About:
This handbook will teach you how to become a better screencaster. Ian and team at ProCasts have been screencasting since 2005, here Ian shares his knowledge to help you improve your skills. All of our techniques are covered along with all of the major software and platforms (including Camtasia Studio, Jing and Screenflow). We focus less on individual packages and far more on the general skills that you need to effectively explain, communicate, sell and support using screencasts.
By Ian Ozsvald (founder of ProCasts.co.uk)

This version:
First Edition (Release 10) published May 2010
You can check to see if your version might be out of date by visiting the homepage: http://thescreencastinghandbook.com

Update Policy:
By buying access now you are guaranteeing that you will receive each update to this eBook including the release of the first full edition of this book. The second edition could well involve the addition of more information, more tools and more resources - I'll let you know how to buy an upgrade when we get to this stage (but that's quite a way off for now!).

Refund Policy:
You are entitled to a full refund at any point, it is as simple as that. You're buying into an experimental book-creation process, if my book doesn't answer your questions then I am happy to fully refund your payment, just email me.

Contact: ian@TheScreencastingHandbook.com
Website: http://TheScreencastingHandbook.com
Google Group: http://groups.google.com/group/thescreencastinghandbook

License:
This handbook is for your personal use, you may not distribute it to others. Group licensing is available upon request.

Copyright:
Ian Ozsvald 2010
# Table of Contents

**Important note - Updates** ................................................................. 9
**Version history** ........................................................................... 9
**Introduction** .................................................................................. 10
  - Ian - Teacher, Founder, Screencaster and Researcher .................. 10
  - What you should get out of this book ....................................... 11

**What's the value of screencasting?** ............................................. 11
  - Tutorials - learning by seeing .................................................. 11
  - Product Tours - seeing your problem solved ............................ 13
  - Visualising Complex Information .......................................... 13
  - Grant & Proposal Support ..................................................... 14
  - Recording a Meeting ........................................................... 14
  - Bug reporting ........................................................................ 14
  - Hardware demos .................................................................... 14
  - Machinima ............................................................................ 15

**Who else is screencasting?** ......................................................... 16
  - Tutorials ................................................................................ 16
  - Students explaining their Python-based Adventure Game project . 16
  - Kids teaching math to kids .................................................... 17
  - Teaching Python Network Programming at University ............ 17
  - Open-source enthusiasts explaining Open Office .................... 17
  - Commercial web and graphics tools taught at Lynda.com ........ 18
  - Commercial emailing service taught at AWeber.com ............... 18
  - Product Tours ........................................................................ 19
  - Google .................................................................................. 19
  - FogBugz ............................................................................... 19
  - TechSmith ............................................................................ 19
  - Crunch .................................................................................. 19
  - Technical Support .................................................................. 20
  - User-created tech support .................................................... 20

**Making a screencast in the next 30 minutes** ................................ 21
  - Watch 'Making a screencast in the next 30 minutes' .................. 21
  - What's possible? ..................................................................... 21
  - To get started - get Jing (5-10 minutes) ................................ 22
  - Next - have a quick practice (3 minutes) ............................... 22
  - Planning your screencast (5 minutes) .................................... 22
  - Who is the audience? ............................................................ 23
  - What are you showing them? ............................................... 23
  - Practice runs ......................................................................... 23
  - Record and deploy (10-15 minutes) ...................................... 23
  - Have a repeatable environment .......................................... 23
  - Record, repeat .................................................................... 23
  - Share your first recording in our Google Group .................... 24
  - Checklist - I have ............................................................... 24
  - Two examples that I made earlier ......................................... 24
  - BuildBrighton example - how to login to MediaWiki with OpenID 24
  - BarCamp Workshop - "Recording the screen in 7 minutes with Jing". 25

**Making a screencast in 2 hours** ................................................ 26
  - Planning ............................................................................... 26
  - Workflow ............................................................................. 26
  - Recording ............................................................................ 26
  - Camtasia Studio 7 (Windows) ............................................... 27
  - BBFlashBack Pro 2.6.6 (Windows) ........................................ 28
  - ScreenFlow 2.1 (Mac) .......................................................... 28
  - Editing ................................................................................... 28
Making a screencast in 1-2 days .................................................. 32

Research - what problem are you solving? .................................. 32
Demo Videos .............................................................................. 32
Tutorials ..................................................................................... 33
Technical support ....................................................................... 33
Planning - how are you going to show your solution? .................. 34
Demo Videos .............................................................................. 34
You need a good script ............................................................... 34
Testimonials .............................................................................. 35
Call to action .............................................................................. 36
Tutorials ..................................................................................... 36
Who is the viewer? ..................................................................... 36
Which key points must be shown? ............................................ 37
Keep the tutorial short with redundant information ................. 37
One video per step ..................................................................... 37
Technical Support ...................................................................... 37
Who is the viewer? ..................................................................... 38
Which key points must be shown? ............................................ 38
Keep the support video short ................................................... 38
Common Pitfalls ......................................................................... 38
Writing the script ........................................................................ 38
Building script ideas .................................................................. 39
Rough notes ............................................................................... 39
Storyboarding ............................................................................. 39
How much needs to be defined? ............................................... 41
Requirement - software demo to increase sales ....................... 42
Requirement - tutorial videos to teach a friendly audience ....... 42
Requirement - quick internal demos for bugs and tutorials ...... 42
Rubber Ducking ......................................................................... 42
Prototype - making the rough-draft ......................................... 43
Nervous? .................................................................................... 43
Recording .................................................................................... 43
Take a breath and smile ........................................................... 43
Pause, take another breath, smile and continue ...................... 43
Screen resolution and your recording area .............................. 44
Suggested recording resolutions ............................................. 45
Choosing other recording resolutions .................................... 45
Setting your recording resolution ............................................ 45
Using Slides and PowerPoint .................................................... 46
Animations .................................................................................. 47
Wallpaper and icons - tidy it up ............................................... 48
Animated effects ........................................................................ 48
Get rid of annoying pop-ups .................................................. 49
Don’t wiggle the mouse ........................................................... 49
Avoid mouse marathons .......................................................... 49
Editing and Producing ............................................................. 49
Cutting away the rubbish ........................................................ 49
Zooms to focus attention ........................................................ 51
Highlights to focus attention .................................................................................. 51
Text annotations to add useful information .......................................................... 52
Intro screens ........................................................................................................ 53
Exit screens ......................................................................................................... 54
Recording the narration ....................................................................................... 56
  Do you need narration? ..................................................................................... 56
  Recording narration as you demo or after? ....................................................... 57
Microphones ........................................................................................................ 58
Environment ......................................................................................................... 58
De-noising the audio and Dynamic Range Compression ..................................... 59
Adding Sub-titles ................................................................................................. 59
  YouTube and .srt files ...................................................................................... 60
Making a .srt (SubRip) file .................................................................................. 61
Adding music ........................................................................................................ 61
  Places to find music .......................................................................................... 61
  Adding the music .............................................................................................. 62
Ideas for improving your screencast ..................................................................... 62
Measuring Results ............................................................................................... 63
Share your video and get feedback ...................................................................... 63
Checklist - I have ................................................................................................. 63

Export - which file formats do you need? ............................................................ 66
What are the best dimensions for exporting the video? ....................................... 66
  Common screen resolutions for viewers ........................................................... 66
  Recommended export resolution ....................................................................... 67
  Video examples of different export resolutions (including SD, HQ and HD) .... 67
    Export notes for the widescreen (16:9) video ............................................... 69
    Export notes for the normal aspect ratio (4:3) video ..................................... 69
Goal - what do you want to achieve with your video? .......................................... 69
  Goal - widest possible distribution .................................................................. 69
  Goal - widespread distribution in the most common video sites .................... 70
  Goal - embedding in your own site and some distribution ............................... 70
  Goal - hosting in your own site ........................................................................ 70
Destinations - where will your video live? ........................................................... 71
  Destination - YouTube ..................................................................................... 71
    YouTube HD settings ..................................................................................... 71
      General export settings .............................................................................. 71
      Camtasia Studio 7 (Windows) export settings ........................................... 72
      Camtasia 1.1 (Mac) export settings ........................................................... 72
      ScreenFlow 2.1 (Mac) export settings ......................................................... 72
    YouTube SD/HQ settings .............................................................................. 73
  Destination - Vimeo .......................................................................................... 73
    Vimeo HD settings ........................................................................................ 73
      General export settings .............................................................................. 73
      Camtasia Studio 7 (Windows) export settings ........................................... 74
      Camtasia 1.1 (Mac) export settings ........................................................... 74
      ScreenFlow v2.1 (Mac) export settings ......................................................... 75
    Vimeo normal (4:3) settings .......................................................................... 75
  Destination - ShowMeDo ................................................................................... 75
  Destination - your site ....................................................................................... 75
  Destination - mobile phones ............................................................................. 75
    iPhone ............................................................................................................ 76
      Settings in common tools: ......................................................................... 76
    Common formats and a guide to their settings ............................................. 76
The Screencasting Handbook

Codecs ............................................................................................................. 77
  H.264 or AVC (mainly used in .MP4 and .M4V files) ........................................ 77
  On2 VP6 (mainly used in newer .FLV files) ..................................................... 78
  H.263 and Sorenson Spark (mainly used in older .FLV files) ......................... 79
  Theora (mainly used in .OGV and .OGG files) .............................................. 79
  DivX and Xvid (mainly used in .DIVX and other files) ................................. 80
  Animation (great for lossless exporting but results in big files) ..................... 80
  Dirac .............................................................................................................. 80
  H.265/H.264+ (still a draft specification, not yet in the real world) .......... 80
Containers ......................................................................................................... 80
  .MP4 ............................................................................................................. 81
  .FLV ............................................................................................................. 81
  .MOV ........................................................................................................... 81
  .AVI ............................................................................................................. 82
  .WMV ......................................................................................................... 82
  .SWF ........................................................................................................... 82
  .OGV and .OGG .......................................................................................... 82
  MKV .............................................................................................................. 82
  .DivX ........................................................................................................... 83
  .CAMREC and .CMREC ........................................................................... 83
  .FBR .............................................................................................................. 83

Distributing your screencast ............................................................................. 84
  Distribution and embedding (with their player) .............................................. 84
  YouTube ...................................................................................................... 84
  Screencast.com ............................................................................................ 84
  Vimeo .......................................................................................................... 85
  Blip.tv ....................................................................................................... 85
  ShowMeDo .................................................................................................. 86
Embedding with your own player ..................................................................... 86
  Camtasia Studio's Player (only for Camtasia Studio for Windows) .......... 86
FlowPlayer ........................................................................................................ 86
  Cost ............................................................................................................. 87
  Documentation ............................................................................................. 87
  Examples of use ........................................................................................... 87
JW Player from LongTail Media ........................................................................ 87
  Cost ............................................................................................................. 88
  Documentation ............................................................................................. 88
  Examples of use ........................................................................................... 88
LightBox, ThickBox and FancyBox for pop-up players ..................................... 88
HTML5 ............................................................................................................ 89
Hosting ............................................................................................................. 90
  Your own FTP hosting ................................................................................. 90
  Amazon's S3 and CloudFront as a cheap Content Delivery Network .......... 91
  Larger content delivery networks like Akamai .............................................. 91
  Streaming media ......................................................................................... 91

Common Workflows ....................................................................................... 92
  ShowMeDo - 'good enough is good enough' tutorials .................................. 92
  ProCasts - professional software tours ......................................................... 92
  Recording Meetings ..................................................................................... 93
  Screencasting a Screencasting Tool ............................................................. 94

Screencasting software .................................................................................... 95
  Windows ...................................................................................................... 95
  Camtasia Studio 7 ....................................................................................... 95
  BBFlashBack 2.7 ....................................................................................... 95
Screencasting tools

Editing software

ScreenFlow 2.1
Camtasia Mac 1.1
Jing
iShowU
SnapzPro
Linux
RecordMyDesktop
xVidCap
Web-apps
ScreenToaster
Screenr (Screencasting for Twitter)
Screencast-O-Matic
ScreenCastle

Windows
Camtasia Studio 7
BBFlashBack 2.7
Adobe Premiere
Sony Vegas
Blender
VirtualDub

Mac
ScreenFlow 2.1
Camtasia Mac 1.1
Adobe Premiere
Blender
Linux
PITiVi
Blender

Screen resolution tools
Sizer (Windows)
MouseZoom (Windows)
Mouse Position Menu (Mac)

Audio editing
Audacity (Open source - Windows, Mac, Linux)
The Levelator (Open source - Windows, Mac, Linux)

For codec and file format information
Video Lan Client (AKA VLC - Windows, Mac, Linux)
MPlayer (Windows, Mac, Linux)
Screen resolution tools
Sizer (Windows)
MouseZoom (Windows)
Mouse Position Menu (Mac)
Audio editing
Audacity (Open source - Windows, Mac, Linux)
The Levelator (Open source - Windows, Mac, Linux)

Flash Video MX Standard from Moyea (Windows)
Microphones ................................................................. 114
Choosing a microphone .................................................. 114
Analogue vs Digital (USB) ................................................. 114
Dynamic vs Condenser .................................................... 115
XLR .............................................................................. 115
Common microphones ..................................................... 115
  Samson C03U, Lapel Mic, QuickCam Communicate Deluxe, Blue
  Snowball (USB), Logitech Headset, AT2020 (USB) .......... 115
  Altec Lansing (3.5mm), Microsoft LifeChat LX-3000 (USB), Samson
  C0U1 (USB) .................................................................. 116
Cheap 3.5mm mic ............................................................. 116
SM58 (XLR, needs USB interface) ..................................... 116
Plantronics 550 (USB) ..................................................... 116
Dexun Pro 58 .................................................................. 117
sE2200A (XLR) ................................................................. 117
ATM73a (mini XLR) .......................................................... 117
Mic Technique ................................................................. 117
Removing noise sources .................................................. 117
  Electrical noise .............................................................. 118
  Audible noise .................................................................. 118
Lip smack ........................................................................ 119
Removing echo ................................................................. 119
Avoiding high-energy "esses" (generally known as sibilants) ...... 119
Avoiding booms and bangs (generally known as plosives) and using a pop-filters .................................................. 120
Avoid breathing into the microphone .................................. 120
Gathering your thoughts and smiling .................................. 121
Extra hardware ................................................................ 122
Mouse vs Trackball? ......................................................... 122
Webcam ........................................................................... 122
  Demo videos .................................................................. 122
  Meetings ....................................................................... 123
Copy holder .................................................................... 123
How Screencasting Works ................................................. 124
  Capturing a screencast .................................................. 124
  Storing the screencast .................................................... 124
Other resources ............................................................ 125
  The Screencasting Handbook's Google Group ................. 125
  ProCasts Tutorials and Critiques .................................. 125
  ShowMeDo.com ............................................................. 125
  Peepcode.com ............................................................... 125
  Railscasts.com ................................................................ 126
  Scrast.net ..................................................................... 126
  ScreenCastsOnline ......................................................... 126
  Lynda.com .................................................................... 126
  JQueryForDesigners.com .............................................. 126
  Books ........................................................................... 127
Tell others about The Screencasting Handbook ................. 128
Professional Screencasting and Demo Videos with ProCasts.co.uk 128
Acknowledgements .......................................................... 128
Do you have the latest version of The Screencasting Handbook? .... 128
**IMPORTANT NOTE - UPDATES**

Are you on the Updates mailing list?

