

The main purpose of the notefile inspector is to repair notefiles by rebuilding their links. However, the inspector is generally useful for checking the health of notefiles, deleting cards and backing up 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 describes each of the three phases in turn and concludes with tips, strategies and pitfalls to watch for.

Using the notefile inspector requires some knowledge of the innards of notefile organization. We recommend that you read Appendix A, Notefile Concepts, before continuing.

Note: The information in this appendix was copied whole from the documentation for a prerelease of NoteCards and has not yet been updated for this release. As such, it may be inaccurate in places.

Running the Notefile Inspector: Phase 1: Scouring the Data Area

To start the inspector on a notefile, first be sure that the notefile is not open. Then select the item **Inspect&Repair** from one of the NoteFile Ops menus (i.e. from the notefile Banner, the MenuBox icon or the notefile FileBrowser). 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. We 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. (Note that closing this window will abort the Inspect&Repair process if you confirm.) You can monitor the progress of phase 1 by watching the prompt window. It will be printing messages like "Processing byte xxxxx of yyyy."

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 cards and the number of deleted ones. If all seems well with the world, the next line will read "All non-deleted

cards look okay." If not, there will be various messages outlining the problems. (See Figure 1.)

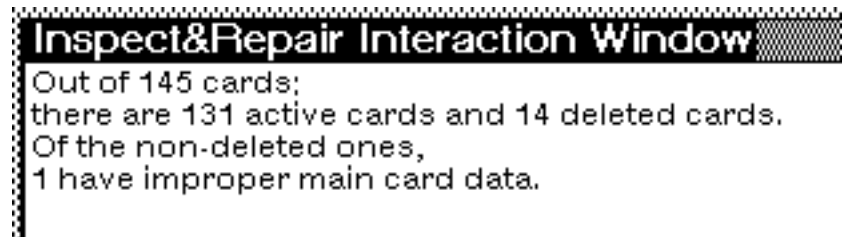


Figure B-1: Snapshot of a sample interaction window

Normally, the messages describing problems are of the form shown in Figure 1, namely a count of the number of cards broken in a given way. Some messages are a bit trickier. If you have fileboxes with bad substances (i.e. main card data), then you're told that if you don't wish to delete or back up such fileboxes to previous versions, then phase 3 will rebuild them. If, however, the filebox is one of the top level boxes (Contents, Orphans, ToBeFiled), then deletion is not an option. (See Section 3.)

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 do load the appropriate files, then run **Recheck Bad Cards** to get the bad cards list recomputed before inspecting any cards.

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. (See Figure 2.) The particular options you get in that menu depend on the state of your notefile and are described below. The last three 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.



Figure B-2: The Phase 1 options menu.

Continue 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 3). Selecting it causes Inspect&Repair to move to phase 3 and rebuild your notefile's links.

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 of rebuilding links.

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.) It requires your confirmation in the prompt window.

Recheck Bad Cards

Recompute the bad cards list by running the last part of phase 1 again. This is useful if you've loaded files containing definitions of previously undefined card types.

Inspect Cards

This brings up a menu of titles of active cards 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 titles for deleted cards as well as active ones. Using this option, one can undelete deleted cards and restoring them to some previous version.

Running the Notefile Inspector: Phase 2: Your Chance to Tinker

After selecting **Inspect Cards** in the interaction window's attached menu, a menu containing numbered titles of notecards pops up and is attached to the interaction window's lower left corner. It contains the first 10 characters of titles 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 100 card titles. If your notefile has more than that, then the menu is composed of several pages each containing around 100 cards. Rapid switching between pages is possible. Figure 3 below shows a sample card inspector menu.

Attached to the upper right corner of the cards inspector menu is a menu containing at least the three options: **Abort**, **Done** and **Search**. If the menu has multiple pages (there are more than 100 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 exchanged for 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. It requires your confirmation in the prompt window. (This is equivalent to choosing

Abort from the inspector window as described in the previous section.)

