Better data quality from your web form

Effective international name and address Internet data collection

by
Graham Rhind

First release, April 2009.
Second release, August 2009.
Check for newer versions at http://www.grcdi.nl/book4.htm

Published by
Graham Rhind
Nieuwe Prinsengracht 80-hs
1018 VV AMSTERDAM
The Netherlands

If you wish to be kept informed (by e-mail) of updates to this work, send your name (and company name, if relevant) to book4@grcdi.nl

Contents

Author's note, acknowledgements and disclaimer

I've been an Internet user for over a decade, and have spent much of that time wailing and gnashing my teeth as I sweated my way through the laborious process of trying to complete poorly designed web forms. Eventually it got too much, so I started the cathartic process of writing about them.

To those web form designers who have provided me with more ammunition than I know what to do with – thanks. But we've seen enough now – perhaps you'd like to use this tome to give some thought to your forms and to improve them? The rest of the world would be ever grateful, and it would give me a chance to write a companion volume containing examples of well thought out forms.

I live in hope.

A great deal of what I have written is common sense, but it is clear that common sense often only becomes obvious when somebody points it out – if that were not the case we would never find any forms with a required *state* field.

With this book I have attempted to fill the gap left by most works about usability by concentrating on the experience that your international customers have with your form, and how it affects them, their relationship with you and your data quality.

I have attempted to do this without going into too much depth about the idiosyncrasies of international personal name and addresses, though they fascinate me so much that this has been hard to do; and have avoided technical discussions – links to get more information are provided where appropriate in the text.

If you have any comments about this book - the contents, the balance of subjects, how it can be improved or what might be missing; if you want to post a rave review or flame me; or you want to know when the new edition will be released – anything, in fact, related to this book – then please send it to me at book4@grcdi.nl

I have used the term "customer" throughout this book, though people completing your form may not (yet) be a customer of yours. (And, judging by the hurdles placed before them by many forms, most of them will never succeed in becoming your customer). I have no semantic problems with using the term "user", though I know it irritates some people; but I have chosen a term that gives those users a more human face – the humanity of your customers is something that is often overlooked in today's business processes, and web forms are no exception to that.

Thanks to Simon Daniels of Percassity Marketing Data Solutions (http://www.percassity.com) for his encouragement and feedback; and to Winfried van Holland, CTO of Human Inference (http://www.humaninference.com) and Gwillim Law (http://www.statoids.com) for their valuable input.

There are many examples of forms in this book, and all have been clipped (or mocked up) in a way to make them as unrecognisable as possible – it's not my intention to embarrass specific companies or individuals through their inclusion in this book. If you recognise your form and are happy to let me use your (company's) name in the book, let me know. Equally, if you think you recognise your site and would like it removed from the book, let me know that too – there are, unfortunately, plenty more examples to fall back on.

About the author

Graham Rhind has specialised for over 16 years in international address and postal code methodologies. He started his data quality career by building the European database of Scitex SA in Brussels, and then became Research and Development Director for OTS Group in The Netherlands. He is now an independent consultant and owner of GRC Database Information.

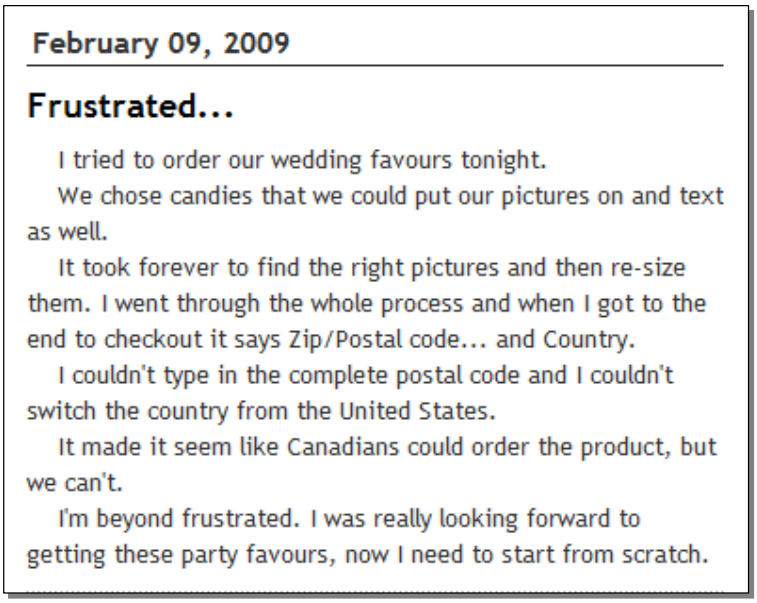
His researches have led him to write three books before this one: "Building and Maintaining a European Direct Marketing Database" (1994) , "Global Source Book for Address Database Management" (1998, updated twice annually) and "Practical International Data Management - A guide to working with global names and addresses" (2001).

He is a regular speaker at conferences and seminars, and has developed a range of software and reference data for optimal international address data standardisation, formatting, validation and de-duplication.

Graham is a charter member of the International Association for Information and Data Quality.

Through GRC Database Information Graham offers a large number of services to his customers, to help them manage globalisation in the most efficient way, with the emphasis on quality.

A state of denial and beyond frustration



February 09, 2009

Frustrated...

I tried to order our wedding favours tonight.

We chose candies that we could put our pictures on and text as well.

It took forever to find the right pictures and then re-size them. I went through the whole process and when I got to the end to checkout it says Zip/Postal code... and Country.

I couldn't type in the complete postal code and I couldn't switch the country from the United States.

It made it seem like Canadians could order the product, but we can't.

I'm beyond frustrated. I was really looking forward to getting these party favours, now I need to start from scratch.

Take a quick tour around some blogs, and it won't be long before you find one venting frustrations about the problems faced when trying to do something as seemingly simple as make an order or fill in a request form online. On an online help forum José is advised to add 00000 as his postal code because he is not allowed to register his hardware unless he has one. Tim, whose credit card was stolen while he was on holiday, has had to lie to his credit card company and make up a postal code for the place he is staying, because his card company won't take his details without it.

Everywhere we look on the Internet, there are forms which seem determined not only not to let the correct information through, but also to prevent a prospect from becoming your customer.

The anecdotes are everywhere[1], and so are the figures which show what a huge problem badly designed web forms are. A survey from 2008, commissioned by Tealeaf and carried out by Harris Interactive[2], found that a hugely significant 87% of adults attempting to carry out an online transaction in the previous year had encountered problems with that process; and that 41% of those would give up on the transaction or move to a competitor as a result of these problems. 84% of those who faced problems would share their experiences with others, on- and offline.

The costs of lost custom to the companies concerned is enormous.

Yet judging by the forms we are faced with online daily, it would seem that few of those responsible for designing or programming forms have ever come across one of these anecdotes, or have ever had problems filling in a form themselves.

[1] For another good discussion on international forms from the customer perspective, see http://evolt.org/node/15118

[2] http://www.tealeaf.com/resources/Harris_2008.asp

Or, more likely, they are in a state of denial or simply complacent.

It is a privilege for your company when a customer chooses to provide you with his or her information through your web form.

You should be bending over backwards to make your forms as quick and painless to complete as possible. In reality, most web forms make customers jump through hoops to fill them in, constantly placing hurdles for them to cross, especially where forms have been designed to fulfil the demands of the back-office processes with little reference to the customer.

It should be seen as a common courtesy to make a web form as usable as possible, coaxing your customer through the process of disgorging his or her information in as frictionless a manner as possible. It never does a company any harm to have happy customers and it is always negative to have unhappy ones. But apart from this, there are very good business reasons for making forms usable. Fewer of your customers will give up on the process before completion, and the more usable a form is, the better the quality of the data that is collected through it will be.

Forms are the poor relations in the art of web usability. Where companies are willing to make large outlays to design and test their sites, forms are given either no thought or merit a fleeting afterthought.

Usability experts often do not mention, or skim over forms, in their books.

When they do mention forms, they usually do not mention international data collection and its effect on usability, or mention it as an afterthought.

And when they do mention international data collection, they refer to the form … and not the form**s**.

For many companies it is easy to forget that your web site can and will be viewed from every corner of the planet, and, unless you are careful to limit your customer base to a single country, you will need to take all customers into account. Usability guru Steve Krug, for example, author of "Don't Make Me Think, A Common Sense Approach to Web Usability", is quick to forget this. He lists, for example, 5 companies who have no tagline[3] on their websites, but suggests that it's forgivable for those companies because they are so well known.[4]

I, a pretty well travelled Briton, have not heard of two of them, though they may be well known within the United States.

It is a common error in web form design to make such assumptions about the world beyond your own country's borders.

[3] a pithy phrase which characterizes the enterprise
[4] Steve Krug, *Don't Make Me Think*, New Riders, Berkeley, CA, USA, 2006 page 106

Krug says, of forms:

> "Whenever you build a new kind of page – particularly forms – you should print the page out and show it to the person in the next cubicle and see if they can make sense out of it. This kind of informal testing can be very efficient, and eliminate a lot of potential problems."[5]

When it comes to forms collecting personal names and addresses from an international audience, this method of testing would do little more than enforce the idea that a form is good and blinker the company to the poor data quality being produced, as the person in the cubicle next to you is likely to live in the same country, have a similar cultural background, and will work for the same company. A printout will not show where fields are too short, any validation or dynamics that are in place and so on.

Perhaps the biggest problem with creating an Internet-based data entry form for your customers is that it's too easy to do. Forms can be laid out in a few minutes by fairly low-grade technical staff, and many are created outside the remit of any company-wide data management regime. Few companies inculcate the importance of data quality to all of their staff, and in others, stake holders in data quality are not usually those who create, or supervise the creation of, company web forms.

Though the form layout itself is easy to create, the back-end/client/server part of the process is less so, and requires some technical skill, so that either the form is created by a technician without a direct stake holding in customer service or data quality; is overlooked or deliberately passed over due to the technical issues (often the case in small companies); or the wishes of the stake holders are not put into practice by the technical staff.

Equally, it is often front line staff, without the clout or motivation to get changes pushed through, who are faced with the daily problems of poor form design. When Tim, who lost his credit card, made up a postal code because the country where he was does not have them, there is little doubt that the call centre worker concerned made no effort to pass on to the powers that be the problem that their badly-designed system was causing them and their customers.

I was once contacted by a receptionist who wanted to know if there was a sure fire way of finding out which country an address was in. After some questioning, I found out that her company had failed to add a country field to its web form, so the receptionist was drafted in to fill in the missing information every time the form had been completed. Because she was but a receptionist, she firstly did not dare to make the suggestion that the form be

[5] Steve Krug, *Don't Make Me Think*, New Riders, Berkeley, CA, USA, 2006, page 145

modified, and later, when she did, she was ignored. It is not easy to identify a country from an address without a lot of background knowledge, so often the receptionist was forced to call the customer to ask them for their country. You do not need to have much imagination to realise how much this dented the company's reputation with its customers or how much of her time, and the company's, was wasted on this exercise.

It's easy to create a form, but it is NOT easy to create a form which will enable your customers to easily enter all the data that they need to provide and will enable you to collect and use high-quality data about your customers. The information that is derived from the low-quality data provided by most forms is then, in its turn, flawed, and it often takes companies several years (or never!) to start addressing the problem of the poor quality of the data collected.

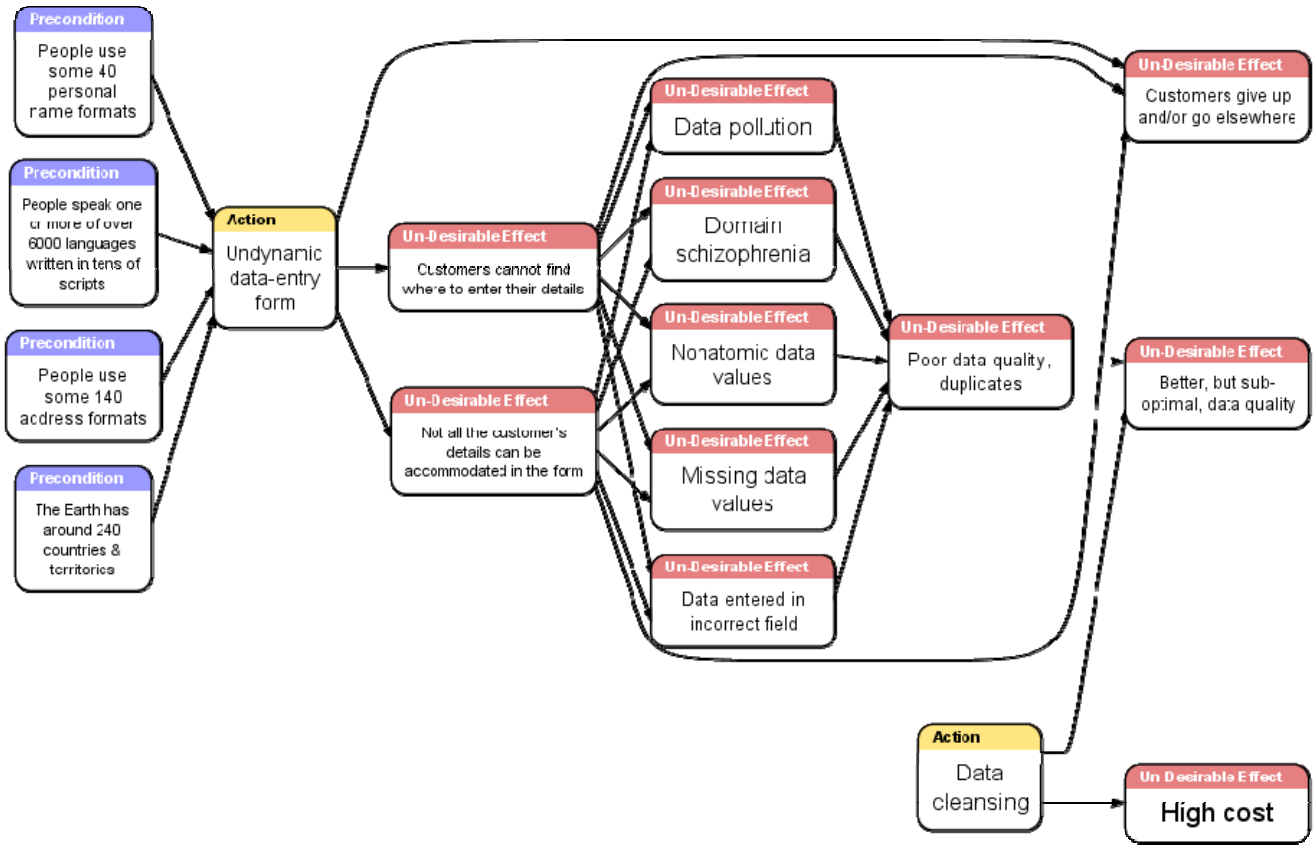
Creating a data entry form, or rather forms, to ensure high-quality data gathering, is not easy, but it is highly recommended that some thought is given to this before a form is thrown together and posted online. It costs more time and money at the creation stage, but it saves enormously on data cleansing, lost customers and lost revenues further down the processing and time lines.