If you've just bought the book then you should automatically get invited to the Handbook Updates mailing list. I use this list to mail out new copies of the book to all purchasers. If you're not sure that you were added to the list (sometimes the automatic invites get lost) then jump to the very last page of the book. Give it a few days from purchase though, I check a little while after you've purchased to make sure the invite was sent.

**VERSION HISTORY**

May 2010 Release 10 - First Edition completed, edits made and bugs fixed based on reader feedback
April 2010 Release 9 - Completed "How screencasting works" and "Microphone technique"
April 2010 Release 8 - Completed "Distributing your screencast", wrote most of "Screencasting software", covered "Common workflows"
January 2010 Release 7 - Added "Export - which file formats do you need?" and extended "Screen resolution and your recording area"
December Release 6 - Merged "A deeper look at the techniques behind screencasting" into "Making a screencast in 1-2 days", added checklists to the three "Making a screencast in..." chapters, added two new examples to "What's the value of screencasting?"
November Release 5 - Wrote "Make a screencast in 2 hours" chapter, added "Distribution" chapter to talk about YouTube, Vimeo, Screencast.com and ShowMeDo.com.
October Release 4 - Added 'Making a screencast in the next 30 minutes' chapter and screencast, started an outline of the "Make a screencast in 2 hours" chapter
September Release 3 - Second release, added examples recordings to Microphones section, expanded the Other Resources list, expanded "What's the value of screencasting"
August 2009 Release 2 - First release, chapter outlines and early chapter drafts
INTRODUCTION

IAN - TEACHER, FOUNDER, SCREENCASTER AND RESEARCHER

I'm told that writing a book is hard and self-publishing even harder. Personally I love to share knowledge and I've built a few projects that do just that so I'm planning on building The Screencasting Handbook in the same way, one chapter at a time. I have 4 years of screencasting knowledge to share and I know a set of others who have strong skills and are kindly sharing their feedback. With luck this book and the associated discussion group will give you just the resources you need to learn new screencasting skills.

Thinking back I remember the wonder of first seeing video on the Internet during the late 1990s. I was doing my undergraduate degree at Swansea University (UK) and NCSA Mosaic had just been installed, I'm sure the video was awful by today's standards but the fact that this video appeared, for free, on my machine, at my request...wow! Now more searches occur in YouTube than in Yahoo! so YouTube would be second only to Google for search-volume if it were classified as a search engine. Video is a part of our world and it will only become more prevalent as bandwidth and processing power improves.

Jump forwards to 2002 - I was Senior Programmer for the Artificial Intelligence firm MASA (A.I. is the other 50% of my passion) with a programming and R&D team split between the UK and France. We had email and landlines but no live video and no easy desktop-recording software. I remember the pain of trying to debug remotely, having a colleague explain (with words!) what they were seeing on screen in complex applications and prompting them for things that might have been relevant. Now we have Skype with Desktop Sharing and tools like Jing and ScreenToaster to make this process easy.

In late 2005 my friend Kyran Dale floated the idea that videos of software ("they're called screencasts" he told me) would be an ideal way to teach programming remotely. We both have a passion for education - Kyran had completed two post-doctoral positions at Sussex University and 10 years after graduating I'm still involved with the Computer Science department. The idea that we could enable people to share their knowledge via these screencasts seemed like a wonderful idea - but no site existed to let us do this. Being programmers we scratched our own itch and on December 31st 2005 we released the first version of ShowMeDo.

Having no prior experience with marketing or business-building the early growth of ShowMeDo was somewhat, well, slow. It did grow and along the way we met some wonderful people (like Horst, Gasto, Alan Pope and HeathenX who join us here in the Handbook's Group), now the site serves 50,000 screencasts a month to an international audience and 100 authors have contributed over 1,000 unique screencasts, almost all crafted purely for ShowMeDo's audience.

One of the hardest experiences when founding ShowMeDo was the process of actually recording the screencasts. I'd sit at home in the evening (ShowMeDo was unfunded so it grew in evenings and weekends) and practice my screencasting. I'd set the computer up with a low-end mic and free software (CamStudio and VirtualDubMod) and proceed to break

2. http://masagroup.net/
out in a sweat. I hadn't even started recording and I was nervous. Heck, I had complete control over deleting any recording I made - yet I was still nervous. I assumed that this was personal but in the ShowMeDo group it became obvious that I wasn't alone - most of us had no idea of the 'right' technique, software, tools or processes - we all just made it up as we went along.

Jump forward a few more years to 2008 when I started ProCasts⁴. Clients had approached me via ShowMeDo to make custom demo videos that explained their software and after a while I broke this out as ProCasts which now occupies most of my time. The act of producing over 140 demo and tutorial screencasts in ShowMeDo gave me the experience to competently demonstrate software that helps clients to sell to, train and support their users.

Now I come full-circle. I've learned an awful lot over the last 4 years by reading, practicing and asking for feedback. Now I'm going to share my knowledge and invite collaboration from others with experience so we can teach you new skills and approaches to improve your screencasting abilities. I'm fully expecting some hard questions to come back from you and the answers will stretch my knowledge in new directions - don't be shy!

WHAT YOU SHOULD GET OUT OF THIS BOOK

You should find value in this book if you want to:
- make screencasts more quickly
- communicate more effectively
- get feedback from more people
- find a wider audience
- have fun sharing your knowledge!

If I'm missing a topic then post in the Google Group. I want to answer your questions so don't be shy in asking. You'll see sections in the book marked with "Reader - I need your feedback:" - here I'm asking for you to give me feedback (preferably in the Google Group). I don't want to waste time with long answers to topics that aren't helpful, also I don't want to write too briefly on topics that you want to read about. I need your feedback.

WHAT'S THE VALUE OF SCREENCASTING?

TUTORIALS - LEARNING BY SEEING

When learning it is generally accepted that if we engage more modalities⁵ then we have a better chance at recalling what we've learned. A 'modality' is a way that information is encoded. Examples include text, images and narration.

One of the hardest ways to learn anything that involves graphics is by reading plain text. Most of the software we use now has a rich user interface with liberal use of graphics, icons, menus and controls that move. If I cast my mind back to school in the 80s we had DOS-based WordPerfect and MS Word with ring-bound user manuals filled with page after page of text with nary a diagram to be seen. As long as you followed the instructions word-for-word and you had the word processor open you stood a chance - but woe betide you if you tried to read the manual at home and remember the steps the next day!

⁴. http://procasts.co.uk/
The problem was the lack of mechanisms that would engage aspects of our memory - no images, no video, no sound, no speech - just plain black and white text. Learning to use software by reading a manual is generally regarded as a punishment!

Imagine also what happens if your manual refers to the wrong version of the software - with text-only manuals it often becomes impossible to recreate a series of actions when the menus and wording in the application changes as we lack all the cues we'd normally use.

Better manuals have graphics - sometimes even in colour. This helps a lot as it engages our visual memory and we have a better chance of recalling the information later. Pictures are still pretty rubbish at describing sequences of actions, especially if you need to interact with the application during that sequence.

Perhaps the best way to learn is when you have an expert sat right down beside you - they show you the application working, answer specific questions and gently walk you through the steps that you need to learn while you do the actions. This invokes many modalities and gives us the best chance of recall.

Screencasts sit between static graphics and having an expert close at hand. We get a visual, moving demo with a friendly voice, we can also have text annotations on screen and animations. The only thing we lack is a way to talk to the expert. If the screencast refers to an older version of the software we still have a chance of following along - often there are enough visual cues, coupled with the narrator's discussion, to give use the clues we need to figure things out. This means that screencasts still have value when software is being rapidly updated (which can't necessarily be said for old-style manuals).

An advantage of screencasts over an expert is that they can be replayed at will every hour of every day. If a commenting system is added then it becomes possible for viewers to leave questions and for the author to reply with an answer. The process is wiki-like where incremental units of information are added to the production (if not to the actual video itself).

We can also add translated subtitles to a screencast so they can be viewed by non-native speakers or replace the audio track with alternate languages.

Elizabeth Daley in Expanding the concept of literacy⁶ (2003, Educause) argues for an increase in the use of multimedia for education because text isn't ideal for many situations. She discusses all forms of multimedia and screencasting as a term was barely recognised back in 2003 but her arguments apply just fine:

"...print carries its own technological bias. Print supports linear argument, but it does not value aspects of experience that cannot be contained in books. Print deals inadequately with nonverbal modes of thought and nonlinear construction."

Later Elizabeth talks about famous video-based scenes from world history and discusses why they are more evocative than a textual description:

"Rich media, with its multiple simultaneous layers, does much more than provide enhancements, illustrations, and tools for enriching, accessing,

---

and transmitting the established literacy...Multimedia and cinema, though sometimes enriched by language, embrace many other elements as co-equal - not only image but also sound, duration, color and design."

She concludes:

"... following from the previous three arguments, those who are truly literate in the twenty-first century will be those who learn to both read and write the multimedia language of the screen"

As an aside, an article at the New York Times reports that a Study Finds That Online Education Beats The Classroom. I'm not going to take sides here but I know that remote learning and video tutorials are a great way to expand the places where we can learn and when (and how often) we can learn. Not being limited to a classroom can only help a student learn at convenient times and it is bound to engage memory and thought processes in different ways to just being in a classroom:

""The study’s major significance lies in demonstrating that online learning today is not just better than nothing — it actually tends to be better than conventional instruction,” said Barbara Means, the study’s lead author and an educational psychologist at SRI International."

**PRODUCT TOURS - seeing your problem solved**

One of the reasons I started making product demos in ProCasts rather than just tutorials for ShowMeDo was that I could see the power in showing a first-time viewer how a piece of software solved their problem.

One of the biggest time wastes I’ve experienced when trying to choose new software is the up-front cost of deciding which packages to try. They all have fancy descriptions and many have screenshots but few are shown in action so you never know if they really solve your need until you download them, sink time into learning them and are able to make a sensible judgment.

If I had an expert on hand, I could ask them about specific use-cases and they could demo the software solving those cases. If they did the right job then I'd know to evaluate the software, if they didn't then I'd know to move on. We can use screencasts to demo your software via your website to new visitors so they see it in action, solving common problems.

**VISUALISING COMPLEX INFORMATION**

I love this simple example of visualising non-trivial, complex (and geeky!) information.

In this video we see an Awesome C64 Visual Debugger (ICU64), the output window visualises the memory contents of the emulated Commodore 64 8-bit computer as a game

is played. You can see memory being filled as the game loads and how memory changes in real-time as the game plays in the neighbouring window.

Imagine trying to explain visual memory inspection using words or screenshots...instead we see the game playing and the memory changing in lock-step. The use of zooms in the debugger is also a great way of showing how nice it is to drill into data for detail once you have the high-level overview.

**Grant & Proposal Support**

Academics have used screencasts to support grant proposals. Screencasts can either add useful context to a proposal - showing past achievements and outlining what a successful outcome might look like - or they can be used as the main media for the proposal.

**Recording a Meeting**

When I attend a meeting with a client I often use my MacBook to record the meeting (obviously with permission!). The MacBook has a built-in video camera and microphone, ScreenFlow can record hours of content easily.

The main value is having a recording of our voices, with the on-screen clock (so I can make notes in my logbook against the time) with a screen recording for when we interact with the client's website. This way I can tally up my written notes with the discussion and screen so I have a complete record weeks later of all that was said. These screencasts would never be published (and generally are deleted once a contract is finished) but are a useful point of reference.

**Bug Reporting**

Sometimes it is far easier to demonstrate a bug than it is to describe it using text and screenshots. When playing with a new tool called vnc2flv I had trouble getting good FLV video output. I recorded a video and added it to a blog post describing the symptoms. In less than a day in the tool's forum I had the answer which turned out to be a simple problem with my choice of video player!

The use of video to demonstrate the problem (you need to see it as it involves a flashing screen which wouldn't make sense in a screenshot) meant it was obvious to some of the more experienced forum members where the problem might be, they gave me some options and their first guess was the right solution. If I'd have tried to describe the problem without a video we'd have been scratching our heads for a lot longer.

**Hardware Demos**

To demo how easy it is to program an Arduino controller I used a web-cam to show the Arduino's LED whilst I re-programmed it for a ShowMeDo demo. This is a simple way to

---

10. [http://groups.google.com/group/vnc2flv-users/browse_thread/thread/4efbf6ea816480be](http://groups.google.com/group/vnc2flv-users/browse_thread/thread/4efbf6ea816480be)
show how easy it is to control something outside of the computer when programming, here's a screenshot:

**MACHINIMA**

Machinima\(^{12}\) is “the use of real-time three-dimensional (3-D) graphics rendering\(^{13}\) engines to generate computer animation”. Red vs Blue\(^{14}\) (YouTube\(^{15}\), wikipedia) is the most striking use that I remember, R vs B started with their first episode W.M.D. PSA\(^{16}\).