Choosing **Done** from this attached menu indicates that you're done tinkering with card parts and wish to return to the main interaction window. This causes the card inspector menu to close and the phase 1 process outlined in the previous section to be performed again (leaving you looking at something like Figure 1). Note, however, that the scanning of the data area is not repeated (it takes way too long). Rather, your changes are made to the in-core index array and the statistics on bad cards are recomputed. This cycle of compute statistics (phase 1) followed by inspect and tinker (phase 2) can be repeated as often as desired. Eventually, you must either abort, end the Inspect&Repair via **End Inspect&Repair**, or continue to phase 3 via **Continue Repair**.

Choosing **Search** from the attached menu allows you to find those cards having titles matching a given string. This works much like the Search card in Notecards. That is, you specify a string and the list of cards whose titles contain that string is printed out. This is handy, for example, if you want to undelete a card, but only know its title. Note that the search is case *ins*sensitive.

The card inspector menu

In the card inspector menu, those titles 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 entries indicating the problem. For example, in the card inspector menu shown in Figure 3, the shaded menu item "10: BNF for Li|S" indicates that the card with title beginning "BNF for Li" has bad substance. 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 meaning is that the current version of that card part is beyond the last checkpoint pointer. (There may have been a crash, for example, thus preventing the notefile from closing normally.)

Figures 3 and 4 show both pages of a two page card inspector including deleted cards. Sometimes deleted cards (e.g. "110: NIL|LSPT" in Figure 4) show up shaded with LSPT suffixes. This is because they were deleted before any data about the card was written to the notefile. Thus there are no valid card parts to back up to for those cards. But often, deleted cards can be backed up to previous versions. (See Figure 6 and discussion below.)

Card inspector: Page 1 of 2				Abort Done Search Previous Page Next Page First Page
1: Contents	2: Orphans	3: To Be File	4: Link Label	
5: Registry	6: Papers	7: Value Cell	8: Value cell	
9: Introducti	10: BNF for LIS	11: BNF for Co	12: Format of	
13: Recomputin	14: Example fr	15: Implicatio	16: Extensions	
17: Formula ex	18: Collect ex	19: Outline	20: Value cell	
21: People	22: Schatz, Br	23: Sharma	24: Dean, Jeff	
25: Kasif, Sim	26: Krovetz, B	27: Liles, Bil	28: Feuerman,	
29: Cook, Lesl	30: Kondo, Jin	31: Schwartz,	32: Ward, Sand	
33: Gregory, R	34: Schwartz,	35: Delisle, N	36: Chapman, G	
37: Hamlet, Di	38: Ellis, Ski	39: Halasz, Fr	40: Brennan, S	
41: Moore, Lee	42: VanLehn, K	43: Carnese, D	44: Notecards	
45: NC in Loop	46: Review of	47: Pros and C	48: Introducti	
49: Discussion	50: Paper outl	51: Document f	52: Reggia, Ji	
53: Wohltman,	54: junk	55: NoteCards	56: Applicatio	
57: Applicatio	58: Introducti	59: Random not	60: NCAPPLICAT	
61: Mail box	62: Question o	63: The progra	64: ARI Exampl	
65: Second exa	66: Summary of	67: Notes from	68: Mont-Reyna	
69: Kim, Scott	70: Weiser, Ma	71: Places	72: REI	
73: Frescoe's	74: Higgins, R	75: Dieter's	76: Pitts, Wel	
77: Leavitt, M	78: Marshall,	79: Kimball's	80: Deerwester	
81: CPSR	82: Moran, Tom	83: Nelson, Te	84: Informatio	
85: Mail	86: Loops	87: Colab	88: Koncepts C	
89: Wijsman, N	90: Todd, John	91: Draft	92: Notes on K	
93: Stats on K	94: Cards	95: Links	96: Notefile	
97: Notes from	98: Thoughts o	99: Document f	100: Notes from	

Figure B-3: A card inspector menu: Page 1.

Card inspector: Page 2 of 2		Abort Done Search Previous Page Next Page First Page
102: Tailorabil	103: Global par	
104: Interlisp	105: Card regis	
106: PI-based f	107: Event hook	
108: Template c	109: Interlisp	
110: Programmer	111: Types mech	
112: Card types	113: Tailorabil	
114: foo	110: foo	
117: asdf	110: FILE OPT	
119: Table of P	120: Table of c	
121: Table of c	122: Support fo	
123: Outline	124: Techniques	
125: Document f	126: PI-based f	
127: Event hook	128: Document f	
129: Thoughts	130: Talks	
131: Talk on NC	132: Types hier	
134: Title slid	135: Levels of	
136: Global par	138: PI sketch	
139: Use of PI	140: Event hook	
141: Types mech	142: Miscellane	
143: IDE exampl	144: Kinds of c	
145: Estrin, De		

Figure B-4: A card inspector menu: Page 2.