This book is not about any general aspects of usability outside of your web form(s). It doesn't look at page layout, navigation, breadcrumbs, colours, fonts, icons, graphics and their position on the page, and so on; though I do stray into that territory at times when the problem is often encountered in forms intended for an international audience. It also does not discuss the various cultural aspects of approaching customers in general. For example, it will not tell you to use the colour red and the number eight on your form for China as they are considered auspicious.

This book concentrates on the specifics of data entry forms for two pieces of information that almost every company will require from their customers: the personal name and postal address; and on those forms which may be completed by people from different countries and cultures (and, on the web, I can't think of any which do not fulfil this condition – even if all your customers are in one country, they will surely have varied ethnic, national and cultural backgrounds). However simple it may seem to collect this information, you will find that it's not as simple as current collection forms seem to think that it is.

If you look around the Internet at the myriad examples of data entry forms available on company websites, you'd be forgiven for thinking that the world contained a single personal name and address format – usually that of the country in which the company collecting the data is based.

The effect of poor web form design on data quality[6]



This is clearly not the case.

There are around 240 countries and territories in the world, serviced by almost 200 postal authorities, most with their own unique ideas on how to best get mail to each point of delivery. Though mail volumes are decreasing, and you are likely to be trying to reach your customers in many other, perhaps electronic, ways, an address (as defined by postal systems) is an important part of your database as it may be used to uniquely identify an individual customer, and if you are selling products, you will need that information to allow

[6] ***Nonatomic data values***: *multiple facts being entered in the same field*
Domain schizophrenia: *fields used for different purposes by the customer to be able to enter all required information- misfielding*

a delivery to be made. There are also many other reasons for striving for maximum data quality – to allow data matching, de-duplication, validation, enhancement, credit checking and so on.

The 240 countries and territories contain people using upward of 130 postal address formats, almost 40 different personal name formats, and speaking one or more of over 6000 languages. They will write their language (if it has a written form at all) using one or more of forty scripts (writing systems).

Examples of different writing systems

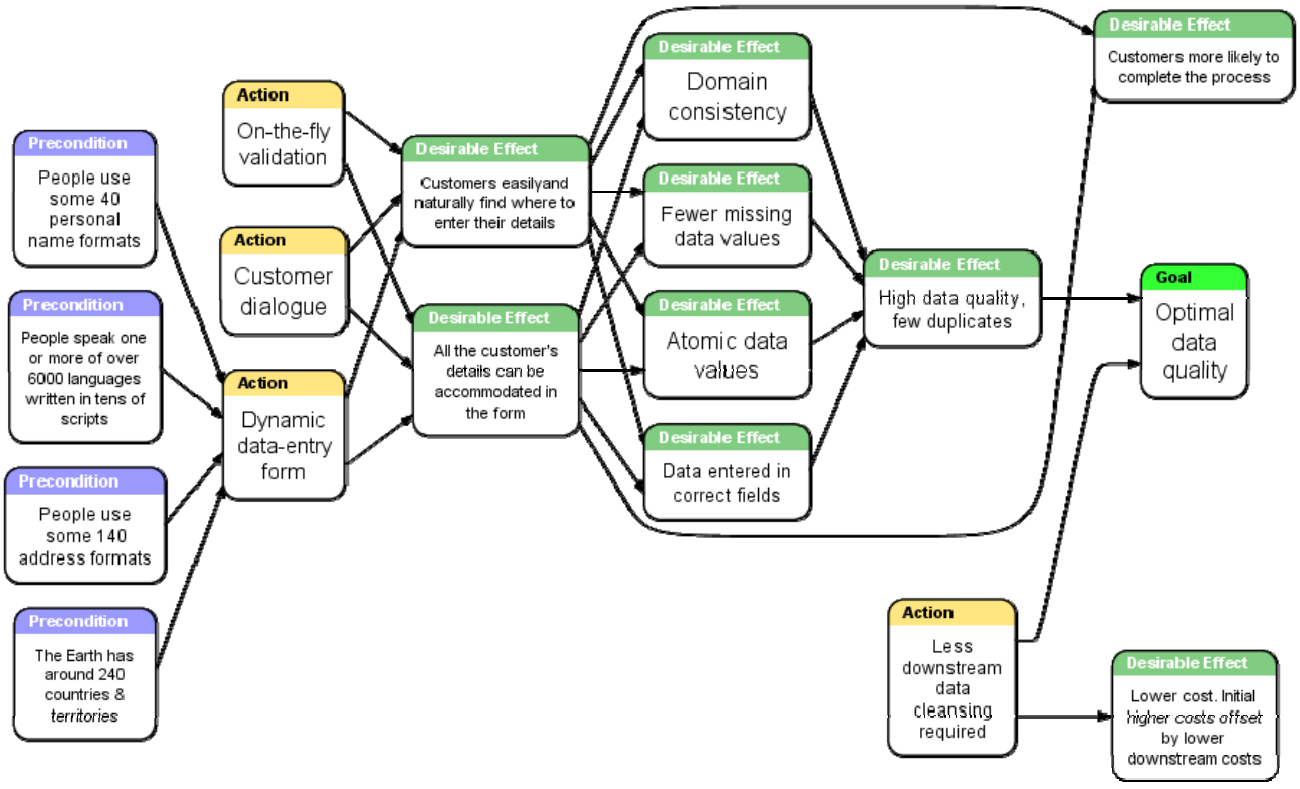
Is your web form able to accommodate each (potential) customer? Can it accept all of these formats, scripts, diacritical marks?

There can be many reasons why you choose to place an data entry form online: to allow customers to contact you, to gather data, to sell something, because your boss told you to do so, to provide the perception to your customers that their opinions count and that you wish to hold a dialogue with them, to make money. What this all boils down to is that you want something from your customer.

There can be many reasons why your customer chooses to provide you with data through an online form: to purchase a product, to ask a question, to gather knowledge, to complain. What it all boils down to is that they want something from you - otherwise they wouldn't go to the time and trouble of providing their information to you.

In other words, you want something from them and they want something from you. And, this being the case, the initial thought that must be given to the form when it goes online is that it must facilitate in every possible way this process of information exchange. It should provide the least possible friction to allowing the customer to start this dialogue with you. A large number of customers using e-commerce sites abandon their shopping carts when they reach forms which are largely unusable for them. Far too many forms are designed (if they are designed at all) for the benefit of the company concerned, without thought being given to how usable they are, or are perceived to be, by the customer. Customers are highly aware that their data is valuable to you and that their time is valuable to them. Forms which create friction lead to customers peeling off from the page before completing the form out of frustration (a black mark against your company) or to them providing poor quality data just to be able to continue to get what they are after from you (at the price of high downstream processing costs for your company).

Better web form design produces better data quality



Your customers claim ownership of their personal information: name, address, contact and identity information. They place a high value on it and expect you to do the same. You need to ensure not only that this information is gathered, stored and manipulated in ways that enable you to make optimal use of them; but that, in a customer-centric manner, you show respect to your customer through these best practices with this data. This allows the building of a profitable relationship and dialogue with the customer.

Given the same preconditions, a well thought out and dynamic form (or set of forms) can have an immense positive effect on your data quality. By asking the country name and the desired field label language beforehand, you can have a form build itself dynamically to suit that person's preference and cultural background. This enables the customer to add all of their data, in fields designed to receive that data, in an order in which the customer feels comfortable; and without requiring data which they cannot provide.

Add to this data validation, such as postal validation, postal code length checks and so on, and a dynamic dialogue with the customer where any doubt exists or double checking is required, and data quality improves many times over. Though cost outlays are greater in the setup phase, the savings downstream, when your company is able to rely on optimal data quality, repays these many times over.

The Solution

I've decided to present my solution to poor international data form design at the start of this book. The reasoning behind these suggestions will become clear as you work through the book and check out the examples. My solution is that:

- You should ask one or more filter questions before loading the page containing the form, or before loading the form if it is on the same page as the filter questions. These must include the country where the person is (or resides, depending upon the purpose of the form).

- Those filter questions may include one asking for the language in which the form is to be presented if you have the form available in more than one. You should never automatically present a form in a national language of the country, but leave that choice to the customer.

- If your form is aimed at both consumers and businesses, and contains fields relevant to only businesses, such as company name, job title and so on, you might also ask whether the customer is filling in the form in a private or business capacity.

- On the basis of the answers given, a form is built which contains only the fields which are relevant to that customer. It contains those fields in the order that is familiar and logical for that person so that their flow through the form is as natural and as frictionless as possible.

- You avoid using automatic tabbing.

- The field labels are in the language requested, and/or are adjusted for the country concerned. They are clear and unambiguous. Where required, an example and/or additional explanation is provided.

- No fields appear on the form that are not relevant and/or which the customer cannot complete.

- No required fields are demanded that the customer is unable to provide a response to. Every customer must be able to provide a response before a field is set to be required.

- Field lengths and field length constraints, validation and formatting rules are all adjusted for the country concerned. Data is processed as much as possible on-the-fly so that no, or as few as possible, formatting demands are placed upon the customer.

- Multiple choice questions (e.g. drop downs and related form elements) contain items relevant for the country concerned, in a logical order. Drop downs are only used where the number of options is finite and all options exist on the list.

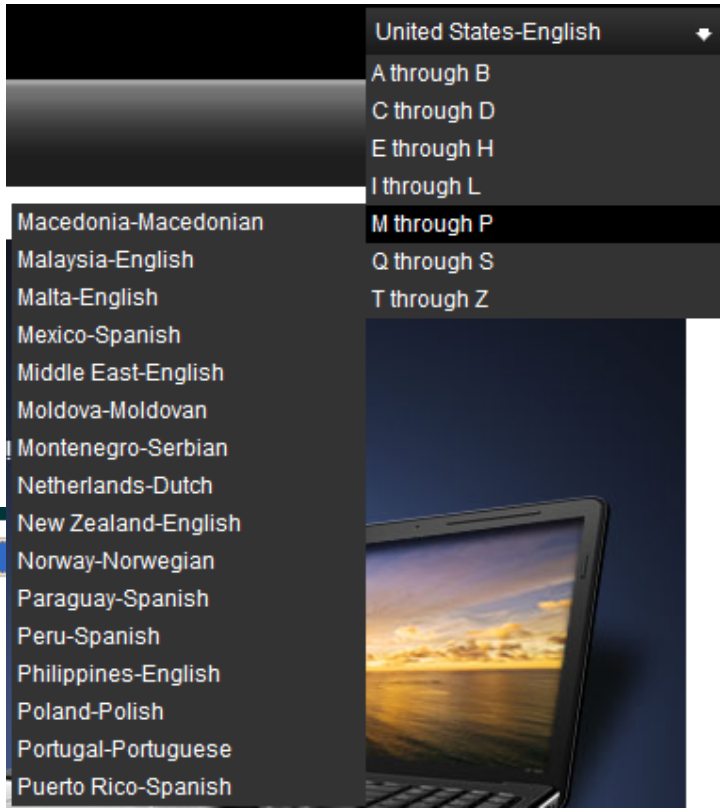
- Error messages are clear, unambiguous, helpful and guide the customer to the solution. They appear in a position on the screen that the customer can see. As far as possible, hold a dialogue with your customer.

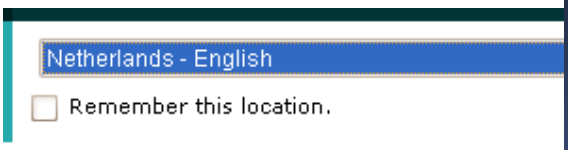
- The form contains no spelling or grammatical errors.

- The form has been thoroughly tested, and one or more people hold responsibility for ensuring that the form and its components remain valid and relevant as the world changes.

The solution

Many companies now allow customers to choose their preferred country and/or language on their websites. Though this leads to localized web pages, in most cases the level of localization of the web forms is very limited, with often only the language being altered – the layout, formatting, validation routines and so on are often left untouched, to the detriment of both the long-suffering customers and your company's success.





Data quality and your form

For a full discussion of international data management, database structures and further, please refer to *Practical International Data Quality*, by Graham Rhind, Gower, Aldershot, 2001. See http://www.grcdi.nl/book3.htm.

One of the first questions you should resolve before designing the form is for whom it is being designed. Are you designing it for your customer? Or are you designing it for your database? Usually your answer will lie on a continuum somewhere between these extremes.

Related to this is your priority regarding the quality of the data. Must data quality be maintained at all cost regardless of the burden that is placed on the customer to achieve this? Or is it more important that the customer completes the form and the quality of what they enter can take a back seat? The former might be the case for a company from which a product needs to be sent, so the address should be as correct as possible to reduce costly returns; whilst a political party might prefer the latter approach as it is interested in getting you on board above and beyond the data quality issues.

The way that a customer perceives what will happen to the data, and their stake holding within it, is also highly relevant. Few customers who have ordered an expensive product from you mind spending a little more effort in getting their details correct as it is in their interest that the product arrives correctly and expeditiously. If asked, however, for information that is not deemed relevant to the task in hand, such as being asked for a postal address when requesting an e-mailed newsletter, many customers will either peel off from the form or (deliberately or through complacency) provide bad data.

Bear in mind also that you can always ask for more information of your customers at a later date, though this should be additional to what you already know and not a re-request for data because of poor form design or back-end validation routines!

Before you start designing your form, you need to be aware of the data quality issues at stake, and how your data will be stored: for international name and address data this is a far more complex story than for national name and address data. These decisions will affect the validation rules that are built into a form. Some companies, for example, strip punctuation from data being entered into their databases, only to find at a later date that their data would have been better and easier to process had it been retained.

To achieve the highest quality data possible, both as a service to your customer but also to enable you to use the data profitably in your business, you need to collect data that is:

- Relevant
- Complete
- Up to date (timely)
- Accurate and correct
- Consistent
- Duplicate poor/unique

To achieve this, integrate these ideas into your processes:

Knowledge

Understand how name and address systems work, what components they contain, and how they are put together, both in your native country and for international data. Only through knowledge will you be able to design systems and storage, and the forms to go on top of them, to recognize and correctly manage each part of the data.

Data storage

The way that personal name and address data is stored is largely dependent upon the uses to which you will put that data, but to some extent it will usually be parsed into its component parts and stored in that way. For example, a telephone number may consist of a country code, an area code and a subscriber's number. As each can change separately from the other (individually, such as the subscriber's number, or as a range, such as the area code), they should be stored and managed in separate fields in a database. This general rule can also be applied to addresses, but to store each component separately for every address type in the world you would need over 100 fields. This is unworkable, and is not required for most uses to which a company will put that data. Those components which are used operationally: a postal code for searching, or a building number for de-duplication, for example, should be stored separately. Follow these guidelines:

- Ensure that the system is able to correctly hold data from any address in the world (if working internationally) or in your own country (if working nationally).
- The data should be parsed into different fields to such an extent that it is possible to use certain key elements, such as thoroughfare name and house number, in processes such as de-duplication and searching.
- The data should not be so fragmented that you, your customer and others working with the data need to have an encyclopaedic knowledge to be able to sort out which piece of information goes where.

