

# Real Life Perl

## Glueing the Pieces Together

Karl "CountZero" Moens

FOSDEM 2013

2<sup>nd</sup> of February 2013



Karl  
"CountZero"  
Moens

I am a lawyer, working in an insurance broker's office.



So my life would be very dull and boring if it was not for:

My Thai girlfriend      and Perl

Who am I?

The Problem

The Solution

L<sup>A</sup>T<sub>E</sub>X

Solution

Lessons  
learned

Karl  
"CountZero"  
Moens

I am a lawyer, working in an insurance broker's office.



So my life would be very dull and boring if it was not for:

My Thai girlfriend

and Perl

Who am I?

The Problem

The Solution

L<sup>A</sup>T<sub>E</sub>X

Solution

Lessons  
learned

Karl  
"CountZero"  
Moens

I am a lawyer, working in an insurance broker's office.



So my life would be very dull and boring if it was not for:

My Thai girlfriend

and Perl



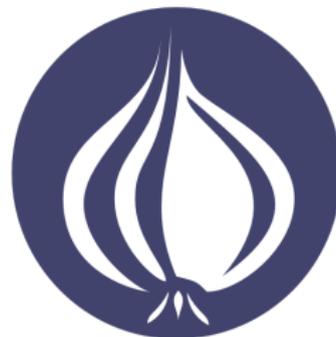
I am a lawyer, working in an insurance broker's office.



So my life would be very dull and boring if it was not for:

My Thai girlfriend

and Perl



Today we will speak about Perl.

Those who are only interested in Thai girlfriends may now quietly leave the room.

Karl  
"CountZero"  
Moens

Who am I?

The Problem

The Solution

L<sup>A</sup>T<sub>E</sub>X

Solution

Lessons  
learned

Today we will speak about Perl.

Those who are only interested in Thai girlfriends may now quietly leave the room.



Our insurance broker's office manages the insurance of many different fleets of ships.

This involves a lot of administration:

- Issuing insurance policies
- Providing "Certificates of Cover" (1 page summary of the policy)
- Making "Extracts of Cover" (more detailed summary)
- Updating and Renewing of fleet contracts

Strange as it may seem, all this work was done manually. A project which included automating part of this work was discontinued after three years and several 100 000 € .

Our insurance broker's office manages the insurance of many different fleets of ships.

This involves a lot of administration:

- Issuing insurance policies
- Providing "Certificates of Cover" (1 page summary of the policy)
- Making "Extracts of Cover" (more detailed summary)
- Updating and forwarding of fleet-lists
- ...

Strange as it may seem, all this work was done manually. A project which included automating part of this work was discontinued after three years and several 100 000 €.

Our insurance broker's office manages the insurance of many different fleets of ships.

This involves a lot of administration:

- Issuing insurance policies
- Providing "Certificates of Cover" (1 page summary of the policy)
- Making "Extracts of Cover" (more detailed summary)
- Updating and forwarding of fleet-lists
- ...

Strange as it may seem, all this work was done manually. A project which included automating part of this work was discontinued after three years and several 100 000 €.

Our insurance broker's office manages the insurance of many different fleets of ships.

This involves a lot of administration:

- Issuing insurance policies
- Providing "Certificates of Cover" (1 page summary of the policy)
- Making "Extracts of Cover" (more detailed summary)
- Updating and forwarding of fleet-lists
- ...

Strange as it may seem, all this work was done manually. A project which included automating part of this work was discontinued after three years and several 100 000 €.

Our insurance broker's office manages the insurance of many different fleets of ships.

This involves a lot of administration:

- Issuing insurance policies
- Providing "Certificates of Cover" (1 page summary of the policy)
- Making "Extracts of Cover" (more detailed summary)
- Updating and forwarding of fleet-lists
- ...

Strange as it may seem, all this work was done manually. A project which included automating part of this work was discontinued after three years and several 100 000 €.

Our insurance broker's office manages the insurance of many different fleets of ships.

This involves a lot of administration:

- Issuing insurance policies
- Providing "Certificates of Cover" (1 page summary of the policy)
- Making "Extracts of Cover" (more detailed summary)
- Updating and forwarding of fleet-lists
- ...

Strange as it may seem, all this work was done manually. A project which included automating part of this work was discontinued after three years and several 100 000 € .

Our insurance broker's office manages the insurance of many different fleets of ships.

This involves a lot of administration:

- Issuing insurance policies
- Providing "Certificates of Cover" (1 page summary of the policy)
- Making "Extracts of Cover" (more detailed summary)
- Updating and forwarding of fleet-lists
- ...

Strange as it may seem, all this work was done manually. A project which included automating part of this work was discontinued after three years and several 100 000 € .

Our insurance broker's office manages the insurance of many different fleets of ships.

This involves a lot of administration:

- Issuing insurance policies
- Providing "Certificates of Cover" (1 page summary of the policy)
- Making "Extracts of Cover" (more detailed summary)
- Updating and forwarding of fleet-lists
- ...

Strange as it may seem, all this work was done manually. A project which included automating part of this work was discontinued after three years and several 100 000 € .

Our insurance broker's office manages the insurance of many different fleets of ships.

This involves a lot of administration:

- Issuing insurance policies
- Providing "Certificates of Cover" (1 page summary of the policy)
- Making "Extracts of Cover" (more detailed summary)
- Updating and forwarding of fleet-lists
- ...

Strange as it may seem, all this work was done manually. A project which included automating part of this work was discontinued after three years and several 100 000 € .

Karl  
"CountZero"  
Moens

Who am I?

The Problem

The Solution

L<sup>A</sup>T<sub>E</sub>X

Solution

Lessons  
learned



I am always looking for ways to improve my karma.  
So I decided to write an application to automate the  
production of certificates, extracts, fleetlists, ...  
And write it in 7 days or less.  
With no budget! And while still doing my normal job!!



