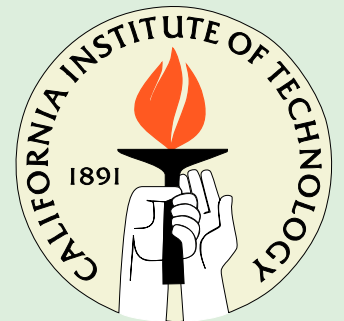


Adobe 'Marked Objects' plugin for WaRMreader

*Ross Moore, Wendy McKay
Thomas Ruark*

*Macquarie University, Sydney
CalTech, Pasadena
Adobe Systems, San Jose*



Outline: .bb file

- extension of X_Y-pic's `\xyimport` command

Outline: .bb file

- extension of X_Y-pic's `\xyimport` command
- facilitates placement of labels over imported graphics

Outline: .bb file

- extension of X_Y-pic's `\xyimport` command
- facilitates placement of labels over imported graphics
- works by having knowledge of the location of interesting places in the imported graphic, stored within a separate file in an easily readable text-based format.

Outline: `.bb` file

- extension of X_Y-pic's `\xyimport` command
- facilitates placement of labels over imported graphics
- works by having knowledge of the location of interesting places in the imported graphic, stored within a separate file in an easily readable text-based format.

Since this file must also contain the Bounding Box information for the graphic, we refer to it as the `.bb` file, even though filename extensions other than `.bb` can (and should) be used, according to the format of the information within the file.

Outline: .bb file

- extension of X_Y-pic's `\xyimport` command
- facilitates placement of labels over imported graphics
- works by having knowledge of the location of interesting places in the imported graphic, stored within a separate file in an easily readable text-based format.

Since this file must also contain the Bounding Box information for the graphic, we refer to it as the `.bb` file, even though filename extensions other than `.bb` can (and should) be used, according to the format of the information within the file.

There's now a plugin module for Adobe's *Illustrator* [2] software that provides an intuitive windowed interface for easy creation of a `.bb` file, by inserting 'Marked Points' and associated information.

Using WaRMreader macros

Before showing how the plugin works within *Illustrator*, we first look at motivation for not including labels within the image itself,

Using WaRMreader macros

Before showing how the plugin works within *Illustrator*, we first look at motivation for not including labels within the image itself, and revisit how the information is used within a L^AT_EX document for placing nicely typeset labels over an imported graphic image.

Using WaRMreader macros

Before showing how the plugin works within *Illustrator*, we first look at motivation for not including labels within the image itself, and revisit how the information is used within a L^AT_EX document for placing nicely typeset labels over an imported graphic image. This will help in understanding the nature of the information that needs to be provided within the .bb file, and consequently gives a motivation for the operations that can be performed with the new plugin tool and its associated windows and dialogs.

Using WaRMreader macros

Before showing how the plugin works within *Illustrator*, we first look at motivation for not including labels within the image itself, and revisit how the information is used within a L^AT_EX document for placing nicely typeset labels over an imported graphic image.

This will help in understanding the nature of the information that needs to be provided within the .bb file, and consequently gives a motivation for the operations that can be performed with the new plugin tool and its associated windows and dialogs.

Study this [composite example](#).

Using WaRMreader macros

Before showing how the plugin works within *Illustrator*, we first look at motivation for not including labels within the image itself, and revisit how the information is used within a L^AT_EX document for placing nicely typeset labels over an imported graphic image. This will help in understanding the nature of the information that needs to be provided within the .bb file, and consequently gives a motivation for the operations that can be performed with the new plugin tool and its associated windows and dialogs.

Study this [composite example](#).