The data should also not be so fragmented that output of the data in the correct address-block format is no longer practical.

Your decisions must be based on how you manage the data and how you can effectively collect that data without additional burdens upon the customer. For example, you

might collect the street address line in a single field, including the building number. This makes the entry easy for the customer, but any validation software that you use may be required to locate and parse the building number on-the-fly, which it will not always be able to do correctly. On the other hand, by moving the onus to the customer – for example, by requesting that they add the building number in a different field to the street address – you are increasing the level of difficulty involved for them to complete the form, but also adding an additional degree of complexity which may in itself cause errors, for example if the customer enters the building number information in the wrong field.

Ideally you would test each of these options to decide which works best for you and, importantly, for your customers.

Unless the customer data is never to be seen by the customer, and if you use it for any other purpose than for sending items via the post which benefit from a discounted postal rate when an address is written in a certain way, and where speed of delivery is important (for example, sending printed publications through the post), the data should always be standardised according to the cultural norms of the customer. In other words, avoid storing a person's personal name and address data all in upper case and full of abbreviations where the customer themselves would prefer a mixed-case version with the words written in full.

It is better to write and store data in full than to use abbreviations. Abbreviations don't travel well – they mean different things to different people and in different cultures. Operationally, data which is written in full is easy to abbreviate, but abbreviated data cannot always be transformed back to its full state – does that *ST* in the address mean *Saint* or *Street*?

Allow punctuation in your data. This makes the data far more readable for customers, but it also makes far more sense operationally. Commas show lists, full stops show abbreviations and so on, and this is useful to know when parsing data.

Store and output your data in mixed-case. This is far more pleasing and readable for the customer, and there is no good operational reason not to do this. Upper-case data is easier to use in processes such as matching and searching, but this can be programmed to be done in memory, leaving the data in its correct state in the data table. Also very important to note is that upper-casing some letters with diacritical marks may result in the loss of that mark as not all code-pages contain upper-case equivalents of lower-cased accented characters. Once a diacritical mark has been lost, it is nigh on impossible to automatically restore it. You can upper case *Kölnstraße* to *KÖLNSTRASSE*, for example, but you can not apply logic via a computer program to know that *KÖLNSTRASSE* should become *Kölnstraße* in mixed-case without potentially causing errors, as "ss" can occur also in mixed-case words in that language.

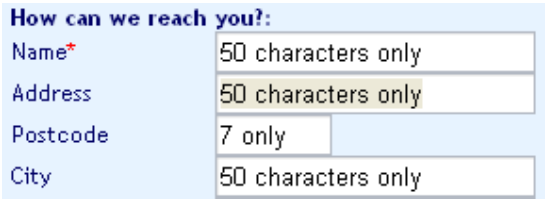
Diacritical marks should always be used. For speakers of most languages they are an important aspect of being able to understand and read text, and incorrect or missing marks don't just make a text less readable, they can change the meaning of the text in negative ways that can rebound very badly upon you. It can be significant to your customer if, for example, in German, you use *Schwul* (gay) in your communication with

them, rather than *schwül* (humid, sultry)!

Always used localised (local-language) forms, particularly of place names. Some companies still think it normal to store place names such as *Köln* as Cologne and *Milano* as Milan, and go to a great deal of expense to do this. There is no reason to do this and it is guaranteed to annoy your customers.

Use full stops when writing acronyms or abbreviations such as B.B.C. Operationally, this allows your systems to see that this is an abbreviation and, for example, to leave the data in upper case. No system can recognise without corroboration that the *CAR* in *CAR & BODYWORK GARAGE* may be an acronym for *Corinthia Automobile Repairs* and that the string would be cased correctly as *CAR & Bodywork Garage* rather than *Car & Bodywork Garage*. This would be recognisable if the string were written *C.A.R. & BODYWORK GARAGE.* If you prefer to output acronyms and abbreviations without full stops, this is easy to program for output.

Field lengths

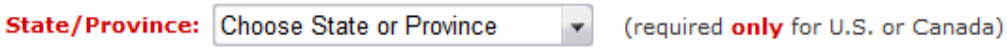


How can we reach you? Not with the information gathered here! This form, on a Dutch website, but aimed at a global audience, restricts input within the form, presumably reflecting the database table field lengths to which the data will be written. The longest postal code in use anywhere, at the time of going to press, is 10. Postal codes within the Netherlands have a length of 7. One can imagine the programmer of this form counting the digits in his home postal code and applying that to the form. It clearly hasn't been properly tested, and nobody has noticed how much truncated (and near useless) data they will be collecting.

Never use numeric fields to store any data which does not need calculations applied to it. Companies storing, for example, telephone number and postal code data in numeric fields soon find that they are losing the initial 0 of their data through this form of storage.

Ensure that the fields are long enough to contain your customer's data. The longest place name in your country might be 30 characters or so, but the longest in the world has 163 characters; people's names and job titles can be long and complicated; the longest country name at the time of writing has 52 characters, and so on. If in doubt, leave more space than might be required. This will greatly aid your data quality, preventing customer irritation and the need for unusual and inconsistent abbreviating or truncating of data.

Synergy between the form and the database



Whilst, thankfully, this form did not require me to add a state that I do not have in my address, the invoice that I received contained:

AMSTERDAM
ZZ
1018 VV

clearly demonstrating that the form is sitting on top of a database that requires that the state field is not left empty.

Personal names

Postal addresses are defined by national systems, based on postal and cultural norms. The nationality or origins of the person living at an address will not affect the way that the address is written and formatted.

This is not the case with personal names. Personal names are not written in one way in one country, and a different way in another. Our mobile world has ensured that the forty or so personal name formats found in the world will be found almost everywhere. You cannot assume that a person in China uses a Chinese naming pattern, a person in Egypt uses a Muslim naming pattern, and so on.

This creates a challenge when requesting a name on a web form.

Many web forms request the name be split into its component parts upon entry, and sometimes this is so out of habit – because others do it. You should, however, ask yourself whether you really will use the personal name data gathered in a way which requires it to be separated.

Many web forms where the resulting form data is to be e-mailed to a company, for example, request both a "first" name and a "last" name, including, until recently, the form on my own website. If asked, most companies would respond that they need to separate that information so that they can respond with either "Dear Graham" or "Dear Mr Rhind". Does this requirement weigh up well against the problems posed by the web form for people whose name does not fit this first name/last name pattern; and the additional data quality issues and potential loss of customers in asking for two fields instead of one to be completed?

In my case, as I do not add information from my web form to a database, and when I had honed my programming skills enough to remove that extra field, I found that I could usually work out from the submitted data which part of a name is which; and when I cannot, "Dear Graham Rhind" does not usually provoke a negative reaction.

If you do not need to store personal name data in separate fields, it is an option to use one field: "Name", and this method of collecting personal name data is used successfully on some e-commerce sites. This allows the customer to write their name and any associated data in the way they want to – with or without a form of address, with or without seniority or academic qualification, with or without middle initials, and so on.[7]

Do not choose this option if you need to store different personal name elements in different fields: the processing of personal name data should be very limited, if it is to be done at all. You may be able to identify name-related data such as forms of address and academic titles, but attempts to identify and split given names and surnames or to assign genders on the basis of names will fail and do fail.

[7] For a discussion of this issue, please see http://www.siliconglen.com/usability/courtesytitles.html

Is that Christopher Robin or Robin Christopher? Cliff Richard or Richard Cliff? George Michael or Michael George? Where do I split the name Tim Brook Taylor to get a given name and a surname? Is that a female American *Jean* or a male French *Jean*? Is that a male Italian *Nicola* or a female British *Nicola*? In every case, without corroborative evidence, there's no way of knowing, so the decision on how to collect and store personal names needs to be made at the data entry stage and not later.[8]

If you do need to store personal name data in its parsed form, be aware of some common errors. Never use field labels on forms with an international audience (solely) indicating relative position of name elements - *prefix, first name, last name, suffix* – unless your only interest is to store the elements in the correct order rather than collecting the same data in the same field.

To clarify this: Most web forms requesting a *prefix* are actually asking you for your form of address (Mr, Mrs etc) as Anglo-Saxons write that in front of their personal names. In the *suffix* they would expect a seniority indicator (Senior, III) or an academic qualification (BA, Ph.D.).

Yet a Japanese customer will write their form of address after their names (and concatenated to it, so it will be most strange to them to be expected to split that off from their name). Germans will write their academic qualifications before their names. I will write my given name first, but a Hungarian or Chinese customer will write it last. Using these field labels could, therefore, cause confusion for your customer and increase the chance that their data is added to the wrong field within the form (and therefore your database), greatly reducing its use to you. If you are using a single form with one set of generic field labels, use terms such as form of address, given name, surname/family name, academic qualification, seniority and so on.

As always, don't be afraid to use longer field labels which explain what you are trying to collect, and examples: *Given name/first name*, for example, if your customers may not be aware what is meant by *given name*.

Equally, do not expect all people to have both a given name and a surname – many people do not (most of the population of Indonesia, for example). Having both a given name and a surname as required fields (and few forms do not) does cause problems for some customers who have to make something up to add to the surname field. Many also have to make middle initials up as a large number of us don't have one.

A large percentage of the world's population do not use the personal name format *given name/surname*, and when asked to split their names are forced to make an arbitrary split, which may vary from name form to name form. This is a common problem in East Asian names and Muslim names.

The message must be that thought needs to be given to how you collect personal names in your web form, before it is put online, as once a name is collected it is impossible to correct

[8] A full discussion of personal names and the dangers of over-processing them can be found in *Practical International Data Quality*, by Graham Rhind, Gower, Aldershot, 2001. See http://www.grcdi.nl/book3.htm

automatically.

Mailing address or street address

Postal systems have two main delivery methodologies, reflected in addresses. The first is to a building, such as a company office or a residence, and these have so-called *street addresses*. The second is to post office boxes, post offices, or through other special routing systems – these have so-called *mailing addresses*.

Many countries have postal deliveries to street addresses. Others have deliveries only to mailing addresses. Many countries have both.

Which do you ask for on your form?

If you are delivering a product and are not able to do that to a mailing address, this should be clearly stated on the form (before you ask for the address!).

In other cases, if you provide a single form which asks for the address, you need to be prepared to get both street addresses and mailing addresses provided through that form. If you provide a separate field for building number, some customers will fill that in with the post office box or mailing bag number, whilst others will write the mailing address in full within the street name field, including the number.

The mailing address field in your database, that created to hold a post office box number or similar, should never be numeric – some mailing addresses in the world contain letters as well as digits.

Mailing addresses

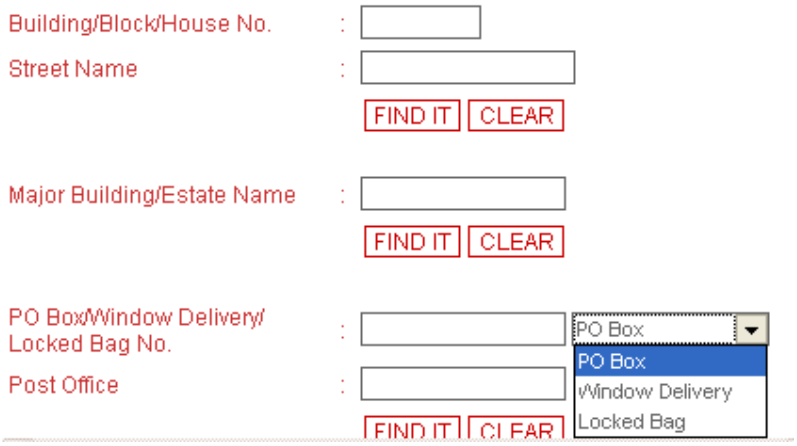
When customers without street addresses have to put their mailing address into a standard street address form, the inevitable result is data which floats between fields, as in this example. Making an option for the customer to add their mailing address data to a single field with a suitable label:

prevents this problem. As presenting customers with multiple fields will always reduce data quality, and producing a clear field label to explain that you would like the mailing address type word (e.g. "P.O. Box") in one field and the number in another is very difficult, it is better to collect the information (the number alone or the number with the mailing address type word) in a single field, and to post-process that data to

standardise it.[9]

If you need to know whether an address is a street address or a mailing address, ask your customer to choose whether they will add a mailing address or a street address (and define them clearly). On that basis alter the input form to increase the ease of completion for the customer and to improve the quality of the captured data for you. You can choose to store that data in a new field or fields, or you can set a flag showing the type of address, and reuse fields (such as thoroughfare name and building number) for this information.

Mailing addresses



This form clearly distinguishes between street addresses and mailing addresses.

---

[9] Address element reference files, such as that described at http://www.grcdi.nl/addresses.htm , can be used for such post-processing work.

Dynamic data-collection forms[10]

It is remarkable how many companies cannot get past the idea that an online Internet page or other digital data collection form has the same possibilities as a form printed on a piece of paper. A form on paper must necessarily be static, fixed and unchanging. Those on the Internet have all the digital possibilities to be dynamic. Unfortunately, most of the forms you will encounter on the Internet are as static as their paper-based counterparts.

The world's 240 or so countries and territories use around 140 address formats, and the world's populace have almost 40 different ways of writing their personal names. A one-size-fits-all approach to web form design to capture this information simply will not work. Those of us trying to fill in the forms have known this for a long time.

Static forms

Static one-size-fits-all forms to collect personal name and address data, such as this one, result in high customer dissatisfaction and low data quality when applied to customers from different countries and with different cultural backgrounds. Even with my Anglo-Saxon heritage, giving me no problems with most forms when a personal name is requested, there will always be other issues which slow or block my progress through the form, making its completion a hassle rather than a smooth and thought-free flow. The order in which the fields are shown are not those that I am used to for my Dutch address, so I must stop and think about what the web designers require (or add the wrong data to the wrong field). Even if I wanted to add the (irrelevant, in addressing terms) province where I live, the State/province drop down does not contain it, so I must state that I live in Arkansas or Yukon Territory, and your company will have to sort that data quality issue out later. If I lived in a country without postal codes, such as Ireland[11], I would have to add something there just to move on – 00000, perhaps, or 90210 … Thus, my address:

[10] Although I refer to forms in the plural, as though you have a different form on many different web pages, and you choose to send a customer to the appropriate page on the basis of their location, language etc., you can naturally program a single form to be dynamic – as long as it appears to the customer in the way that is ideal for them to provide you with their data.

[11] Most Irish will tell you that Dublin has postal codes, but these are, in fact, sorting codes.

Nieuwe Prinsengracht 80-hs
1018 VV AMSTERDAM

will enter your database as

Nieuwe Prinsengracht 80-hs
AMSTERDAM 1018 VV YT

The further addresses deviate in pattern and contents from that in the form and the worse the experience is for the customer, the more that will leave that page at that point without completing your web form, and the worse will be the data collected.

Sites can recognise me when I visit their pages. They can inform me of what I have bought, what I might like to buy, and alter their pages according to my browsing history. They can tell me the time and the weather where I am.

