# NCGuidedTourCard

By: Randy Trigg (Trigg.pa@Xerox.COM)

Stored: {qv}<notecards>1.3L>Library>NCGuidedTourCard, .lcom, .ted
Written: August 1, 1988 by Randy Trigg
Last updated: May 5, 1989 by Peggy Irish.
Loads NCTableTopCard package.

## INTRODUCTION

The author of a complex NoteCards network is sometimes faced with the problem of conveying its meaning to future online readers using the same hypermedia environment. The NoteCards guided tours library package provides a way to reify and manipulate branching paths through a NoteCards network, thus allowing readers to explore the notefile under the guidance of the notefile's author.

A guided tour is a graph whose nodes are tabletop cards (see the documentation for the NCTableTopCard NoteCards library package) and whose edges are GuidedTour links connecting the cards. Because the tabletop cards may have other links connecting them to other parts of the network, the guided tour is a subgraph of the subnetwork containing those cards. The guided tour card is implemented as a specialization of the NoteCards browser card. Thus, creating and editing a guided tour is almost identical to creating and editing a browser. The biggest change is the ability to "run" the guided tour by means of a row of buttons across the top of the browser window. In this way, readers and authors access the tour through the same graph-based interface.

Readers interested in learning more about the history of and motivation for the guided tour package (as well as the tabletop facility) are directed to the paper "Guided Tours and Tabletops: Tools for Communicating in a Hypertext Environment," by Randall H. Trigg, Conference on Computer-Supported Cooperative Work, September 1988. (Copies can be obtained by writing to SSLPapers.pa@Xerox.com.)

## LOADING THE NCGUIDEDTOURCARD PACKAGE

Loading the NCGuidedTourCard package is done in the same way as loading any

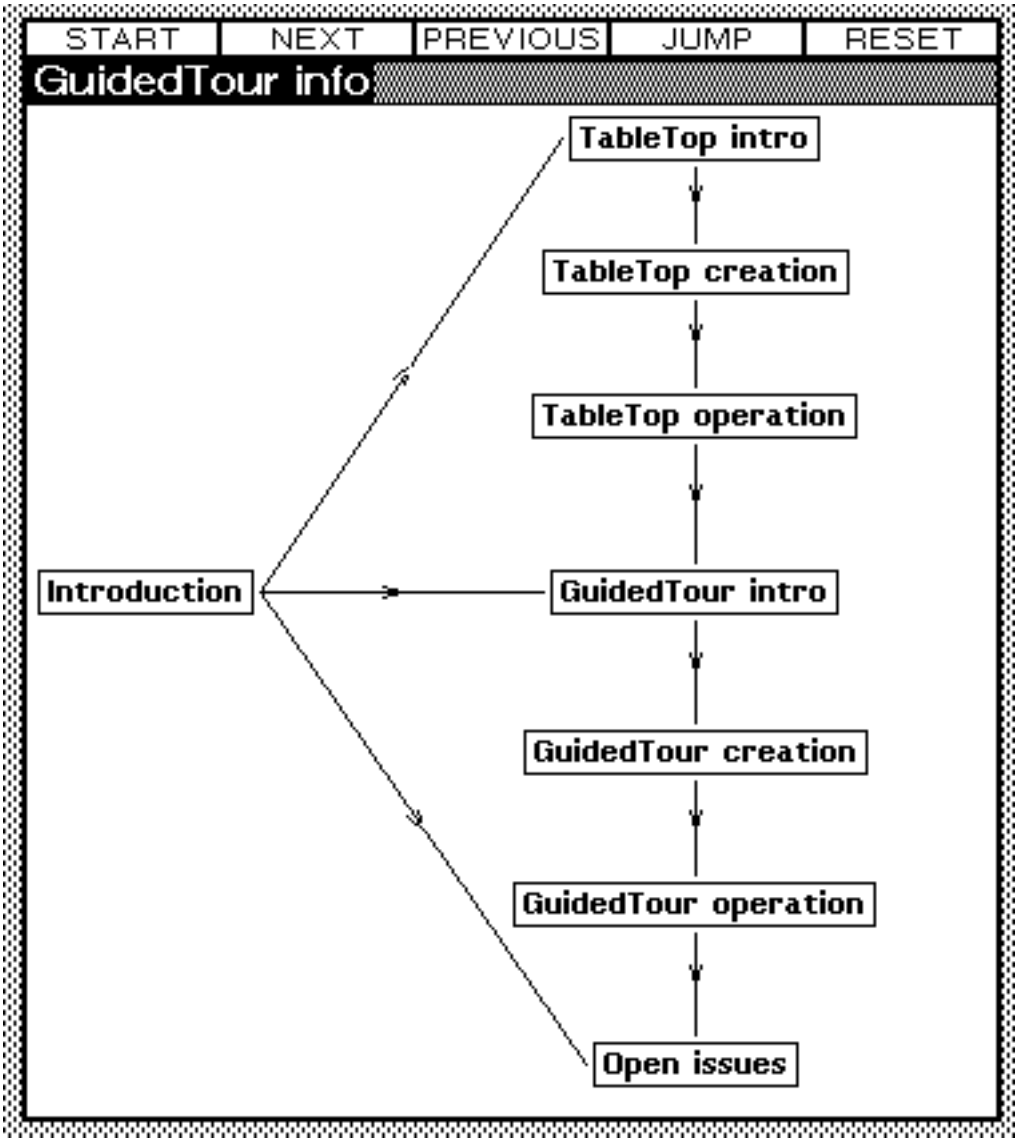
other NoteCards card type library package. For example, you can type the following at an Interlisp exec:

