Inspecting and Repairing Notefiles

Xerox Corporation

Randy Trigg

[First written: 8/9/85 Randy Trigg]
[Last updated: 8/26/85 Randy Trigg]

This document describes the Notefile inspector facility available via the Inspect&Repair option on the Notefile Ops menu in NoteCards Release1.2i.

The old Repair Notefile facility rebuilt the links in a Notefile from the contents of card substances. This was used whenever a notefile was thought to have inconsistent links. The problem was that notefiles with inconsistent links often had other problems that caused Repair to break. Thus the motivation for developing the notefile inspector documented here was to verify a notefile's readability before invoking the link rebuilder. As it turns out, this inspector is useful generally for checking the health of a notefile, deleting cards and backing up other cards (or more precisely, card parts) to previous versions. Thus, you may want to use the inspector even if your notefile is healthy and doesn't need its links rebuilt.

The notefile inspector has three separate phases: reading the notefile's data area searching for healthy card parts, allowing the user to make modifications, and rebuilding the links. The process can be aborted after phase 1 or 2 if desired. This document begins with a brief discussion of the organization of a notefile. Then follows sections describing each of the three phases. Finally, I outline some tips, strategies and pitfalls to watch for.

1. What you need to know about a notefile's innards.

1.1 Notefile structure.

A notefile consists of two parts, the index and the data area. Each card in the notefile has an entry in the index. An index entry has 5 parts, a status field and 4 pointers. The status field specifies whether the index entry is free or occupied by an active or deleted card. There is one pointer in the index entry to each of the 4 parts of a card: substance, title, links and property list. These point into the data area. Whenever you change, say, the title of a card, the new title is written to the end of the data area and the index entry title pointer for that card is updated to point to the new location in the data area. Thus, in general, a notefile's data area grows every time any part of any card is changed. To throw away the old versions of card parts, it is necessary to compact the notefile.

1.2 Card IDs.

Every card in the notefile has a unique ID, e.g. NC00023. The top level fileboxes; **Contents**, **Orphans**, and **To Be Filed** have IDs NC00001, NC00002, and NC00003, respectively. Note that these boxes cannot be deleted. The IDs from NC00004 through NC00020 are unused. Currently, an old ID is never reused, even if its card is deleted and the notefile is compacted. Thus, if the inspector shows no entry in the card inspector menu for some ID, it is because that card has been deleted. If you've asked to show deleted cards and it still doesn't appear, it's because the notefile has been compacted since that card was deleted.

1.3 Card parts.

Of the four parts of a card, the title and property list are simplest. The title is simply a string while the property list is a list of attribute value pairs. If you have not been attaching properties to your cards, then the inspector will only show those properties that the system maintains. Currently the only such property is "Updates" with value equal to a list of dates on which the card was updated going chronologically backwards from the front of the list to the back.

The substance of the card is simply its contents. Thus a text card's substance is a text stream, a browser card's is its graph, etc. These are stored on the file in a manner appropriate to the substance type. Thus a text substance on the notefile looks like the way TEdit writes out text streams (text followed by "looks" information).

1.4 Links on the notefile.

The links of a card are divided into three groups: **to links**, **from links**, and **global links**, where the global links form a subset of the to links. The to links of a card are those links pointing from this card to some other card. The from links are those links pointing from another card to this one. Finally, the global links are those to links that are global, that is, they point from this card as a whole to another card. (Global links are the ones that aren't anchored in the source card's substance. Source links are an example.)

The confusing thing about links out on the notefile is that they are stored in several places. All links are stored as to links with their source card and as from links with their destination card. Furthermore, links that are not global are also stored within the substance of their source card inside of a link icon. Links that are global are also stored on the global links list of their source card. Thus, all links are stored in three places: as a to link on the source card, as a from link on the destination card and either in the substance of the source card or on its global links list. If these three records of a link don't agree for some reason, then we say that the notefile is **inconsistent** and needs its links rebuilt.

1.5 The links rebuilder.

The third phase of the inspector rebuilds the links of a notefile as follows: First it removes all the to and from links for every card. Then it reads the substances for each card and recreates to links and from links by looking at the links found inside the link icons in the substance.