And yet these same sites have proved themselves incapable of creating a dynamic form for information gathering which adjusts itself to the needs of the customer and to their location, or, more often, even understanding that there is a need for data-collection forms to be dynamic. The site from which I order my books, based in the UK because they have no Dutch site, and held up by all as a beacon of dynamic usability, still cannot correctly manage my Dutch address in either its forms or on the products it sends out.

There are a number of preconditions (i.e. givens, something you and your customer have no control over) relating to the data you are attempting to capture, that most companies are either unaware of or choose to ignore. Given these conditions, a static form is clearly going to give problems for the customer. Unless they come from the country, use the language and have name and address formats for which the form was designed, they will always have issues when trying to enter their details.

Looking at most company websites, one would be forgiven for believing that the whole world conformed to the same patterns as those within the host country of the company - that the whole world writes their name just so, that they all have ZIP codes, that they all have states, and so on.

Static forms (or a single one-address-structure-fits-all form) increase the friction between your customers and what they, and you, want to achieve through the form. The usability of the form is decreased the moment your customer comes from a place other than the nation whose patterns you have fixed onto that form.

In a form which is dynamic, a number of aspects of the form can be altered on the basis of what is known about your customer and what they enter.

The most important aspects of the form which can be dynamically defined are:

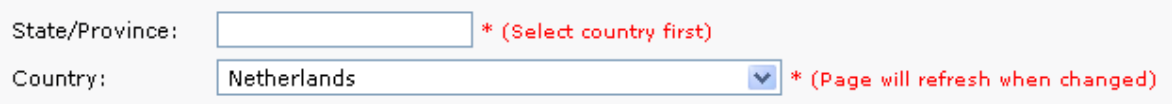
- The language of the field labels, based upon:
    - the language that the user requests to see the form in (and NOT the language

of the country in which the customer is to be found at that moment). Many dynamic systems make an assumption that a person in country X speaks the national language of country X. This assumption not only ignores minority linguistic groups, visitors, holiday makers and so on, but also the mobile nature of the world's inhabitants – many people live in countries where they do not speak (one of the) national language(s).

- Which information the customer is asked for and which they are not, based upon the norms of the country concerned. Asking customers for their state and postal code when they have neither will irritate the customer; and
- The order and layout in which the fields appear on the screen

---

Flawed dynamic forms



That's great! So why make the customer jump around the form in this way? Why not just swap the position of the fields on the page?

This form should have informed me of this before I scrolled through the list of American states and Canadian provinces and territories!

Ideally, the form layout should reflect as much as possible the familiar pattern of the data being gathered, for example the order in which a personal name is written, or an address block. Though I favour forms that recreate a familiar pattern on the screen, such as an address block, some usability experts counsel against putting fields next to each other horizontally. Whatever you decide, it is important that the fields are in the correct (familiar) order for that customer.

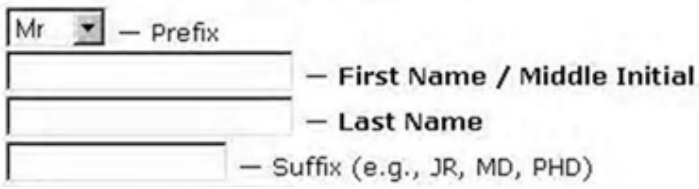
Consider how your customer will read your page and form. If they speak a language using the Latin script, for example, they will read normally from top left to bottom right, and this will be the order of field completion which will be natural for them. They will see field labels written to the left or above the fields[12], and be confused, or pass over, by those written elsewhere. Bear in mind, though, that people using languages and scripts with different writing directions, such as east Asian pictogrammic scripts, Arabic and Hebrew, will need your forms to be laid out differently, for example with the field labels

[12] For an example of the difference label placement in forms can make for usability and data quality, see http://www.uxmatters.com/mt/archives/2006/07/label-placement-in-forms.php

to the right of the fields.

If you will be presenting your form in more than one language, it is often useful to place the field labels above the fields so that the different lengths of the field labels in different languages do not alter the field placement. This is also a useful option if you recreate an address block structure in your form with fields next to each other horizontally on the form.

Reading order



The order in which elements are placed on the form applies not only to fields. The elements should follow our natural reading order – in the case of most languages written in Latin script, from top left to bottom right. By placing the field label after rather than before its field, this form forces the customer to flick their attention from field to label and back to the field in an unnatural way, increasing frustration but also encouraging error.

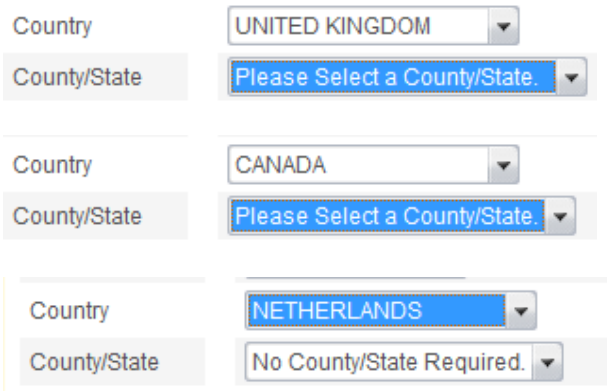
In a static form your customers will be asked to enter their details in an unfamiliar order. They will have demanded of them that they add data that they do not have (such as a state or a postal code), and often there is no place in the form for data they do need to provide, which they then must either leave it out or add into a space or field where it does not belong.

If this customer does complete the form, it will cause all sorts of data headaches for you, though it often takes some time for that to be realised. The resultant poor quality and duplicate-rich data can only be sub-optimally corrected with expensive software or routines. These can produce better data, but can never produce data with a quality that might have been realised had data validation and customer dialogue been a part of a dynamic data entry form.

Your data entry forms should be dynamic. You shouldn't be asking customers to provide information which is irrelevant or which they cannot provide. Good examples are asking for a "state" or a "postal code" from customers in countries which do not have them. Ask your customer first for their preferred language and the country in which they reside, and program your input form to dynamically rebuild itself so that the customer is asked only for relevant data, in the order in which they expect to enter it, and with the correct input masks and validations systems working on it. A surprising number of companies ask for the country at the bottom of their web form and then rebuild the form – after the customer has entered all their information.

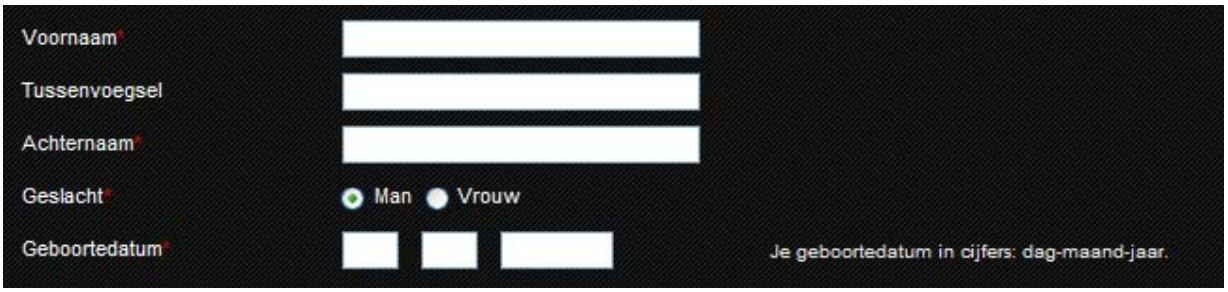
I would advise against creating forms that rebuild (entirely) as your customer adds their information. Many people find it highly disconcerting for the form that they are happily filling in to suddenly disappear and, some time later, re-appear … but not quite the same. Initial panic can be replaced by irritation, despite your good intentions. Don't be afraid of asking enough screening questions (country, language etc.) before building the form, so that the form that is built is correct and requires no other alterations whilst your customer is working on it. Injecting information onto the page, such as a validated address or other feedback, works well, provided that the page doesn't have to rebuild entirely to do this.

It costs more to create a dynamic form - more knowledge is required and the programming is more extensive - but the quality of the results will pay back this initial investment in the shortest time.



This form demonstrates simple dynamic change to the state field according to the country name specified. A further improvement would be if the instructional default field text also changed, to ask for a county in the United Kingdom and a province or territory for Canada. Much better would be for that field not to appear at all for those countries where that data is it not required.

Over-enthusiastic feedback



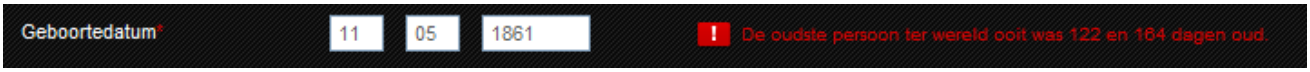
This form is an excellent example of a dynamic page. As can be seen from the screenshots below, it gives clear and graphic feedback to ensure that the customer fills in the form correctly:

Translation: Fill at least 6 characters in

Translation: This username is still available

It does, though, suffer from "over-enthusiastic" feedback – it fails to wait for a field to be completed before flagging up errors:

The error that the date is not in the correct format is flagged while the customer is still typing, causing a moment of panic. It does, though, have an amusing way of flagging up any obvious errors:



Translation: The oldest person who has ever lived was 122 years and 164 days old.

Data validation[13]

Validation can be an impediment to users, but it need not be. It is only an impediment when it has been poorly thought out or implemented, forming a barrier between your customer and them completing the form, such as when it does not take account of international differences.

Make a rule: validate data upon entry wherever and whenever possible. This is far more effective, and a great deal cheaper, than validating after data collection. Validation upon entry also allows a dialogue with your customer to ensure maximum data accuracy.

Validate anything that can be validated at the entry stage, on any characteristic. You may not have the facility to know, for example, whether a telephone number exists, but if it contains more digits than are possible for that country, this can be flagged and corrected through feedback and dialogue with your customers. Check that that a bank account number passes validation tests, that addresses are postally valid and correctly formatted, and so on.

If you add validation to a field, make sure that the customer knows any limits to the number of characters etc. – this is better than punishing them afterwards with an error message.

Validation

On-the-fly validation prevents examples like this one, where I carefully dictated the spelling of my Dutch address to a British call centre operative. Every element is incorrect – name, street name, place name, layout – except the postal code (though the format is wrong) and the building number. Without those two pieces of information, this delivery would never have arrived.

Don't be afraid to partially validate when full information is not available. If you are not

[13] By this I mean checks that the data being added, as far as can be tested, reflects the real-world constructs to which it refers; and/or that the information is fit for purpose. For a fuller discussion of this, and a comparison with *verification*, see http://en.wikipedia.org/wiki/Verification_and_Validation

able to validate that a bank account number is valid, for example, but do know the maximum length of a bank account number in that country, validate for that – even partial validation will greatly improve data quality.

Data cleansed after collection can never have the same quality as data cleansed during collection. During collection there exists the opportunity to have a dialogue with your customer. Any errors, typos, incomplete fields, unvalidated information and so on can be fed back to the customer for immediate correction. Once the form has been completed and the data sent to the database, you need to rely mainly on batch cleansing mechanisms to clean any errors. Without the dialogue with the customer, the level of correction can never be as high, and in some cases errors can be introduced into the data through over-zealous processing.

States … again

Data validation need not be complex or costly. Upon being asked in a form, yet again, for a State, in my advanced condition of frustration I typed "We don't have states …". A simple validation routine could have checked whether I had entered more than an allowable two letters, and, if two letters were entered, that they were valid. Without a routine, the data was accepted, the database polluted, and my mailing piece received with my comment intact (see above). My confidence level in this (short-lived) software company was not enhanced by this experience!

Let's take an example. A customer enters:

S. Speelberg St 17

A validation routine might find and present a corrected version:

Stephen Spielberg Street 17

or request clarification from the customer. Without validation, this will be injected as is into your database. Any cleansing software run over this record at a later date will need to decide what this street is without the benefit of a dialogue with your customers. It may find and correct this address, but it might make errors:

South Speelberg Saint

Data cleansing after collection is expensive, not just in terms of the software, hardware and so on, but also in terms of the loss of opportunity that you would have had had your data already been clean. The extent of the challenge of cleansing data which has been collected without cleansing at the data collection stage should not be under-estimated. In some cases, the quality has proved to be so poor that companies have chosen to discard all data already collected and to start the process again – with improved data collection procedures!

Consistency should always be strived for. Decide upon formats. Even if a format is found to be incorrect, it is easier to correct consistent data at a later stage to achieve data quality.

Avoid the processing or over-processing of personal name data and company data. Names are highly personal and have huge varieties of spelling, punctuation and casing. Processing these to standardise them (and, in the process, destroying their accuracy) will be highly detrimental to your processes and your relationship with your customer. Gender can generally not be accurately derived from a name, especially not when international data is included. If gender information is required, it should be a question on your web form, and not assigned later. If it is imperative that you collect personal name data accurately, this can only be done at the data collection stage.

Over-processing

	Original Address	Corrected Address
Name:	Graham Rhind	Mr. Graham Rhind
Title:	Owner	Owner
Company name:	GRC Database Information	Grc Database Information

This company neatly fed back my validated postal address for confirmation, but by processing other information they introduced errors which were not in the original data – in this case, they "corrected" the casing of my company name and assigned a form of address that I prefer not to use.

Rapid addressing

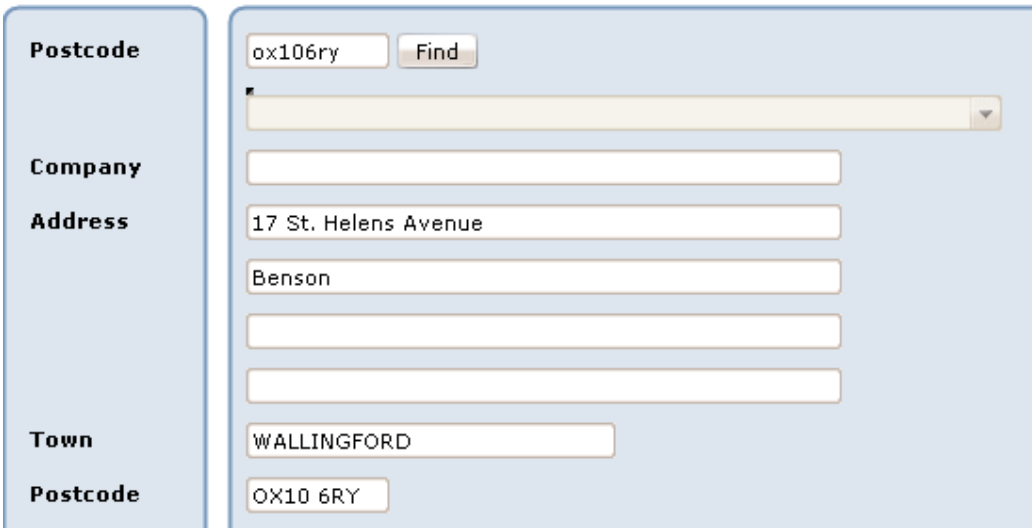
For a limited number of countries, online forms can be built upon postal validation systems to allow rapid addressing. In these cases the form does not need to follow the input norms of the country concerned, but asks for a number of pertinent address elements, allowing the software to return a full address, thus saving keystrokes for the customer and improving quality for you.[14]

Rapid addressing

In the form below, for customers in the United Kingdom, only a postal code needs to be added, after which the customer can choose their particular address from the drop down. As rapid addressing systems are never

[14] For another look at rapid addressing in forms (and a source of some of the examples in this book!), check out http://www.goodusability.co.uk/2009/03/using-address-finders-in-web-forms/

complete or completely accurate, the customer is allowed to modify or enter their own data if the rapid addressing software fails to find a match.