The field started when enthusiasts would act out scenes inside the game and record the output, they’d then add a voice-over and they where then producing a new type of movie. The first well-known example of a machinima film was Diary of a Camper\(^{17}\) (wikipedia).

---

15. http://www.youtube.com/show/redvsblue
WHO ELSE IS SCREENCASTING?

TUTORIALS

Below we'll take a look at four examples of tutorial videos covering open-source tuition in schools, office-application tutorials, commercial skill training and commercial product training.

Students explaining their Python-based Adventure Game project

How often do you see students so engaged in their work that they will record themselves explaining their new adventure game, written in Python, in 2 minutes? Horst is a teacher in Austria who encourages his students to explain their projects inside ShowMeDo. Mira created 'Simple graphic adventure game with python and easygui, part 1 (English)' which runs for 2 minutes with spoken English and English subtitles to teach you how her adventure game works including a look at the underlying Python code.

Mira uses a web-cam to record her image while she gives a live, unedited demo of her game. The game uses a simple graphical tool-kit to present descriptions, actions and images. The fonts are larger than usual so they are recorded clearly, the English sub-titles make it easy to read if you have trouble with Mira's accent. Having watched the video the night before I found I could still recall the structure of it the following morning in part because it mixes an enthusiastic speaker with video and interesting images - there are many things here that engaged my memory.

The second video in the sequence shows Mira and her sister Teresa, they form quite a double-act as they explain the second version of the game. Horst has a homepage for the project which links to both the videos and source code along with a discussion.

ShowMeDo allows contributions from anyone, the motto is 'good enough is good enough' - this embodies the idea that if you can make a recording that teaches a new skill to the viewer then it is acceptable as new content. Production quality can be variable, viewing some of the screencasts will give you an idea of what's possible by untrained screencasters with various skill levels, often with home-built collections of equipment starting from bundled mics and free software.

ShowMeDo's Python section has 505 screencasts in 141 series, you'll find a variety of techniques from many authors.

Horst used Ubuntu Linux and gtk-recordmydesktop with a good webcam and a cheap mic. The desktop was set to 1024x800, ekiga showed the webcam and geany was the IDE. Horst used the production notes in ShowMeDo which includes details for recording, editing, cropping, audio manipulation, logos and output.

Horst has added the he "prefers recording the sound directly with the webcam using the command: mplayer tv:// to display a webcam window".

**Kids teaching math to kids**

[MathTrain.com](http://www.techsmith.com/community/education/real-users/mathtrain.asp) was established in 2006 by Eric Marcos to help kids teach math to other kids. There's a nice *write-up* at TechSmith as Camtasia Studio was the main tool, the videos were hosted in a [Moodle](http://moodle.org/) e-learning system.

**Teaching Python Network Programming at University**

Tim Bower uses screencasts to complement his book, the Network Programming Study Guide. His [online notes](http://www.sal.ksu.edu/faculty/tim/NPstudy_guide/index.html#guide) explain that the videos show you processes that are described in the book:

> "Providing instructional videos regarding the construction of programs and demonstrating the usage of various programs. Currently, all of the videos are available through K-State Online, which requires enrollment in the class, and some of the demonstraion videos are available through ShowMeDo."

His [Python Network Programming](http://showmedo.com/videotutorials/series?name=J8Pgfi4nA) series in ShowMeDo has 4 videos. In the first screencast he shows a live client/server chat system, he uses a large font and gives a very clear demo. He uses a mouse highlight which is a nice way of helping you see exactly where the mouse is.

In the second video he shows coloured syntax as he explains some of the Python code, the third video is a very thorough 18 minute walk-through.

Tim used an Altec Lansing head mic (discussed in the Microphones chapter) with 3.5mm (not USB) connector. The fourth video sounds different as Tim used a C01U mic rather than the Altec Lansing. For recording Tim used Camtasia Studio 4.

**Open-source enthusiasts explaining Open Office**

When was the last time you used an open-source Office application suite and found the docs hard to follow...and then found screencasts that walk you through all the usual operations? Dai has created one of ShowMeDo's most popular series with this [Introduction to OpenOffice Base](http://showmedo.com/videotutorials/series?name=AXggL6j0a) 28, this is the kind of feedback he gets:

> "Down to earth knowledge share that can help me to create database. Direct to the point teaching minimizes learning time. Thanks a lot." - rommel

---

Dai has used an opening and closing title which adds a professional edge to the production, this is a very easy step with a tool like Camtasia Studio and I believe that the assets came from the openoffice.org website.

ShowMeDo's OpenOffice section\(^{29}\) has 10 series each containing 6-28 screencasts, you'll find a variety of techniques from 10 authors.

**Commercial web and graphics tools taught at Lynda.com**

Lynda.com\(^{30}\) is a well-respected repository of screencast training material for commercial web and graphics tools, it is widely known in the web-design and graphic-design world. Lynda's content base has been growing for 10 years, the production process is typically higher-quality than ShowMeDo's and only a few authors are chosen to add to the site.

Typically an author is flown to their studio to record in their environment, the recordings include full-body videos mixed with straight screencasts. This will give you an idea of what's possible for untrained presenters coupled with experienced producers.

This page lists a course on 'Premiere Elements 7 Essentials'\(^{31}\), the underlined links are free content that you can view without a subscription. If you watch 'Understanding the workflow' you'll see an upper-body recording of the author which transitions to a PowerPoint-like presentation. The 'Relinking missing media' episode is a screencast that walks you through a process.

**Commercial emailing service taught at AWeber.com**

AWeber is a commercial service that manages an email distribution list (if you receive emails from TheScreencastingHandbook then you're on our AWeber list). AWeber has fully embraced screencasts as a way of teaching users both how to use their software and how to effectively market using emails.

AWeber's video tutorials are smoothly produced, typically on a Mac. They use a variety of speakers and the production quality can vary a bit e.g. Getting Started\(^{32}\) uses zooms but Send Email Newsletters\(^{33}\) doesn't - so it is hard to read the screen! The narration is typically clear, well paced and noiseless so it is easy to listen to and the screencasts flow smoothly.

I highlight AWeber's use of screencasts because I found that they made it incredibly easy for me to learn how to setup an email list. Prior to TheScreencastingHandbook I'd never setup a list in this way - not only did the screencasts explain how to complete the goals, they also taught me why I should care about certain steps.

---

PRODUCT TOURS

Google are a new adopter of screencasting, we'll look at one of their recent examples below along with TechSmith's productions for their screencast-related products. I also discuss the dual-narration FogBugz demo and one of our ProCasts productions.

Google

When Google released their new Chrome browser they knew they needed to have a quick, easy way to tell people why they should try it rather than stick with their usual browser. They created 10 very short screencasts (each 15-30 seconds) that give a quick tour of each of their main features.

These videos are great if you just want a quick idea of their features and they are very easy to make but they don't give you an example of a typical session, so you don't necessarily see how their features will fit into your way of working.

FogBugz

Joel and another team member use the very interesting approach of giving a product tour with two speakers with an informal, jokey approach. The product is FogBugz, a well-known and respected bug tracking tool for software developers. The 12 minute screencast is a bit of a long watch, the fun narration kicks in at around 0:33 but listen to the first half-minute to get a feel for Joel's approach.

Inside ProCasts we took inspiration from Joel's approach and used a dual narration for our tour screencast for MockupScreens. I took on the role of the 'contractor' and my colleague Richard was the 'client', we also used animations and a hand-drawn slide to add extra visual touches as we demonstrated a typical session mocking-up a user-interface.

TechSmith

TechSmith are one of the grand-daddy suppliers of screencasting software, their Camtasia product is well-known as being the most widely-used screencasting tool on Windows. All of TechSmith's products have tour videos, you can see the Camtasia Tour here. They have a separately recorded narration track with nice music played over screencast recordings and end with a full-screen video recording of an employee.

Crunch

Inside ProCasts we created a 4 minute tour for Crunch's homepage to show a first-time visitor how their software helps freelancers to record invoices, payments and tax calculations. They combine an accountancy firm with an invoice-tracking tool, that's the first time such a service has been created in the UK. The screencast includes an animated introduction and a short animation to explain the concept.

TECHNICAL SUPPORT

Screencasts aren’t often seen for technical support but when used properly they can really save you time which means you have more time to build your project.

Coursework.info provides background material for coursework assignments. Their users have a variety of capabilities, some have trouble using the Internet and aren’t comfortable with credit cards and on-line billing. Their problem was a small but significant volume of support emails and calls from users who had finished using the service but couldn’t figure out how to cancel their recurring payment.

These callers needed handholding which took approximately 15 minutes per user. Inside ProCasts we created a 3 minute screencast that walks the user through the process of cancelling their subscription, we used a slow, clear walk-through with a calm, confident voice and several on-screen annotations. The result was that most viewers were able to cancel their subscription and the support chap regained several days a month to spend more time helping other users.

The live version is only for logged-in members of coursework.info, you can see our copy on the ProCasts examples page, scroll down to ‘Cancelling your coursework.info Subscription’ and click the icon.

User-created tech support

YouTube run a clever channel - in YouTubeHelp's Channel your technical support videos are shown as official support videos. Despite YouTube’s size they ask their users to make their technical support content! This certainly emphasises YouTube's do it yourself attitude.

38. http://procasts.co.uk/examples.html
39. http://www.youtube.com/user/YouTubeHelp
MAKING A SCREENCAST IN THE NEXT 30 MINUTES

Here you'll learn how to use the free Jing from TechSmith to make a short screencast, hosted on the web so anyone can see it, in the next 30 minutes.

Requirements - Windows or Mac, a free sign-up to Jing, 30 minutes.

If you're on Linux try ScreenToaster - it works for me on Ubuntu 9.04 (ScreenToaster uses Java). The user-interface is a little different to Jing's so the process of planning, practicing and making the recording are a little different - it is certainly very easy to use ScreenToaster. ScreenToaster also works on Windows and Mac.

WATCH 'MAKING A SCREENCAST IN THE NEXT 30 MINUTES'

You can find the companion screencast here (6 minutes). It runs you through a live recording session using Jing on Windows showing you how to:

- Tell Jing to start recording
- Select the right area of the screen
- Create your demo (see the finished recording: Making a screencast in the next 30 minutes - Multimap Demo)
- Stop recording
- Upload to screencast.com
- Use the supplied short URL in your browser (and how to get a reminder of the URL)
- Watch the resulting screencast in screencast.com

The screencast runs for about 8 minutes, it shows bits of the recording of the Multimap Demo (just enough to entice you, not enough to bore!) and you can verify the final recording with the link above. The audio for the Multimap demo sounds a bit rough, that's a Jing problem that TechSmith have mentioned. I used the free Jing service so this demo should be as close to something you can record as possible.

For this demo I practiced the walkthrough several times (as discussed below), then recorded the session live. I used Camtasia 6 to record the full desktop (1280x960 resolution) using Windows XP, the microphone is an sE2200A.

You'll note in the Multimap demo that the Jing recording timer and control bar is shown in the bottom-left corner - that's because my recording region was too large so the timer couldn't be shown below the window. Rather than retake the screencast I chose to leave it in as a visual reminder to you to leave space below the video - otherwise the timer gets in the way of your production. If you record a smaller region of the screen then the control bar appears below the recording region, so it doesn't get recorded into the final video.

WHAT'S POSSIBLE?

Jing is a free tool that records 5 minutes of audio and video to make an SWF screencast, hosted by screencast.com. You can pay a small fee (about $15USD) for Jing Pro which lets you create MP4 screencasts which can then be edited. Neither version of Jing has any editing features. The screencast.com account lets you easily share the results with others over the web.
I used the free Jing to make a short screencast for BuildBrighton, my local geek hackerspace ('they make robots!'). The screencast shows new users how to login to the wiki using OpenID so they can make edits. The screencast is linked on the site's homepage or you can go direct to this page in screencast.com. As noted below this screencast was made as the 3rd of 3 takes (the other 2 had mistakes!), each taking about 5 minutes.

TO GET STARTED - GET JING (5-10 MINUTES)

You need to install Jing, visit JingProject.com and, click the 'Download Now!' button and let it install. On Windows it might also download the .Net environment from Microsoft which is large (about 100mb) so this might take a while. The Mac installer is pretty quick.

Once it is installed, the first time Jing runs it will ask you for your login details. If you don't have them already it'll let you make a free account (required) for screencast.com which will host your screencast.

NEXT - HAVE A QUICK PRACTICE (3 MINUTES)

The JingProject.com homepage has a tour video showing you how to get started (I grabbed this link on Sept 28th 2009 - let me know if it goes out of date!). If you watch the video you'll see how Jing is represented as a 'yellow sun' on the edge of the computer screen, how to start recording, how to define and area and how to record the screencast.

At this stage it doesn't matter what you'll record - just mouse the mouse around the screen and talk out loud. All we want to do is confirm that you know how to go about making a screencast and that it correctly records your screen and narration. After recording you can preview the output and upload (if you want) to screencast.com.

I've found the process to be flawless, the only confusing bit is after recording when it has finished upload...it gives you a link in your clipboard to the URL at screencast.com (for easy pasting into an email) but if you don't see the dialog box, you can get lost. I've noticed that if I try to upload it a second time (using the same three upward-pointing arrows) it tells me that it is already uploaded and it reminds me of the URL - this is a useful confirmation step if you've become lost at this late stage.

To confirm the upload open a new tab in your browser and paste in the URL, you'll be directed to screencast.com and your video will be shown.

PLANNING YOUR SCREENCAST (5 MINUTES)

When you first start to screencast there will be many things you might be unsure of - don't let these stop you! In the words of Nike - "Just Do It!". Take a look below at the "Two examples I made earlier" to see how easy it is to just have a few tries and get a good take. It is far better to deploy a screencast that is unpolished but helps people rather than never publish something that you keep meaning to improve.

Remember - you only have at most five minutes for your recording, it is absolutely fine to do short recordings too (e.g. 30 seconds). Shorter is better as it keeps you more focused.
Who is the audience?

This is the most important question - if you can't answer it then you risk pitching your screencast to the wrong people who might not have the right experience to understand what you're showing them. Spend a minute thinking about who your real users are, normally if you're solving a simple problem that can be explained in a few minutes then you'll have a well defined target audience.