I am always looking for ways to improve my karma.  
So I decided to write an application to automate the  
production of certificates, extracts, fleetlists, . . .

And write it in 7 days or less.

With no budget! And while still doing my normal job!!



I am always looking for ways to improve my karma.  
So I decided to write an application to automate the  
production of certificates, extracts, fleetlists, . . .  
And write it in 7 days or less.

With no budget! And while still doing my normal job!!



I am always looking for ways to improve my karma.  
So I decided to write an application to automate the  
production of certificates, extracts, fleetlists, . . .

And write it in 7 days or less.

With no budget! And while still doing my normal job!!



I am always looking for ways to improve my karma.  
So I decided to write an application to automate the  
production of certificates, extracts, fleetlists, . . .  
And write it in 7 days or less.  
With no budget! And while still doing my normal job!!

---

## Constraint

---

## Solution

---

Only 7 days to provide a working solution

RAD language (Perl, of course)

No budget

Free and Open Source based

To be used by admin people who are Windows trained and have no clue about CLI

???

Minimal install on user's PC

Install Perl, ... on shared network drive

---

---

## Constraint

---

## Solution

---

Only 7 days to provide a  
working solution

RAD language (Perl, of  
course)

No budget

Free and Open Source  
based

To be used by admin  
people who are Win-  
dows trained and have  
no clue about CLI

???

Minimal install on user's  
PC

Install Perl, ... on  
shared network drive

---

---

## Constraint

---

---

## Solution

---

Only 7 days to provide a  
working solution

RAD language (Perl, of  
course)

No budget

Free and Open Source  
based

To be used by admin  
people who are Win-  
dows trained and have  
no clue about CLI

???

Minimal install on user's  
PC

Install Perl, ... on  
shared network drive

---

---

## Constraint

---

## Solution

---

Only 7 days to provide a working solution

---

RAD language (Perl, of course)

---

No budget

---

Free and Open Source based

---

To be used by admin people who are Windows trained and have no clue about CLI

---

???

---

Minimal install on user's PC

---

Install Perl, ... on shared network drive

---





---

## Constraint

---

## Solution

---

Only 7 days to provide a  
working solution

---

RAD language (Perl, of  
course)

---

No budget

---

Free and Open Source  
based

---

To be used by admin  
people who are Win-  
dows trained and have  
no clue about CLI

---

Minimal install on user's  
PC

---

Install Perl, ... on  
shared network drive

---



---

## Constraint

---

## Solution

---

Only 7 days to provide a working solution

---

RAD language (Perl, of course)

---

No budget

---

Free and Open Source based

---

To be used by admin people who are Windows trained and have no clue about CLI

---

???

---

Minimal install on user's PC

---

Install Perl, ... on shared network drive

---

Who am I?

The Problem

The Solution

First Day

Second Day

Third Day

Fourth Day

L<sup>A</sup>T<sub>E</sub>X

Solution

Lessons  
learned

Let's find out how this business is done now.

- 1 Finding and analyzing the various documents to be produced; 3 types of documents found: fleetlist, extract of cover, certificate of insurance
- 2 These documents are all based upon a standard format (same per fleet; but different per insurance policy), however some ships may have additional clauses
- 3 Per fleet there is one spreadsheet with technical data and one MSWord-file with "names" and "capacities"; these files are maintained by different admin people
- 4 Sometimes the client requires documents for the whole fleet, sometimes only for one or more vessels in the fleet
- 5 Vessels are usually written in Dutch; however the fleet might be a mix of Dutch and English vessels

Let's find out how this business is done now.

- 1 Finding and analyzing the various documents to be produced; 3 types of documents found: fleetlist, extract of cover, certificate of insurance
- 2 These documents are all based upon a standard format (same per fleet; but different per insurance policy), however some ships may have additional clauses
- 3 Per fleet there is one spreadsheet with technical data and one MSWord-file with "names" and "capacities"; these files are maintained by different admin people
- 4 Sometimes the client requires documents for the whole fleet, sometimes only for one or more vessels in the fleet
- 5 Vessels are regularly added and deleted from the fleets; technical data will be updated regularly: this must be easy to do

Who am I?

The Problem

The Solution

First Day

Second Day

Third Day

Fourth Day

L<sup>A</sup>T<sub>E</sub>X

Solution

Lessons  
learned

Let's find out how this business is done now.

- 1 Finding and analyzing the various documents to be produced; 3 types of documents found: fleetlist, extract of cover, certificate of insurance
- 2 These documents are all based upon a standard format (same per fleet; but different per insurance policy), however some ships may have additional clauses
- 3 Per fleet there is one spreadsheet with technical data and one MSWord-file with "names" and "capacities"; these files are maintained by different admin people
- 4 Sometimes the client requires documents for the whole fleet, sometimes only for one or more vessels in the fleet
- 5 Vessels are regularly added and deleted from the fleets; technical data will be updated regularly: this must be easy to do

Let's find out how this business is done now.

- 1 Finding and analyzing the various documents to be produced; 3 types of documents found: fleetlist, extract of cover, certificate of insurance
- 2 These documents are all based upon a standard format (same per fleet; but different per insurance policy), however some ships may have additional clauses
- 3 Per fleet there is one spreadsheet with technical data and one MSWord-file with "names" and "capacities"; these files are maintained by different admin people
- 4 Sometimes the client requires documents for the whole fleet, sometimes only for one or more vessels in the fleet
- 5 Vessels are regularly added and deleted from the fleets; technical data will be updated regularly: this must be easy to do

Let's find out how this business is done now.

- 1 Finding and analyzing the various documents to be produced; 3 types of documents found: fleetlist, extract of cover, certificate of insurance
- 2 These documents are all based upon a standard format (same per fleet; but different per insurance policy), however some ships may have additional clauses
- 3 Per fleet there is one spreadsheet with technical data and one MSWord-file with "names" and "capacities"; these files are maintained by different admin people
- 4 Sometimes the client requires documents for the whole fleet, sometimes only for one or more vessels in the fleet
- 5 Vessels are regularly added and deleted from the fleets; technical data will be updated regularly: this must be easy to do

Let's find out how this business is done now.

