The \texttt{pmdb} Package

D. P. Story
Email: dpstory@acrotex.net

processed June 7, 2021

Contents

1 Introduction 1

2 The main code 2
  2.1 Package options and package requirements 2
  2.2 Special attention to Thor 3
  2.3 Some switches 3
  2.4 Form field creation 3
    2.4.1 Marginal edit source creation 3
    2.4.2 Marginal checkbox creation 4

3 \texttt{\textbackslash pmInput} states 5
  3.1 Other pushbutton creations 6
  3.2 Defining the \texttt{\textbackslash pmInput} command 7

4 Field JavaScript 8

5 Document JavaScript 10

6 Index 11

7 Change History 12
  1 (*package)

1 Introduction

This package addresses the issue of a poor-man’s database. Educators who use \LaTeX to construct exams and homework sometimes have a collection of problems. Each problem is in its own TEX file. The educator creates a document and \texttt{\textbackslash inputs} several of these packaged questions. This package attempts to provide some “user interface” to the questions and provides a mechanism of selecting which questions are to be included in the document.
What does this package do? For a document that inputs content using the \LaTeX command \texttt{\input}, the same content can be input using the command \texttt{\pmInput} (capitalized). When content is input by \texttt{\pmInput}, a check box is created in the margin at the insertion point of the content. The check boxes so created can be checked or cleared. When the user clicks on push button provided by this package, a list of all selected \texttt{\input} statements are listed in the JavaScript console. This list can then be copied and pasted into another document the author is developing. If you \texttt{Ctrl+Click} on a check box, the associated content is opened in the default TEX editor. In this way, the document author can see a typeset of the content and decide whether the content should be included in the developing document.

The DB stage: When you have a collection of questions (or content) in various files and want to use this package to \texttt{\input} them into your document, the following comments are apropos:

PDF creators: Any PDF creator current in the \LaTeX world is valid for use with this package.

PDF viewers: The ideal viewer is Acrobat, however, Adobe Reader and PDF-XChange Editor can also be used. In the case of Adobe Reader, there is an annoying security dialog box that appears each time you use the \texttt{Ctrl+Click} feature of the check box; the \texttt{Ctrl+Click} feature does not WORK with PDF-XChange Editor.

The production stage: After the document has been assembled (using pmdb), and you build your final document (perhaps an exerquiz quiz), the end user can use an appropriate PDF reader. If an exerquiz quiz is used, then a minimum of Adobe Acrobat Reader is required.

2 The main code

2 \edef\th@dquoteCat{\the\catcode'"}
3 \catcode'"=12\relax
4 \newif\ifpmdbmode \pmdbmodetrue

2.1 Package options and package requirements

dbmode The default option is \texttt{dbmode}. When in effect, check boxes appear in the margins at each \texttt{\pmInput} point.

5 \DeclareOption{dbmode}{\pmdbmodetrue}

!dbmode A convenient way to turn off the creation of the check boxes is to simply place an exclamation point (!) in front of the \texttt{dbmode} option.

6 \DeclareOption{!dbmode}{\pmdbmodefalse}

tight When this option is taken, the checkboxes are tight against the text box area.

7 \newif\ifpmdbtight \pmdbtightfalse
8 \DeclareOption{tight}{\pmdbtighttrue}
The default for the package, the checkboxes are placed to extreme left (or right) in the margins.

```latex
\DeclareOption{!tight}{\pmdbtightfalse}
\ProcessOptions
\RequirePackage{eforms}
```

### 2.2 Special attention to Thor

One motivation for this package is to support the thorshammer package, to that end we make the following assignment, if Thor is not present. This is to prevent stoppage: if you are inputting a \RespBoxEssay question that is accompanied by the \essayQ command, defined in thorshammer.

```latex
\def\pmdb@ckThor{\@ifundefined{essayQ}{\let\essayQ\@gobble}{}}
\AtBeginDocument{\pmdb@ckThor}
```