(FILESLOAD (FROM NOTECARDS) NCGUIDEDTOURCARD)

Alternatively, you can add the atom NCGUIDEDTOURCARD to the global var NOTECARDSLIBRARYFILES in your init file. This will ensure that NCGUIDEDTOURCARD is loaded when bringing up a fresh sysout. Finally, if a link to a guided tour card is followed and the NCGUIDEDTOURCARD package is not currently loaded, then it will be auto-loaded by the system.

**"RUNNING" A GUIDED TOUR**

Operating or "running" a guided tour is done via the five buttons arrayed along the top of the guided tour card as shown below.



In each of the following operations, the current displayed tabletop is automatically closed down before bringing up the next. (This is made more efficient by not closing those cards shared by the current and next tabletops.)

START - Clicking on START causes the first tabletop in the tour to be brought up. For simple linear or singly-rooted tree structures there is an obvious "first" tabletop. Some tours, however, might have multiple roots, i.e. multiple nodes without incoming edges. In such cases, the user is asked to choose from a menu of possible starting tabletops.

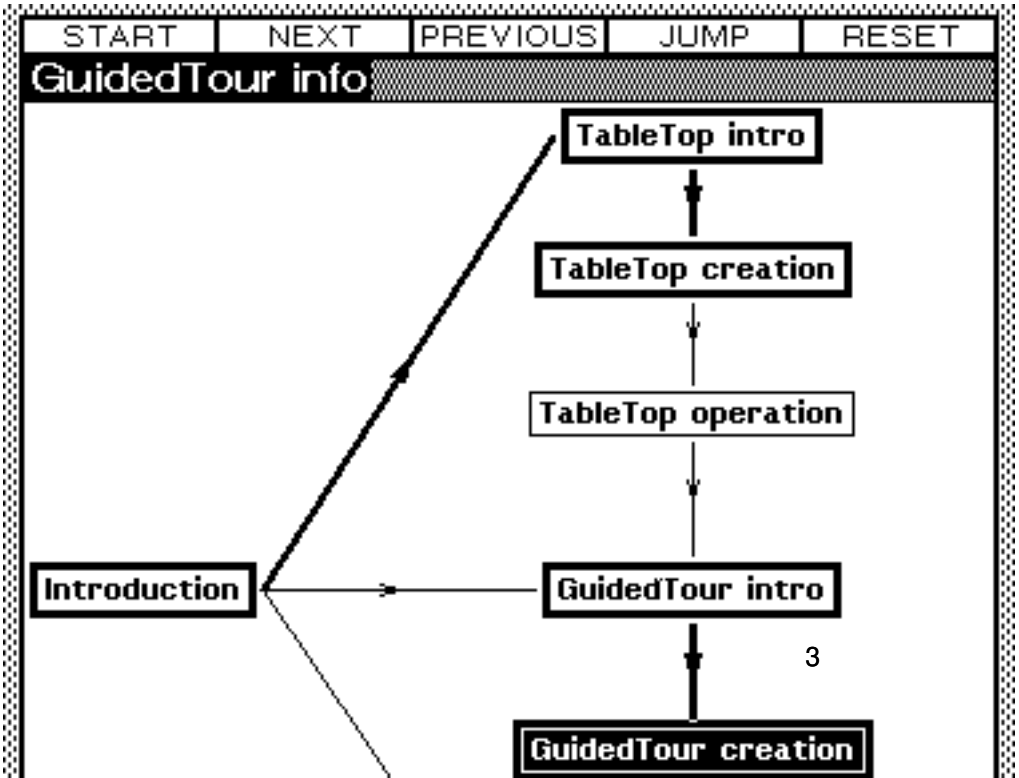
NEXT - This is the standard way to move along a path. Clicking NEXT simply closes down the current tabletop and brings up the next one on the path. If the current node is a branch point having multiple outgoing edges, then the user is asked to choose from a menu of possible next tabletops. In any case, the new current node is highlighted as is the edge that was followed.