- ① Finding and analyzing the various documents to be produced; 3 types of documents found: fleetlist, extract of cover, certificate of insurance
- ② These documents are all based upon a standard format (same per fleet; but different per insurance policy), however some ships may have additional clauses
- ③ Per fleet there is one spreadsheet with technical data and one MSWord-file with "names" and "capacities"; these files are maintained by different admin people
- ④ Sometimes the client requires documents for the whole fleet, sometimes only for one or more vessels in the fleet
- ⑤ Vessels are regularly added and deleted from the fleets; technical data will be updated regularly: this must be easy to do

Who am I?

The Problem

The Solution

First Day

Second Day

Third Day

Fourth Day

L<sup>A</sup>T<sub>E</sub>X

Solution

Lessons  
learned

- 1 The core of the application will be a templating engine. Perl has an abundance of these. I will choose `Template::Toolkit`, it is easy to use, very flexible and can contain some business logic in the template itself. And I already have the “Badger book”.
- 2 The data for each fleet will be maintained in two spreadsheets. The MSWord file will be transformed into a spreadsheet.  
`Spreadsheet::ParseExcel::Simple` makes reading Excel spreadsheets very easy.
- 3 The template will produce a L<sup>A</sup>T<sub>E</sub>X-file and will have to be “compiled” into a final PDF file. There are a number of Perl modules to run the T<sub>E</sub>X-engine.
- 4 The Windows users will like to see a GUI for choosing which vessels to “run”. Wx, Tk/Tcl ??? I never used these before.

Who am I?

The Problem

The Solution

First Day

Second Day

Third Day

Fourth Day

L<sup>A</sup>T<sub>E</sub>X

Solution

Lessons  
learned

- 1 The core of the application will be a templating engine. Perl has an abundance of these. I will choose `Template::Toolkit`, it is easy to use, very flexible and can contain some business logic in the template itself. And I already have the “Badger book”.
- 2 The data for each fleet will be maintained in two spreadsheets. The MSWord file will be transformed into a spreadsheet.  
`Spreadsheet::ParseExcel::Simple` makes reading Excel spreadsheets very easy.
- 3 The template will produce a L<sup>A</sup>T<sub>E</sub>X-file and will have to be “compiled” into a final PDF file. There are a number of Perl modules to run the T<sub>E</sub>X-engine.
- 4 The Windows users will like to see a GUI for choosing which vessels to “run”. Wx, Tk/Tcl ??? I never used these before.

- 1 The core of the application will be a templating engine. Perl has an abundance of these. I will choose `Template::Toolkit`, it is easy to use, very flexible and can contain some business logic in the template itself. And I already have the “Badger book”.
- 2 The data for each fleet will be maintained in two spreadsheets. The MSWord file will be transformed into a spreadsheet.  
`Spreadsheet::ParseExcel::Simple` makes reading Excel spreadsheets very easy.
- 3 The template will produce a L<sup>A</sup>T<sub>E</sub>X-file and will have to be “compiled” into a final PDF file. There are a number of Perl modules to run the T<sub>E</sub>X-engine.
- 4 The Windows users will like to see a GUI for choosing which vessels to “run”. Wx, Tk/Tcl ??? I never used these before.

- 1 The core of the application will be a templating engine. Perl has an abundance of these. I will choose `Template::Toolkit`, it is easy to use, very flexible and can contain some business logic in the template itself. And I already have the “Badger book”.
- 2 The data for each fleet will be maintained in two spreadsheets. The MSWord file will be transformed into a spreadsheet.  
`Spreadsheet::ParseExcel::Simple` makes reading Excel spreadsheets very easy.
- 3 The template will produce a L<sup>A</sup>T<sub>E</sub>X-file and will have to be “compiled” into a final PDF file. There are a number of Perl modules to run the T<sub>E</sub>X-engine.
- 4 The Windows users will like to see a GUI for choosing which vessels to “run”. Wx, Tk/Tcl ??? I never used these before.

Who am I?

The Problem

The Solution

First Day

Second Day

Third Day

Fourth Day

L<sup>A</sup>T<sub>E</sub>X

Solution

Lessons  
learned

- Starting the day with cleaning up the “technical” spreadsheet and transforming the MSWord file into a spreadsheet. Note to self: make sure that the ship’s names are the same in both files.
- Writing the code to read in both spreadsheets and “objectifying” the data. `Moose` is a great module: just describe the data-structure and `Moose` practically builds the objects for you. With some `after` magic the data gets imported as soon as the name of the spreadsheet is added to the object.
- I realize it is not a good idea to hard-code the paths and names of the spreadsheets: add a configuration file in YAML-format to contain this info. Use some `BUILD` magic to parse the config-file and stuff the object with the file-paths, triggering the loading of the data in the object.

Who am I?

The Problem

The Solution

First Day

Second Day

Third Day

Fourth Day

L<sup>A</sup>T<sub>E</sub>X

Solution

Lessons  
learned

- Starting the day with cleaning up the “technical” spreadsheet and transforming the MSWord file into a spreadsheet. Note to self: make sure that the ship’s names are the same in both files.
- Writing the code to read in both spreadsheets and “objectifying” the data. `MOOSE` is a great module: just describe the data-structure and `MOOSE` practically builds the objects for you. With some `after` magic the data gets imported as soon as the name of the spreadsheet is added to the object.
- I realize it is not a good idea to hard-code the paths and names of the spreadsheets: add a configuration file in YAML-format to contain this info. Use some `BUILD` magic to parse the config-file and stuff the object with the file-paths, triggering the loading of the data in the object.

Who am I?