This form provides a drop down of possible addresses as I type my information into it, reducing keystrokes for the user and increasing the quality of the data collected. This can be applied to other fields, such as forms of address or job titles.

For example, for addresses in Canada or The Netherlands, the postal code and the house number only can be requested, and should be sufficient to return a full address.

Note, though, that the number of countries where rapid addressing is possible using a very small number of address elements is limited.

Be sure to provide feedback to the customer so that they can check that the returned address is the correct one – they could easily have made a typo in their details which, if not checked and corrected, could invalidate their data.

Postally formatted addresses (for example, all upper case, abbreviated, without diacritical marks) are not always the way the customer likes to see their information written – allow the customer some leeway to alter formats.

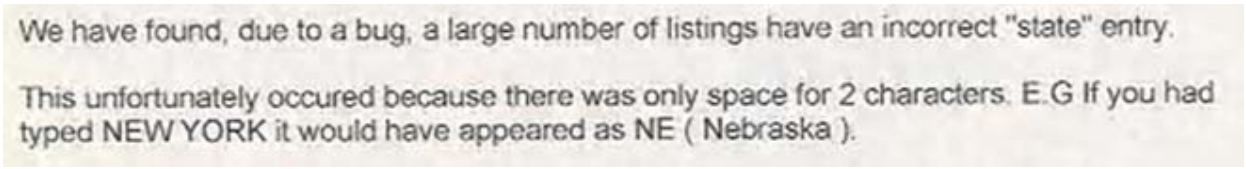
There is a time lag before new and altered addresses enter postal databases, a further time lag before those addresses enter the software provider's database, and, if you use in-house software rather than a web service, a further time lag before that change is delivered to you. If the address that the customer enters cannot be found in the postal database, have a form prepared to allow the customer to add their address manually – formatted and

ordered in the correct manner for that country and language region!

Providing an escape route

This form is clear and well designed, and provides an escape route for customers to add their address manually should the rapid addressing software fail to work. One blogger reported his frustrations at failing to be able to have his address validated by online rapid addressing software only because he used a full stop after ST. In that case he was not able to add his address manually, and the company concerned lost a customer.

Embarrassment



Lack of validation at entry often means that the only way to correct data is to grovel back to the customer for information, as this company did.

Multilingual or unilingual? Languages

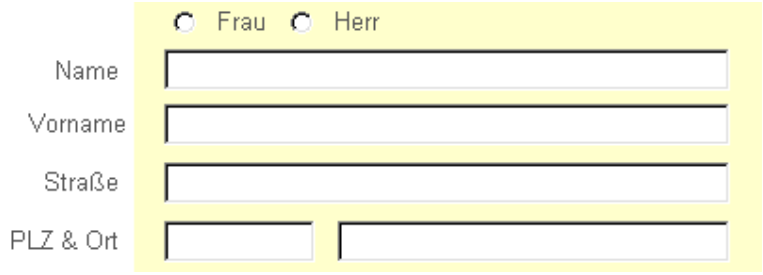
Your web forms should be multilingual, as far as is possible, to take account of the various languages spoken in the country or countries in which you operate. However, consider that the language that an individual speaks is often not the same as that of the country in which they are situated at that moment – we live in a mobile world. Be sure to ask your customer for their preferred language rather than assuming it.

There are over 6000 languages still spoken on our planet. You will regularly be faced with several hundred of these when you collect data from your international customer base. Choosing to maintain your web form in a single language will have clear implications for the quality of the data collected:

- However widespread the language, there will be large numbers of people who do not speak or understand that language. They will either not be able to fill in the form, or will make an attempt to fill it in, making errors and polluting your data.
- However clear and explanatory field labels are, people for whom the form language is a second language will make mistakes and data corruption is inevitable.

Matching the form and its language to the country

It's not only American companies who are guilty of forgetting that there is a world beyond their country's borders. A German food manufacturer sells its products all over the world, and prints its web site URL on its product packaging. For many years, however, should you have wished to contact the company for any reason, you would have been faced with a form in German, optimised for German addresses. The message seemed to be that the company had no interest in any customer outside Germany.



It took some years for the company to realise how this was affecting their image and their customer relations. They then overhauled their web site to ask the customer in which country they are (though not their preferred language), and then presenting the customer with a form optimized for that country (though not in the preferred language of that customer).

This will have a positive effect on their customer relations and on the quality of the data they are collecting.

Clearly, a multilingual approach to a web form will increase data quality, as your customers will better understand what data that you are requesting from them, the field labels and so on. If you choose to make your form multilingual, you must be sure that you are capable of reading, understanding and responding to customer input in each of those languages. Posting a web form in Finnish and then responding in English is not acceptable!

If choosing the multilingual approach:

- Translate the instructions and field labels, but adapt these to a locality. The localisation of the form is more than a matter of translation. For example, the Dutch used on a form for Belgium might be different to that for The Netherlands; British English differs from American English and so on. In Dutch, *woonplaats* is always used to mean postal town, even though its direct translation is *place of residence.* A direct translation of *postal town* would cause confusion for the customer filling in the form. Using the local idiom increases accuracy and data quality.
- Check your translations. Having somebody who thinks they speak the language have a go at it will usually end up in tears. I've seen "postal code" labelled as "area code" in some forms because the correct term wasn't known, and this will invite customers to enter their telephone area codes.

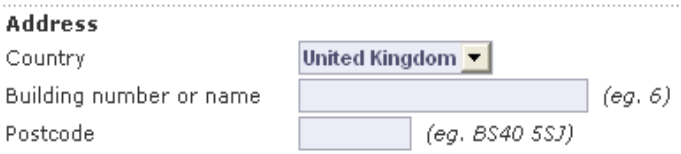
Localisation

Localisation of web forms is about more than just changing the language. In this form, being filled in by British customers, a Zip is requested. ZIP codes are the name given to postal codes in the United States. In Britain they are called Postcodes.

Remember also to match the localisation programming with your form language. This form provided Dutch-

language buttons on an English-language form.

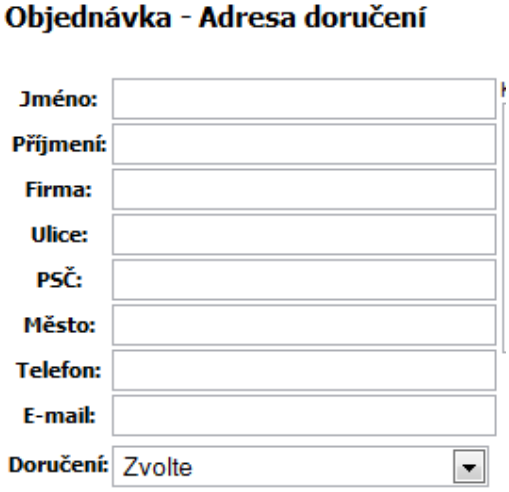
The site below did the same thing with the form's field labels when it found that I was in The Netherlands. It makes no sense to suddenly change languages on the basis of a person's location anyway, but only changing half of the labels clearly indicates that this form was not sufficiently tested!

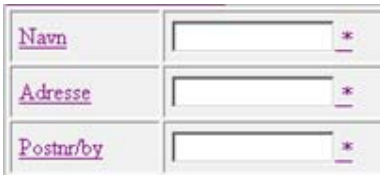


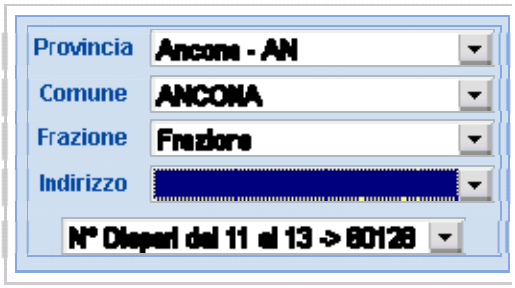
If you work with international data, your system needs to be made ready for data written in different writing systems (alphabetic, pictogrammic etc.), written in different code pages, with different diacritical marks (accents) and written in different directions on the page. Often data has been transliterated (written in a different script than its original form) and, as there are few standardised transliteration systems, this data is often difficult to compare with other transliterations or to the original.

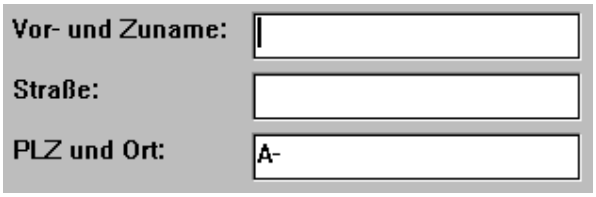
Languages

Forms such as these, in the language chosen by the customer, in a format relevant to the country concerned (not always the case in these examples) improve the clarity of the form for the customer and improve the quality of the data gathered.









A partial success

This form requests a country and language and then partially rebuilds the form to reflect these choices. It is unfortunate that this is only partial (multiple languages on the same form can be very confusing, even if somebody speaks all of those languages) and that it often does this into a language of the country chosen before the customer has had a chance to choose a language. For Finland, for example, this will upset the speakers of the second national language, Swedish, which is not provided as a language option. Furthermore, the label of the field requesting the language choice is also changed, causing problems for those customers who do not understand Finnish.

Clarity

Ensuring clarity for a form within a single country and language region is already a challenge. What may be perfectly obvious to one person is quite unclear to many others. Witness the use of the field label "Title". In one exercise in data collection in a single country where a form of address was expected, 6% still added their job title to that field.

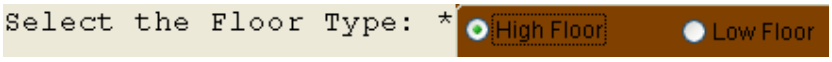
Consider, then, the implications when your customers can be in any country, have a kaleidoscope of cultural backgrounds and speak one or more of any number of languages. Ensuring clarity then takes on a whole new dimension, and without considering it at the design stage, any lack of clarity will have major effects on the quality of the data you will collect.

You can greatly reduce the friction created by a form by concentrating on clarity. The clearer the instructions, field labels, error messages and so on, the easier the form filling becomes for your customer, and the more likely they are to complete the process and to provide you with best quality data.

Ensure that your customer understands why you need certain data and what you intend to do with it. Like many people, I get deeply suspicious when asked for a postal address when I am not asking for something to be sent to me via the post – such as when I wish to ask a technical question in a forum, subscribe to an e-mailed news letter and so on. Ask only for information that you need, and if the need will not be clear to the customer, inform them why you are asking it. This increases trust levels in your company.

Fortunately, our brains work differently

I rarely cease to be amazed by how something can seem obvious to one person yet be a mystery to another. Caroline Jarrett and Gerry Gaffney[15] use this question on a hotel booking form as an example of a clear set of options:



Now, I live in a country full of dinky houses next to canals, with windmills and people wearing clogs. OK, no clogs. And few windmills. But also few high rise buildings. When I first saw this questions (without reading the accompanying text) I could not imagine why somebody booking a hotel would be asked if they wanted the floor in their room to be raised or not – it's not a question I've ever seen on a Dutch site or for any hotel I've ever stayed in. It took some time before I realised that the question was trying to ascertain whether I wanted my room to be on the ground floor (or storey) or a higher floor (or storey). So, whilst this was clear to one set of people, I personally would have re-worded it to make it clearer to more.

No form can be completely clear to everybody who will complete it, but any consideration of how meanings can be interpreted differently by different people will be to the good of your form and the data quality.

---

[15] Caroline Jarrett & Gerry Gaffney, "Forms that Work", 2009, Morgan Kaufmann, Burlington, MA, USA, p. 96

If you need to send error messages to the customer, make it clear what the error is and inform them how they can correct it. Don't just tell that there is an error, and what it is – tell them how to go about fixing it. Some forms simply colour a field red when there has been an error. This is guaranteed to irritate because it informs the customer that something is not right, but not what. (Note also that the colour red does not have the same warning connotation in all cultures, and takes no account of the possibility of colour blindness in your customer). Other pages print error messages high or low on the page, away from the form, so that the customer, who may not see this, is left confused and wondering why they still have the (apparently) completed form in front of them.

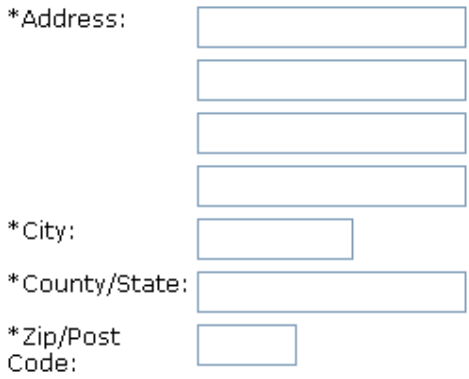
Not scanning

Remember that most people do not scan the whole web form before they start to fill it in. They read from top left to bottom right (if they use Latin script), filling it in as they go. So, when they get to a field like this:

they will fill in their address, like this:

Nieuwe Prinsengracht 80-hs
1018 VV AMSTERDAM

When they continue they find:



As they have already filled in their postal code and city in the "Address" fields, they will either have to go back and correct it (few do), or fill that data in again, causing them frustration and producing duplicated information for your database. "Street address" would be a better, clearer, label in this case.

When it comes to field labels, don't be afraid to explain what is required and to give examples. Usability experts will tell you that short, pithy field labels, and a minimum amount of text, improves the experience for the customer. What that does not do is

help a customer work through a (possibly confusing) form and to provide you with full and correct data. Clarity does not always mean verbosity. *Title* is a short field label which is full of ambiguity. *Job title* is still short but explains exactly what is required. Field labels are discussed in more detail later.

Examples

ENTRER – TOUR – IMMEUBLE – BATIMENT – RESIDENCE – ZONE INDUSTRIELLE

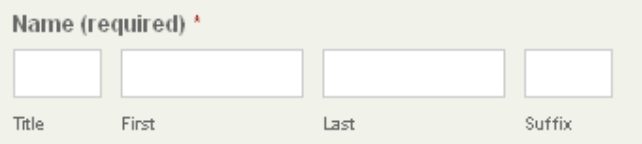
Don't be afraid to make it clear what information you expect in each field, and to use examples.

Relative field lengths, too, can be used to make a form more intuitive for the user – short pieces of information are shown on the screen as short fields, longer pieces as longer fields.

Finally, it is highly advisable to ask the customer to enter the information that you are trying to collect. Many forms are, for example, embarrassed to ask for a customer's gender, and try to get around this by asking for a form of address[16] and then trying to divine their gender from that. This is usually very clumsily done by asking the customer to choose their form of address from one (male) form and one (female) form. This annoys customers whose form of address is not one of these. When customers are able to choose their real form of address, a gender can often not be divined from it because it is gender-neutral – Professor or Doctor, for example.