- Image with no labels.
- Example of a .bb file.
- Image with marked points.
- Image with mathematical labels.
- Example of **WaRMreader** coding.

no labels in image

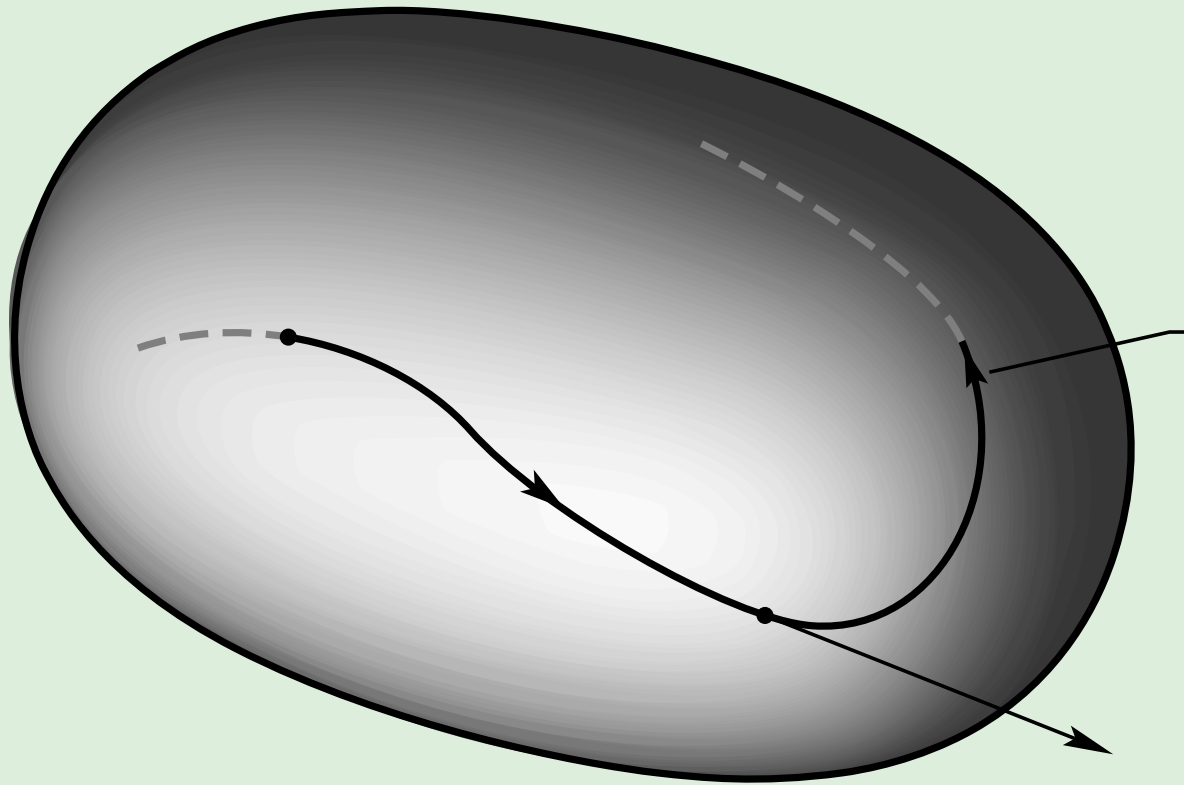


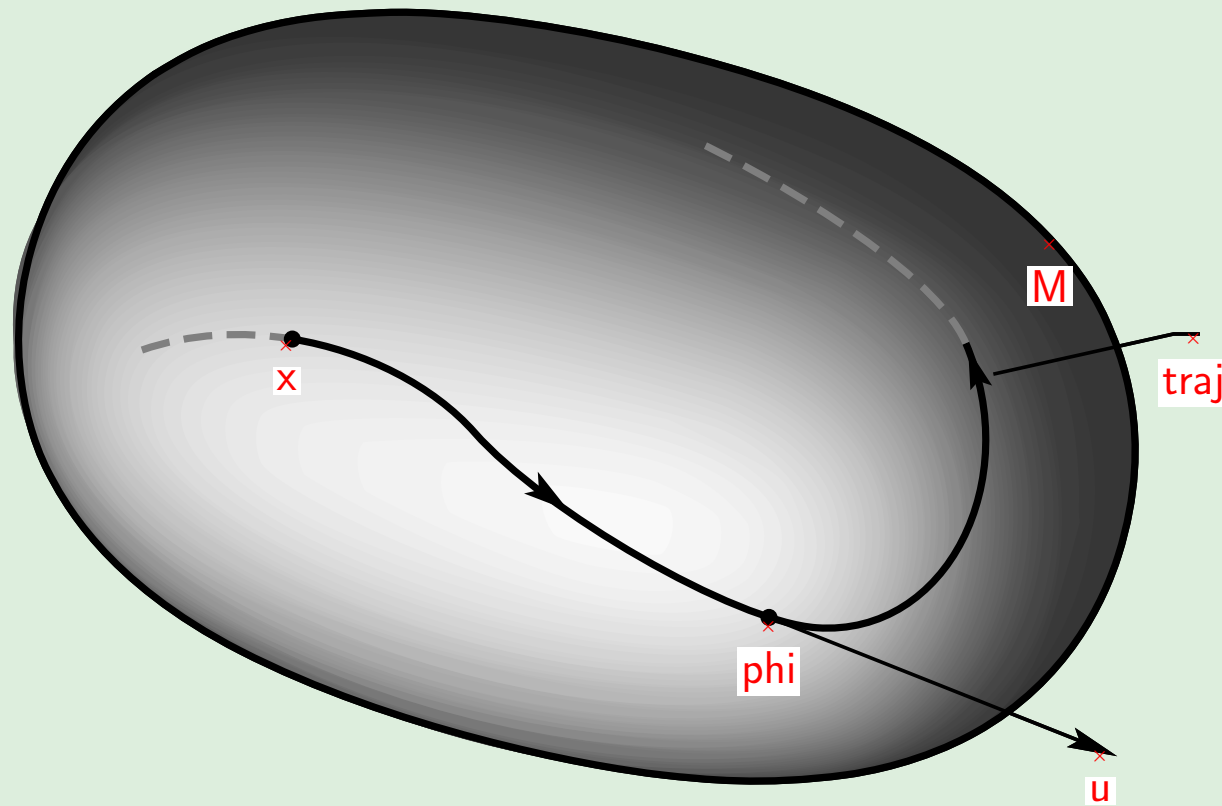
Image created with Adobe's *Illustrator* program, with no visible labels. Such labels will be added later, when the image is included within a L^AT_EX document.

contents of .bb file

```
%%Creator:Adobe Illustrator 9.0, Macintosh 2000
%%Title: Fig8_2_1.eps
%%Date: 7/6/2001 7:16:6 PM
%%BoundingBox: 0 0 165 108
%%Coordinates: LL
%%StartMarkedPoints
%%MarkedPoint:(144,75) : point(0,0) : M %M, manifold
%%MarkedPoint:(164,62) : point(0,0) : traj %trajectory of\\fluid partic
%%MarkedPoint:(151,4) : point(0,0) : u %u(\\varphi_t(x),t)
%%MarkedPoint:(105,22) : point(0,0) : phi %\\varphi_t(x)
%%MarkedPoint:(38,61) : point(0,0) : x %x
%%EndMarkedPoints
```