### 2.3 Some switches

**Some Booleans and counters.** The \ifpmdbFP switch is set to true when the path to the resource is a full path; otherwise, it is set to false. The \ifpmdbDQs switch is set to true of double-quotes are detected; otherwise it is false.

```latex
\newif\ifpmdbFP \pmdbFPfalse
\newif\ifpmdbDQs \pmdbDQsfalse
```

### 2.4 Form field creation

In version dated 2021/01/03, we introduce a button to edit the resource directly in the default application. If can be turned on with \editSourceOn and off again with \editSourceOff. The default is off. These commands can be placed anywhere in the document and affect all subsequent insertions of \cbSelectInput.

```latex
\newif\ifeditSource \editSourcefalse
\def\editSourceOn{\editSourcetrue}
\def\editSourceOff{\editSourcefalse}
\newcount\pmdb@Cnt
```

#### 2.4.1 Marginal edit source creation

First up is the design for the edit pushbutton or link annotation.

```latex
\editSourceBtn[\text{\textbf{options}}]{\text{\textit{(wd)}}}{\text{\textbf{(ht)}}}{\text{\textit{(txt)}}} The button to show the current referenced content in the default application.
\newcommand{\editSourceBtn}[3][\def@editSourceBtn\#1]{\pushButton[\A{/S/Launch/F(\#1)}]{\protect\A#1}{\text{\textbf{#2}}}{\text{\textit{#3}}}}
```

```latex
\editSourceLnk[\text{\textbf{options}}]{\text{(wd)}}{\text{(ht)}}{\text{(txt)}} The link version of the button.
\newcommand{\editSourceLnk}[4][\def@editSourceLnk\#1]{\raisebox{1bp}{\setLinkBbox[\A{/S/Launch/F(\#1)}]{\A#1}{\text{\textbf{#2}}}{\text{\textit{#3}}}{\text{\textbf{#4}}}}}
```

3
Declare the appearance of this push button and link

\useEditBtn \[\TU{View in default editor}\] \{11bp\} \{11bp\}
\useEditLnk \[\linktxtcolor{red}\H{N}\]{11bp}{11bp}{E}

We provide a control for setting the annot used to edit the source: \useEditBtn, a pushbutton is used; \useEditLnk, a link is used. The default is to use a pushbutton.

\def\useEditBtn{\def\@editSourceBorL{\@editSourceBtn}}
\def\useEditLnk{\def\@editSourceBorL{\@editSourceLnk}}
\useEditBtn

2.4.2 Marginal checkbox creation

\cbSelectInput\{\path\} creates a checkbox with a tool tip of \path. The mouse up action \ccBoxMU fixes up relative paths, and defines a Ctrl+Click action. When the checkbox is so clicked, the \path is opened in the default browser. The \path can be relative or absolute. This command is used within \insertCkBx; its \path argument is passed to it from \insertCkBx.
\def\pmCBPresets#1{\def\pm@CBPresets{#1}}
\pmCBPresets{}

Here is the checkbox that appears in the margins and the edit source button ifeditSource is true.
\def\cbSelectInput#1{\mbox{\ checkBox\[\TU{#1}\presets{\pm@CBPresets}\}
\cmd{\bParams{#1}{\the\pmdb@Cnt}\eParams}}
\AAmouseup{\ccBoxMU}{pmdbCkBx.\the\pmdb@Cnt}{11bp}{On}\ifeditSource\olBdry\@editSourceBorL{#1}\fi\global\advance\pmdb@Cnt\@ne\}
\insertCkBx\{\method\} The argument of this macro describes the method of inserting the checkbox. The default definition works well for a straightforward document, where you are inputting ordinary LaTeX code (such as sections or chapters).
\def\insertCkBx#1{\def\@insertCkBx##1{#1}}
\def\pmAlignCB{\ifpmdbtight\if@reversemargin\def\pmAlignCBAlt{\ifodd\value{page}\leavevmode\hfill\else\fi}\else\def\pmAlignCBAlt{\ifodd\value{page}\hfill\else\fi}\fi\else\if@reversemargin\def\pmAlignCBAlt{\ifodd\value{page}\hfil\else\hfil\fi}\else\def\pmAlignCBAlt{\ifodd\value{page}\hfill\else\fi}\fi\fi}
\pmAlignCB
\altCBMargins Placement of checkboxes in the margin. \pmAlignCB controls the margin placement. \altCBMargins alternates the margin placement, forces the checkbox to the extreme left (on odd pages) and extreme right (on even pages).
\def\setCBsMarg{%
\ifpmdbtight
\if@reversemargin
\def\pmAlignCBAlt{\ifodd\value{page}\leavevmode\hfill\else\fi}\else
\\def\pmAlignCBAlt{\ifodd\value{page}\leavevmode\hfill\else\fi}\else
\\def\pmAlignCBAlt{\ifodd\value{page}\hfil\else\hfil\fi}\fi
\else
\if@reversemargin
\\def\pmAlignCBAlt{\ifodd\value{page}\hfil\else\hfil\fi}\else
\\def\pmAlignCBAlt{\ifodd\value{page}\hfil\else\hfil\fi}\else
\fi
\fi
\fi
\fi
This is the default declaration. It works well when you are inputting content that goes into horizontal mode. We insert the check box at the beginning of the first paragraph. When you are inputting files that come into a list environment, this method does not work satisfactory.