PREVIOUS - Clicking here provides the reader with a menu of tabletops already visited. These appear in the order in which they were visited. Clicking on an item in the menu effectively JUMP's to that point in the tour.

JUMP - This operation is used to leave the current path by jumping to an arbitrary tabletop selected by the user from the guided tour graph (again after closing down the current tabletop).

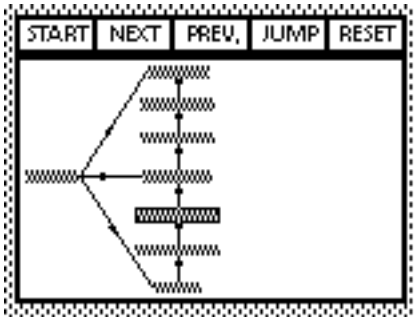
RESET - This operation closes the current tabletop and resets the "state" of the guided tour. In particular, any highlighting of nodes and links in the guided tour graph is turned off and the PREVIOUS list is cleared.

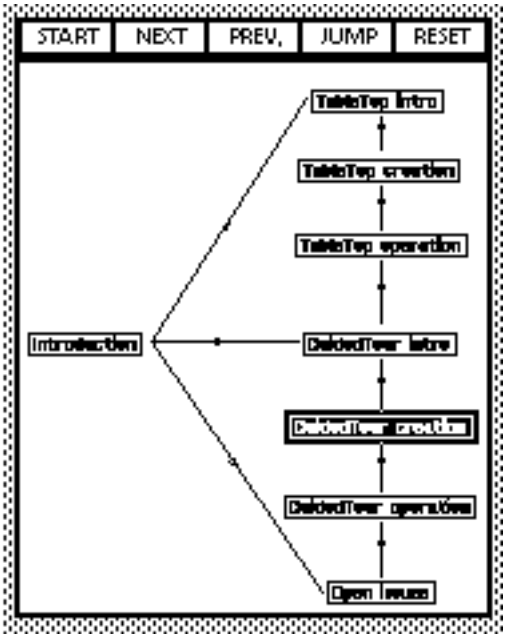
At any point in time, the graph structure displayed in the guided tour card provides various indicators of the reader's place in the tour.



The node in the graph whose tabletop is currently open is highlighted in reverse-video. The nodes for tabletops previously visited are drawn with heavier borders. Finally, the edges corresponding to links followed using the NEXT command are displayed in bold. The guided tour card appearing above shows a point in time when the reader has visited five cards. Starting at "Introduction," the top branch was taken, two tabletops were visited, and then a JUMP was made to "GuidedTour intro" whereupon NEXT was again used to move to the presently displayed tabletop "GuidedTour creation."

This standard mode of operating a guided tour (using the buttons along the top of the card) is extended in several ways. First, because screen space is at a premium and because the window containing the guided tour graph is often large, the user can run the tour from the guided tour window's shrunken icon, which is a miniture version of the guided tour window. This icon may be shaped to suit the user's needs and to fit the available screen space. The current tour stop is highlighted in reverse-video in the shrunken icon.