Your customers are not fools and understand what you are trying to achieve, and often don't appreciate the way you are trying to achieve it. An honest request for the data you are trying to gather generally works better.

Clarity



Clarity is required at all points of the form. On this form, the asterisk indicates that a personal name is required, but then gives four fields in which to enter name data. Are all of these fields required? Or just some of them? And if the latter, which ones? If I have no suffix to my name, can I leave that empty, or will that bring up an error when I press the button to send you my information? Or should I save time by entering something, anything, just to make sure, and in doing so make a mess of your data quality?

Instructions must also be crystal clear. How many times haven't I found instructions like this:

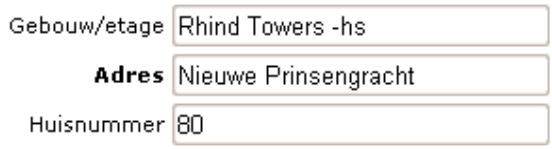
All applicable fields are required.

[16] Known also as *salutation* or *honorific* or, on too many web forms, as *title*.

Applicable to me? Applicable to you? Applicable to my country? What?
How have these companies programmed a check that all applicable fields have been filled in? Through some sort of mind meld between the form and your brain? You can be sure that when you get to the end of a form like this and try to commit your data, the applicability of the fields will become apparent and that it will be the web site's applicability that counts, not that of the customer!

This dynamic form creates confusion because the programmers changed the field labels but not the position of the fields on the screen.

The label "Gebouw/etage" is requesting "Building/floor". In my street address: Nieuwe Prinsengracht 80-hs, the –hs indicates the floor, and it is positioned after the house number in my address. If my house also had a name, such as "Rhind Towers", that would be positioned above the street line. This leaves me with a conundrum on how to fill in this form:



This is not only foreign to me, it is also not what was intended by the designers of the form. The field labels lack clarity.

Asking for what you want

Many web forms attempt to gather one type of information by asking for another. In this case, this company is asking for a form of address when what they are trying to find out is the customer's gender. Your customer is not stupid, and this is usually done in such a clumsy way that it causes more friction than it would do if you had asked the question you wanted the answer to.

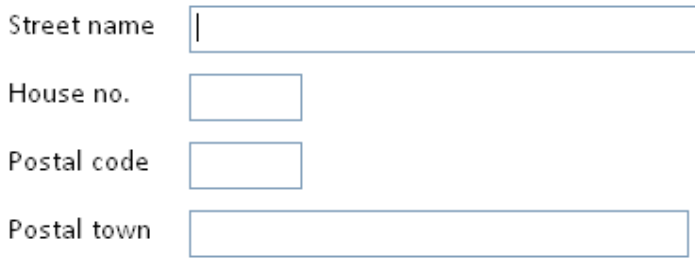
In this case the form will frustrate those customers who have a title, such Dr or Professor, not shown. Secondly, there are data quality implications, as female surgeons, barristers etc. in some countries may be addressed as Mr. It is always best to bite the bullet and to ask for, in this case, a person's gender.

Element layout and tab order

As mentioned above, if you choose to create dynamic forms, the order of the fields on the form can, and should, be adjusted to the cultural and national norms of the customer, so that the field layout reflects the familiar layout of the data being entered.

Field positioning

Replicating the position of the elements of the data you are gathering, for example the address block, in the familiar pattern (as above for The Netherlands), can increase the ease with which the customer can complete the form, and the quality of the data entered, rather than simply making a column of fields in this way:



You should only show on the form those fields that the customer can complete, hiding those which have no relevance. Forms which ask customers to complete a field "if applicable" are missing the point – the field should not be shown if it is not applicable. Equally, forms that change labels, such as the instruction for a state field that it is not required for that country, would have been better if they had been built on the basis of country, so that that field would not have been shown in the first place.

If technically possible, make sure that you position the cursor in the first field automatically when somebody enters your form. Even better, fill the background of the currently active field with a different colour. Every little aid for the customer helps.

Avoid automatic tabbing, that is, having the cursor move to the next field when the required number of characters for input has been reached. This brings inconsistency to the input methodology and confuses the customer. They may tab their way manually through the form and then suddenly find themselves in the wrong field because they did not notice an automatic tab working (not many people look constantly at the screen and the cursor as they add information).

For example, on this form where a Dutch postal code[17] is requested:

[17] Dutch postal codes have the format 9999 AA

the company concerned considered an automatic tab after four digits had been entered. If the customer was not aware that the cursor had already moved to the next field and pressed tab manually, they would then proceed to add the final two letters of the postal code to the wrong field. This might remain unnoticed (with resultant poor quality data gathered), or would require them to back up and correct their data, creating unnecessary friction and frustration.

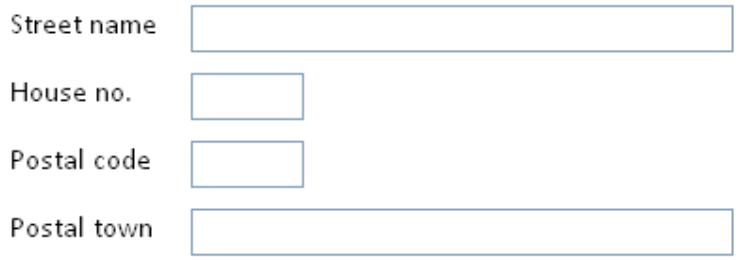
Tab order

This form, already awkwardly formatted for any user, has its tab order from left to right and then down. In other words, the customer who tabs between fields is asked for the information in the order First Name, Company Name, Last Name, Address, Email Address, City Name, Phone, Zip, Country! Unless the customer has their wits about them, this is a sure recipe for the customer data to be inserted in the wrong fields

---

Visually, it is always a good idea that the relative field size on the screen reflects the relative length of that data in reality. Postal codes and building numbers, for example, are generally shorter pieces of information than street addresses and place names. Showing the fields to collect postal codes and building numbers relatively shorter on the form will give an additional clue to the customer about the data expected, reducing errors and increasing data quality.

Relative field sizes



Regardless of the size of the field in your database, the length of the field on the forms can give a visual reflection of the relative length of the data being collected, which aids clarity for the customer. The form above, where the fields collecting the house number and postal codes – relatively short pieces of information – are shown shorter, is much more intuitive than the same fields below which all have the same length. This is often done for design purposes, but cosmetics must take a back seat to clarity in forms.

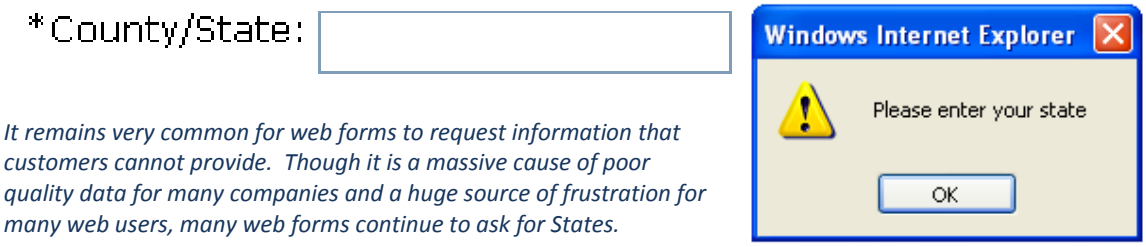
Required fields

Make a field required only if every customer presented with that field on a form is able to provide that information, or be willing and prepared to face the data quality and customer relations penalties resulting from not heeding this advice. Commonly, data collection forms insist, for example, on a postal code and still, far too often, for a state, which locks out large numbers of people. There are many countries without postal code systems (including some economically significant ones), and in some others with postal code systems, the general populace either do not know their own code, or do not even know that there is a system. Many more countries do not have states or provinces, or have them but do not use them in addresses.[18]

Forcing any customer to enter data that they do not have will result in one of two actions:

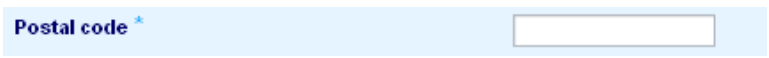
- the customer gives up and leaves the form and your site, with an appropriate alteration in their opinions of your company
- the customer is keen to complete the form and get the goodies, so they enter something, anything, to move on, to the obvious detriment to your data quality.

This is such an issue that you will find regular references to it, and solutions, in support forums all over the Internet.



It remains very common for web forms to request information that customers cannot provide. Though it is a massive cause of poor quality data for many companies and a huge source of frustration for many web users, many web forms continue to ask for States.

Many other examples exist, however. Though most countries have postal codes, many do not. The frustration that this causes is illustrated by the inhabitants of one British Overseas' Territory lobbying to be given a postal code simply to allow them to purchase via the Internet, an option usually closed to them previously by ill-thought out web forms



---

[18] For full information about postal code systems see The Global Sourcebook for Address Data Management – http://www.grcdi.nl/book2.htm

- If you are non-USA, but need to fill in a web site form which asks for a Zip Code, try any of the following and see if it accepts it:

    1. Your Post Code
    2. Your Post Code with zeroes in front to make it 5 digits
    3. n/a
    4. 00000
    5. 90210

Advice abounds all over the Internet on how to get around poorly designed forms with inappropriate required fields. The damage to your data and to your customer's image of you could be greatly reduced with a little planning and forethought.

The mystery of titles

Why, exactly, would a field asking for (in this case) a form of address need to be a required field?[19] *There is a cultural element here – in some cultures it is deemed impolite to address people without a form of address. In others, though, the opposite is true. It is often an individual choice – I, personally, prefer that no titles, prefixed or suffixed, be added to my name. Given the flawed decision making processes currently followed in most businesses, it may be that the original intention was to discover the gender, and that during the design stage stakeholders began suggesting their own gender-neutral additions, such as Dr, but that the required status of the field was never altered or questioned.*

The use of a form of address is a personal preference, and there are no good reasons for not respecting that. To allow those preferring a form of address be used, an optional field requesting "How would you like to be addressed? (e.g. Mr, Mrs …)" or "If you prefer to use a title in front of your name, please enter it here:" would suffice well (though be aware that in some cultures the form of address comes after the name).

---

[19] For an excellent outline and opinion piece on this issue, see http://www.siliconglen.com/usability/courtesytitles.html

Drop downs and other multiple-choice form elements

I have no intention of getting involved in the ongoing debate about whether to use drop downs, radio buttons, check boxes, text fields or hyperlinks for multiple-choice fields in forms. Whichever one(s) you choose to use, there are some rules which should not be overlooked. For the purpose of brevity, I refer to drop downs only, but the rules apply to all multiple-choice form elements.

Do not use drop downs except when the list of options is exhaustive, mutually exclusive, and the drop down contains every option. If you use a drop down and do not, or cannot be sure, that you have included all possible options, use also a text field with the drop down to allow customers to specify their option, if missing from the drop down. Drop downs are commonly used for, for example, forms of address (Mr, Mrs etc.) and job titles. In neither case is it possible to add every eventuality to the drop down, and this frustrates customers.

Exhaustive and mutually exclusive

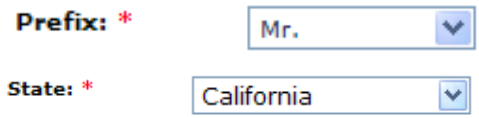
This drop down, a required field on this form, contains the most extensive collection of forms of address which I have yet seen on an Internet form. It contains 224 forms of address stemming from civil life, the armed services, religions, nobility, governments, academia and the judiciary, with some from non-English-speaking countries. Yet still it breaks the rules in that it is not an exhaustive list and the options are not mutually exclusive. I have collected over ten times this number of forms of address without actually making too much effort to find them, so there will always be people whose form of address is not on this list. Equally, were my title to be Professor Sir, how would I indicate this, when I may only choose one?

Every appropriate option should be added – but not more than that. Some lists cannot resist adding extra entries which are irrelevant and confusing. Country lists which include, for example, *European Union*, as well as the countries of the Union individually, allow users to choose one or the other … or both (breaking the rule of mutual exclusivity). Each added entry degrades data quality and increases friction for the customer – they have more options to scroll through to find their choice.

Do not add a default value to fields or have drop downs default to the first item on a list rather than to an empty item. Many companies will tell you how they discovered, to their cost, the tremendous skew this results in with the collected data.

Drop downs should be ordered in a logical way, often alphabetically, in the language in which the form is being read. Customers having to search long and hard for the entry appropriate to them in drop downs is one of the most common complaints, and a source of much data quality reduction.

Pre-selection



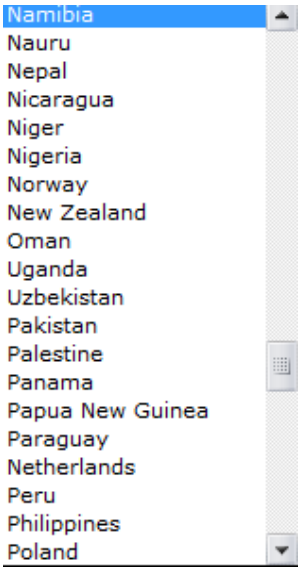
Pre-selecting a choice in a drop down is an invitation to the customer to save their time and effort and to simply skip on to the next field. The owners of this site will find that an abnormally large percentage of the people filling in their form are males from California. Or, at least, appear to be …

List ordering

When using drop downs, ensure that the list is in a logical (usually alphabetic) order. This company has put United Kingdom and United States of America at the top probably because they are its largest markets. It looks as though Japan, at the bottom, was added as an afterthought.

Country lists often show the most bizarre list ordering. This list has most country names in English, but shows Belgium in Dutch (and not in French or German, their other national languages), Denmark in Danish and so on. Switzerland is Die Schweiz, under D, and nowhere near the S where German-speaking Swiss would look for it, and is not in any other of Switzerland's four national languages.

In the example below, the order has been fixed in the original language (French), and not re-ordered when shown in a different language.

