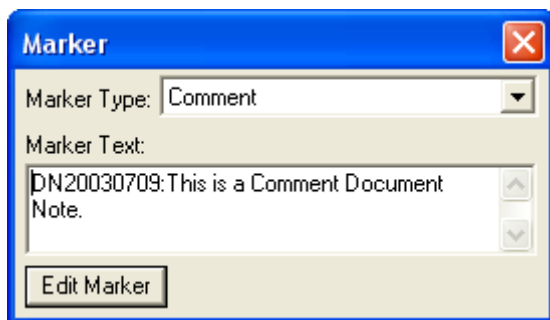
I have already created simple scripts that I can share with you. You can then edit these scripts to meet your specific needs, for example to change the hot key used.

Comment Document Note Script

One script causes ||| (pressing the pipe key 3 times) to open the document note window, where you enter the text of the note.



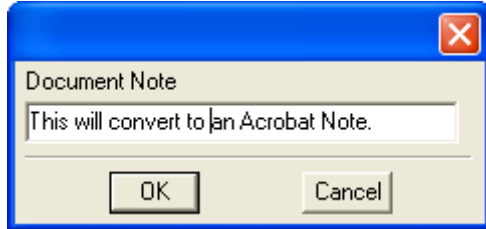
When you close the note, a Comment marker is created with the relevant formatting information, including the time taken from the system down to the day.



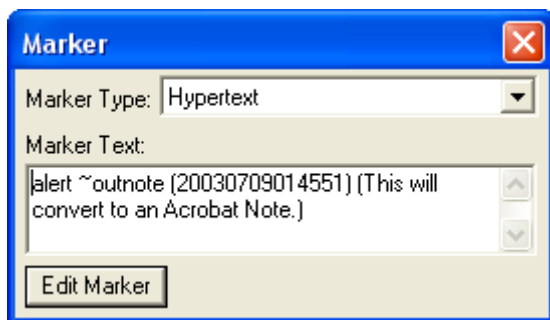
When the Index of Markers is generated, the prefix DN and date in yyyyymmdd format are added as level 1 followed by all of the document notes created that day as level 2. Depending on how you setup the IOM Reference page, the page number can be added as a hyperlink to where the note marker is embedded.

Acrobat Document Note Script

A second script causes `||}` (pressing the pipe key twice followed by the right bracket) to open the document note window, where you enter the text of the note.



When you close the note, a Hypertext marker is created with the relevant formatting information, including the time taken from the system down to the second.



When you print the file or book with Generate Acrobat Data selected, the marker is written to the PostScript file. When you distill, TimeSavers converts these markers to Acrobat notes, which are hidden from view, but visible in the Comments/Annotations tab. The full version of Acrobat can also generate reports that you can print or save as a PDF. When you embed this type of document note, Acrobat distiller will not be able to distill the file on machines where TimeSavers is not installed.

Load Script

The third script loads the other two scripts to memory each time you open FrameMaker.

Can I share my source files with other writers?

In theory yes. However, if you use TimeSavers shortcuts to embed the documents, the Acrobat notes can only be created when distilling on a machine having TimeSavers with the TS-short.ini definitions used. For many, this is a small price to pay for time saved and value added compared using a more traditional version management system.