At the top of the first page of any notefile's card inspector, there are entries for the top level boxes and special cards. These are labeled

"1: Contents", "2: Orphans", "3: To Be File", "4: Link Label", and "5: Registry".

Note that if the inspector shows no entry in the card inspector menu for some card, 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.

If you button card title entry 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 cards cannot be deleted and thus their popup menus only contain the Inspect option. These are the 5 special cards mentioned above.

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 entry brings up a card parts inspector for that card.

The card parts inspector

Figures 5 and 6 below show examples of card parts inspectors. A card parts inspector is composed of four attached menus arranged vertically and one attached operations menu atop the stack.

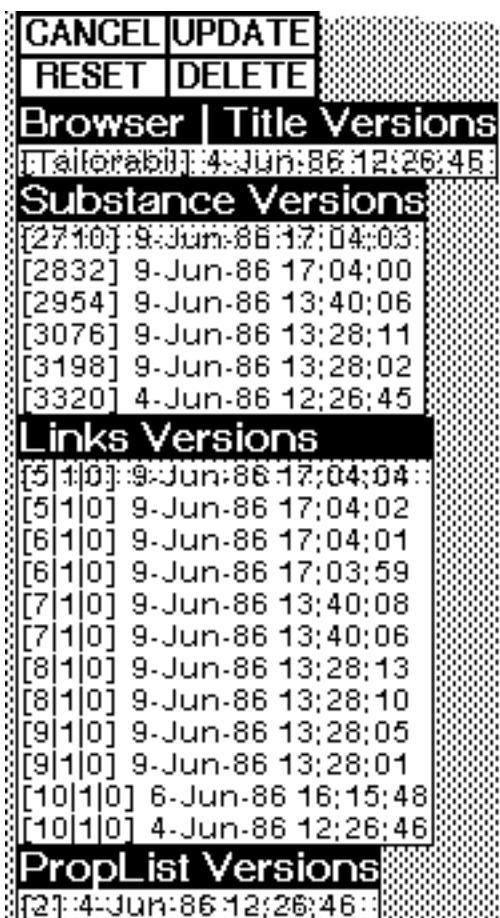


Figure B-5: Card parts inspector for a browser card.

In Figure B-5, the four menus contain entries for every valid version of card parts for the browser card with title "Tailorability". The top menu is for versions 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 6 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. (Very old notefiles may show "NO DATE AVAILABLE" in place of a real date.)

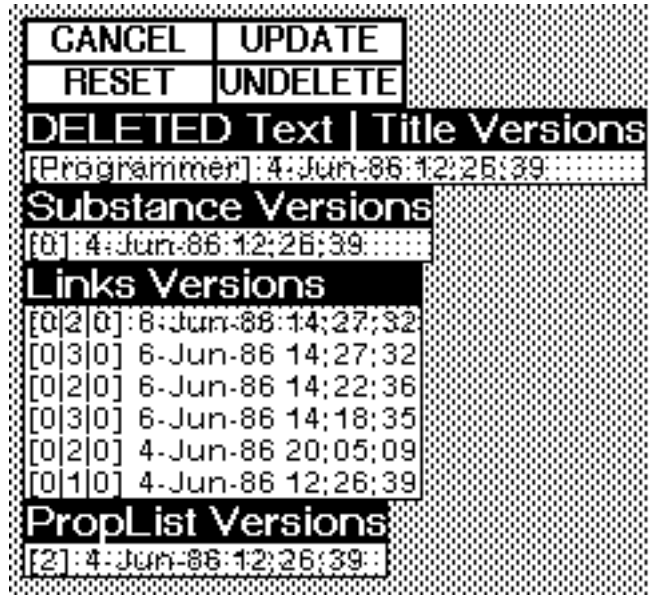


Figure B-6: Card parts inspector for a deleted card.