The Problem

The Solution

First Day

Second Day

Third Day

Fourth Day

L<sup>A</sup>T<sub>E</sub>X

Solution

Lessons  
learned

- Starting the day with cleaning up the “technical” spreadsheet and transforming the MSWord file into a spreadsheet. Note to self: make sure that the ship’s names are the same in both files.
- Writing the code to read in both spreadsheets and “objectifying” the data. `Moose` is a great module: just describe the data-structure and `Moose` practically builds the objects for you. With some `after` magic the data gets imported as soon as the name of the spreadsheet is added to the object.
- I realize it is not a good idea to hard-code the paths and names of the spreadsheets: add a configuration file in YAML-format to contain this info. Use some `BUILD` magic to parse the config-file and stuff the object with the file-paths, triggering the loading of the data in the object.

Who am I?

The Problem

The Solution

First Day

Second Day

Third Day

Fourth Day

L<sup>A</sup>T<sub>E</sub>X

Solution

Lessons  
learned

- Now that we have the object structure done, we can write the template files.
- Start by writing a L<sup>A</sup>T<sub>E</sub>X-file on basis of the MSWord-“Extract of Cover”-file. All it really needs is adding L<sup>A</sup>T<sub>E</sub>X structure commands and wrapping it inside an already existing style file importing our house-style. Looks good! 😊
- Doesn't look good. 😞 All accented characters are wrong. Changing all those by hand into their L<sup>A</sup>T<sub>E</sub>X equivalents is bo-o-o-o-oring. Ah, Thai girlfriend is on Yahoo Messenger. Spend the afternoon chatting. 😊 😊
- I realize the spreadsheets also contain accented characters. I cannot ask the maintainers of these files to input accents in L<sup>A</sup>T<sub>E</sub>X-style (ö => \" {o}). Let's sleep over it.

Who am I?

The Problem

The Solution

First Day

Second Day

Third Day

Fourth Day

L<sup>A</sup>T<sub>E</sub>X

Solution

Lessons  
learned

- Now that we have the object structure done, we can write the template files.
- Start by writing a L<sup>A</sup>T<sub>E</sub>X-file on basis of the MSWord-“Extract of Cover”-file. All it really needs is adding L<sup>A</sup>T<sub>E</sub>X structure commands and wrapping it inside an already existing style file importing our house-style. Looks good! 😊
- Doesn't look good. 😞 All accented characters are wrong. Changing all those by hand into their L<sup>A</sup>T<sub>E</sub>X equivalents is bo-o-o-o-oring. Ah, Thai girlfriend is on Yahoo Messenger. Spend the afternoon chatting. 😊 😊
- I realize the spreadsheets also contain accented characters. I cannot ask the maintainers of these files to input accents in L<sup>A</sup>T<sub>E</sub>X-style (ö => \" {o}). Let's sleep over it.

Who am I?

The Problem

The Solution

First Day

Second Day

Third Day

Fourth Day

L<sup>A</sup>T<sub>E</sub>X

Solution

Lessons  
learned

- Now that we have the object structure done, we can write the template files.
- Start by writing a L<sup>A</sup>T<sub>E</sub>X-file on basis of the MSWord-“Extract of Cover”-file. All it really needs is adding L<sup>A</sup>T<sub>E</sub>X structure commands and wrapping it inside an already existing style file importing our house-style. Looks good! 😊
- Doesn't look good. 😞 All accented characters are wrong. Changing all those by hand into their L<sup>A</sup>T<sub>E</sub>X equivalents is bo-o-o-o-oring. Ah, Thai girlfriend is on Yahoo Messenger. Spend the afternoon chatting. 😊 😊
- I realize the spreadsheets also contain accented characters. I cannot ask the maintainers of these files to input accents in L<sup>A</sup>T<sub>E</sub>X-style (ö => \“{o}). Let's sleep over it.

Who am I?

The Problem

The Solution

First Day

Second Day

Third Day

Fourth Day

L<sup>A</sup>T<sub>E</sub>X

Solution

Lessons  
learned

- Now that we have the object structure done, we can write the template files.
- Start by writing a L<sup>A</sup>T<sub>E</sub>X-file on basis of the MSWord-“Extract of Cover”-file. All it really needs is adding L<sup>A</sup>T<sub>E</sub>X structure commands and wrapping it inside an already existing style file importing our house-style. Looks good! 😊
- Doesn't look good. 😞 All accented characters are wrong. Changing all those by hand into their L<sup>A</sup>T<sub>E</sub>X equivalents is bo-o-o-o-oring. Ah, Thai girlfriend is on Yahoo Messenger. Spend the afternoon chatting. 😊 😊
- I realize the spreadsheets also contain accented characters. I cannot ask the maintainers of these files to input accents in L<sup>A</sup>T<sub>E</sub>X-style (ö => \“{ö}). Let's sleep over it.

Who am I?

The Problem

The Solution

First Day

Second Day

Third Day

Fourth Day

L<sup>A</sup>T<sub>E</sub>X

Solution

Lessons  
learned

- **CPAN CPAN CPAN!** If everything else fails (and even well before that), check CPAN. `LaTeX::Encode` escapes and encodes utf-8 text into L<sup>A</sup>T<sub>E</sub>X entities.
- In a few minutes all MSWord files are saved as text files and L<sup>A</sup>T<sub>E</sub>X-encoded. Turning these into a L<sup>A</sup>T<sub>E</sub>X-file takes only a few hours.
- All the variable information in the L<sup>A</sup>T<sub>E</sub>X-file is replaced by `Template::Toolkit` tags and variables.
- Adding two lines to the script to run the template through `Template::Toolkit` and it produces a finished L<sup>A</sup>T<sub>E</sub>X-source file
- Add a back-quotes command to compile the L<sup>A</sup>T<sub>E</sub>X-source file into a PDF-file.

Who am I?