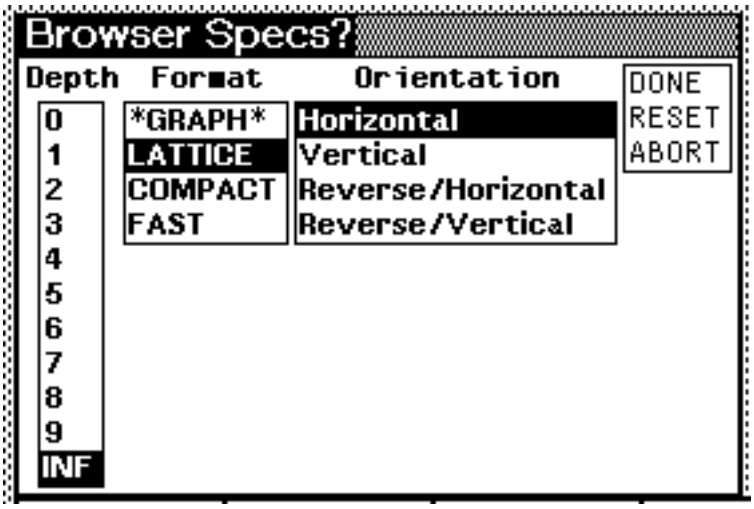

Second, it may be desirable to "circumvent" the tour by peeking at the cards in the next tabletop or by bringing up a tabletop without closing down the current one. This can be done directly by middle-buttoning the link icons appearing in the guided tour graph. (See the NCTableTopCard documentation for a description of tabletop operations.) Finally, it is important to emphasize that you can stray from the tour at any time by following links emanating from cards in the tabletop. Such explorations deeper into the notefile don't affect the current state of the tour.

## CREATING AND MODIFYING GUIDED TOURS

There are two common styles of constructing tours. In the first, authors create an empty guided tour card and build and link together tabletops using the guided tour's editing menu. In the second, authors build a guided tour over already constructed tabletop cards. In any case, the elements of guided tour construction include creating guided tour cards, creating or including tabletops (and other tour nodes), linking together nodes in the tour, and modifying the tour.

*Creating a new guided tour card*

Creating a guided tour card is very much like creating a Browser card. Just choose GuidedTour from the NewCards menu on the notefile icon. If you want the guided tour to be empty, then choose DONE when asked to specify the tour's roots. Otherwise, select a set of starting cards (usually tabletops). Unlike browser cards, guided tour cards don't require you to specify a set of link types to follow. There is a default set of link types (initially the single GuidedTour link type) used by guided tour cards during creation and Recompute. Because of this, the BrowserSpecs stylesheet for guided tour cards doesn't contain the usual columns for link types.




*Adding nodes to a guided tour*

All editing of the guided tour contents is done through the guided tour editing menu.

[illegible]0[illegible]>8| [illegible]F[illegible]0|[illegible]3

▮‘▮‡l▮▮▮▮▮▮F▮▮▮▮▮3>|  ▮▮▮@▮F▮▮▮▮▮▮▮▮▮▮▮▮▮▮@▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮
▮▮▮▮▮▮F▮▮▮▮▮▮▮▮▮▮▮▮▮▮@▮F▮▮▮▮▮▮▮▮▮▮▮▮▮▮@▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮ 08 ⅛l‘ ▮▮
▮▮▮F▮▮

▌08Þ▌l‘▌▌▌@▌F▌▌

[illegible]

▌7l ▌Þ‘▌▌▌@▌F▌▌

▌07l▌Þ'▌▌▌@▌▌▌▌

[illegible]

*Node*s in the guided tour are usually tabletops, but can be arb**itrary card** types. To add an existing card as a node in the guided tour, use the Add Node menu item just as for standard browsers. To create a new tabletop card from scratch, click on the Create TT Card item. You will be asked for cards to be included in the tabletop just as though you created the tabletop from the NewCards menu on the notefile icon. By pulling *across on the Create TT Card menu item, you ca*n create guided tour nodes which are cards of other types. If you use such non-tabletop cards in your tour, you'll find that they generally act similar to tabletop cards that contain only one card. (But see the Handy Hooks section below to change this default behavior.)

Linking together nodes

Nodes in the guided tour can be linked toget**her using the Create** Link menu item. Normally, you will not be asked to specify a link type when creating a link using the editing menu. (If, however, the default set of link types has been changed to have greater than one element, the**n a menu of link types is popped up for you** to select from.) Note also that new links created from the editing menu are always "Global" links.

Modifying the g**uided tour and specifying roots**

**As with standard browser cards, guided tours can be modified by adding new nodes a**nd edges, removing old ones, rearranging nodes spatially within t**he tour, and annotating the tou**r using labels. In addition, it's usually a goo**d idea** t**o specify explicitly what the roots of the tour sh**ould be. (Roots can be (re-)specified by selec**ting t**he Change Browse**r Roots item from th**e middle button title bar menu of the guided tour card.) Other**wise, tour readers clicking on START have to choose fr**om a system-constructed menu of all nodes in the to**ur tha**t don't have inc**oming edges.**

**PROGRA**MMER'S INTERFACE TO GUIDED TOUR CARDS

To create a new guided to**ur card programmatically, call NCP.CreateCard just as you woul**d to create a browser card:

(NCP.CreateCard 'G**uidedT**our NoteFile T**itle NoDisplayFlg** Props ParentFileBoxes ParamList InterestedWindow)

See the documentati**on on NC**P.CreateCard for details on the args.