3 \textit{pmInput states}

The switch \texttt{ifqzInput} is set to true when \texttt{InputQuizItems} is expanded; otherwise, it is set to false.

\begin{verbatim}
\newif\ifqzInput \qzInputfalse
\def\InputParas{\global\qzInputfalse
\insertCkBx{\ifpmdbmode
\everypar{\marginpar{\pmAlignCB\cbSelectInput{##1}}\global\everypar{}}\fi}}
\InputParas
\def\InputPros{\global\qzInputtrue
\let\pmHook\pmHook@qzItems
\saveQNo\z@
\insertCkBx{\def\cbInQzMargin{\cbSelectInput{##1}}}%}
\end{verbatim}

Set the default to be \texttt{InputParas}

\begin{verbatim}
\InputParas
\def\InputPros{\global\qzInputtrue
\let\pmHook\pmHook@qzItems
\saveQNo\z@
\insertCkBx{\def\cbInQzMargin{\cbSelectInput{##1}}}%}
\end{verbatim}

The \texttt{InputPros} is a command that declares the next \texttt{pmInput} is for questions for an \texttt{eqexam} document. Currently this command is still under development.

\begin{verbatim}
\def\InputPros{\global\qzInputfalse
\insertCkBx{\ifpmdbmode
\def\prior@questionsHook{\marginpar{\pmAlignCB
\cbSelectInput{##1}}\global\everypar{}}\fi}}
\end{verbatim}

The \texttt{InputPros} is a command that declares the next \texttt{pmInput} is for items in a \texttt{quiz} environment of \texttt{exerquiz}

\begin{verbatim}
\newcount\saveQNo \saveQNo\z@
\def\pmHook@qzItems{\let\item@pmOld\item
\def\item@pmNew{\item@pmOld\itemhook\let\item\item@pmOld}%
\let\item\item@pmNew}
\def\InputQuizItems{\global\qzInputtrue
\let\pmHook\pmHook@qzItems
\saveQNo\z@
\insertCkBx{\let\item@pmOld\item
\let\item@pmNew{\item@pmOld\itemhook\let\item\item@pmOld}%
\saveQNo\z@
\insertCkBx{\def\cbInQzMargin{\cbSelectInput{##1}}}%}
\end{verbatim}
This command declares that the next \pmInput macros are for items in an list environment 2019/12/09 v0.4

\def\pmHook@item{\let\item@pmOld\item
\def\item@pmNew{\ifx\pmiarg\@empty
\ifx\pm@Brk\ef@YES
\def\pm@next{\item@pmOld\[]}
\else
\let\pm@next\item@pmOld
\fi
\else
\def\pm@next{\item@pmOld\[\pmiarg\]}%
\fi\pm@next\itemhook\let\item\item@pmOld}%
\let\item\item@pmNew