The Problem

The Solution

First Day

Second Day

Third Day

Fourth Day

L<sup>A</sup>T<sub>E</sub>X

Solution

Lessons  
learned

- **CPAN CPAN CPAN!** If everything else fails (and even well before that), check CPAN. `LaTeX::Encode` escapes and encodes utf-8 text into L<sup>A</sup>T<sub>E</sub>X entities.
- In a few minutes all MSWord files are saved as text files and L<sup>A</sup>T<sub>E</sub>X-encoded. Turning these into a L<sup>A</sup>T<sub>E</sub>X-file takes only a few hours.
- All the variable information in the L<sup>A</sup>T<sub>E</sub>X-file is replaced by `Template::Toolkit` tags and variables.
- Adding two lines to the script to run the template through `Template::Toolkit` and it produces a finished L<sup>A</sup>T<sub>E</sub>X-source file
- Add a back-quotes command to compile the L<sup>A</sup>T<sub>E</sub>X-source file into a PDF-file.

Who am I?

The Problem

The Solution

First Day

Second Day

Third Day

Fourth Day

L<sup>A</sup>T<sub>E</sub>X

Solution

Lessons  
learned

- **CPAN CPAN CPAN!** If everything else fails (and even well before that), check CPAN. `LaTeX::Encode` escapes and encodes utf-8 text into L<sup>A</sup>T<sub>E</sub>X entities.
- In a few minutes all MSWord files are saved as text files and L<sup>A</sup>T<sub>E</sub>X-encoded. Turning these into a L<sup>A</sup>T<sub>E</sub>X-file takes only a few hours.
- All the variable information in the L<sup>A</sup>T<sub>E</sub>X-file is replaced by `Template::Toolkit` tags and variables.
- Adding two lines to the script to run the template through `Template::Toolkit` and it produces a finished L<sup>A</sup>T<sub>E</sub>X-source file
- Add a back-quotes command to compile the L<sup>A</sup>T<sub>E</sub>X-source file into a PDF-file.

Who am I?

The Problem

The Solution

First Day

Second Day

Third Day

Fourth Day

L<sup>A</sup>T<sub>E</sub>X

Solution

Lessons  
learned

- **CPAN CPAN CPAN!** If everything else fails (and even well before that), check CPAN. `LaTeX::Encode` escapes and encodes utf-8 text into L<sup>A</sup>T<sub>E</sub>X entities.
- In a few minutes all MSWord files are saved as text files and L<sup>A</sup>T<sub>E</sub>X-encoded. Turning these into a L<sup>A</sup>T<sub>E</sub>X-file takes only a few hours.
- All the variable information in the L<sup>A</sup>T<sub>E</sub>X-file is replaced by `Template::Toolkit` tags and variables.
- Adding two lines to the script to run the template through `Template::Toolkit` and it produces a finished L<sup>A</sup>T<sub>E</sub>X-source file
- Add a back-quotes command to compile the L<sup>A</sup>T<sub>E</sub>X-source file into a PDF-file.

- **CPAN CPAN CPAN!** If everything else fails (and even well before that), check CPAN. `LaTeX::Encode` escapes and encodes utf-8 text into L<sup>A</sup>T<sub>E</sub>X entities.
- In a few minutes all MSWord files are saved as text files and L<sup>A</sup>T<sub>E</sub>X-encoded. Turning these into a L<sup>A</sup>T<sub>E</sub>X-file takes only a few hours.
- All the variable information in the L<sup>A</sup>T<sub>E</sub>X-file is replaced by `Template::Toolkit` tags and variables.
- Adding two lines to the script to run the template through `Template::Toolkit` and it produces a finished L<sup>A</sup>T<sub>E</sub>X-source file
- Add a back-quotes command to compile the L<sup>A</sup>T<sub>E</sub>X-source file into a PDF-file.

L<sup>A</sup>T<sub>E</sub>X may have to run multiple times before the output file (PDF or DVI) stabilizes:

- At least twice if there is a TOC at the front, or internal references, or (hyper)links
- Three times if the document includes a bibliography or an index
- Four or five times if there are complex tables or floating figures

Karl  
"CountZero"  
Moens

Who am I?

The Problem

The Solution

L<sup>A</sup>T<sub>E</sub>X

Solution

Lessons  
learned

L<sup>A</sup>T<sub>E</sub>X may have to run multiple times before the output file (PDF or DVI) stabilizes:

- At least **twice** if there is a TOC at the front, or internal references, or (hyper)links
- **Three** times if the document includes a bibliography or an index
- **Four or five** times if there are complicated tables spanning multiple pages





L<sup>A</sup>T<sub>E</sub>X may have to run multiple times before the output file (PDF or DVI) stabilizes:

PAGE 3

DEPARTMENT	COURSE	DESCRIPTION	PREREQS
COMPUTER SCIENCE	CPSC 432	INTERMEDIATE COMPILER DESIGN, WITH A FOCUS ON DEPENDENCY RESOLUTION.	CPSC 432

- At least **twice** if there is a TOC at the front, or internal references, or (hyper)links
- **Three** times if the document includes a bibliography or an index
- **Four** or **five** times if there are complicated tables spanning multiple pages



L<sup>A</sup>T<sub>E</sub>X produces many auxiliary files which litter your hard-disk.

In between runs you may need to run indexing programs or bibliographic databases

After having (re-)compiled a L<sup>A</sup>T<sub>E</sub>X-file you must manually start a viewer to see the result.



L<sup>A</sup>T<sub>E</sub>X produces many auxiliary files which litter your hard-disk.

In between runs you may need to run indexing programs or bibliographic databases

After having (re-)compiled a L<sup>A</sup>T<sub>E</sub>X-file you must manually start a viewer to see the result.



L<sup>A</sup>T<sub>E</sub>X produces many auxiliary files which litter your hard-disk.

In between runs you may need to run indexing programs or bibliographic databases

After having (re-)compiled a L<sup>A</sup>T<sub>E</sub>X-file you must manually start a viewer to see the result.