The link rebuilder is also able to rebuild bad filebox substances. It does this by looking for all cards in the notefile with from links from the bad box and creating a new substance for the box containing only links to those cards. This process loses any text that the box might have contained as well as scrapping the original ordering of links. Nonetheless, in some cases this may be preferred to backing up the substance to a previous version or to deleting the box altogether.

The links rebuilder can rebuild the notefile's list of link types in a similar manner. That is, it records the set of link types seen on valid links and replaces the old links types with the new set. Note that this throws away any link types for which there are no links in the notefile.

Finally, the links rebuilder can rebuild bad global links for a card. It does this by looking for any cards with from links from the bad card that are global. This assumes that the card at the destination end has good links. Thus, if the cards at both ends of a global link have unreadable links, then there is no way to recover that link.

The inspector provides the option of having the links rebuilder phase rebuild bad filebox substances, bad link types, and bad global links. See Section ?? below.

2. Running the Notefile inspector: Phase 1: Scouring the data area

To start the inspector, first be sure that there is no open notefile. Then select the item **Inspect&Repair** from the NoteFile Ops menu. There is one option available at this level by "pulling to the side" called **ReadSubstances**. This ensures that substances of all cards pronounced valid by the inspector are readable. If this option is not invoked, then a check is still run on the length of the substance, but not on its contents. Unfortunately, the ReadSubstances option requires MUCH more work by phase 1. I recommend that you only use this option if Phase 3 (link rebuilding) breaks with some error like "Bad Piece Tbl" from TEdit. In that case, up-arrow out of the break and start the Inspect&Repair process over again, this time using the slower but more comprehensive ReadSubstances option.

Selecting Inspect&Repair will invoke phase 1 of the inspector, wherein the data area of the notefile is scoured for valid card parts. A record of all such parts is kept and statistics printed out at the end. You'll be asked to position the window in which those statistics as well as later inspector communications will be printed. You can monitor the progress of phase 1 by watching the prompt window. It will be printing messages like "Processing byte xxxxx of yyyyy."

When phase 1 has completed and you've positioned the interaction window, statistics on your notefile will be provided. You'll be told the total number of card IDs used and the number of those that are currently associated with active and deleted cards. (The rest are free and will never be reused. See Section 1.2 above.) If all seems well with the world, the next line will read "All active cards look okay." If not, there will be various messages outlining the problems. (See Figure 1.)

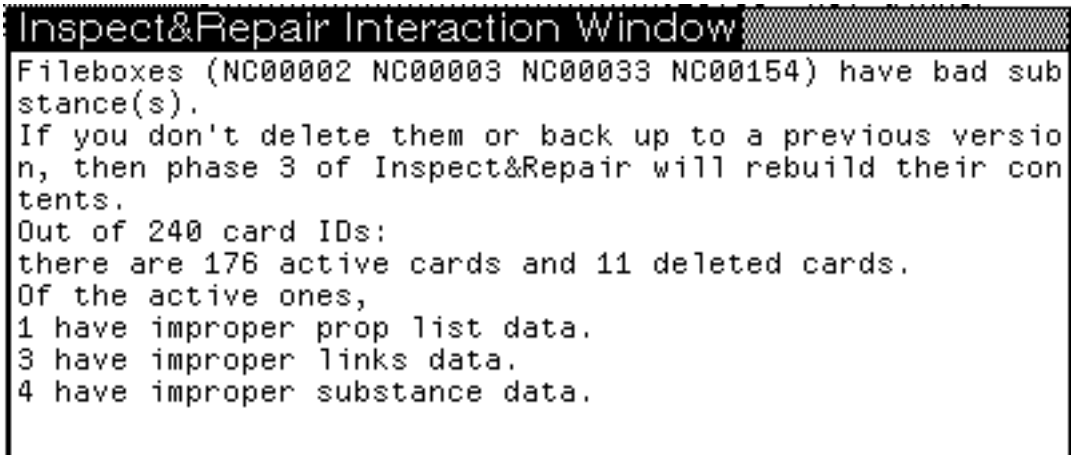

Figure 1: Snapshot of a sample interaction window

Note that several fileboxes have bad substances and that a special message is printed on their behalf. This indicates that if you don't wish to delete or back these up to a previous version, then phase 3 will rebuild them. (See Section 1.5 above.)