\def\ItemHook#1{\def\itemhook{#1}}
\def\InputItems{\global\qzInputfalse
\let\pmHook\pmHook@item
\insertCkBx{\def\cbInQzMargin{\cbSelectInput{##1}}}
\ItemHook{\leavevmode\ifpmdbmode
\marginpar{\pmAlignCB\cbInQzMargin}\fi}}

Place checkbox and input \langle path \rangle \langle path \rangle
\ckBxInput{(path)} Places the check box and inputs the \langle path \rangle.
\let\pmHook\relax
\def\ckBxInput#1{\@insertCkBx{#1}\
\ifpmdbDQs\def\donext{\pmHook\input{"#1"}}\else
\def\donext{\pmHook\input{#1}}\fi\donext}

3.1 Other pushbutton creations

This package provides two form fields that are used for the DB step.
\displayChoices[options]{(wd)}{(ht)} inserts a push button whose action is to display the selections in the console window. The argument \langle wd \rangle can be empty, in which case, the width of the field is determined from the \CA key.
\newcommand{\displayChoices}[3][{}]{\pushButton{\TU{\displayChoice@TU}\CA{\displayChoice@CA}#1}A\mouseup{\sldInputs}A\protect\AA}
\{\sldInputs\}{#2}{#3}

Easy access to the \CA and \TU keys, for language localization, are provided. These are \displayChoiceCA{(text)} and \displayChoiceTU{(text)}.
\clrChoices[{\textit{options}}]{{\textit{wd}}}{{\textit{ht}}} inserts a push button whose action is to clear all check boxes (and underlying JavaScript variables) created by this package. The argument \textit{(wd)} can be empty, in which case, the width of the field is determined from the \texttt{CA} key.

\begin{verbatim}
\newcommand{\clrChoices}[3]{\pushButton{\clrChoices@TU\CA{\clrChoices@CA}{#1}\AA\mouseup{\clrAction}\protect\AA\{\Inputs\}{#2}{#3}}}
\clrChoicesCA{Clear Choices}
\clrChoicesTU{Clear all check boxes created by pmdb}
\end{verbatim}

### 3.2 Defining the \texttt{pmInput} command

\texttt{pmInput\star[\langle\textit{arg}\rangle]\langle\textit{path}\rangle} is the main user-interface for inputting a file; here, the macro’s name is \texttt{pmInput}, ultimately it calls \texttt{input} with the same path. Paths with spaces must be enclosed in double quotes (\texttt{pmInput\{my cool problem.tex\}}) and the extensions must always be used. It the \texttt{\star} option is specified, \texttt{pmInput} gobbles up all remaining arguments and does nothing otherwise. The optional argument \texttt{\langle\textit{arg}\rangle} is only obeyed when \texttt{\InputItems} is in affect; it is passed on to the underlying \texttt{\item}, as in \texttt{\item[\langle\textit{arg}\rangle]}. The \texttt{\langle\textit{path}\rangle} argument is the path to the content to be input; if the \texttt{\langle\textit{path}\rangle} contains spaces, then the path must be enclosed in double quotes ("my cool path/file.tex").

\begin{verbatim}
\newcommand\@gobbleOR[2]{}
\def\pmInput{\@ifstar{\@gobbleOR}{\pmInput@i}}
\def\pmInput@i{\@ifnextchar\[%\]{\let\pm@Brk\ef@YES\inputConta}{\let\pm@Brk\ef@NO\inputConta}}
\let\pm@Brk\ef@NO
\def\inputConta{\bgroup\@makeother"\inputContb}
\newcommand\inputContb[2]{\egroup\def\pmiarg{#1}\inputConti#2;;}
\def\inputConti{\@ifnextchar"\{%\global\pmdbDQstrue\removedqs\}{\global\pmdbDQsfalse\removesemis}}
\def\removedqs"#1";;{\inputContii{#1}}
\def\removesemis#1;;{\inputContii{#1}}
\end{verbatim}

Determine if double quotation marks are used.

\begin{verbatim}
\def\inputContii#1{\isItFullPath#1:@nil\doinput{#1}}
\end{verbatim}

Determine if this is a full path, we do this by searching for a colon (\texttt{\colon}). Following the search for the colon, pass on to the final step of \texttt{\doinput}.