(NCGT.StartTourAtRoot GTCard)

Equivalent to hitting ST**ART on guided tour card GTCard.**

**(NCGT.NextTourS**top GTCard Don'tCloseCurrentFlg)

Equivalent to **hitti**ng NEXT on guide**d tour card GTCard e**xcept that if Don'tCloseCurrentFlg is non-nil, then current tab**letop won't** *be closed down firs*t.

(NCGT.PreviousTourStop GTCard Don'tCloseCurrentFlg)

Equivalent to hitting PREVIOUS on guided tour card GTCard except that if Don'tCloseCurrentFlg is non-nil, then current tabletop won't be closed down first.

(NCGT.JumpToTourStop GTCard Don'tCloseCurrentFlg NextNode)

Equivalent to hitting JUMP on guided tour card GTCard *except that if Don't*CloseCurrentFlg is non-nil, then current tabletop won't be closed down first. Also if NextNode is a valid node in the graph, then user won't get bugged; we'll jump directly to that card.
(NCGT.ResetTour GTCard Don'tCloseCurrentFlg)
Equivalent to *hitting RESET on* guided tour card GTCard except that if Don'tCloseCurrentFlg is non-nil, then current tabletop won't be closed down first.
Handy Hooks
GuidedTourBringUpFn. If this prop is on a card type's atom, then when cards of that type are encountered in guided tours, they are brought up by calling this function. (This is useful when you don't want the default behavior for stops in the tour that aren't tabletop cards.) This function is expected to return the list of cards it brings up.
GuidedTourCloseDownFn. If this prop is on a card type's atom, then when cards of that type are encountered i*n guided tours, they are* closed down by calling this function. (Note: This prop doesn't seem to be called anywhere, as of the version of October 19, 1988. )
GTNextTourStopFn. This function is called by NCGT.NextTourStop, and its job is to "filter" the list of possible next tour stops from the *current tour stop.* The list of next cards (the destination cards of the NCGT.GuidedTourLinkTypes links from the current tour stop) is passed to the *GTNextTourStopFn, w*hich returns some sublist of the next cards. Both the card type atom of the card in the current tour stop and the car*d type atom of the gu*ided tour card are checked for the GTNextTourStopFn. If both card type atoms have a GTNextTourStopFn, only the one from the current tour stop cardtype is called.
GTCreateGuidedTourLinkFn. If this prop is on the card type atom of the guided tour card, then this function is called whenever a new link is created using the graph edit menu of the guided tour card. This is a way of getting control over *the type of link use*d, whether it's global or local, etc.
GTResetGuidedTourFn. If this prop is on the card type atom of the guided tour card, then this function is called whenever the guided tour is reset.
GTBringUpCachingFn. This is an NCP.CardUserDataProp of a guided tour card whi*ch is called each time a card* or tabletop is brought up.
GTCloseDownCachingFn. This is an NCP.CardUserDataProp of a guided tour card which is called each time a card or tabletop is closed down. If this prop is non-nil, all cards in the guided tour stop which are being closed are merely undisplayed -- they are still cached. It is the job of the **GTBringUpCachingF***n and the GTCloseDownCac***h**ingFn to decide which of these cached but undisplayed tour stop cards should be uncached.
GTCleanUpCachingFn. This is an NCP.CardUserDataProp of a guided tour card which is **calle**d when a guided tour card is closed. Its job is to clean up whatever cached but undisplayed guided tour stop cards are still laying around from the GTBringUpCachingFn and the GTCleanUpCachingFn.
ImmediatelyGoToNextGTStopFn. Whenever a new **guided t**our stop card is brought

up, its card type atom is checked for this function. If the function exists and returns non-NIL, NCGT.NextTourStop is immediately called. Thus, for example, it is possible for a us**er to trave**l through a whole guided tour, by merely pressing a guided tour's Start button.

Global Variables

NCGT.GuidedTourLinkTypes

This variable is initially bound to the single atom GuidedTour. The link types on this list will be the only legal link types users will be able to create in a guid