If there are cards having user-defined types whose type definition code has not been loaded, then you'll get a message to that effect, something like "<n> cards have unknown card types (FOO BAR)." At this point you should load the lisp files containing the definitions of the unknown card types. If not, then these cards will show up with bad substance in phase 2. If you have a card for which no substance versions could be read, then you'll also get unknown card type messages for it (reading something like "<n> cards have unknown card types (NIL)"). This is because the inspector couldn't find a card type on the notefile for that card.

A menu of options appears attached to the upper right corner of the interaction window. The particular options you get in that menu depend on the state of your notefile and are described below. The first two options appear in all cases. The other two may or may not be present in the menu you get. In any case, you should select one of the options before attempting any other NoteCards-related work.

ABORT: Choosing this option aborts the Inspect&Repair process entirely, throwing away any changes you might have made (such as card deletions or back ups.)

INSPECT CARDS: This brings up a menu of active card IDs with which you can inspect, delete, or back up particular cards. There is a "pull-across" menu item called INCLUDE DELETED CARDS, which if selected will include card IDs for deleted cards as well as active ones. Using this option, one can undelete deleted cards and restore some previous version.

END INSPECT&REPAIR: This option is only available if it seems that you don't need to continue to the link rebuilding phase. You will not get this option if you've deleted any cards, or generally if there are problems with the notefile. Choosing this option causes the Inspect&Repair process to end gracefully (via a normal checkpoint and close notefile), thus skipping phase 3, rebuilding links.

CONTINUE INSPECT&REPAIR: This option is only available if the notefile is in fairly good health (i.e. okay except for fileboxes to rebuild or global links to rebuild - see Section 1.5 above). Selecting it causes Inspect&Repair to move to phase 3 and rebuild your notefile's links.

3. Running the Notefile inspector: Phase 2: Your chance to tinker

After selecting INSPECT CARDS in the interaction window's attached menu, a menu containing Notecards IDs will pop up and be attached to the interaction window's lower left corner. It will contain IDs for all active cards and possibly deleted cards as well if you selected the submenu item INCLUDE DELETED CARDS described above. The menu can hold some 200 card IDs. If your notefile has more than that, then the menu will be composed of several pages each containing around 200 IDs. Rapid switching between pages is possible.

Attached to the upper right corner of the cards inspector menu is a menu containing at least the two options: ABORT and DONE. If the menu has multiple pages (there are more than 200 active cards in the notefile), then the attached menu will also include the items NEXT PAGE, PREVIOUS PAGE, and FIRST PAGE. Selecting these causes the current menu to be swapped with either the next menu, previous menu, or first menu, respectively.

Clicking ABORT causes the entire Inspect&Repair process to quit, throwing away any changes you've made. (This is equivalent to choosing ABORT from the inspector window as described in Section 2.)

Choosing DONE from this attached menu indicates that you're done tinkering with card parts and wish to return to the main interaction window. Normally, this causes the card inspector

menu to close and the phase 1 process outlined in Section 2 to be performed again. Thus the data area will be rescoured and new statistics on the health of your notefile will be printed. This cycle of scour data area (phase 1) followed by inspect (phase 2) can be repeated as often as desired. Eventually, you must either abort, end the Inspect&Repair process gracefully, or continue to phase 3. Because phase 1 can be quite slow for large uncompacted notefiles, there is one optimization: if you've made no changes in phase 2, then the data area scouring is not repeated in phase 1. Rather, the old information from the last scouring is recovered and used instead.