\begin{verbatim}
\def\inputContii#1{\isItFullPath#1:@nil\doinput{#1}}
\end{verbatim}
A command to detect presence of a colon.
135 \def\isItFullPath#1:#2\@nil{%  
136 \def\@rgii{#2}\ifx\@rgii\@empty  
137 \global\pmdbFPfalse\else\global\pmdbFPtrue\fi}

Final step, if the switch \ifpmdbmode is true, we insert the check box \ckBxInput;  
otherwise, we pass \langle path \rangle to \input.
139 \def\doinput#1{\ifpmdbmode\def\donext{\ckBxInput{#1}}\else  
140 \ifpmdbDQs\def\donext{\input{"#1"}}\else  
141 \def\donext{\input{#1}}\fi\fi \donext}

During the development of this package, the original command name used was \Input. There are a few users that use this old definition; the command \Input is defined in other package, in particular in the srcltx. So we allow the use of \Input if \Input is not otherwise defined.
143 \def\pmInputWarni{\PackageWarningNoLine{pmdb}{The command \string\Input \space is already defined.\MessageBreak  
144 \string\Input \space is already defined.\MessageBreak}}
145 \def\pmInputWarnii{\PackageWarningNoLine{pmdb}{Letting \string\Input \space to \string\pmInput. You are \MessageBreak  
146 \string\Input \space to \string\pmInput. You are \MessageBreak encouraged to use the supported\MessageBreak  
147 \string\Input \space instead}}} \def\pmInputChk\@ifundefined{Input}{\let\Input\pmInput  
152 \pmInputWarnii}{\pmInputWarni}} \AtBeginDocument{\pmInputChk}

4 Field JavaScript
\ccBoxMU This is the JavaScript action of the check box, used in \cbSelectInput This code uses two parameters \p(1)=\langle path \rangle and \p(2)=\langle cnt \rangle (the\pmdb@Cnt).
154 \begin{defineJS}\makeesc\@\makecmt\%}{\ccBoxMU}
155 @ifpmdbFP%  
156 event.target.userName="@p(1)";  
157 @else%  

This part of the code is Windows specific. Don’t know enough about Mac OS to form the proper path.
158 // device independent path  
159 var pos=this.path.lastIndexOf("/");  
160 var thispath=this.path.substring(0,pos+1);  
159 \<drive>/user/documents/.../myfolder/  
161 pos=thispath.indexOf("/",1);  
162 var drive=thispath.substring(1,pos);  
163 var platform=app.platform;
Windows platform

```javascript
if (platform=="WIN")

    <drive>:\user\documents/...\myfolder/

thispath=drive+":"+thispath.substring(pos+1);

MacOS platform: I'm not familiar with the MacOS file system, so we'll just assume it is the same as with Windows.

```javascript
if (platform=="MAC")

    <drive>/user/documents/.../myfolder/

```javascript
thispath=drive+"/"+thispath.substring(pos+1);

```javascript
event.target.userName=thispath+("@p(1)");@fi

```javascript
if (event.modifier){

  aebTrustedFunctions(this,aebLaunchURL,%

  {cURL: "file:///+event.target.userName});

  try {

    catch(e) {

      console.show();

      console.println("The Ctrl+Click action is not supported, %

      installation of aeb\_pro.js or aeb-reader.js is required.");

    }

    (2020/04/29) Add a shift event for quizzes.

  }

```javascript
else {

    this.gotoNamedDest("@currQuiz.+@p(2)+1");@fi

```javascript
else {

    if (event.target.isBoxChecked(0)){

      _oSPaths["pmdbCkBx.@p(2)"]=%

      ["@p(1)",@ifpmdbDQs true@else false@fi];

      _aInputs[@p(2)]=true;

      _numInputs++;

    }else{

      _oSPaths["pmdbCkBx.@p(2)"]=null;

      _aInputs[@p(2)]=false;

      _numInputs--;

    }

    event.target["_boxState"]=!event.target.isBoxChecked(0);

  }