What are you showing them?

This is also very important - the answer most definitely isn't "everything!". You want to be able to show them one clear topic in the few minutes of this screencast. It might be an overview of how something works, or might be a walk-through for how to solve one problem. Make sure you have a clear answer in mind to this question.

Practice runs

Next do a few practice runs without recording. You'll go through the actions and speak out loud. If you can do this off-the-cuff with a clear explanation then you'll sound absolutely fine in the recording. Speak slowly (but not artificially slowly) and clearly, keep it conversational, don't worry about "ums" and "ahs" or pauses.

Reader: If you're having a problem at this point, come join us in the Google Group and ask for feedback. If common questions keep coming up then I'll know to write more in this section.

RECORD AND DEPLOY (10-15 MINUTES)

Have a repeatable environment

Sometimes with software it is easy to delete whatever you just did and start afresh - using a text editor which you start from scratch would fall into this category. Sometimes you use a package where you can't delete the previous takes - using a scheduling tool where you can't delete past tasks (so they show up in subsequent takes of your screencast!) would fall into this category.

Take a moment to make sure that you've got an environment where you can easily do a recording, delete it, then record again, without visual glitches that will make the final screencast hard to watch.

Record, repeat

All you have to focus on is recording the screencast in one clean take. If you find you're struggling with five minutes of recording, consider splitting the screencast into two halves which are easier to record. You don't have to join them together, users are quite capable of watching two screencasts rather than one. Generally speaking shorter screencasts are better as they tend to be more focused.

Because Jing doesn't let you make any edits you won't have an opening or closing title slide. Also you can't edit out any audio glitches like breaths, "ums", "errs" or mistakes. Don't worry about these little problems, users generally don't care about these little glitches.
if the screencast makes sense and helps them solve their problem or understand new software.

So - try a recording, watch it, if it doesn't work then try again. It is as simple as that. When you have one that works, send it up to screencast.com (just press the three-up-arrows button in Jing) and

**Share your first recording in our Google Group**

If you'd like feedback on your recording or you'd like to share your thoughts on your process (or - even better - to suggest improvements to the guide above!) then share the link in our [Google Group](#). Everyone is friendly, just say that you have made your first recording and ask us to give some feedback, we'll see if we can offer some helpful advice.

**CHECKLIST - I have...**

- figured out the aim of this screencast
- thought about what my viewer will need so they will definitely learn something new from me
- run through the demo without recording so I know I can do it all without getting lost
- made notes on paper so I keep with the flow of the demo when I do get lost (but see the point above!)
- recorded a prototype and the sound records ok and the screen's text is readable
- embedded the prototype recording into a page so others could definitely see it - now I won't get frustrated by discovering a problem with embedding later
- dealt with external sources of possible noise, my mobile phone is off, windows and doors are shut, fans and fridges are unplugged and the cat is out of the building
- viewed the final recording and nothing was missed out so the aims I had are definitely met

**TWO EXAMPLES THAT I MADE EARLIER**

**BuildBrighton example - how to login to MediaWiki with OpenID**

As I mentioned above in "What's possible" I made one example for [BuildBrighton](#), my local robot-building hackerspace here in Brighton. The screencast is linked on the homepage (or [here](#) at screencast.com).

My process involved some practice and then three takes, the third take was a good one and that's what you see above. I'm pretty sure I used my sE2200A mic and I recorded on Windows XP. All in the start-to-finish time was 30 minutes. I didn't have a written script but I did build a sequence of notes which I refined with each new practice run.

At first I practiced the logins - the reason for the screencast is that logging in to MediaWiki using OpenID is non-intuitive. I wanted to make it easier for new users to get started. I did about three practice logins (without recording), each time building up a better story in my head and refining the approach.

Next I tried one recording - almost immediately I changed the story mid-flow because I thought of something new to say...and I wrecked the recording. No problem, I just canceled the recording and started again.
The second take worked better but I remember trying to over-explain the process of the OpenID login (I didn't have a script, just notes). The act of making-it-up-as-you-go-along means sometimes you start to blather on with too much detail. I canceled this attempt.

The third attempt went fine (I'd had five practice goes before this!), I just uploaded it and added it to the BuildBrighton wiki.

The above sequence should make it clear why some practice runs and note-taking is important. Always know what you want to achieve and make your explanation stick to that goal. I tried to explain OpenID too much because that part of the script wasn't clear in my head - by making the mistake and then iterating on my plan I soon solved that problem.

**BarCamp Workshop - "Recording the screen in 7 minutes with Jing"

At BarCamp Brighton 4 in August 2009 I ran a [30 minute workshop](http://thescreencastinghandbook.com/blog/screencasting-in-7-minutes-with-jing-workshop-at-barcamp-brighton-4/) to get people started using Jing. The entry talks about what we covered and it has a video (recorded on a second MacBook sitting opposite me) so you get to see me explaining everything, if you're curious.

The entry also has a link to Jez's 2 minute example screencast that he made from my direction. He made this using Jing on his Mac, he walks through the processes of searching Wikipedia for 'RMS' to show the viewer how to learn about the 'RMS Titanic'. He hadn't screencasted before but had wanted to for internal communication, once he saw how easy it was he reported that he'd definitely be using Jing internally.

MAKING A SCREENCAST IN 2 HOURS

If you have two hours spare and you've already tried the '30 minutes lesson' in the previous chapter then you're ready for the next step. With an extra couple of hours you're in a position to try a more powerful tool which allows you to edit out mistakes.

Here I'll cover Camtasia Studio 6 (I'll use Windows but it works in a similar way on the Mac) and BBFlashBack Pro for Windows and ScreenFlow for Mac. Camtasia costs around $300USD and BBFlashBack Pro costs £109GBP (approximately $160USD) and ScreenFlow is $99USD. All are available as trial versions.

Camtasia Studio is the grand-daddy on Windows, it works well for normal productions. Generally I recommend this if you have the budget.

BBFlashBack is the number two, it also works well for normal productions (see my BBFlashBack review). It lacks some of the features of Camtasia (e.g. denoising the audio, adding title screens) but otherwise has all the usual editing features and export options. If you are screencasting occasionally it might be better to use this as it is fairly powerful and cheaper than Camtasia.

ScreenFlow 2 is one of the major packages on the Mac. It has a good editor, version 1 wasn't as powerful as Camtasia Studio 6 on Windows but it has lots of nice graphical effects. Version 2 adds lots of new editing features, I've only used Version 1 (I've yet to upgrade) but the demo videos on Telestream's site show its power. The steps below are based on my experience with ScreenFlow 1 and (from what I can see) they're entirely relevant to ScreenFlow 2.

You won't have time to do much more planning (the creative development process takes more time, we'll discuss that later). You will have time to try editing out simple mistakes, cleaning up the audio a touch and creating things like title screens.

PLANNING

With just two hours to play with you won't have extra time to plan a more complex screencast (but we'll cover these details later for when you have more time).

You could spend some more time expanding the ideas that you want to cover and putting them to a friend or colleague to get their feedback. Keep working to reduce as much fluff as possible so you're showing the minimum amount of material that explains just what the user needs.

You might also want to think some more about what the viewer really needs to see, making the right screencast completely flows from addressing the needs of the viewer.

WORKFLOW

You'll find workflow discussion in the Google Group e.g.:

- work flow strategy covering the idea of recording audio and video in steps in one session, with many notes from other screencasters

work flow expectations covering some thoughts on screencasting in 2 hours or less
optimal workflow ideas covering a long discussion of various workflow approaches for high-quality screencasts
Silvio Grosso's Linux/Windows workflow describes how he uses a virtual Linux instance on Windows and records and edits using open-source tools

I'll assume that you've already prepared your demo on paper and practiced it a few times (as discussed in the previous chapter). Remember to take a deep breath and smile (smile!) before your start speaking - a friendly smile comes through at the start of the recording.

When speaking remember that you can edit out breaths, 'ums', 'errs' or background noises as long as there is no on-screen action happening at the same time! I tend to say a sentence and then take a breath (which I can cut out later), if I hear background noise (e.g. a car or a sea-gull here on my coast) then I wait until the noise passes and cut out that entire section.

Generally I'd advise sticking to 5 minutes of recording for a tutorial or 2-3 minutes for a product demo.

RECORDING

Each of the major packages make it easy enough to record the screen. Here we'll cover the straightforward process of recording the desktop ready for editing in the next section.

When you get the option, choose to record the application's window that you're demoing rather than the entire desktop. The goal is to record only the visual elements that are necessary so the user gets to focus on the key visuals, rather than irrelevant stuff on your desktop.

Do remember that you often have a Pause feature in screencasting tools - this lets you pause recording part-way through so you can catch your breath and focus on what comes next.

Camtasia Studio 7 (Windows)

TechSmith provide a 3 minute Record Full Screen (http://www.techsmith.com/learn/camtasia/7/record-full-screen/) video which shows you how to record your first video. Their Learning Centre (http://www.techsmith.com/learn/camtasia/7/) links to a larger set of in-depth videos that teach you about recording, editing, producing and sharing.

1. Create a new project
2. Start recording using 'Record the screen'
3. Select a window (better) or full desktop, press the red 'rec' button
4. Let countdown pass
5. Take a deep breath, smile, wait half a second and then start the demo
6. Hit F10 to end recording or click the Stop button, Save the video
7. Record another take or a new scene as needed
8. Click Edit to be returned to the editor
BBFlashBack Pro 2.6.6 (Windows)

BBSoftware have two videos showing how to record and edit using FlashBack 2 on their Demo Videos page.

1. Start the Recorder by pressing the red record button on the toolbar
2. Step through the wizard, perhaps make sure the recording toolbar appears in the Windows toolbar rather than floating on the screen
3. Choose a window (better) or the full desktop
4. Wait a moment as it starts recording
5. Take a deep breath, smile, wait half a second and then start the demo
6. Click the Stop button, save the video
7. Record another take or a new scene as needed (record new scenes with Add Additional Recording in the File menu)
8. Record more scenes or replacement scenes as necessary

ScreenFlow 2.1 (Mac)

Telestream have a great set of demo videos (http://www.telestream.net/screen-flow/demos.htm) that show ScreenFlow in action. The Overview video will give you a good idea of what's available and it shows how easy it is to make a first recording.

1. Start ScreenFlow and a dialog appears
2. Decide whether you're recording the webcam, mic and computer audio, press the Record button
3. The full desktop is recorded after a brief countdown
4. You'll see a small icon in the top bar showing a video-camera with a dot in it, this means recording is activated
5. Smile, take a deep breath, wait half a second and then start the demo
6. Click 'Stop Record' after clicking the video-camera icon in the top bar or use Shift Command 2

Editing

Let's look at easy editing steps that will make your videos look a whole lot more professional than a simple Jing recording.

Your aim here will be to remove obvious mistakes (like narration errors or bad scenes), remove dead scenes (where nothing interesting happens so the video is shorter), clean-up the audio (if offered in the package), add title screens (if offered in the package) and to add zooms and annotations to focus the user's attention on key areas.

Finally you will export a finished video which you can upload to another site or host yourself.

Camtasia Studio 7 (Windows)

TechSmith's Learning Centre (http://www.techsmith.com/learn/camtasia/7/) has a series of videos for editing, producing and sharing.
1. Play the video to verify that your audio and video were recorded - you'll see two timelines at the base of the editor. Did you miss anything out that might force you to re-record?
2. Play the video, identify sections that need cutting (e.g. dead areas or mistakes), zoom in on the timeline (with the magnifying glasses) if needed
3. Highlight regions of the timeline and Cut them (Ctrl-X) to take them out of the timeline
4. Create a Title Clip to add title pages and closing credits. Add these to the start and end of the video - a title page is good for the title, date and author, a closing credit might be useful for attribution, license and website links
5. Add Zoom-n-Pan to focus the viewer's attention or use Smart Focus for auto zooms (but you'll probably need to tweak these by hand)
6. Add callouts (after doing zooms as they're dependent on screen location) to highlight parts of the screen and add text annotations
7. Use the noise-reduction function to reduce background hum in the audio. You get better results by manually selecting a region of noise (with no speech) but the automatic de-noiser works ok
8. Export the finished video using 'Produce and share...'. MP4 is a good choice for uploading to sites like YouTube, FLV is good if you're hosting your own video (it will play on almost all computers), SWF is useful if you have interactive elements (only available in Camtasia and not in other packages) but the video quality can suffer.

**BBFlashBack Pro 2.6.6 (Windows)**

BBSoftware have a video showing you how to edit using FlashBack 2 on their Demo Videos page. To edit your recording:

1. On the FlashBack toolbar you'll see a small arrow which expands into a drop-down menu - here you'll find a quick link to your recent recordings. Choose the one you want to edit
2. Play the video to verify that your audio and video were recorded - you'll see two timelines at the top of the editor. Did you miss anything out that might cause you to re-record?
3. Play the video, identify sections that need cutting (e.g. dead areas or mistakes)
4. Highlight regions of the timeline and delete them (Del) to take them out of the timeline (don't use Ctrl-X as that acts as Close!)
5. Add zooms via the 'Add Zoom/Pan' button and remember to click 'Apply Effects' or you won't see the zoomed result
6. Export the finished video using 'Export...' in the File menu. MP4 is a good choice for uploading to sites like YouTube, FLV is good if you're hosting your own video (it will play on almost all computers). Remember to leave Apply Effects enabled when you export or your zooms won't be applied in the exported video!

If your audio is a bit noisy then you'll want to export the audio track and clean it up:

1. Right click on the audio track, it is probably labeled 'Microphone'
2. Click 'Export Track to File...', choose a filename for the MP3
3. Open Audacity and import the track
4. You can easily denoise the audio to remove background hum and perform other edits like dynamic range compressions inside the free Audacity, see this blog post for details
5. Export the audio from Audacity as MP3 or WAV (if you don't have an MP3 encoder installed)
6. Go to the start of the timeline
7. In the Sound menu choose 'Add Sound...', select the cleaned audio file
8. Use the grey drag-handle that's at the start of the audio track to drag it around if it needs lining up
9. You'll probably want to 'Mute Track' on the old audio track or 'Delete' it entirely

**ScreenFlow 2.1 (Mac)**

Telestream’s demo videos ([http://www.telestream.net/screen-flow/demos.htm](http://www.telestream.net/screen-flow/demos.htm)) include four on editing, five on actions (like zooms) and three on exporting, these will give you lots of background on using ScreenFlow.