As T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X *compile* your text input into an output file, why not use a *make-like utility* to help you?

`latexmk` is a Perl script for running L<sup>A</sup>T<sub>E</sub>X the correct number of times to resolve cross references, etc;

It also runs auxiliary programs (`bibtex`, `makeindex`) if necessary, and `dvips` and/or a previewer as requested. It has a number of other useful capabilities:

Its current released version is 4.35 of 11<sup>th</sup> of November 2012.

<http://www.phys.psu.edu/~collins/software/latexmk-jcc/>

or on CTAN (Comprehensive T<sub>E</sub>X Archive Network):

<http://www.ctan.org>

As T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X *compile* your text input into an output file, why not use a *make-like utility* to help you?

`latexmk` is a Perl script for running L<sup>A</sup>T<sub>E</sub>X the correct number of times to resolve cross references, etc;

It also runs auxiliary programs (`bibtex`, `makeindex`) if necessary, and `dvips` and/or a previewer as requested. It has a number of other useful capabilities:

• It can generate a PDF file directly from the source files.  
• It can generate a PDF file from a DVI file.

Its current released version is 4.35 of 11<sup>th</sup> of November 2012.

<http://www.phys.psu.edu/~collins/software/latexmk-jcc/>

or on CTAN (Comprehensive T<sub>E</sub>X Archive Network):

<http://www.ctan.org>

As T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X *compile* your text input into an output file, why not use a *make-like utility* to help you?

`latexmk` is a Perl script for running L<sup>A</sup>T<sub>E</sub>X the correct number of times to resolve cross references, etc;

It also runs auxiliary programs (`bibtex`, `makeindex`) if necessary, and `dvips` and/or a previewer as requested.

It has a number of other useful capabilities:

- start a previewer and then run L<sup>A</sup>T<sub>E</sub>X whenever the source files are updated

Its current released version is 4.35 of 11<sup>th</sup> of November 2012.

<http://www.phys.psu.edu/~collins/software/latexmk-jcc/>

or on CTAN (Comprehensive T<sub>E</sub>X Archive Network):

<http://www.ctan.org>

As T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X *compile* your text input into an output file, why not use a *make-like utility* to help you?

`latexmk` is a Perl script for running L<sup>A</sup>T<sub>E</sub>X the correct number of times to resolve cross references, etc;

It also runs auxiliary programs (`bibtex`, `makeindex`) if necessary, and `dvips` and/or a previewer as requested. It has a number of other useful capabilities:

- start a previewer and then run L<sup>A</sup>T<sub>E</sub>X whenever the source files are updated
- clean your workspace of unnecessary auxiliary files

Its current released version is 4.35 of 11<sup>th</sup> of November 2012.

<http://www.phys.psu.edu/~collins/software/latexmk-jcc/>

or on CTAN (Comprehensive T<sub>E</sub>X Archive Network):

<http://www.ctan.org>

As T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X *compile* your text input into an output file, why not use a *make-like utility* to help you?

`latexmk` is a Perl script for running L<sup>A</sup>T<sub>E</sub>X the correct number of times to resolve cross references, etc;

It also runs auxiliary programs (`bibtex`, `makeindex`) if necessary, and `dvips` and/or a previewer as requested. It has a number of other useful capabilities:

- start a previewer and then run L<sup>A</sup>T<sub>E</sub>X whenever the source files are updated
- clean your workspace of unnecessary auxiliary files

Its current released version is 4.35 of 11<sup>th</sup> of November 2012.

<http://www.phys.psu.edu/~collins/software/latexmk-jcc/>

or on CTAN (Comprehensive T<sub>E</sub>X Archive Network):

<http://www.ctan.org>

As T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X *compile* your text input into an output file, why not use a *make-like utility* to help you?

`latexmk` is a Perl script for running L<sup>A</sup>T<sub>E</sub>X the correct number of times to resolve cross references, etc;

It also runs auxiliary programs (`bibtex`, `makeindex`) if necessary, and `dvips` and/or a previewer as requested. It has a number of other useful capabilities:

- start a previewer and then run L<sup>A</sup>T<sub>E</sub>X whenever the source files are updated
- clean your workspace of unnecessary auxiliary files

Its current released version is 4.35 of 11<sup>th</sup> of November 2012.

<http://www.phys.psu.edu/~collins/software/latexmk-jcc/>

or on CTAN (Comprehensive T<sub>E</sub>X Archive Network):

<http://www.ctan.org>

As T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X *compile* your text input into an output file, why not use a *make-like utility* to help you?

`latexmk` is a Perl script for running L<sup>A</sup>T<sub>E</sub>X the correct number of times to resolve cross references, etc;

It also runs auxiliary programs (`bibtex`, `makeindex`) if necessary, and `dvips` and/or a previewer as requested. It has a number of other useful capabilities:

- start a previewer and then run L<sup>A</sup>T<sub>E</sub>X whenever the source files are updated
- clean your workspace of unnecessary auxiliary files

Its current released version is 4.35 of 11<sup>th</sup> of November 2012.

<http://www.phys.psu.edu/~collins/software/latexmk-jcc/>

or on CTAN (Comprehensive T<sub>E</sub>X Archive Network):

<http://www.ctan.org>

As T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X *compile* your text input into an output file, why not use a *make-like utility* to help you?

`latexmk` is a Perl script for running L<sup>A</sup>T<sub>E</sub>X the correct number of times to resolve cross references, etc;

It also runs auxiliary programs (`bibtex`, `makeindex`) if necessary, and `dvips` and/or a previewer as requested. It has a number of other useful capabilities:

- start a previewer and then run L<sup>A</sup>T<sub>E</sub>X whenever the source files are updated
- clean your workspace of unnecessary auxiliary files

Its current released version is 4.35 of 11<sup>th</sup> of November 2012.