Figure B-6 shows a card parts inspector for a deleted text card and thus allows the UNDELETE option. By clicking UNDELETE, this card can be backed up to its last version.

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. In addition, each menu item contains some 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.

CANCEL

This cancels this card parts inspector, throwing away any changes made.

UPDATE

This closes the card parts inspector, incorporating 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.

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 CANCEL and then inspecting this card again from the card inspector menu.)

Note that cards that can't be deleted (like the top level fileboxes) 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. (See Figure B-7 below.) For example, inspecting a title version brings up the Interlisp inspector on a record containing the title, date and card object. Similarly, inspecting a links or prop list version brings up the Interlisp inspector on a record containing the lists of links (i.e. to links, from links and global links) or the prop list. Note that changing values in these Interlisp inspectors has no affect on the notefile and is ignored.

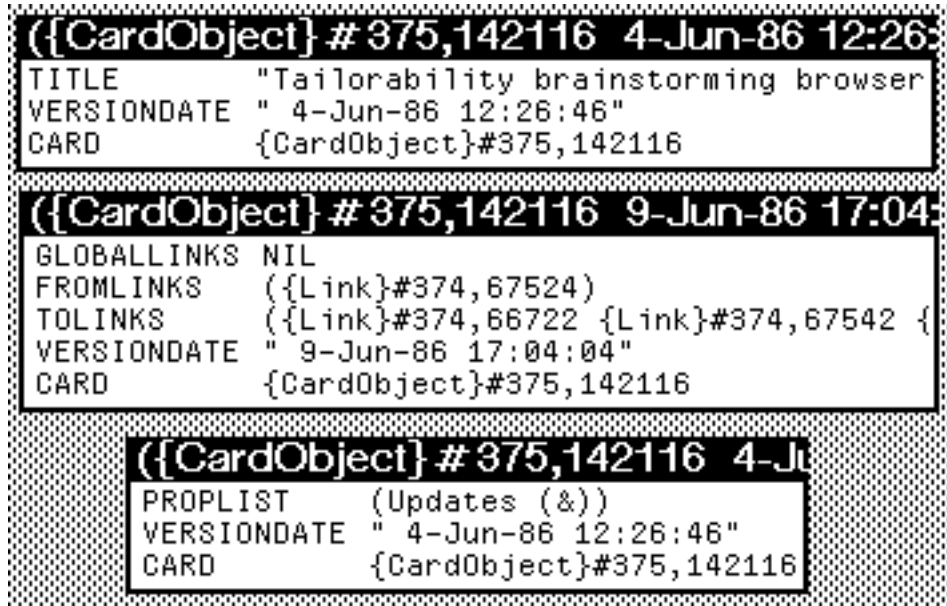


Figure B-7: Card part version inspectors for title, links and prop list.

All substance versions for card types inheriting from Text, Sketch and Graph can currently be inspected. (This includes most 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 to the selected one. This changes a card part's current version to another legal one. (This change can be undone by resetting or canceling the card parts inspector as well as by later aborting the card inspector or interaction window.)

Running the Notefile inspector: Phase 3: Rebuilding your links

To complete the Inspect&Repair process, select the **Continue Repair** option from the interaction window menu. This invokes phase 3, the links rebuilder. Normally, this simply rebuilds links from card substances. 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.)

First the links rebuilder 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 card's 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.

Note that the links rebuilder automatically files any unfiled cards in the ToBeFiled box. A message is printed in that case.

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 was inadvertently deleted. Or back up a card that was accidentally changed to its original version.

Inspect & Repair after a crash

If you try to open a notefile and you get the infamous "Work was done since last checkpoint..." message, then you're given the chance to run Inspect&Repair on the notefile. This is a good idea

whenever you wish to recover that post-checkpoint work. The inspector will integrate the changes you made since that checkpoint. (You do have to continue through phase 3, however.)

Compacting

It used to be the case that old versions of card parts were in the notefile but inaccessible. Thus, there was little reason not to compact a 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 closes the notefile before starting

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 title and close the notefile. Then, run Inspect&Repair, find the card's entry in the inspector menu using the Search facility, 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 Section 3), and others that it can't (e.g. bad substance for non-filebox). 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