1. Play the video to verify that your audio and video were recorded - you'll see two timelines at the base of the editor. Did you miss anything out that might cause you to re-record?
2. Play the video, identify sections that need cutting (e.g. dead areas or mistakes)
3. Use the Ripple Delete option to delete a section of audio & video
4. Apply call-outs to emphasise the mouse or zoom on areas of importance
5. Add Text Boxes to put text annotations onto the screen
6. Change the Video Properties to apply global zooms, transitions, reflections, shadows and angled screens if you want them (but they might just get in the way!)
7. Export the finished video, MP4 is a good choice for uploading to sites like YouTube, ScreenFlow doesn't offer FLV exports - most Macs play MP4 but if Windows or Linux viewers are in your audience then think about getting an FLV converter as they describe or try using ffmpeg. The do support the newer .F4V format which uses MP4 video, so their Flash export won't work on older Flash machines that lack the MP4 decoder (but those machines are in the tiny minority now!).

**Distributing**

Now you have the option of hosting the video yourself or having it hosted by an external provider. When you're starting out it is easiest to let someone else handle the hosting and to use their embedding code so we'll cover that here (self-hosting options are discussed later).

The easiest options by far are YouTube and Vimeo. Both are really easy to upload to and reliably give you embed code to put into your own site. This means you don't have to worry about bandwidth usage or javascript/html code creation to get the video into your site.

See the Distribution chapter later in this book for details on getting your video into YouTube, Vimeo and other sites.

For public distribution you'll probably use YouTube or Vimeo (but only for non-commercial work), for private hosting you'll probably use Screencast.com.

**Share your video and get feedback**

Once you've made your video you might want to share it in the Google Group to get some feedback. Feel very welcome to share a link, in particular outline any problems you had and areas that you'd like to improve - someone is bound to have good ideas to help you improve your technique.
CHECKLIST - I have...

- Prepared...
  ◦ read the "Making a screencast in the next 30 minutes" chapter
  ◦ figured out the aim of this screencast
  ◦ thought about what my viewer will need so they will definitely learn something new from me
  ◦ run through the demo without recording so I know I can do it all without getting lost
  ◦ made notes on paper so I keep with the flow of the demo when I do get lost (but see the point above!)

- Prototyped...
  ◦ recorded a prototype and the sound records ok and the screen's text is readable
  ◦ checked that the video is as short as it can be - 2-3 minutes for a demo is great or up to 5 minutes for a tutorial
  ◦ during the prototype I made sure to leave in sections where I could take a breath and gather my thoughts, knowing that I can edit these sections out later
  ◦ embedded the prototype recording into a page so others could definitely see it - now I won't get frustrated by discovering a problem with embedding later
  ◦ dealt with external sources of possible noise, my mobile phone is off, windows and doors are shut, fans and fridges are unplugged and the cat is out of the building
  ◦ practiced smiling when I start to record so I sound happy, relaxed and interested rather than bored or rushed
  ◦ read the narration a few times so I don't rush or go too slow

- Recorded...
  ◦ made sure I don't wiggle the mouse during recording to draw attention to parts of the screen
  ◦ made smooth, sweeping mouse movements when moving between screen elements
  ◦ viewed the final recording and nothing was missed out so the aims I had are definitely met

- Edited...
  ◦ cut dead scenes from the recording in the editor
  ◦ added some zooms so the viewer’s attention is directed to the important screen elements
  ◦ added highlights to emphasise important parts of the screen when the rest should be faded or blurred
  ◦ used text annotations and arrows to add extra information visually which back-up the narration
  ◦ checked the narration and removed background noise and silenced 'ums' and 'errs' and loud breaths

- Exported...
  ◦ exported the final video using FLV, MP4 or SWF (or maybe MOV or WMV) and viewed the exported video for any artefacts that can creep in during the export process

- Distributed...
  ◦ thought about distributing the video internally and/or externally (which could be public or private)
  ◦ read the Distributing your screencast chapter
Making a screencast in 1-2 days

Having created your first screencasts and learned a few editing techniques you'll probably want to make better screencasts in less time. Here we'll go through the main techniques that will help you make the right screencast for your audience without costing you a lot of time.

Research - what problem are you solving?

Knowing which problem you're solving - and for who - is critical to making the right screencast. Let's look at the needs of viewers for Demo Videos, Tutorials and Tech Support.

Demo Videos

Demo videos, particularly those that go into the homepage, are the stock-in-trade for ProCasts. The field is new and most software products on the Internet don't yet have a demo video - but they're an incredibly powerful way of explaining to a first-time visitor why your product is useful to them. This makes more visitors stay in your site for longer, learning more and converting into trial users (or remembering more of what you do so they can tell a friend!).

A typical first time visitor at a product website typically has a problem in mind that they're solving (and if they don't - then they're not going to buy anything). The goal of your site is to explain quickly and clearly that your product solves their problem so they can stop their search and spend their money with you, rather than elsewhere.

The website's design, copy and screenshots are all a part of this process and everyone has these elements. Few websites have screencasts - and those that do tend to communicate their message more effectively, making it easier for a visitor to tell if the product fits their needs.

Take a look at this screencast for MockUpScreens - doesn't it give you a better feel for their graphical mock-up tool and how it can be used by clients and contractors to design the interfaces for new computer systems?

Typically a demo video is:

- Short - 1-3 minutes is about the longest attention span you can get from a first-time visitors
- Focused - it shows you what you can do that solves your problem
- Relevant to your needs - you'll pay attention if it speaks of your problem and how it is solved with their product
- Helpful - at the end it will show you how to take a step forwards

A demo video IS NOT:

- A tutorial - save these for users who want to learn about your product, not the ones who need convincing
- Long - you will lose most viewers if your video is over 3 minutes long
- A feature list - write a feature list in bullet points, don't talk through a feature-list in a video
Your homepage video may be used by a reviewer to talk about your product, CNET covered BrandWatch and TechCrunch cover ZTail and both include a ProCasts video. It certainly helps to use a site like YouTube and to make the video available - Balsamiq's Company page includes links to their videos which makes it easy for bloggers to get an embeddable video for their review.

Note that a demo video does need to be professional. It doesn't have to cost a lot if you're prepared to put in lots of time (think about investing days to a week of your own time), the main requirement is that you consider the needs of the viewer and you use a good mic (good audio quality is very important!).

**Tutorials**

Tutorials are a great way to teach a user how to perform a set of actions that would be hard to describe using text and screenshots. Some products are simple enough to be taught in a couple of bullet-points, others are incredibly well designed so they lead the user through the application without any thought. Many, however, are not - this is especially true of new applications that haven't had the benefit of many iterations and lots of user feedback.

Tutorials are easy to make, you can create them using free tools (like Jing), host them using YouTube (so the hosting is free and embedding is super-easy) and replace them whenever updates occur. The goal of your tutorial is to tell the user what you'll teach them (even better: show it in 30 seconds), then lead them through the steps they take to do it themselves.

If you solve a particularly frustrating or novel problem you might find the user blogging about your tutorial so do host copies of your video with YouTube and Vimeo. If your tutorial shows the viewer how to do something that they cannot do with a competitor product - so much the better!

If you have many things to teach then break the videos down into a series of tutorials. This makes them easier to watch and easier to email as links, blog and tweet. In ShowMeDo I've created over 140 tutorial videos, many are in series like OpenOffice for Word Users (containing 11 videos), each tutorial builds upon the last and lets users build their through 40 minutes of tutorial in easy steps. Lynda.com does the same thing for graphic-design and Internet related tutorials.

**Technical support**

Technical support videos are perhaps the easiest to create. The production quality doesn't have to be very high (it helps, but it isn't required) as the users are already using your product. Typically they just want to be shown how to solve the problem that is annoying them so they can move forwards.

In ProCasts we created a set of support videos for Apex Auctions, they're in the site's footer and here in YouTube:

1. Creating an account
2. Registering for an auction
3. Bidding
4. Changing your password
You want to be making short, focused videos that clearly explain how to solve the one main problem that the user is experiencing. The videos are probably 1-3 minutes long but could be longer for complex situations.

Videos work well when the explanation would be hard to understand if explained with lots of text and images. Videos aren't the only answer - bad usability in a website can be solved with some sensible user-experience design but screencasts can be an easy fix to help users solve their problem with very little investment on your part.

**PLANNING - HOW ARE YOU GOING TO SHOW YOUR SOLUTION?**

**Demo Videos**

The point of the screencast is probably to give first-time site visitors the reasons they need to stay longer in your site and learn more (and, later, to try your product). You need to get into their head to tell an appropriate story and answer their questions.

Make a list of all the benefits of your software and think about the usual demo you give in person or over a WebEx presentation. Now rebuild that demo to be 3 minutes long and think about the classic AIDA 42 formula:

- Attention - tell them what you do in the first 1-2 sentences
- Interest - tell them the problem they're facing and that you solve it in the next few sentences (and maybe show them if it is quick to do)
- Desire - walk them through solving that problem using your application
- Action - tell them what to do next, this is your Call To Action

**You need a good script**

The most important thing you need to create a good sales screencast is a good script. It has to give the viewer all of the necessary information within 3 minutes so they know that your product is worth further investigation. You're not really aiming to get them to buy at this point, instead you're giving them the necessary reasons to stick around on your site and learn more.

Generally the user has a short list of sites to investigate (probably from Google searches) and they're looking for reasons to cut sites out of their list - you want your site to stay under their consideration, so make it easy for the user to realise that you solve their problems.

What sort of compelling story can you tell in 2-3 minutes that will press the viewer's buttons and make them think "Holy Moley! This is the product for me! I don't need to go back to Google because this will solve all of my problems!". If you can come up with a demo that gets this reaction then your bounce-rate will decrease and more first-time visitors will convert into trial users.

Making the above script takes practice. Probably you want to talk to friends and give demos, then boil down the essence of the demo into a 2-3 minute script. You have to get into the head of your potential users. What problems do they have? What sort of

demonstration gets the point across the quickest? What do they see when they're at competitor sites and what on your site makes you unique and better?

Remember that you're not giving a tutorial on the homepage, you're giving a concise sales message that shows the user why you solve their problem. Don't be afraid to skip steps and merge scenes together, all you want the user to realise is that your product is the one they have to learn more about. You can give them tutorial screencasts later when they've signed-up.

Testimonials

Do you have testimonials? Adding proof from others that your product is well regarded, probably near the end of the screencast, is a great way to alleviate some of their fears about trying something new. For WebFaction's homepage demo video we added three testimonials, they float onto the screen roughly 1 minute after each other and each testimonial backs-up a point that we make in the narration:

... for web devs @webfaction My favourite hosting provider: smart web panel, SSH, awesome service, affordable pricing."

Ryan McGreal via Twitter
Call to action

Finish with a Call to Action - show the user what to click to get started and who to email if they have questions. If they’ve watched your video all the way to the end then they’re probably interested - make it easy for them to get started and to get their questions answered. For the WebFaction demo we end on their sign-up form showing the user how to join their web-hosting system (along with a testimonial from another happy user!):

![WebFaction sign-up form](image)

Tutorials

Tutorial videos are a great way of teaching new techniques to the viewer. Make sure you keep the video focused on their needs and boil it down to the essential topics.

Who is the viewer?

Your tutorial has to teach the viewer a new technique. The worst thing you could do is to make a video that is too simplistic or too complex so make sure you know who you’re teaching.

An easy way to identify your target viewer is to build descriptions on index cards of example
viewers. Label the cards them with descriptive titles like 'first-time user Fred', 'converted from competitor Colin', 'long time partially-sighted user Patricia'. On each card write down what the user knows and where they are getting stuck. Also write down ideas about their general level of experience, this helps when you're writing the script so you use the right vocabulary - industry experts will know and accept more jargon than first-time users.

**Which key points must be shown?**

If you build the index cards described above then you'll have a good idea about the key points that have to be shown. You're probably best starting with the absolute beginners - they're the ones who are more likely to abandon your software if they can't make good progress. Show them how to move forwards with solving their own problems and they'll invest more time in your software.

Later you can cover the more complicated areas where the videos are useful to fewer people (but these tend to be very useful to those few people!).

**Keep the tutorial short with redundant information**

Your tutorial should be short - probably 3-10 minutes in length. If you have many things to show then break them down into separate videos - remember that people tend to have a short attention span and they just want to know how to solve their current problem.

During the tutorial it is nice if you can show alternative ways to achieve the same same goal. With desktop software there is usually a menu item, a keyboard short-cut and a right-click context menu item that do the same task. If you can weave these alternatives into the demo then your viewer will often recognise the approach that they use (which gives them confidence) and they'll learn new ways to do the same job. For web-apps you might have different ways to navigate through the screens to get to the same result, showing these alternatives helps the user to learn to experiment with your software.

**One video per step**

Do make sure you stick to covering one topic per video. In ShowMeDo I often create longer series of 10 or more videos where each video builds upon the last. This way a user can watch 5-10 minutes worth of content, practice, repeat if needed and then move on when ready.

By keeping the video short and focused you are more likely to help the user solve their current problem so they can get back on with whatever else they were doing, without getting in their way.

**Technical Support**

Technical support videos are an easy way to show the user how to do difficult things without re-engineering your product or web-site. In the longer term it is often better to use usability-design principles to make the product easier to use but this can be impractical, in this case showing the user how to solve their problem with a focused video is a great second-best approach.
Who is the viewer?

The most important thing for a good support video is to address the needs of the viewer - so who is your viewer? Have a think about who they are, what they know about your software and what problems they tend to have.

An easy way to identify your target viewer is to build descriptions of example viewers. Get some index cards and label them with descriptive titles like 'frequent user Fiona', 'part time user Penny', 'new computer user Nevil'. On each card write down what the user knows, what they have problems with, what they're trying to do with your software and how you could solve their problems.