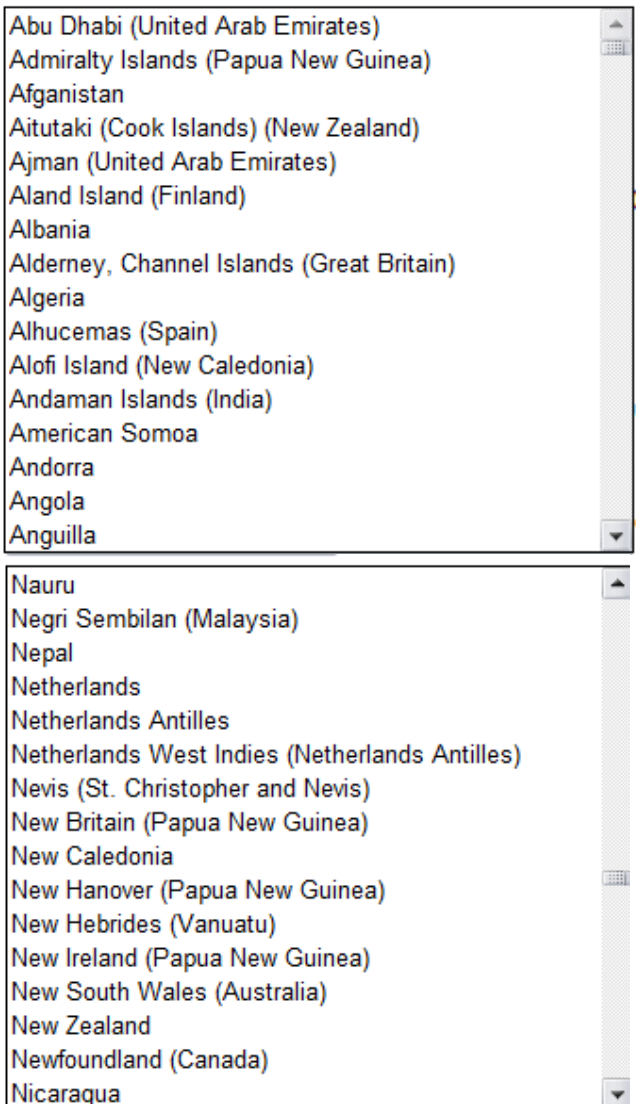


Drop down list contents

The contents of any drop-down list that you place on the form should contain relevant data only, and each entry should be mutually exclusive. This country drop down contains country groupings such as the European Union and the European Monetary Union. This gives me three choices instead of the single choice I should have: The Netherlands. This type of list produces confusion in the user and in the data being gathered.

Overenthusiastic drop downs!

The contents of drop downs need to be complete, and whilst it is better to err on the side of adding too many options than too few, you can have too much of a good thing. This e-commerce site decided that around 240 countries and territories was not enough, and added a range of bizarre content to their lists:



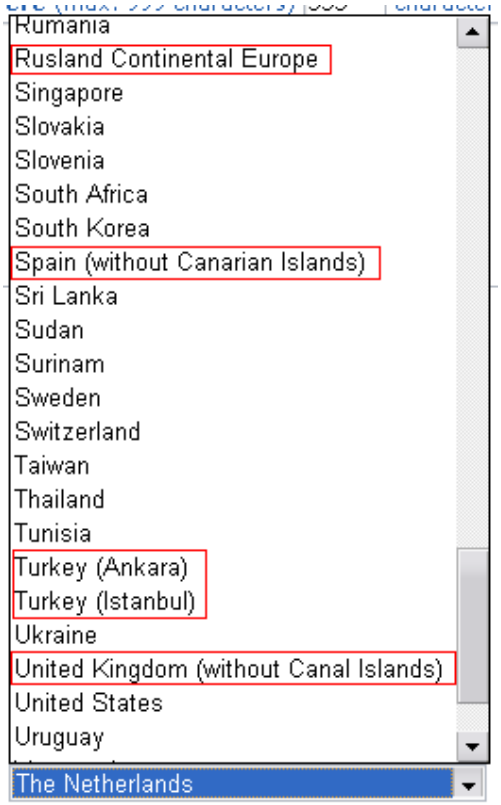
Take Admiralty Islands, or Aitutaki, for example. These places never normally appear on country lists for the simple reason that they are not countries, have no ambitions for independence, and have no geographical imperative to be handled differently (as, for example, some overseas' territories do). And whilst the 614 inhabitants of Alofi Island may be delighted with their inclusion in the list, it makes the list very long for all other users and could cause considerable confusion when it comes to sorting out the data collected.

Drop downs are for your customer's benefit as well

Remember who is using the form: your customer. The e-commerce site with the over-enthusiastic drop-down contents also removed places and added the place-holder "No Longer Available" to their drop-down. This may spark the interest of your customer (it left me wondering where else on this planet (or another) this company had found "countries" to remove from the list; but if nothing else it shows that the form was not tested.

A lesson to us all

The drop down on this site deserves a chapter by itself! It is a text book example of how to ruin both your data and your relationship with your customer in one fell swoop. (The red highlights in the screen dumps are mine).



The list is ordered alphabetically, but contains non-country information in what is supposed to be a country code list (such as Euro Airport). It contains spelling errors (Canal Islands instead of Channel Islands, Canarian Islands instead of Canary Islands etc.). It contains data in mixed languages (Rusland is Russia in Dutch). It contains multiple entries for some countries (Turkey Ankara, Turkey Istanbul) and misses countries and areas (anybody in Turkey outside Ankara or Istanbul, for example).

The company concerned quickly changed their form (see below), but replaced it with one where the country names have been listed from a politically correct source without giving thought about how the names will be viewed by the inhabitants of the country concerned themselves – Macedonians, for example, who call their country Republic of Macedonia, are hostile to any reference to their country as Former Yugoslav Republic of Macedonia.

Similarly, the reference to Taiwan as a province of China. And, the list contains the country of Serbia and Montenegro, which had not existed for two years when this screen capture was taken.

As an aside, many forms use drop downs because they increase the chance of getting accurate and correctly formatted responses from the customer. "This simple technique prevents mistakes"[20] trumpets one book confidently. Apart from mistakes that can arise from simply clicking on the wrong option, I have often seen cases where a carefully chosen option in the drop down list has been changed through using the scroll wheel on the mouse. When a drop down is highlighted the scroll wheel will scroll up and down the list of options. When it is not, it will scroll up and down the page within the browser window. Customers must therefore click outside the drop down to unselect it before being able to use the scroll wheel to scroll down the form. This is often not noticed and is the cause of inaccurate entries in drop downs.

[20] Matthew Linderman & Jason Fried, "Defensive Design for the Web", 2004, New Riders, Berkeley, CA, USA, p. 66

Field labels

One of the advantages of choosing to employ dynamic forms is that the field labels can be altered to match the country and language of the person concerned. Instead of asking for a *ZIP code* in The United Kingdom, you could request a *Postcode*; instead of a *State*, you can request a *County*, and so on.

Non-dynamic forms don't present you with that advantage, and require much thought to be given to field labels, which need to be clear for all customers, wherever they are from.

You need to be very careful and culturally neutral when choosing field labels. Don't shy away from using descriptive labels and explaining what data you are looking for. This is always preferable to collecting poor data due to misunderstandings. Labels such as "Title" (Job title? Form of address? Academic title?), "Position" (Job title? Sitting down? In front of my PC?) and so on, which can be interpreted in several ways, should never be used. This is especially the case if your form is not translated into each language (and no form could ever be translated into every language).

You need to avoid locally specific (e.g. *ZIP Code*), non-universal and culturally-loaded (e.g. *Christian name*) labels.

Non-specific and vague titles

Field labels like this are not specific enough and allow the customer to lose focus, having to think of what is required, and lead to the collection of poor quality data.

Move the onus of formatting from your customer to your form: avoiding overtaxing your customers

Any usability expert will tell you that the more you ask your customers to do to provide you with the data, the more likely they are to bail out. If you need to collect data in a particular way for it to have a meaning (as opposed to having that need because your database is built that way – not the same thing at all[21]), by all means explain clearly to your customer what you expect – that is far better than leaving them in the dark.

However, there is little worse than making customers format data in a particular way to suit your programming and database needs, just because you don't, for whatever reason, add some simple programming to the back-end to do this formatting for them. Worse still is making your customers enter data in a format that is unfamiliar and incorrect to them, such as adding postal codes, normally containing a space, without that space. And possibly worst of all is expecting customers to consult a different page to find out how you expect them to enter that data.

These things are guaranteed to irritate and cause some customers to peel off and go elsewhere.

Collecting data in fewer fields and parsing and formatting at the back-end can also increase data accuracy, whilst making it easier for the customer to complete the form.

A large Dutch online catalogue retailer collected the personal names of their customers in three fields: the given name, the preposition (van, van de) often found in Dutch names, and the rest of the surname. The prepositions are often found stored separately in Dutch databases for reasons such as alphabetical sortation rules.

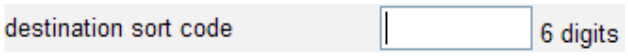
As the company found that some of its customers added their full surnames twice or added other errors, they decided to collect the surname in a single field, including the preposition, and to parse the preposition using software at the back end (the number of possible prepositions is limited, so this was not too difficult to apply). They experienced a more than 40% drop in errors in names being entered. Though some of this may be attributed to a confusing field label on the original form, it does demonstrate that moving the onus of formatting away from the customer can improve data quality.

[21] For example, if you need a date in the format dd-mm-yy because your database structure requires that, and any other format such as dd/mm/yyyy will cause your programme to go boing, that's a problem that you should not be laying on to the shoulders of your customer. However, to be meaningful you need to know whether a date such as 01-11 is 1st November or 11th January, and as such you need to explain to the customer the order in which to enter the data, or use a less error-prone system such as a calendar pop-up.

Forcing the customer to think

This form requests a Dutch postal code, which has the format 9999 AA. It expects that the customer enters the numbers and that they then tab to the next field to enter the letters (though this is not explained anywhere on the page). Regardless of how the data is to be stored within the database behind this page, it is a very simple task to correct any non-standard entry on-the-fly without burdening the customer with your requirements. An entry of 1018 vv can be upper cased, 1018VV can have a space added to it at the correct place, and one of 1018 VV can have, if required, the two parts of the postal code stored separately.
This form would allow people to enter their full postal code within the first postal code field, causing data quality problems within the database.

This bank does the same thing:

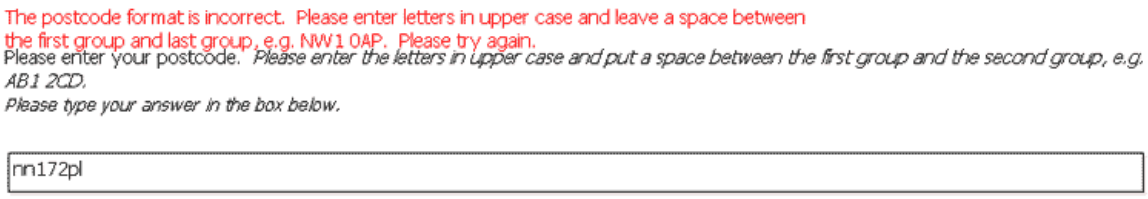


Though sort codes in the British banking system are always written and shown in the format 99-99-99, this bank requires me to stop, think and add it in an unexpected format: 999999. It would be no problem for the bank to extract the hyphens after entry if so required, thus unburdening the customer.

And for most forms requesting credit card information….

Any non-numeric data added to this field could very easily be removed automatically, regardless of how the customer chose to enter the information[22].

In the example below, the error message in red shows just how this company has designed this form for their own convenience rather than that of their customers. It may well have taken more time to program the form to produce the error message than it would have done to simply upper case and format the postal code entered, whilst the requirement that the customer goes back and corrects their "mistakes" leads to frustration and possibly to them quitting the page and you losing a customer.



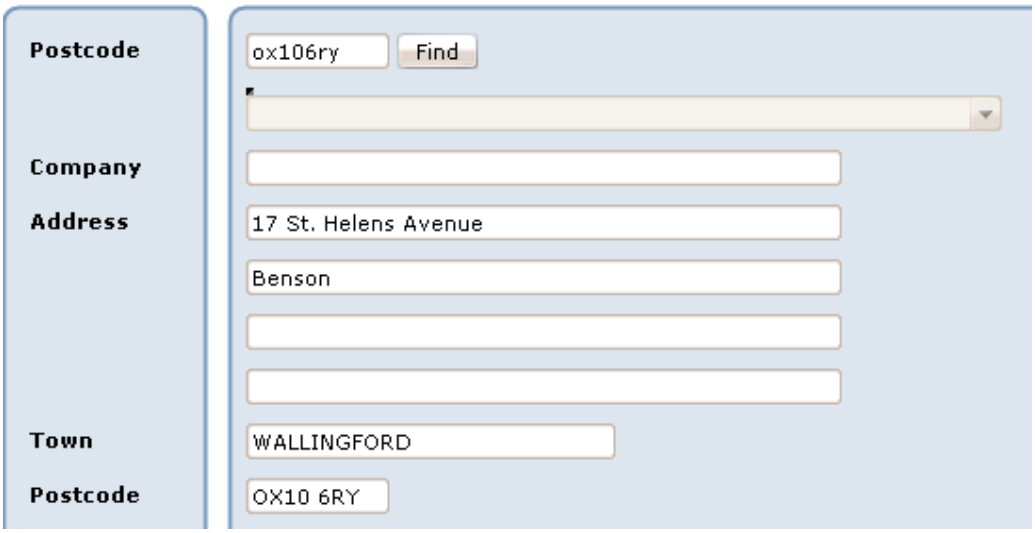
---

[22] Some customers use auto-complete software on web forms. If the form does not accept the data in the form stored in the auto-complete software (which will usually be that which is natural for the customer), you force the customer to re-enter the data and create another layer of frustration.

Algeria	Length of postal code must be exactly 5 numbers, without spaces or gaps.
Australia	Length of postal code must be exactly 4 numbers, without spaces or gaps.
Austria	Length of postal code must be exactly 4 numbers, without spaces or gaps.

Very few people who know their own postal code will not know what it looks like or how it is normally written. In this example, a customer must travel to a new web page to discover not how their postal code is written, but how this company's web form expects the code to be written, in fact the ONLY way it can be written for the customer to actually make an order.

The form below, a rapid addressing entry form for UK customers, can accept a UK postal code in any format – in this case, wrongly cased and with a missing space – and find the address without putting any onus on the customer to enter the code is a specific manner.



Feedback – holding a dialogue with your customer

One of the best ways of increasing the quality of the response and the data from your web form is to hold a dialogue with the customer completing it. This feedback can take many forms. It may occur as the customer enters or leaves a field, or when the user commits the data through pressing the "Send" button. It is important that the feedback has clarity. Some forms show an "error" in input through, for example, colouring a field red. This is not sufficient – the customer needs to know what exactly is wrong with what they have (or have not) entered and also how it can be fixed. Other sites print error messages at the top or bottom of the page or of the browser window, leaving the majority of customers who fail to spot these completely perplexed as to why they are unable to continue with the process upon which they have embarked.

Postal Code does not appear to be valid.

This sort of error message completely lacks clarity. In what way is the postal code invalid? Are they saying that the postal code does not exist, that it is postally invalid? Is the formatting wrong? Did I miss a space? Add a space? Help!

Other feedback can improve the quality of data gathered, without necessarily imposing extra steps or checks on the customer. There are websites, for example, with rapid addressing requesting only a building number and postal code, but which do not then make any effort to feed back or request confirmation of the address they have located with that information. If the customer makes a typo in either piece of information at that point, without feedback they cannot know that and the products that they are ordering from you could be sent to somebody else.

The simple feedback of printing the found address at the point on the form where the customer is (perhaps with a message that this is the address that they have noted for you), will encourage the customer to make alterations if they see that the information is not correct. It is not necessary to add the extra step of requesting the customer to confirm the information if you do not want to burden the customer further, though this step would increase data quality. The decision on whether to do that depends on how important getting that information correct is to your customer. If they are ordering something from you, it is in their best interest to make sure that the address is correct, and in those cases an extra "is this correct" button is usually harmless.