```

```javascript
\end{defineJS}

\sldInputs This is the mouse up action for a push button. It lists all selected content and displays them in the console window of Acrobat/Reader, used in \displayChoices.
```

\begin{defineJS}[\makecomment]{\sldInputs}
console.clear();console.show();
if (_numInputs==0) console.println("No inputs selected");
else {

```
```
for(var i=0;i<_aInputs.length;i++){
    if (!/_aInputs[i]){
        if (_oSPaths["pmdbCkBx."+i][1])
            console.println(\\input\\{"+_oSPaths["pmdbCkBx."+i][0]%206 +""}\\{"});
        else
            console.println(\\input\\{+_oSPaths["pmdbCkBx."+i][0]%209 +""}\\{"});
    }
}

clrAction Mouse up action to clear the checkboxes and to re-initialize internal JS variables. Used in clrChoices.

var _oSPaths=new Object;
var _aInputs=new Array;
var _numInputs=0;
this.resetForm("pmdbCkBx");

5 Document JavaScript

var _oSPaths=new Object;
var _aInputs=new Array;
var _numInputs=0;
this.resetForm("pmdbCkBx");
### 6 Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers in roman refer to the code line of the definition; numbers underlined refer to the code lines where the entry is used.

**Symbols**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>%</td>
<td>154, 198</td>
</tr>
<tr>
<td>@editSourceBorL</td>
<td>28, 29, 36</td>
</tr>
<tr>
<td>@editSourceBtn</td>
<td>20, 28</td>
</tr>
<tr>
<td>@editSourceLnk</td>
<td>23, 29</td>
</tr>
<tr>
<td>@gobbleOR</td>
<td>121, 122</td>
</tr>
<tr>
<td>@insertCkBx</td>
<td>39, 103</td>
</tr>
<tr>
<td>@makeother</td>
<td>127</td>
</tr>
<tr>
<td>@rgii</td>
<td>136</td>
</tr>
<tr>
<td>!dbmode (option)</td>
<td>2</td>
</tr>
<tr>
<td>!tight (option)</td>
<td>3</td>
</tr>
<tr>
<td>{</td>
<td>205, 208</td>
</tr>
<tr>
<td>}</td>
<td>206, 209</td>
</tr>
<tr>
<td>_</td>
<td>177</td>
</tr>
</tbody>
</table>

**A**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>\A</td>
<td>21, 24, 25</td>
</tr>
<tr>
<td>\AA</td>
<td>108, 115</td>
</tr>
<tr>
<td>\AMouseup</td>
<td>35, 108, 115</td>
</tr>
<tr>
<td>@altCBMargins</td>
<td>4, 52, 60</td>
</tr>
<tr>
<td>@AtBeginDocument</td>
<td>13, 60, 153</td>
</tr>
</tbody>
</table>

**B**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>\bParams</td>
<td>34</td>
</tr>
</tbody>
</table>

**C**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>\CA</td>
<td>108, 115</td>
</tr>
<tr>
<td>\cbInQ2Margin</td>
<td>79, 82, 99, 101</td>
</tr>
<tr>
<td>\cbSelectInput</td>
<td>4, 33, 65, 70, 79, 99</td>
</tr>
<tr>
<td>\ccBoxMU</td>
<td>8, 35, 154</td>
</tr>
<tr>
<td>\checkBox</td>
<td>33</td>
</tr>
<tr>
<td>\ckBoxInput</td>
<td>6, 103, 139</td>
</tr>
<tr>
<td>\clrAction</td>
<td>10, 115, 214</td>
</tr>
<tr>
<td>\clrChoices</td>
<td>7, 114</td>
</tr>
<tr>
<td>\clrChoices@CA</td>
<td>115, 117</td>
</tr>
<tr>
<td>\clrChoices@TU</td>
<td>114, 118</td>
</tr>
<tr>
<td>\clrChoicesCA</td>
<td>7, 117, 119</td>
</tr>
<tr>
<td>\clrChoicesTU</td>
<td>7, 118, 120</td>
</tr>
<tr>
<td>\cmd</td>
<td>34</td>
</tr>
</tbody>
</table>