If you've clicked DONE, but there are still cards with bad prop lists, titles, or links (because you haven't either deleted them or backed up their card parts to previous versions), you will be asked up to three questions. The list of card IDs of cards with bad prop lists is printed out and you're asked whether it's okay to set the prop lists of those cards to NIL. Then, the list of card IDs of cards with bad titles is printed out and you're asked whether it's okay to set the titles of those cards to "Untitled." Finally, card IDs for cards with bad links are printed out and you're asked whether it's okay to set the global links to nil. (Global links will be rebuilt as much as possible in phase 3. See Sections 1.5 and 4.) If you want phase 3 to be able to run, it is necessary to either answer yes to these questions or fix each of the bad card parts by hand in phase 2.

3.1 The card inspector menu

In the card inspector menu, those IDs corresponding to deleted cards have a line drawn through them. Those having some sort of problem appear shaded. In addition, an upper-case letter suffix is attached to such IDs indicating the problem. For example, a shaded menu item NC00023SL indicates that ID NC00023 has bad substance and bad links. The letter codes are S, L, P, and T indicating bad substance, links, property list, and title, respectively. If such a letter code appears in lower case, then the indication is that the current version of that card part is beyond the last checkpoint pointer. For example, NC00023t indicates that NC00023's current title was changed since the last checkpoint. (There may have been a crash, for example, thus preventing the notefile from closing normally.)

In addition to menu entries for each card ID, there is also one entry labeled LNTYPES allowing you to inspect (and possibly back up to a previous version) the Link Types for this notefile.

If you button an ID in the card inspector menu, then a popup menu allows up to two choices **Inspect** and/or **Delete**. If the card is currently deleted (has a line drawn through it), then the Delete option is replaced by **Undelete**. Certain card IDs cannot be deleted and thus their popup menus only contain the Inspect option. These are the top level file boxes NC00001, NC00002, NC00003. The link types menu entry LNTYPES also does not allow deletion.

Choosing Delete or Undelete from this popup menu causes the card to be deleted or undeleted, respectively and the line through the menu item either drawn or undrawn. Note, however, that this action (and all others) can be undone by choosing ABORT from either the card inspector menu or the interaction window menu.

Choosing Inspect from the popup menu for a card ID entry brings up a card parts inspector for that card.

3.2 The card parts inspector

Figure 2 below shows an example of a card parts inspector. It is composed of four attached menus arranged vertically and one attached operations menu atop the stack.

Figure 2: A card parts inspector

The four menus contain entries for every valid version of card parts for the card with ID NC00027. The top menu is for version of titles and below that are menus for versions of the card's substance, links, and prop list. For example, the Substance submenu contains entries for three versions of the substance of this card. The current version of each card part is shaded. Each menu entry gives the date that that version was written if available or the string "NO DATE AVAILABLE" if there is no date on the notefile. (The latter is the case for old notefiles prior to the time we began recording card parts write dates.)

If the current version of the card part is bad, then the menu entry will be a string so indicating, for example, "BADSUBSTANCE."

The title of the top menu includes the card's type and ID. In addition, each menu item contains a bit of information, in square brackets, before the date. In the title versions menu, this information is the first few characters of the title. In the substance versions menu, it is the number of bytes in the substance. In the links versions menu, it is a triple of numbers giving the number of to links, from links, and global links for this card. Finally, the proplist versions menu includes the number of entries on the property list for this card (i.e. twice the number of attribute-value pairs).

Atop the stack of menus is an attached menu of operations, described below.

ABORT: This aborts this card parts inspector, throwing away any changes made.

UPDATE: This closes the card parts inspector, effecting any changes (backing up to previous versions of card parts) you might have done.

DELETE: This option closes the card parts inspector and deletes the card. (Again, this can be undone by choosing ABORT from the card inspector menu.)

UNDELETE: For cards that have been deleted, this option appears instead of DELETE. Choosing it causes the card to be undeleted.

RESET: This causes the selections in the submenus to be restored to the values they had when the card parts inspector was first brought up. (Equivalent to doing ABORT and then inspecting this ID again from the card inspector menu.)

Note that cards that can't be deleted don't have the DELETE option on their card parts inspector.

Buttoning an entry in a submenu of a card parts inspector pops up a short menu unless the entry is for a bad version (e.g. "BADSUBSTANCE"). This menu contains at least the entry **Inspect** and possibly **Change Selection**, if the selected entry is not the same as the current one (i.e. not shaded).