<http://www.phys.psu.edu/~collins/software/latexmk-jcc/>

or on CTAN (Comprehensive T<sub>E</sub>X Archive Network):

<http://www.ctan.org>

As T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X *compile* your text input into an output file, why not use a *make-like utility* to help you?

`latexmk` is a Perl script for running L<sup>A</sup>T<sub>E</sub>X the correct number of times to resolve cross references, etc;

It also runs auxiliary programs (`bibtex`, `makeindex`) if necessary, and `dvips` and/or a previewer as requested. It has a number of other useful capabilities:

- start a previewer and then run L<sup>A</sup>T<sub>E</sub>X whenever the source files are updated
- clean your workspace of unnecessary auxiliary files

Its current released version is 4.35 of 11<sup>th</sup> of November 2012.

<http://www.phys.psu.edu/~collins/software/latexmk-jcc/>

or on CTAN (Comprehensive T<sub>E</sub>X Archive Network):

<http://www.ctan.org>

As T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X *compile* your text input into an output file, why not use a *make-like utility* to help you?

`latexmk` is a Perl script for running L<sup>A</sup>T<sub>E</sub>X the correct number of times to resolve cross references, etc;

It also runs auxiliary programs (`bibtex`, `makeindex`) if necessary, and `dvips` and/or a previewer as requested. It has a number of other useful capabilities:

- start a previewer and then run L<sup>A</sup>T<sub>E</sub>X whenever the source files are updated
- clean your workspace of unnecessary auxiliary files

Its current released version is 4.35 of 11<sup>th</sup> of November 2012.

<http://www.phys.psu.edu/~collins/software/latexmk-jcc/>

or on CTAN (Comprehensive T<sub>E</sub>X Archive Network):

<http://www.ctan.org>

No work done on this project. I had to attend to some claimfiles.



Who am I?

The Problem

The Solution

L<sup>A</sup>T<sub>E</sub>X

Solution

Fifth Day

Sixth Day

Seventh Day

Lessons  
learned

No work done on this project. I had to attend to some claimfiles.



Business as usual.

Who am I?

The Problem

The Solution

L<sup>A</sup>T<sub>E</sub>X

Solution

Fifth Day

Sixth Day

Seventh Day

Lessons

learned

No work done on this project. I had to attend to some claimfiles.



Business as usual.

Karl  
"CountZero"  
Moens

Who am I?

The Problem

The Solution

L<sup>A</sup>T<sub>E</sub>X

Solution

Fifth Day

Sixth Day

Seventh Day

Lessons  
learned

No work done on this project. I had to attend to some claimfiles.



Business as usual.

Who am I?

The Problem

The Solution

L<sup>A</sup>T<sub>E</sub>X

Solution

Fifth Day

Sixth Day

Seventh Day

Lessons

learned

No work done on this project. I had to attend to some claimfiles.

Business as usual.

Who am I?

The Problem

The Solution

L<sup>A</sup>T<sub>E</sub>X

Solution

Fifth Day

Sixth Day

Seventh Day

Lessons

learned

No work done on this project. I had to attend to some claimfiles.

Business as usual.

Who am I?

The Problem

The Solution

L<sup>A</sup>T<sub>E</sub>X

Solution

Fifth Day

Sixth Day

Seventh Day

Lessons

learned

I produced some documents and they seemed OK.

Time to write the GUI shell around the script.

I discover I am too “old school” to write GUI shells. I will never be able to learn how to do it in a few hours time.

Who am I?

The Problem

The Solution

L<sup>A</sup>T<sub>E</sub>X

Solution

Fifth Day

Sixth Day

Seventh Day

Lessons

learned

I produced some documents and they seemed OK.

Time to write the GUI shell around the script.

I discover I am too “old school” to write GUI shells. I will never be able to learn how to do it in a few hours time.

I produced some documents and they seemed OK.  
Time to write the GUI shell around the script.  
I discover I am too “old school” to write GUI shells. I will never be able to learn how to do it in a few hours time.



Who am I?

The Problem

The Solution

L<sup>A</sup>T<sub>E</sub>X

Solution

Fifth Day

Sixth Day

Seventh Day

Lessons

learned

Instead of a GUI, I will show a numbered list of the vessels in the CLI and allow input of a list of these numbers which will produce the documents for these vessels.

`Parse::Range` will allow input such as `1, 3, 5-20` which parses as `1, 3, 5 to 20`. Cool! That will impress the admin girls for sure.

I run a whole fleet and show this to the admin girls for checking.

Who am I?

The Problem

The Solution

L<sup>A</sup>T<sub>E</sub>X

Solution

Fifth Day

Sixth Day

Seventh Day

Lessons  
learned

Instead of a GUI, I will show a numbered list of the vessels in the CLI and allow input of a list of these numbers which will produce the documents for these vessels.

`Parse::Range` will allow input such as `1, 3, 5-20` which parses as 1, 3, 5 to 20. Cool! That will impress the admin girls for sure.

I run a whole fleet and show this to the admin girls for checking.

Who am I?

The Problem

The Solution

L<sup>A</sup>T<sub>E</sub>X

Solution

Fifth Day

Sixth Day

Seventh Day

Lessons  
learned

Instead of a GUI, I will show a numbered list of the vessels in the CLI and allow input of a list of these numbers which will produce the documents for these vessels.

`Parse::Range` will allow input such as `1, 3, 5-20` which parses as 1, 3, 5 to 20. Cool! That will impress the admin girls for sure.

I run a whole fleet and show this to the admin girls for checking.



The admin girls found about 20 errors in the documents.

- Some typos: easily corrected
- They like the dates in "human" format rather than '2012-10-25'. Nothing that can't be easily solved with the `DateTime` module.
- Some vessels have additional special clauses or a different insurance arrangement. This logic was put inside the template: no need to change the application itself. `Template::Toolkit` allows even raw Perl to be included if necessary.

All I have to do is to install some shell scripts on the admin girls PCs so they can run the application themselves.

The admin girls found about 20 errors in the documents.