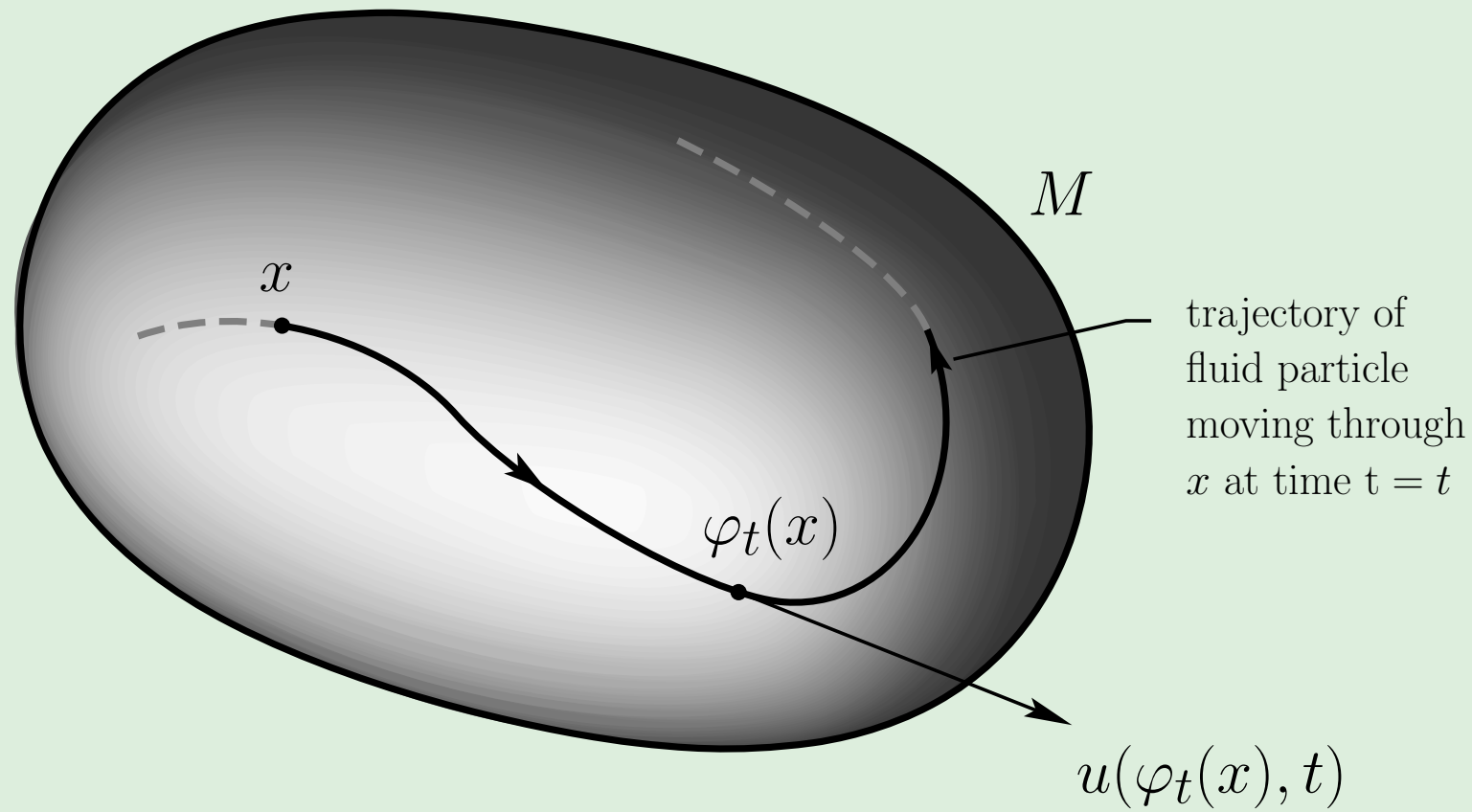
Contents of the file `examp1.bb` defining the ‘marked points’ within the imported image.

the 'marked' points



This shows the locations of 'marked points' whose identifying names and coordinates have been read from the `examp1.bb`.

mathematical labels



Imported image, fully annotated with meaningful labels, positioned using intuitive X_y-pic/WaRMreader notation.