Choosing Inspect allows further inspection of the details of the selected card part version. For example, inspecting a title version brings up the Interlisp inspector on a record containing the title, date and card ID. Similarly, inspecting a links or prop list version brings up the Interlisp inspector on a record containing the lists of links (for to links, from links and global links) or the prop list. Note that if you wish to continue inspecting a links version down to the single link level, choose to inspect the link as a NOTECARDLINK. This is somewhat more communicative about record field names. Note also that changing values out in the Interlisp inspectors has no affect on the notefile and is ignored.

All substance versions for cards having substance types TEXT, SKETCH and GRAPH can currently be inspected. (This includes all cards except those having user-defined substance types, like the NCFile card.) Inspecting a card's substance version will bring up a window showing a copy of the substance. (Note that changes to this copy have no affect on the notefile.) Any links in the substance of the card will show up as bracketed strings describing the link.

Choosing Change Selection from the card part version popup menu causes the current selection to be changed, thus backing up the card to the selected version. (This change can be undone by resetting or aborting the card parts inspector as well as by later aborting the card inspector or interaction window.)

4. Running the Notefile inspector: Phase 3: Rebuilding your links

To complete the Inspect&Repair process, select the **Continue Inspect&Repair** option from the interaction window menu. This invokes phase 3, the links rebuilder. Normally, this simply rebuilds links from card substances (see Section 1.5). In certain circumstances, it may do extra work as well. If your link types list is bad, and you didn't back it up to a previous version, then phase 3 will rebuild it. If there are fileboxes with bad substances that you haven't either deleted or backed up to previous versions, then phase 3 will rebuild them. Finally, if there are cards with bad links that you haven't backed up or deleted, then phase 3 will rebuild those links as well. (It rebuilds ALL to links and from links anyway. For those cards, it will rebuild global links as well.) Again, for details, see Section 1.5.

5. Tips and hints for using Inspect&Repair

This section contains a list of strategies and tips for using Inspect&Repair. For the most part, they are ordered from the useful and obvious to the esoteric. Several of these are implicit in the first four sections of the document, but are repeated here for emphasis and completeness.

When in doubt, abort! All your changes will be lost, but then if you're uncertain about what's happened this is the safe course. Often, in fact, you may simply want to check the health of your notefile and abort without tinkering.

Fixing versus tinkering. There are two main ways to use the inspector, either for fixing a broken notefile, or tinkering with a healthy one. The latter case occurs when you wish to recover some card that you inadvertantly deleted. Or back up a card that you inadvertantly changed to a previous version.

Compacting. Old versions of card parts have always been stored in notefiles, but up till now have been inaccessible. Thus, there was little reason not to compact your notefile often. Now there is a tradeoff between the need to save space by compacting versus the need to be able to back up using Inspect&Repair. Probably the safest course is to keep a backed up copy of the pre-compacted notefile around until you have confidence that the compacted one is healthy and that you have no need for previous versions of any of its cards.

Inspect&Repair can't run when notefile is open. This means that if you are working in your notefile and notice a card you'd like to inspect a previous version of, you must record the card's ID and close the notefile. Then, run Inspect&Repair, find the card ID in the inspector menu and tinker with it as desired.

Fixing enough problems to allow phase 3 to run. You can't run phase 3 unless Inspect&Repair thinks your notefile is above a certain threshold of health. There are certain problems it can handle (e.g. bad filebox substances, see Sections 1.5 and 4), and others that it can't (e.g. bad title). You have to decide either to fix these sorts of problems yourself in phase 2, let phase 3 attempt to rebuild them, or just abort the whole thing (always an option).

Sometimes these decisions can be tricky. For example, suppose a filebox's substance is bad. Call it BadBox. Should you (a) delete BadBox altogether, (b) back its substance up to a previous version, or (c) allow phase 3 to rebuild it by looking for from links in other cards from BadBox? Option (c) may not be advisable if there was important text in BadBox or if the order of cards in BadBox was important. On the other hand, option (b) may be of little use if the last good version is too out of date (or if there is no good version at all).