Check your spelling!

Check the spelling on your form! It is remarkable how many forms are posted with spelling errors and typos. Each error will have an effect on your customer's opinion of your company.

Spelling

Check the spelling on your form before you post it – errors are embarrassing!

Test the form!

Forms should always be viewed and tested before being posted online, for their technical correctness, their usability for the user, and their applicability to your customer base.

Testing forms for their international applicability, however, brings challenges that most companies cannot easily overcome. You could march ten thousand Americans into your testing room and let them play with your form, and the odds would be that none of them would point out that ZIP code isn't used to mean postal code anywhere outside the United States, that customers in Ireland aren't able to fill that required field in and that the postal code field doesn't accept letters, so that's a problem for their Canadian and British brethren.

You should try to get people from different countries to test your form, though getting somebody from every country and language group is difficult to achieve. For this reason it is imperative that you acquire and apply the knowledge necessary to avoid being caught out on such simple issues as required fields and postal codes lengths; but also provide a field for customers to make comments on the form (also a great way for them to vent their frustration after having to fill in your form) so that your customers can act as your testers as well. Importantly: read those comments and act on them!

You can also improve forms in your testing by being deliberately awkward, obstreperous, pushed for time and short-tempered. Try to find names and addresses from other countries and see how they fit into your form. Play devil's advocate.

Testing

It is completely obvious to me, to you, to your programmers, to the Chief Executive Officer, in fact to anybody who sees this form that offering the opportunity to find an address on the basis of postal code/building number only AFTER the customer has filled in their details will cause form rage at every level. So why are forms like this so common? It is down to a lack of testing along with a divorcing of those who are responsible for collecting and using the data from the technical staff who produce the form itself. Also, though this is common sense, common sense is often only obvious after somebody else has pointed it out.

Note: the arrow is not on the original form

If the country had been asked first, the "Find Address (UK Only)" button could have been removed for any

country other than the United Kingdom, reducing any potential feeling of being snubbed that people from other countries might have in coming across that button.

This form presented me with this message when I entered my (valid) Dutch postal code:

Postal Code does not appear to be valid.

After some experimentation, it became clear that the form, despite accepting addresses from anywhere in the world, and including a country drop-down, would only accept British postal codes. Many customers would have given up at that stage. I entered a British postal code to proceed, polluting the company's database. Clearly, the form had not been tested.

An example

To demonstrate how the suggestions made in this book might be put into practice, let's look at a typical form and see how it might be improved to make it useful for an international audience, to reduce its friction so that more customers complete it, and to improve the quality of the data gathered from it.

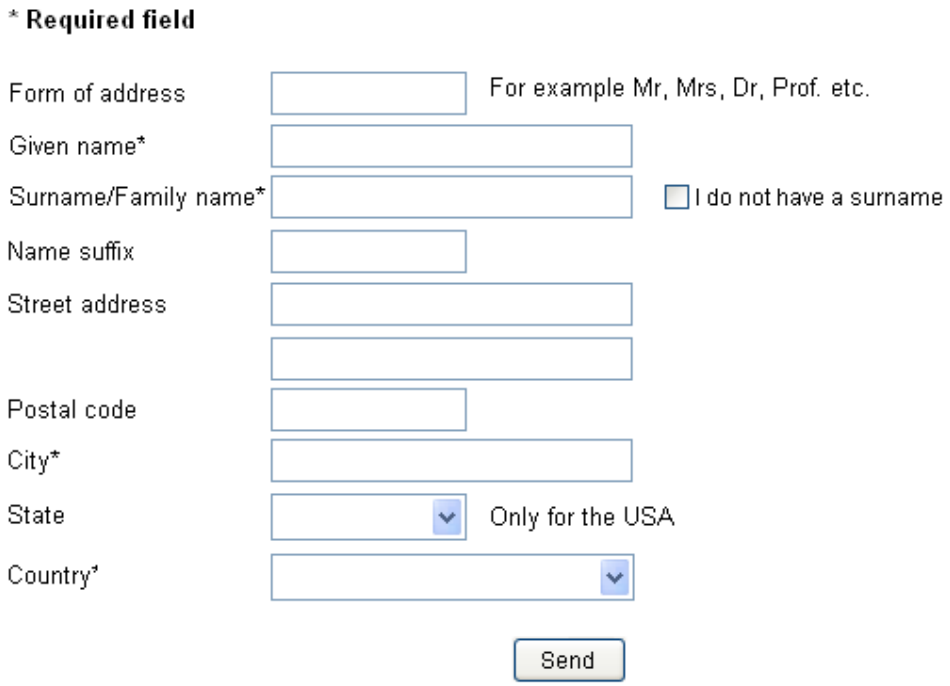
The original form

This simple form has a number of issues which will make it unsuitable and frustrating for many customers, as well as reducing your data quality:

- The form uses field names indicating relative position – first name, last name etc.
- The prefix (form of address) is a required field. Its drop down cannot contain all possible forms of address, so some customers will not find their form of address on the list.
- Last name (surname) is a required field, whereas some people do not have a surname.
- Zip code (more correctly ZIP Code) is a name for postal codes used only in the United States. It will not always be understood (and will often annoy).
- The postal code field is a required field even though not all countries have postal codes.
- The state field (with a drop down containing states for only one country) is a required field, though most people do not live in states
- The country field is asked last.

- The drop downs default to the first item on the list, allowing most customers to skip over that field and leave the default in place.
- All the fields are the same size on the screen, reducing any visible clues as to what belongs in each field.

The same (non-dynamic) form with the main errors ironed out



This form is more acceptable for an international audience:

- The field labels are more generic: given name, postal code etc. However, this has its own consequences – how many Americans realise that their ZIP code is a postal code?
- Form of address is no longer a required field and there is no drop down. The customer can leave that empty if they don't want you to use a form of address.
- Some extra explanation is given where a field label may not be clear or when a field is only for use in a single country
- When surname is essential information for a company, it is problematic to make the field optional as this encourages customers to skip over it. In this example the surname remains required, but this can be overridden through choosing an *I do not have a surname/family name* check box, encouraging most people to enter their surname and providing an escape for those that cannot.
- Postal code is no longer a required field.
- State is no longer a required field
- The drop downs no longer default to a value
- The field lengths on the screen give a visible clue as to what information is required.

However, the potential for errors and confusion remain, so a form built on the basis of a country will work much better:

A form built for customers from The Netherlands, in English

This form has been built for customers from The Netherlands:

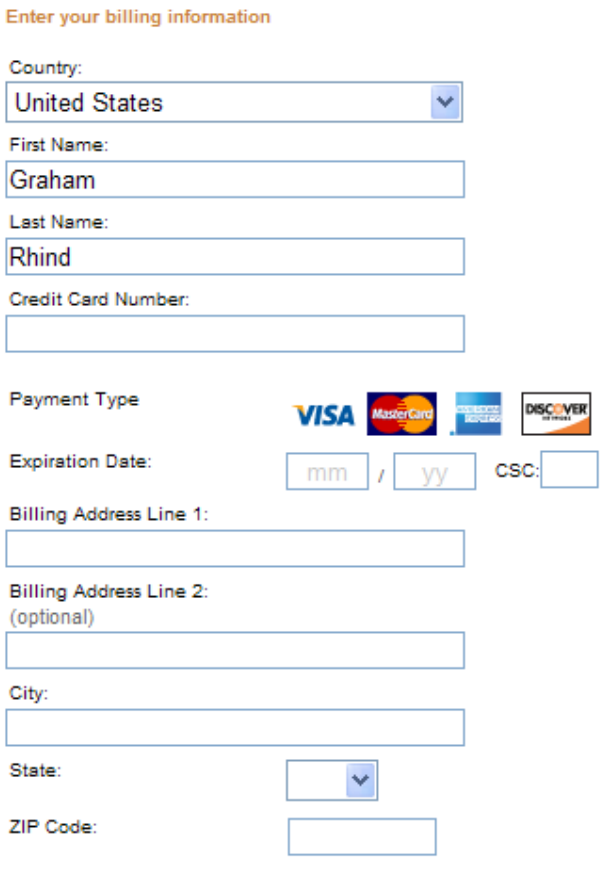
- The country is asked first, and the form is built on that basis
- The fields are presented in the correct and familiar order, under each other or formatted across the page
- Only those fields that a person from this country can fill in are presented – so, no State field, for example
- Fields which all customers can complete are made required
- Some countries have strict rules about naming conventions, and/or are intolerant of alternative name forms for their residents, and for those countries a surname/family name may be expected to exist and can be made required.
- Though the form is not in a local language, field labels can be altered to make them better understood in that country – Postcode, for example, instead of Postal code

Even better for Dutch-speaking customers in The Netherlands would be for the form to be presented in their language:

A form built for customers from The Netherlands, in Dutch

This form now leaves little to chance. The field labels are clear, the fields are familiar and presented in a familiar way, and the amount of friction to fill this form in for this customer is very low, compared to their negative experience were they trying to fill in the form shown in the initial screen dump above.

A partial success



This e-commerce company has got it partially correct. Its forms request the country first and (though only after an unacceptable delay) the form rebuilds itself to suit the country concerned.

Saisissez vos informations de facturation

Pays :
France
Prénom :
Graham
Nom :
Rhind
Type de paiement
Carte Bancaire
MasterCard
Cofinoga
Numéro de carte :
Date d'expiration : mm / yy
Cryptogramme visuel :
Adresse (ligne 1) :
Adresse (ligne 2) :
(facultatif)
Code postal :
Ville :

Though the layout is inconsistent and far from perfect for the country concerned, it shows only those fields that the customer would be able to complete – the state field, for example, is not shown in this French example.

This is only a partial success because the company does not ask for a language preference, and automatically assumes that the customer speaks the main national language of country concerned, not an assumption that will always be appreciated. If the customer does not understand that language, they have no escape route to another language. For countries where multiple national languages exist, such as Switzerland or Belgium, the form simply continues in English.

Enter your billing information

Country:
Switzerland
First Name:
Graham
Last Name:
Rhind
Credit Card Number:
Payment Type
Expiration Date: mm / yy CSC:
Billing Address Line 1:
Billing Address Line 2:
(optional)
Postal Code: CH-
City:

The dynamic world – maintenance

Form sorted, validation set, tested and posted on the web, data coming in.

Finished?

No.

Your form may be suitable for the world as it is now, but it needs to keep up with changes; and data cleansing and data quality processing do not stop once the data has been collected.

The world and its systems are dynamic to a surprising degree. Ensure that the tables and masks being used for validation are up-to-date – make sure that there is a team or staff member to take stewardship of this.

What can change? Countries come, go and change their names. Address formats change and postal code systems are introduced or revamped; and the legal situation in countries can change on, for example, personal naming.

If you use drop downs, ensure that the lists are up to date. How many companies keep up with the ever-changing names of countries – how many still have Yugoslavia in their lists, or Serbia-Montenegro?

There are many changes to systems each year, and you need to keep yourself informed of these changes. The best source for this information is the Global sourcebook for Address Data Management – http://www.grcdi.nl/book2.htm

Often it is change within a company which causes problems within web sites and their forms. Carefully crafted forms get bulldozed when companies merge, when web sites are overhauled or when a company rebrands itself. After years of working to get a form correct, a moment of folly can set the company back years.

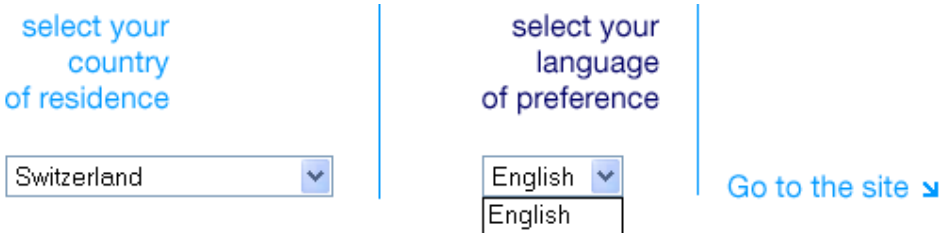
Staying current

Serbia and Montenegro ceased to exist in 2006 yet you would be hard pushed to find a form anywhere on the

web which has yet got round to splitting these countries in its country list 3 years later and counting

---

Don't forget these principles when you overhaul your web site



When a company has understood and applied the principles of good international data management, it is important that other pressures within the company do not compromise these over time. This company had, for many years, this pre-selection screen, sending customers to local and localised web forms according to their country and language preferences.

After a merger with another company, the redesigned website (appearing to put presentational uniformity above either data quality or the needs of its customers) contained a single form design regardless of the country of the customer, with many obstacles placed before the customer. Irish customers, for example, were forced to enter a postal code even though they had none:

This will result, in this case, either in a major data quality problem for the company or the loss of those customers to local Irish competitors, who did not make the same error:

Dos and Don’ts in your web form – a checklist

In summary, here are some of the most important points to bear in mind to enable your web form to delight your customers and to bring you better data quality:

DO

- Make your form dynamic so that it presents the relevant fields in the familiar order to suit that customer
- Give thought on whether you need to ask for personal names in separate fields, or if one field will do
- Ask whether the address is a street address or mailing address, and take this into account for your form design and data management processes
- Validate at the data collection stage
- Localise your forms, by language and by idiom, and check the translation
- Ensure clarity – provide clear field labels, instructions and examples
- Ask for the information that you are trying to gather
- Show fields on the form in a size related to the size of the data being gathered
- Format the data using back-end routines, rather than placing this onus upon your customer
- In non-dynamic and non-localised forms choose generic, culturally-neutral, clear and descriptive field labels.
- Hold a dialogue with your customer, and provide them with feedback within the form.
- Check the spelling on the form
- Test the form
- Maintain the form to keep up with changes in world legal and postal systems

DON’T

- Try to make a one-size-fits-all form
- Present a form automatically in a language of the country where the customer is located unless that is what the customer requests
- Field labels:
    - Avoid field labels indicating the relative position of elements, such as *first name*, *last name*.
- Required fields:
    - Be careful about making a *family name/surname* field a required field, as many people do not have surnames.
    - Don’t show fields on the form which are not relevant for that customer.
- Don’t try to extrapolate from one type of data (e.g. form of address) to another (e.g. gender)
- Avoid automatic tabbing
- Don’t show all fields on the screen as the same length.

- Do not use drop downs except where the list of possible options is exhaustive and mutually exclusive and the drop down contains each of these options.
- Do not add more options to drop downs than are necessary
- Order drop downs in a logical way – grouped, or alphabetically in the language of the form.
- Do not allow changes within companies to undo improvements that you have made in your form

The ***Global Sourcebook for Address Data Management*** by Graham Rhind contains a huge range of knowledge that you will need to develop globally valid personal name and address input forms. Its contents include such information as personal name, address and postal code formats for all countries; form of address lists, sample form layouts and field labels in local languages. See http://www.grcdi.nl/book2.htm for full details.