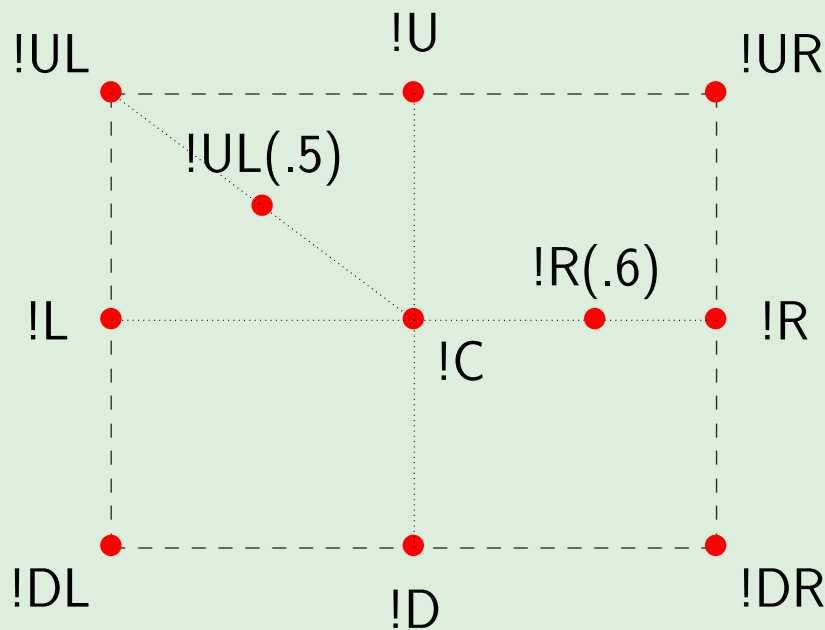
WaRMreader code

Placement of mathematical labels over the **unlabelled image**.

```
\begin{center}
\def\Fname{examp1}%
\renewcommand{\xyWARMinclude}[1]{%
  \includegraphics[scale=.95]{#1}}
\begin{xy}
  \xyWARMprocessMoEPS{\Fname}{pdf}
  \xyMarkedImport{\Fname}% repeat name for pdfTeX
  \xyMarkedPos{x}*++!D{x}
  \xyMarkedPos{phi}*++!D!L(.3){\varphi_t(x)}
  \xyMarkedPos{M}*+!LD{M}
  \xyMarkedPos{u}*+!UL{u(\varphi_t(x),t)}
  \xyMarkedPos{traj}*++!L!U(.5){\parbox{6em}%
    {\scriptsize trajectory of\\fluid particle\\
    moving through\\$x$ at time $\mathrm{t}=t$}}
\end{xy}
\end{center}
```


Xy-pic modifiers

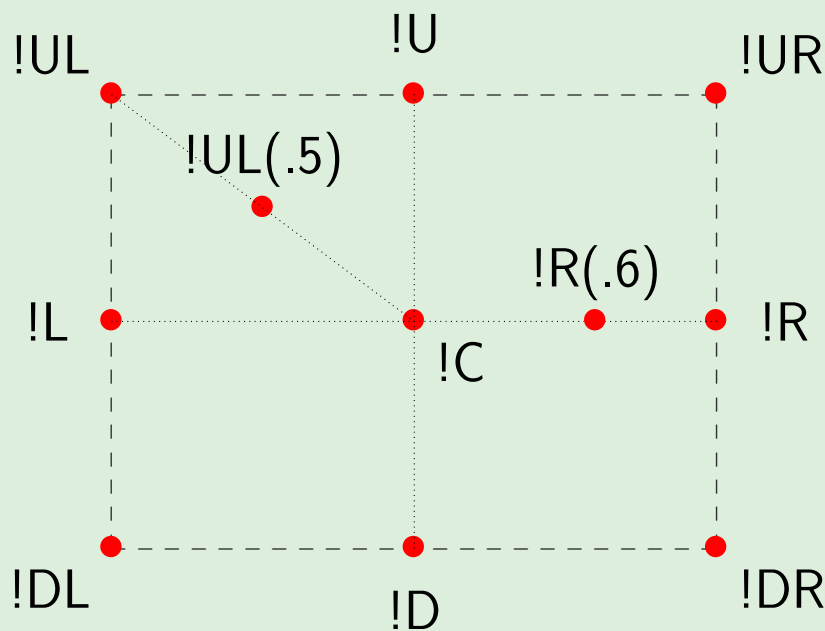
Placement of the labels is best done using a `\xyMarkedPos` command, defined in `warmread.sty`, for each marked-point using its identifier.



Detailed positioning is achieved using short Xy-pic ‘modifiers’ for the center (`!C`), edges (`!L`, `!R`, `!U`, `!D`), and corners (`!UL`, `!UR`, etc.). Arbitrary locations (`!R(.6)` etc.), can be used to specify the anchor point for positioning of an object over a marked point.

Xy-pic modifiers

Placement of the labels is best done using a `\xyMarkedPos` command, defined in `warmread.sty`, for each marked-point using its identifier.



Detailed positioning is achieved using short Xy-pic ‘modifiers’ for the center (`!C`), edges (`!L`, `!R`, `!U`, `!D`), and corners (`!UL`, `!UR`, etc.). Arbitrary locations (`!R(.6)` etc.), can be used to specify the anchor point for positioning of an object over a marked point. (For more details, consult the Xy-pic Reference Manual [5].)

Changing styles

Be aware that the **WaRMreader** [4] macros provide alternative ways to position labels. But the above method using Xy-pic modifiers is the most flexible for ensuring that the labels can be easily positioned over parts of the image which are both near to the feature being labelled, and are clear of other prominent features.