Which key points must be shown?

If you build the index cards described above then you'll have a good idea about the key points that have to be shown. You probably want to make the videos that will save you the greatest amount of time, so figure out which topics affect the majority of your users. Once you solve the most common problems you can drill down to solve the less frequent issues. You're bound to know which topics cause problems because users will email, phone, tweet and blog with their experiences.

There will be users who don't get in contact when they have a problem - these users are more likely to abandon your software if they don't get the help they need. Do make sure to encourage viewers to get in contact in the hope that you can catch one of these viewers before they get too annoyed.

Keep the support video short

Your video should be short - probably 3-5 minutes in length. If you have many things to show then break them down into separate videos - remember that people tend to have a short attention span and they just want to know how to solve their current problem.

Common Pitfalls

Here are some common pitfalls that you really want to avoid:

- Making the video too long - make it as short as it can be, but no shorter
- Focusing on flashy effects and forgetting the message - it is all about the message and solving the viewer's problem
- Explaining each feature or step without context - tell a story to give the viewer context, it'll help them follow your video and remember it later
- Mute - videos are far more effective when you include narration (how many TV adverts do you see that are mute?), just make sure you have a decent microphone
- No call to action - remember to tell your viewer what they should do next (especially in demo videos!), you've got their eyeballs and they are interested so point them at the next step they should take (e.g. download a demo or contact your sales line)

Writing the Script

Having thought about what you need to show the user you'll now want to create a script. The script can be as light-weight or fully-defined as you want. Depending on the nature of
the screencast you might find that just a tiny bit of planning is required since everything you need to see is in your head - or you might find that you need to spend days building ideas, expanding concepts, creating script elements and discussing progress with your team.

Building script ideas

In the previous section we talked about identifying the key points that have to be shown in the screencast along with its intended goal. Now you have to expand these ideas into a series of events that will be demonstrated.

For tutorials and support videos you can often begin with the problem at hand and end having taught the user how to solve the problem. This gives you the start and end points, you then break down the steps that are necessary to go from the start to the end.

For product demo videos you have two choices - you can tell a relevant story or you can show the benefits (not features!) of your product. Inside ProCasts we tend to work with a story - it is a great tool for showing the viewer that we understand their problem and that the product solves this problem. This approach is also very memorable. This approach is also harder - you need to think about the real needs of the viewer and craft a story that will interest them.

The easier approach is to create a tour of the benefits of your product. The benefits that you consider are only beneficial to the viewer if they are relevant, so you need to know who is viewing your video. You can use the benefits to show the viewer why using your product will make their life better, faster or easier.

You may be tempted to show the features of your product - avoid this. End users don't care about features - they have problems that need solving and they want to see that your product solves their problem. Features are easy to talk about but they are harder for a viewer to relate to their problem, you have to do the work converting the feature into a benefit that is relevant to the viewer.

Rough notes

The easiest approach is to jot down a sequence of steps that you have to cover. The more depth you add to the notes, the more likely you are to have a high-quality end product as you’re less likely to forget key items or flub your lines. The trade-off is that you might find the notes too complex to follow during the demo, so you might sound stilted. You'll have to experiment to figure out the right amount of notes that work for you.

Storyboarding

Wikipedia defines a storyboard as "... graphic organizers such as a series of illustrations or images displayed in sequence for the purpose of pre-visualizing a motion picture, animation, motion graphic or interactive media sequence, including website interactivity.", they were initially developed at Disney. They are great for sharing ideas amongst a team of people and for visually developing the ideas that are key to the video.

A hand-written storyboard might look like the following. This shot is taken from TechSmith's blog in a discussion about storyboarding by Betsy Weber, it was developed by
You can see that the storyboard contains visual elements that demonstrate what is shown coupled with a description. Normally you'd also add narration elements and you might define the purpose of the scene so you know that every element counts towards the goal of the screencast.

If you want to print your own storyboard sheets then this [free online storyboard generator](#) site will make a one page PDF document with as many sections as you need, using a 4:3 or 16:9 aspect ratio.

When working with a distributed team an equivalent approach might be to share a document. The following screen contains some narration and actions for a fictional product, the template is the 'screencast storyboard' from OpenOffice. This template can be expanded with images and links to Internet pages and mailed between a team.
Inside ProCasts we often shared a Google Doc with a client. The document starts with a description of the problem (often 'to create a homepage demo video') and then we expand all the information we have about the problem. Next we expand the notes with prototype stories and screenshots and over time we build up the final script. A Google Document has a full revision history and it can be shared and edited at the same time - this is great for when you’re working via a Skype connection with several people.

The resulting script might look a little like the following - narration is in quotes and italics and actions are indented. We start each section with a scene number and the intention and then expand the full details so we know every step that has to be completed. When working as a team it is imperative that everyone (e.g. the lead designer, the scripter, the narrator) knows what the other team members will do during the production:

---

**Scene 1 - show the user how to enter their name**

"We enter our name to start the process"

  click the Go button
  type 'Ian Ozsvald' into the Name box
  click Next button

"and now our account is ready, we can start using X to solve problem Y..."

---

**How much needs to be defined?**

Once you start to plan you’ll want to know how much detail you ought to cover. Depending on the type of demo I’m working on I’ll use different levels of detail, here’s a rough guide:
**Requirement - software demo to increase sales**

This is the sort of work we do in ProCasts for clients. Development can involve a team of 2-5 people along with the client with a time frame of a few days to a few weeks. Much of the work goes into researching the needs of the viewers so we can tell a useful, engaging story that will help show the viewer why this software is useful to them.

To produce the screencast we'll create many versions of the script (5-10 iterations is quite likely) and each version fully defines the on-screen actions, the aims of each scene and the exact wording that goes into the narration. These videos often include hand-crafted animations which can take days to create.

The narration is always recorded separately by one of our narrators and edited into the timeline by our video producer. This alone can be quite a time-consuming process.

**Requirement - tutorial videos to teach a friendly audience**

For ShowMeDo tutorials I'll record a set of 3-10 videos that last 5 minutes each. Typically a series of 10 videos will take 2 days of research, 1 day of practice, 1 day or recording and 1 day of production and uploading.

I use less post-production than for ProCasts sales demos, often I will not use separate animations but instead I'll use slides and live demos. Generally I record the audio whilst acting in the screencast. When acting I leave gaps of 10-20 seconds so I can catch my breath and smile, these are then edited out later.

**Requirement - quick internal demos for bugs and tutorials**

When the screencast has a small, friendly audience it is easiest just to record it 'off the cuff'. Generally I'll make notes and run through a few practice runs, then just record the demo. Generally I won't edit it afterwards (or I'll just do some trims to get rid of dead scenes), then the screencast gets deployed internally as a movie file (normally these aren't uploaded to the Internet). From start to end this might take 30-60 minutes.

**Rubber Ducking**

If you have trouble with the script - try Rubber Ducking. Imagine you take a yellow rubber duck out of your desk drawer, put it on the monitor, then explain the problem you're solving to the duck. This technique comes from programming - rather then distracting a colleague by telling them about your coding bug and then going 'Aha! I know what the problem is...' half-way through your explanation (then you promptly return to your desk, leaving them puzzled and interrupted), instead you can take this imaginary rubber duck out of your draw and explain your bug to it instead.

If you use this technique for the demo and you think your duck understands the problem they face and how you solve it, within 3 minutes, then you've probably got a good-enough story to tell to someone else.
**Prototype - Making the Rough-Draft**

The prototype video is only for you and anyone who might provide feedback. You need to have a prototype rather than just a storyboard before people can really give a sensible critique of what you’re showing.

You are far better off quickly creating a light-weight prototype and getting feedback than endlessly trying to improve on the concept and story. A prototype video is real and tangible, people never quite get what you have in mind if you just talk them through drawings and notes. It is far better to iterate quickly after a prototype than to hear bad news after you've laboured on the finished video.

**Nervous?**

You might feel nervous about making the recording, you just need to get on with it and put your nerves behind you.

For the first year or two of founding ShowMeDo I often felt that someone would take me to task on the quality of my videos. I’d get so worried that I'd fail to even make the prototype. What I learned was that it is far better to make something (even if it isn’t perfect) than nothing at all - you have to make it and get it out there. You can always iterate and improve on your videos later if required.

**Recording**

The advice for getting on with recording the real video is the same as for recording the prototype - just get on with it. It is easy to keep iterating on the story or worry about how your voice will sound - it is much better to keep making progress. Everything can be improved at a later date if required.

**Take a breath and smile**

The start of the screencast is rather important - you set the viewer's expectations using the tone of your voice. You probably want to be friendly, helpful and chatty rather than bored, monotonous and sad. You need to start with a smile!

When I start a screencast I always remember to:

1. Sit upright and take a breath
2. Smile
3. Remember my first line
4. Start by speaking slowly with a friendly tone (and often with an introduction that says who I am)

**Pause, take another breath, smile and continue**

You will probably forget where you are from time to time or just get tired - this isn't a problem if you have an editing package. Just pause and take a break. You want to be quiet as you start the pause and when you're finishing so you can cleanly cut the pause away, inbetween you can be as noisy as you want.
You might find that a glass of water and a lip salve are helpful. Water can help a dry throat - say a few words before you start the narration again or the excess water in your mouth can be heard (it sounds 'watery'). The lip salve can help if your lips get dry - dry lips can make you lick your lips during the narration, this makes a lip-smack sound on the microphone. You won't hear it when recording as you're used to ignoring that sound but it'll be very clear in the final recording (and they're devilishly hard to edit out!).

Remember to reposition yourself the same way when you continue your recording - you want your mouth to be the same distance from the microphone. If you move your mouth to a different place (e.g. you feel relaxed and so you start to slouch in your chair, changing your posture) then the volume of your narration will be different and your tone might change.

If background noise interrupts during your recording then just pause and let it pass. Sometimes you'll have to re-do the scene, other times the background noise can barely be heard.

If the background noise interrupts your acting then you can continue the visual elements whilst narrating and then say the last piece of audio again (without doing any further acting). This way you'll have a back-up narration for this segment, you will have to do some audio editing to move the sound-track around without disturbing the video elements that you've already recorded but this tends to beat starting again. Always say more than you might need if you're repeating the narration, there's nothing worse than repeating some of it only to find it wasn't enough and you have to start the recording all over again.

**Screen resolution and your recording area**

Broadly speaking there are two ways to define what you're recording - you're either capturing the entire desktop or just one window. Often you'll record at a higher resolution to the one that you play back with so you want to record the source video with as small an area as possible.

The worst example I once saw was of a widescreen dual-monitor desktop where the entire desktop was recorded and the video was played back in YouTube. Approximately 2,560 pixels by 800 were recorded and played back at 320x240, so the source recording was shrunk by a factor of 8. This meant that all the text was 8 times smaller - the video endlessly zoomed in and out in an attempt to let you see what was happening. The effect was dizzying and very unpleasant.

The ideal situation is to make it really easy for the viewer to read all of the text in the final video. You want to select the smallest region of your screen to record so your text shrinks the least. In ShowMeDo the videos play back at 640x480, often I’d record at 800x600 and sometimes at 1024x768. These sizes preserve the aspect ratio so the video isn't stretched on the horizontal or vertical axis and text is still very readable.

Do also read "Common screen resolutions for viewers" in "Export - which file formats do you need?" later in the Handbook, this will get you thinking about the resolution of the exported video.
Suggested recording resolutions

If your screencasting tool lets you record just your application's window then this is normally the best choice. Set your application's window to be 1024x768 and make your recording.

If you export at 640x480 later then the text will be legible and you'll have recorded a large enough area that you application can (probably) be shown in its entirety.

Be aware that if you record larger areas then you'll need to export at a larger size - but you can't go much beyond 640x480 before people with laptops no longer see all of your video (see "Common screen resolutions for viewers" for details).

Choosing other recording resolutions

One choice you will need to make is between the regular 4:3 aspect ratio or the 16:9 widescreen ratio. Examples of the recording resolutions are shown below.

If you want to host the video as HD in Vimeo or YouTube then you do want to think about these ratios - see "Export - which file formats do you need?" for a discussion of export resolutions for different hosting sites. Make sure you choose a 16:9 recording resolution if you want to display the video at 16:9 and stick to 4:3 if you want to display at 4:3. If you don't, you'll end up either with a stretched/squeezed video or a letterboxed video (with extra black bars along with sides or the top).

To choose a 4:3 recording resolution simply multiply your recording width by 0.75, see the examples of normal 4:3 aspect ratio recording resolutions:

<table>
<thead>
<tr>
<th>Width</th>
<th>Height (Width * 0.75)</th>
</tr>
</thead>
<tbody>
<tr>
<td>640</td>
<td>480</td>
</tr>
<tr>
<td>800</td>
<td>600</td>
</tr>
<tr>
<td>1024</td>
<td>768</td>
</tr>
<tr>
<td>1280</td>
<td>960</td>
</tr>
</tbody>
</table>

To choose a 16:9 recording resolution you need to multiply your recording width by 0.5625, see these examples of 16:9 widescreen recording resolutions but note that the 1920x1080 resolution is likely to be too big for regular playback on normal computers with readable text:

<table>
<thead>
<tr>
<th>Width</th>
<th>Height (Width * 0.5625)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1280</td>
<td>720</td>
</tr>
<tr>
<td>1920</td>
<td>1080</td>
</tr>
</tbody>
</table>

Setting your recording resolution

Your choice of recording area will completely depend on how small a window your application can be recorded in (some web-apps don't work in a browser that's under 1024x768 pixels) and on your final video hosting platform. Your best option is to make a trial recording, upload it to your video hosting platform and see how it looks. If you can't read the text clearly then you'll have to try again with a smaller recording region.
If you're wondering about the right screen size then see this Google Group thread. It starts off with a general discussion about screen sizes and ends up with low-level Linux screen sizing tips.

Some Mac screencasting tools (e.g. ScreenFlow) can't record just a window, instead the application records the whole screen. The easiest way to record the right material is to set your screen resolution to the smallest useful size so everything is on screen during recording.