**D**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>dbmode (option)</td>
<td>2</td>
</tr>
<tr>
<td>\DeclareOption</td>
<td>5, 6, 8, 9</td>
</tr>
<tr>
<td>\displayChoice@CA</td>
<td>108, 110</td>
</tr>
<tr>
<td>\displayChoice@TU</td>
<td>107, 111</td>
</tr>
<tr>
<td>\displayChoiceCA</td>
<td>6, 110, 112</td>
</tr>
<tr>
<td>\displayChoices</td>
<td>6, 107</td>
</tr>
<tr>
<td>\displayChoiceTU</td>
<td>6, 111, 113</td>
</tr>
<tr>
<td>\doinput</td>
<td>134, 139</td>
</tr>
<tr>
<td>\donext</td>
<td>104–106, 139–142</td>
</tr>
<tr>
<td>\editSourceBtn</td>
<td>20, 23</td>
</tr>
<tr>
<td>\editSourcefalse</td>
<td>16, 18</td>
</tr>
<tr>
<td>\editSourceLnk</td>
<td>23, 27</td>
</tr>
<tr>
<td>\editSourceOff</td>
<td>3, 18</td>
</tr>
<tr>
<td>\editSourceOn</td>
<td>3, 17</td>
</tr>
<tr>
<td>\editSourcetrue</td>
<td>17</td>
</tr>
<tr>
<td>\efNO</td>
<td>125, 126</td>
</tr>
<tr>
<td>\efYES</td>
<td>87, 124</td>
</tr>
<tr>
<td>\egroup</td>
<td>128</td>
</tr>
<tr>
<td>\eParams</td>
<td>34</td>
</tr>
<tr>
<td>\essayQ</td>
<td>12</td>
</tr>
<tr>
<td>\everypar</td>
<td>64, 65</td>
</tr>
</tbody>
</table>

**E**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>\ifreversemargin</td>
<td>42, 47, 54</td>
</tr>
<tr>
<td>\ifeditSource</td>
<td>16, 36</td>
</tr>
<tr>
<td>\ifpmdbDQs</td>
<td>15, 104, 140</td>
</tr>
<tr>
<td>\ifpmdbFP</td>
<td>14</td>
</tr>
<tr>
<td>\ifpmdbmode</td>
<td>4, 63, 68, 80, 100, 139</td>
</tr>
<tr>
<td>\ifpmdbtight</td>
<td>7, 41, 55, 57</td>
</tr>
<tr>
<td>\ifquizInput</td>
<td>61</td>
</tr>
<tr>
<td>\input</td>
<td>144, 148, 151</td>
</tr>
<tr>
<td>\inputConta</td>
<td>104, 105, 140, 141</td>
</tr>
<tr>
<td>\inputContb</td>
<td>124, 125, 127</td>
</tr>
<tr>
<td>\inputConti</td>
<td>127, 128</td>
</tr>
<tr>
<td>\inputContii</td>
<td>128, 129</td>
</tr>
<tr>
<td>\inputItems</td>
<td>132–134</td>
</tr>
<tr>
<td>\inputParas</td>
<td>6, 97</td>
</tr>
<tr>
<td>\inputProbs</td>
<td>5, 67</td>
</tr>
<tr>
<td>\inputQuizItems</td>
<td>5, 76</td>
</tr>
<tr>
<td>\insertCkBx</td>
<td>4, 39, 63, 68, 79, 99</td>
</tr>
<tr>
<td>\item</td>
<td>73–75, 84, 93, 94</td>
</tr>
<tr>
<td>\item@pmNew</td>
<td>74, 75, 85, 94</td>
</tr>
</tbody>
</table>
7 Change History

v0.4 (2019/12/09)
  General: Added \InputItems ..................... 6
  Modified to work when no points are specified . 5
v0.5 (2020/04/29)
  General: Added event.shift for quizzes ........ 9
v0.6 (2021/01/03)
  General: Added \editSourceOn and
  \editSourceOff ............................. 3
v1.0 (2021/06/07)
  General: Upgrading v0.6 to v1.0 for publication . 2