Changing styles

Be aware that the **WaRMreader** [4] macros provide alternative ways to position labels. But the above method using Xy-pic modifiers is the most flexible for ensuring that the labels can be easily positioned over parts of the image which are both near to the feature being labelled, and are clear of other prominent features.

Using different styles is then a matter of changing just L^AT_EX styles, and perhaps an Xy-pic style macro; e.g.

```
\def\Fname{examp1}%  
\mathversion{bold}\LARGE\bfseries  
\renewcommand{\xyWARMinclude}[1]{%  
  \includegraphics[scale=1.3]{#1}}%  
\begin{xy}  
  ...
```

Advantages

The primary advantages of using this method to label graphics are as follows.

Advantages

The primary advantages of using this method to label graphics are as follows.

- The style, size, and font-face of labels placed over imported images necessarily matches that used in the surrounding document.

Advantages

The primary advantages of using this method to label graphics are as follows.

- The style, size, and font-face of labels placed over imported images necessarily matches that used in the surrounding document.
- Since labels are added within the L^AT_EX manuscript body, the wording of labels can be changed without need to alter the underlying graphic in any way.

Advantages

The primary advantages of using this method to label graphics are as follows.

- The style, size, and font-face of labels placed over imported images necessarily matches that used in the surrounding document.
- Since labels are added within the L^AT_EX manuscript body, the wording of labels can be changed without need to alter the underlying graphic in any way.
- The same image can be used many times over; in various settings, using different sizes, styles or text-faces for the labels. There is no need to make any edits at all within the image file itself.

Avoids font issues

A subsidiary advantage is that the method of labelling is unaffected by compatibility problems in font technologies within the software used to create the graphic images.

Avoids font issues

A subsidiary advantage is that the method of labelling is unaffected by compatibility problems in font technologies within the software used to create the graphic images.

For example, there are known problems with the use of Computer Modern fonts in artwork created with earlier versions of *Illustrator*. When such images are viewed in recent versions of Adobe software, either font substitutions occur, or some glyphs may fail to appear.

Avoids font issues

A subsidiary advantage is that the method of labelling is unaffected by compatibility problems in font technologies within the software used to create the graphic images.

For example, there are known problems with the use of Computer Modern fonts in artwork created with earlier versions of *Illustrator*. When such images are viewed in recent versions of Adobe software, either font substitutions occur, or some glyphs may fail to appear. With the **WaRMreader** method such problems become irrelevant, since no text fonts are needed at all within the imported images.

‘Marked Objects’ Plugin Tool

⟨switch to Adobe *Illustrator* demonstration⟩

Getting help from Adobe

When Wendy first approached Adobe with the proposal for a plugin to create .bb files, Tom requested that she provide a work-flow for how the module would be used.

- Title page of proposal for the plug-in.
- Work-flow for using the proposed plug-in.
- Proposed interface changes for accessing the MO-pen and viewing 'marked object' information.
- Further proposed interface changes for using the proposed plug-in.

Bibliography

References

- [1] Abraham, R., J.E. Marsden and T. Ratiu; “Manifolds, Tensor Analysis and Applications”. Applied Mathematical Sciences Series, Vol. 75; Springer–Verlag, New York, (1988) 654pp.
- [2] Adobe Systems Inc.; Adobe *Illustrator* 9 (latest version 9.02); purchase online from <http://www.adobe.com/products/illustrator/main.html>.
- [3] Moore, Ross and Wendy McKay; “Convenient Labelling of Graphics, the WARMreader Way”. TUGboat Vol. 20 (3), 1999; pp. 262–271.
- [4] Moore, Ross; “What is **WaRMreader?**”, examples, documentation and macros for warmread.sty; available online at <http://www-texdev.mpce.mq.edu.au/WARM/> and <http://cds.caltech.edu/~wgm/WARM/reader2001.html>.

Bibliography

- [5] Rose K. and R. Moore; Xy-pic Reference Manual; accompanies the Xy-pic diagram macros for T_EX; version 3.7 available from [CTAN::/macros/generic/diagrams/xypic/](http://CTAN::macros/generic/diagrams/xypic/).