If you don't have 16:9 resolutions (e.g. on my MacBook I have 16:10 resolutions like 1280x800) then choose the closest relevant width and read "Export - which file formats do you need?" to learn about 16:9 Mac exports.

With Windows recording tools (Camtasia Studio and BBFlashBack) you can click an application's window or select a screen region. Use a tool like Sizer (see the "Screencasting tools" chapter) to set your application to the right size, select it and then record just the application. This reduces the need for editing later and you can capture with a 4:3 or 16:9 aspect ratio.

**Using Slides and PowerPoint**

Slides are a great way to introduce concepts that can't be easily explained by showing your application. Gibraltar Software create a programming tool that helps programmers to debug applications. They introduce the idea with the following animated slide and then go on to show how the application works with a traditional screencast. The slide sets the scene so it is easier to understand43 what follows:

![Gibraltar Software Screenshot](image)

You can generate slides in any application but PowerPoint (and the Mac and OpenOffice equivalents) are probably the easiest tools. Keep it simple and don't spend ages talking over a static slide (this is boring!). Keep it moving and use a slide for 5-10 seconds and then step to the next one or, preferably, get on with the demo.

**Animations**

Animations are another great way to explain higher-level concepts, here's two we created in ProCasts. The first explains the 24 hour processing model for a [fund management system](#), characters are shown using the system as a shadow denoting the changing timezones moves across the screen:

The second explains how the [web-application for Crunch](#), coupled with an accountancy firm, creates a freelancer's book keeping system:
If you're using animations then either buy simple stock animations or work with someone who can create a high-quality production. Animations can look awful if they're done on the cheap and that will harm your screencast.

The above animations were created in Adobe Flash and exported as .MOV video files. These were then imported into Camtasia Studio and Adobe Premiere, inside the editors they were just treated as more assets to be edited together.

**Wallpaper and icons - tidy it up**

Wallpaper can be distracting, you're best using either a neutral colour or a clear logo. Scenes of natural beauty or your favourite art will get in the way of your demo and your viewer will wonder why it is relevant.

Desktop icons can also be distracting - everything that's on screen should be there for a reason. Spare desktop icons and windows get in the way, tidy these up and hide them before you record.

**Animated effects**

Modern operating systems apply many visual effects like sliding, exploding and zooming windows, fade-in drop down menus and swish desktop-switching fades. These can look
good but they only work if the video plays back with a high enough frame-rate. If your playback rate is less than 10 frames per second then desktop animations will, at best, look very gritty and lumpy. At worst they will look glitchy and confusing.

Many video sites have a fast frame rate (which you need for 'real world' video) but if you're producing your own videos and you're playing with the frame rate, make sure it isn't too low. It might be easier to disable desktop effects whilst recording.

**Get rid of annoying pop-ups**

There's nothing worse than getting half-way through a screencast and suddenly having one of these alerts pop onto the screen:

- "Windows (/Mac/Linux) Update Is Now Ready"
- "New Java Version Available"
- "New Version of Flash Ready To Download"
- "Anti-virus Updates Ready"
- "You're Running Low On Disk Space"
- "You Have Unused Icons On Your Desktop"

Often you can't disable these. My general practice is to let the machine update itself and then reboot it or at least to wait 10 minutes after boot-up before I start recording. I've been hit by the Windows Update message more times than I care to remember.

**Don't wiggle the mouse**

If you want to draw the viewer's attention to an element of the screen - don't wiggle the mouse. This looks awful. Use an editor and apply a zoom or highlight - they're more effective and look far slicker.

For a clear example see the video in his blog post: [http://thescreencastinghandbook.com/blog/avoiding-mouse-wiggle-use-zooms-and-spotlights/](http://thescreencastinghandbook.com/blog/avoiding-mouse-wiggle-use-zooms-and-spotlights/)

Wiggling mice are visually distracting, you can use a zoom to focus the viewer's attention completely on one area or a spotlight so one part of the screen is greyed-out - this gives you context when discussing the spotlight section (you'll see a clear example in the video above!).

**Avoid mouse marathons**

Sometimes you'll be referring to screen elements that are a long way from each other. Don't move the mouse to one, talk about it, then move it back to the other, talk about that, then back to the first, then back to the other...This looks silly. You're better off using an editor and applying a highlight or zooming and panning the focus.

**EDITING AND PRODUCING**

**Cutting away the rubbish**

It is very important to trim away all of the dead footage in your screencast. One mark of a bad video is where the viewer has to watch drawn-out, slow scenes that should have been chopped away. The worst example would be the waiting during a software installation
sequence - the bit with the hour glass that inevitably takes a few minutes should *chopped away* after you record.

For episode two of the *Introduction to Writer* tutorial series in ShowMeDo the installation sequence for OpenOffice lasts for 40 seconds, the majority of it discusses the configuration options. The part where the progress bar is shown runs for just a few seconds whereas the actual installation time at this point is several minutes:

Any section that has a long pause is a candidate for trimming - even if the long pause was just you gathering your thoughts. If it 'feels a bit awkward' when you watch the video, trim it down.

Obviously you also have to trim out any mistakes. If removing the mistakes means that you get a jump with the action on-screen - perhaps the mouse moved during a mistake so by removing the mistake you see the mouse in very different positions on adjacent frames - consider adding a fade. Sometimes it doesn't matter, other times the difference is jarring and it is nice to use a quick fade (lasting less than a second) just so the viewer knows that the difference wasn't in their imagination.

Zooms to focus attention

Zooms let you move into a screen to draw the user's attention to a detail. In the following screenshot we've zoomed in from a full web-browser shot so the viewer can read the details of a task in this scheduling system:

Without the zoom the user would have heard the narration discuss the task dialog but they may not have seen it clearly. Don't forget that whilst a user will be able to read everything in your application, in a video that's presented at 640x480 pixels everything will be smaller, especially small dialog boxes.

Highlights to focus attention

Highlights are a great way of focusing the user's attention on a detail of the screen when you want it to be shown in relation to other information in the scene. In the following screenshot from a programming demo for ResolverOne called "Black-Scholes and Value at Risk with RunWorkbook" I've used a highlight to fade the rest of the screen away whilst the Pricer column is discussed. In the same scene the other columns are discussed, for this part of the video it is useful to focus the viewer's attention on just this column:

Text annotations to add useful information

Text annotations are a great way to give the user additional information - you can back-up the point you are discussing in the narration with a visual reminder so the viewer has a better chance of understanding what you're showing them.

In the demo video for VisualSVN[^46] I show three computers (the server and two client machines), I use text annotations to remind the viewer which machine we're looking at during the screencast. At the start of the video I make the point about the three machines clearly by discussing it and printing it on screen:

[^46]: http://www.visualsvn.com/visualsvn/demo/
Intro screens

An introductory title screen can tell the viewer what the video is about. This is useful if you have more than one video and if the video won't play just from your web-page - if the user finds the video "out in the Internet" then a title screen can tell them whether the video is relevant to their needs.

In our open-source advocacy video for the AdblockPlus Firefox plug-in we have a simple title screen that uses the title of the plug-in (so YouTube viewers know what they're watching if they stumble upon the video), a short sentence that tells you what you'll see and the AdblockPlus logo:

You can add a date to the title screen if the software is likely to age - if viewers watch an out-of-date video that *almost looks like it might work* then they will get confused but a date-stamp on the title page will help. Version numbers will also help. This is less of a problem if the software version is clearly shown during the video.

You might also want to add contact details and licensing information.

**Exit screens**

For the [AdblockPlus](http://adblockplus.org/) video we finished it with a simple exit screen that attributes the plug-in to Wladimir Palant, the video to our [ProCasts](http://procasts.co.uk/) and shows the [Creative Commons By Attribution](http://creativecommons.org/licenses/by/3.0/) license:
For our client Crunch.co.uk we simply end with the URL that a viewer needs to visit to join their service and their phone number:
You might also want to show:
- your email address
- your blog
- you Twitter name
- your web-site’s video page (if you have other videos)

**Recording the narration**

Narration is an essential part of a screencast. It is important to spend some time on your audio setup and to think about how you’ll speak and how you’ll record the narration.

**Do you need narration?**

It is essential to have narration. You can have a mute screencast (e.g. see my [WindowTabs screencast critique](http://blog.procasts.co.uk/2009/05/critique-of-windowtabs-screencast-useful-but-mute/)) but you’re missing an essential method of convincing your user that your product is the one they need or that your tutorial is easy to follow. It goes without saying that the narration has to be clear and easy to understand, preferably in a similar accent to the one recognised by your target viewers.

See also my [critique](http://blog.procasts.co.uk/2009/05/critique-of-windowtabs-screencast-useful-but-mute/) of Orchestrate’s original tour video, created by the CEO. He sounds...
bored and it makes the screencast difficult to watch. Remember to keep your voice animated and interested - imagine you're giving a live demo to potential clients who are sitting around your desk - they're nodding enthusiastically at you and want to learn more!

If you really don't want to do your own narration, think about out-sourcing the generation of an mp3 to a site like voices.com. Note that you'll have to edit the recording into your video and that you risk having the narration sound like it was recorded by someone who doesn't know what the application does (when this happens it sounds very odd!).

For tutorials you definitely want to add narration and it is fine to use your own voice. If you're worried about an accent then just speak slowly and be as clear as possible.

**Recording narration as you demo or after?**

There are two main techniques to recording your narration, you can record it whilst acting or you can record it afterwards. Your choice can drastically alter your production times and also the quality level of the recorded audio.

It is easiest to record the narration whilst you're acting as everything comes together at the same time. It is hard however to *do this well* - it tends to take a lot of practice. It can be nerve-wracking too as you don't want to get it wrong because you know a re-record is behind you if you make a mess of it!

My best advice is to take it really slowly if you record the audio whilst you record the video. Leave yourself lots of gaps where you can take a breath and gather your thoughts and edit these out later. This is certainly the fastest method if you're happy with the output - I record my ShowMeDo tutorials this way (with gaps for breaths and to check my notes).

For a higher quality finish it is easier to record the video component - again leaving lots of spare footage - and then record the narration later. When you record the narration later it makes sense to use a tool like Audacity, you can keep speaking your sentences until each one is perfect. Next you cut up the recording into sentence or paragraph-long chunks and save these as separate .wav or .mp3 files. Finally you import these into your screencast editor and line up each piece of audio on the timeline.

This second approach is obviously more time consuming and fiddly. You should consider this approach if:

- you find speaking whilst acting to be too hard (this is the case for many people and it took me a while to learn the skill)
- you want your narration to be perfect

For ProCasts all of our audio is recorded separately. Either we use Camtasia Studio 6 to join the audio with the video (for simpler productions) or we use a more comprehensive editor like Adobe Premiere for fine control. Often I'll create a prototype of the narration so we can create a prototype video and later I'll employ one of our narrators to do a more polished voice-over for the finished video.

With some editors (e.g. Adobe Premiere) if you replace the sound samples by copying new ones over the old ones you can replace the entire audio track easily. This means that you can re-record the audio and give the sentences the same file names and then Premiere uses

---

52. [http://blog.procasts.co.uk/2009/05/critique_of_orchestrate/](http://blog.procasts.co.uk/2009/05/critique_of_orchestrate/)
the new samples in place of the old samples. This can work with Camtasia Studio but it notices that the files have changed, sometimes it has trouble importing the new versions.

**Microphones**

The key to recording good audio is to have a good microphone (think: Garbage In, Garbage Out). Your source recording has to be as good as possible:

"audio has even been shown to affect the audience perception of the quality of a presentation more than the visuals. ... So, sound has the power to raise (or lower) audience perception of visuals, but visual doesn’t have the power to change how the audience perceives the audio." - [Creating Passionate Users](http://showmedo.com/videotutorials/video?name=1100000&fromSeriesID=110), January 2005

Generally this means getting rid of the el cheapo mic that came bundled with your PC (typically a tiny mic that connects with a 3.5mm jack plug) and replacing it with a better USB mic.

The 3.5mm jack plugs are a source of noise as the analogue signal goes via an analogue-to-digital converter in the PC, along the way it picks up electrical noise. USB connectors transfer a digital signal so no extra noise is introduced and you want to **minimise noise sources**! Spend £50 GBP (approx. $75USD) on a good USB mic and the quality of your narration will noticeably improve.

See the "Microphones and hardware" chapter for more details.

**Environment**

When recording audio you'll want the quietest possible environment. Be aware that a decent mic will pick up the sound of passing cars, people talking in the background, air-conditioning (and PC fans), mobile phones, aeroplanes and refrigerators. Some of these sources cause sound, some also produce electrical noise (e.g. mobile phones). Be sure to record in a room with no other people, at a quiet time of day, with the windows shut and mobile and electrical devices turned off.

Constant background noises like a PC fan can be removed in any good editing package (including Camtasia and Audacity). You are always better off if you can reduce the number of noise sources, many can't be easily edited out (e.g. a ringing phone, background conversation, bird calls) and will affect the listener's perception of your video.

For an example of background noise, listen to the first 30 seconds of Siddhi's [Django tutorial](http://showmedo.com/videotutorials/video?name=1100000&fromSeriesID=110) screencast at ShowMeDo, you can clearly hear the background street noise. You do not want this sort of noise in the narration that's designed to convince the viewer that your product has a high quality and is worth paying for! Note that I'm not knocking Siddhi's production (he's had very strong positive reviews and has made a great set of resources for ShowMeDo), his work is entirely appropriate for ShowMeDo's tutorial requirements - but you wouldn't want this on your homepage for fear of putting off purchasers.

---

De-noising the audio and Dynamic Range Compression

Unless you're working in a sound-proof booth you're bound to have background noise in your narration recording. Common background noise includes

- Ambient background noise like computer fans
- Street/office noise (e.g. voices, doors closing, cars driving past)
- Electrical interference

Removing ambient background noise is easy if you can export the audio track and you have an hour or two to spare. Watch my Editing Audio from Camtasia with Audacity\(^5^4\) video - it isn't specific to Camtasia at all (but was made with a request from TechSmith), it just focuses on Audacity. This tutorial will show you how to:

- remove background noise (you can do this for all the audio or just sections)
- silence periods with breathing and lip-smack noises
- fade in and out to cover-up harsh cuts
- use dynamic range compression to improve the low and high pitch balance
- normalise the volume levels (this is a manual process, see The Levelator below for an automatic equivalent)
- de-ess using the volume control to soften strong sibilants

If you have exported your audio then a better tool for dynamic range compression is The Levelator\(^5^5\) (Windows, Mac, Linux). You simply drag in your audio file and it will quieten the loud sounds (like the start of sentences where you might speak with more gusto) and amplify the quieter parts so the overall track has a more balanced volume. You don't have any choices to make, it does it all for you.