- **Some typos: easily corrected**
- They like the dates in “human” format rather than '2012-10-25'. Nothing that can't be easily solved with the `DateTime` module.
- Some vessels have additional special clauses or a different insurance arrangement. This logic was put inside the template: no need to change the application itself. `Template::Toolkit` allows even raw Perl to be included if necessary.

All I have to do is to install some shell scripts on the admin girls PCs so they can run the application themselves.

The admin girls found about 20 errors in the documents.

- Some typos: easily corrected
- They like the dates in “human” format rather than '2012-10-25'. Nothing that can't be easily solved with the `DateTime` module.
- Some vessels have additional special clauses or a different insurance arrangement. This logic was put inside the template: no need to change the application itself. `Template::Toolkit` allows even raw Perl to be included if necessary.

All I have to do is to install some shell scripts on the admin girls PCs so they can run the application themselves.

The admin girls found about 20 errors in the documents.

- Some typos: easily corrected
- They like the dates in “human” format rather than '2012-10-25'. Nothing that can't be easily solved with the `DateTime` module.
- Some vessels have additional special clauses or a different insurance arrangement. This logic was put inside the template: no need to change the application itself. `Template::Toolkit` allows even raw Perl to be included if necessary.

All I have to do is to install some shell scripts on the admin girls PCs so they can run the application themselves.

The admin girls found about 20 errors in the documents.

- Some typos: easily corrected
- They like the dates in “human” format rather than '2012-10-25'. Nothing that can't be easily solved with the `DateTime` module.
- Some vessels have additional special clauses or a different insurance arrangement. This logic was put inside the template: no need to change the application itself. `Template::Toolkit` allows even raw Perl to be included if necessary.

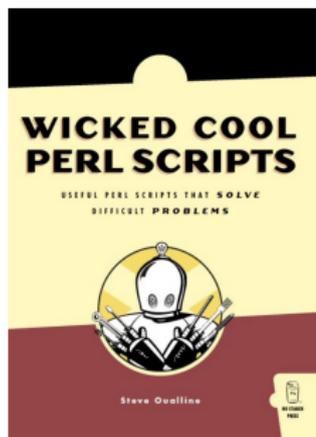
All I have to do is to install some shell scripts on the admin girls PCs so they can run the application themselves.

The admin girls found about 20 errors in the documents.

- Some typos: easily corrected
- They like the dates in “human” format rather than '2012-10-25'. Nothing that can't be easily solved with the `DateTime` module.
- Some vessels have additional special clauses or a different insurance arrangement. This logic was put inside the template: no need to change the application itself. `Template::Toolkit` allows even raw Perl to be included if necessary.

All I have to do is to install some shell scripts on the admin girls PCs so they can run the application themselves.

- Perl makes difficult things possible, fast.
- The Power of CPAN is awesome!
- Perl glues many different open and closed source technologies together.
- Perl is very much alive and kicking, but hides itself well.
- Perl Programmers always get the nicest girls.



- Perl makes difficult things possible, fast.
- The Power of CPAN is awesome!
- Perl glues many different open and closed source technologies together.
- Perl is very much alive and kicking, but hides itself well.
- Perl Programmers always get the nicest girls.



**CPAN** always  
finds a way

- Perl makes difficult things possible, fast.
- The Power of CPAN is awesome!
- Perl glues many different open and closed source technologies together.
- Perl is very much alive and kicking, but hides itself well.
- Perl Programmers always get the nicest girls.

TRULY, THIS WAS  
THE LANGUAGE  
FROM WHICH THE  
GODS WROUGHT  
THE UNIVERSE.



Karl  
"CountZero"  
Moens

Who am I?

The Problem

The Solution

L<sup>A</sup>T<sub>E</sub>X

Solution

Lessons  
learned

- Perl makes difficult things possible, fast.
- The Power of CPAN is awesome!
- Perl glues many different open and closed source technologies together.
- Perl is very much alive and kicking, but hides itself well.
- Perl Programmers always get the nicest girls.

The screenshot shows the Forbes.com website interface. At the top, there is a search bar with the text "Search: Forbes.co" and a search icon. Below the search bar, the "Forbes.com" logo is displayed in large white letters on a dark blue background. Underneath the logo, there are navigation links for "U.S.", "EUROPE", and "ASIA", along with the text "HOME PAGE FOR TH". A secondary navigation bar contains links for "HOME", "BUSINESS", "TECH", "MARKETS", and "ENTREPRENEUR". Below this, there are links for "Video", "Blogs", "E-mail Newsletters", and "People Tracker". The main content area features a snippet from Associated Press with the headline "HASH(0x8f01d48)" and the byline "By JULIE WATSON 03.18.07, 10:35 PM ET". At the bottom, there is a "Popular Videos" section with a blue header and a snippet of a video titled "Venezuelan Presic radical policies pos".

Karl  
"CountZero"  
Moens

Who am I?

The Problem

The Solution

L<sup>A</sup>T<sub>E</sub>X

Solution

Lessons  
learned

- Perl makes difficult things possible, fast.
- The Power of CPAN is awesome!
- Perl glues many different open and closed source technologies together.
- Perl is very much alive and kicking, but hides itself well.
- Perl Programmers always get the nicest girls.



Karl  
"CountZero"  
Moens

## A job well done!

Who am I?

The Problem

The Solution

L<sup>A</sup>T<sub>E</sub>X

Solution

Lessons  
learned



The boss and the office girls are grateful. Thank you Perl!  
Now if only I could write a script to answer the 500+ emails  
that arrived while I was writing Perl scripts.

Karl  
"CountZero"  
Moens

A job well done!

Who am I?

The Problem

The Solution

L<sup>A</sup>T<sub>E</sub>X

Solution

Lessons  
learned



The boss and the office girls are grateful. Thank you Perl!  
Now if only I could write a script to answer the 500+ emails  
that arrived while I was writing Perl scripts.