Ambient background noise can be removed automatically using Camtasia Studio 6 on Windows. Camtasia Studio 6 includes tools for audio enhancement, they have a great audio tutorial page\(^5^6\).

After you have recorded your audio you can use the Audio Enhancements page to 'automatically try to detect noise in the timeline'. This will scan the timeline for regions where you aren't speaking and it will try to create an audio filter that removes the constant background noise that it samples. These results can be variable - I tend to use the manual process on the same screen where you select a region of pure noise, ask it to Remove Noise and then confirm the result.

Adding Sub-titles

Subtitles are a great way to make your video more accessible to viewers who don't speak English. Subtitles were originally developed to help the hard of hearing, with the rise of video on the Internet they're also a great way to make it easy for non-native speakers to follow what you're showing.

Adding subtitles depends on the player that you're using. You can either add them as a separate file (often using the SubRip\(^5^7\) .srt format) so the player adds them as an overlay or you can hardcode them into the video.

---

If you use a separate file you have the advantage that search engines can read them which might help your ranking, the downside is that different versions of Flash can render the subtitles in different ways (possibly depending on the fonts that are available on the viewer's machine). It can also be possible to supply different subtitles for different languages.

If you hardcode the subtitles then you can only supply one language, you do however get to control how the subtitles are rendered.

Aside from YouTube I have limited experience with using subtitles. If you’d like me to discuss further approaches then post your request into the Google Group and I’ll do some research. My recommendation is to use a separate file so you get a little bit of SEO benefit.

**YouTube and .srt files**

For [WebFaction's](http://www.webfaction.com/demos/tour) homepage demo we use YouTube for hosting, YouTube makes it very easy to add subtitles in many languages. To add subtitles simply:

1. login to YouTube
2. visit My Videos and find the video you want to work on
3. click Captions
4. click Add New Captions or Transcript
5. browse for a captions file, select the language of the captions, Upload File
6. you can keep adding new captions in the same language to replace an older version

If you watch the WebFaction demo then you'll see that the subtitles appear and disappear at specified times. The .srt (SubRip) format is a very simple caption format and it is very popular, YouTube understands it and there are many programs to help you make the file.

The start of the subrip file for the WebFaction subtitles looks like this:

1
0:00:01,800 --> 0:00:04,500
Jim wanted to quickly setup a website with his own applications,

2
0:00:04,800 --> 0:00:07,500
with full Secure Shell access and helpful support

3
0:00:08,000 --> 0:00:09,500
WebFaction were recommended and he

4
0:00:09,500 --> 0:00:14,500
liked the examples posted by other users in the Show Off Your Site forum, so he setup an account

The file is a plain text file, each section has a scene number (e.g. 1, 2, 3, 4), a timecode (from --> to) and a line of text.

---

Keep the line of text short, if it is longer than a normal line of text then it takes up too many lines on screen and looks silly (and it is hard to read it quickly!).

**Making a .srt (SubRip) file**

As I mention that are many programs that help you to make a .srt file. In my experience it is (shockingly!) still easier to hand-build your own .srt file. Here's my process:

1. add the first line of narration using a guessed *from* and to *time*
2. open a video player like *VLC*, open your video, open the subtitles
3. watch the video, decide if your subtitle is lined up appropriately
4. modify the .srt file to adjust the timecodes, repeat until you're happy
5. add a blank line
6. add the next line of narration and continue

This is a tedious process. If you've found a program that makes this easier then I'd love to hear about it - post it into the Google Group and I'll add it here and acknowledge your input. If you'd like to see a screencast of this process then post into the Google Group and I'll produce one.

**Adding music**

Some screencasts work well with a backing track, the homepage demos that we created for *Crunch* 59 and *ButterfieldFulcrum* 60 both use music. Backing tracks don't tend to work well for tutorials but for demos they can help with setting the mood and pace. If you're not sure feel very free to post a question into our Google Group to see what everyone thinks.

**Places to find music**

You need to have permission when you use music created by somebody else. You can either find tracks that are free for private use without payment or you can buy royalty-free licenses to commercial tracks.

Remember that you're not buying music for you but for your viewer. Think about what they might like and where they might play it (something loud and violent might not work well on an office computer's speakers!).

Here's a list of free legal music sources:

- OpSound
- Jamendo
- FreeMusicArchive
- Live Music Archive at the Internet Archive
- Creative Common's audio section (for more sources)

Alternatively you can license royalty-free tracks, the typical cost will be $5USD to $50USD, this seems to vary more on the artist than the length of the song or the quality of the music! Generally royalty free tracks can be played as many times as you like in your finished video, as long as you've bought a license. Here are a list of some stock music providers, inside ProCasts we've bought music from the first two:

59. http://crunch.co.uk/
60. http://bfgl.com/
Adding the music

This step can be a little more tricky. Some editors (including Camtasia) let you add a second audio track so you can just add your music. The problem is that often you want to control the music's level (how loud it is) - at the start of the video you might intro with the music but by the time you start to speak you might want the music to be quieter. Later if you have a section of the demo with no narration you might want the music to become louder and then quieter, with a fade out right at the end of the video.

Most screencast editors don't let you add an envelope for the volume on a track so you have to use an external editor. Audacity is great for this, here's a YouTube video by user Sonovert which shows you how to edit the volume for music tracks with the envelope tool.

When using music typically I start it loud, fade it down after a few seconds (to match the start of the narration) to a low level, keep it at the low level throughout the video and then make it fade to nothing after the narration has finished so it gently fades away in the last few seconds.

Once you've set the envelope in Audacity you just export a new audio file (often as .mp3 or .wav) and import this track into your screencast. You may have to iterate a few times to get perfect timing but the results are worth it.

Ideas for improving your screencast

For Demo Videos:
- Are you telling a compelling story or just stepping through a feature list? Tell a story that's relevant to the viewer and show them how you solve their problem - they'll pay far more attention as there's something in it for them
- Don't try to address dis-similar markets with the same video or you'll confuse your viewers. Know your viewers and focus on addressing their questions. If you have to make two videos (e.g. one for Corporate clients and one for casual home users) - do so
- Use the homepage video to tempt the user to stick around for longer, use tutorials once the user has decided to try the product - don't use a tutorial on the homepage or you'll bore all the undecided visitors!

For all videos:
- Mix different types of media - use still images, screenshots and animations - they'll stimulate the brain in different ways
- If you have trouble with the script, turn to a professional copy-writer. A copy-writer can help you with the message, structure and language - this is especially important if you're writing English but it isn't your native tongue

61. http://www.youtube.com/watch?v=FYA6LmhAzGE
• Remember that **improving the quality** of the audio has a larger impact on the viewer's perception of quality than improving the visuals - use a good mic and speak clearly with enthusiasm
• If you have real problems with the voice-over, out-source the job (e.g. to [voices.com](http://voices.com))
• There’s no excuse for smudgy, low-bitrate videos any more. Bandwidth is cheap and YouTube (amongst others) offer very clear hosting for screencasts
• Get feedback from your users, use this to iterate and make a v2 or make several targeted videos

**Measuring Results**

You've probably discovered how important it is to test the results of changes you make to your website. Screencasts for marketing should definitely be tested - you have to invest resources creating them and you need to know how well they perform against other changes you could make with the same resources.

The simplest test is a classic A/B page test (e.g. using [Google Website Optimiser](http://google.com/websiteoptimizer)). In test A you might have a large 'Show the Tour' play button, in the B test you might have a screenshot in the same place but **no tour video** - which one drives more sign-ups and trials?

Having confirmed that more conversions occur when you have a screencast (if they don't - have a really hard think about why not!), think about how you can persuade more visitors to watch the video. YouTube is useful here - you can use the Insight page to see how many views you get each day. Try variants on the screenshot - do you just need a 'play' symbol? How about adding some text ('Watch the 3 minute Tour?') and telling the user that you're solving their problem 'See how AcmeSolution solves Problem X in under 3 minutes'?

If you've used YouTube to host the video, take a look to see if people are commenting and leaving a rating. This feedback could come from a cold visitor who found the video in YouTube and who hasn't seen your website - did they understand the message you were sending?

When you are testing, I'd love to hear your results! If you can share your experiences, I'd be happy to give my thoughts - increasing user-views is a subtle topic and I'm keen to hear about what works and what fails. Share your findings in our [Google Group](http://google.com/group) or contact me directly.

**Share your video and get feedback**

Once you've made your video you might want to share it in the [Google Group](http://google.com/group) to get some feedback. Feel very welcome to share a link, in particular outline any problems you had and areas that you'd like to improve - someone is bound to have good ideas to help you improve your technique.

**Checklist - I have...**

• Prepared...
  ◦ read the "Making a screencast in the next 30 minutes" and "Making a screencast in 2 hours" chapters
  ◦ figured out the aim of this screencast
thought about what my viewer will need so they will definitely learn something new from me
run through the demo without recording so I know I can do it all without getting lost
made notes on paper so I keep with the flow of the demo when I do get lost (but see the point above!)

• Scripted...
  ◦ For a Demo Video I’ve listed all the Benefits that I have to show or explain and I’ve figured out a nice story to tell
  ◦ For a Tutorial or Support Video I’ve listed all the steps that must be shown
  ◦ Read the script to someone else and they agree that it covers everything that’s needed
  ◦ Written the script to include all the narration, actions and notes that I need

• Prototyped...
  ◦ recorded a prototype and the sound records ok and the screen's text is readable
  ◦ checked that the video is as short as it can be - 2-3 minutes for a demo is great or up to 5 minutes for a tutorial
  ◦ during the prototype I made sure to leave in sections where I could take a breath and gather my thoughts, knowing that I can edit these sections out later
  ◦ embedded the prototype recording into a page so others could definitely see it - now I won't get frustrated by discovering a problem with embedding later
  ◦ dealt with external sources of possible noise, my mobile phone is off, windows and doors are shut, fans and fridges are unplugged and the cat is out of the building
  ◦ checked that the computer isn’t under a heavy load so the fan is running slowly and quietly
  ◦ practiced smiling when I start to record so I sound happy, relaxed and interested rather than bored or rushed
  ◦ read the narration a few times so I don’t rush or go too slow
  ◦ a glass of water to hand in case my mouth goes dry
  ◦ vaseline or lip-salve to hand in case ‘lip smack’ sounds come through in the prototype

• Recorded...
  ◦ cleared the recording area of icons and windows that aren't useful
  ◦ set my background image to a neutral colour or a useful image like a logo
  ◦ checked that all the latest Windows, Flash, anti-virus and Java updates are applied so no annoying pop-ups appear
  ◦ checked that if windows are opened during my screencast, they appear in the recording region
  ◦ cleared my browser's form-field cache and URL history if I'm using the browser (so nothing silly appears!)
  ◦ made sure I don't wiggle the mouse during recording to draw attention to parts of the screen
  ◦ made smooth, sweeping mouse movements when moving between screen elements
  ◦ viewed the final recording and nothing was missed out so the aims I had are definitely met

• Edited...
  ◦ cut dead scenes from the recording in the editor
  ◦ added some zooms so the viewer's attention is directed to the important screen elements
• added highlights to emphasise important parts of the screen when the rest should be faded or blurred
• used text annotations and arrows to add extra information visually which back-up the narration
• checked the narration and removed background noise and silenced 'ums' and 'errs' and loud breaths
• only used one or two types of fade throughout the video and each is quick (about one second long)

• Produced
  ◦ Added an opening title slide with the title, my name, email address and date
  ◦ Added a closing title slide with a website and call-to-action details (if they’re relevant) and the license

• Exported...
  ◦ exported the final video using FLV, MP4 or SWF (or maybe MOV or WMV) and viewed the exported video for any artefacts that can creep in during the export process

• Distributed...
  ◦ thought about distributing the video internally and/or externally (which could be public or private)
  ◦ read the Distributing your screencast chapter
**EXPORT - WHICH FILE FORMATS DO YOU NEED?**

**WHAT ARE THE BEST DIMENSIONS FOR EXPORTING THE VIDEO?**

As discussed in "Screen resolution and your recording area" in "Making a screencast in 1-2 days" your goal is to record enough of the screen so everything the viewer needs to see is recorded - and to export at a resolution that is easy for the viewer to watch.

Often you will want to record a large part of your screen to show your application, it would seem natural to export your video at the same resolution. The problem you face is that many people in the world still have small screens.

**Common screen resolutions for viewers**

The following table shows the screen resolution breakdown for ShowMeDo.com's 50,000 monthly visitors:

<table>
<thead>
<tr>
<th>Screen resolution (pixels)</th>
<th>Percentage of monthly visitors</th>
</tr>
</thead>
<tbody>
<tr>
<td>less than 1024x768</td>
<td>7%</td>
</tr>
<tr>
<td>1024x768</td>
<td>15%</td>
</tr>
<tr>
<td>1280x800</td>
<td>20%</td>
</tr>
<tr>
<td>1280x1024</td>
<td>16%</td>
</tr>
<tr>
<td>greater than 1280x1024</td>
<td>42%</td>
</tr>
</tbody>
</table>

The viewers with screens smaller than 1024x768 include netbook and smart phone users (the smallest reported resolution for December 2009 was 320x240), the ShowMeDo website doesn't render properly below 1024x768 so these users are ignored (unless their screens allow scrolling).

Both 1024x768 and 1280x800 restrict the depth of the screen to around 768 pixels (800 pixels is only a few more than 768) so a sensible common lower-limit to bear in mind is that your video should be viewable on a screen that is 1024x768 pixels. The screenshot below was taken on my 1280x800 MacBook screen showing a 772x579 demo video. 93% or more of my site's visitors will be able to see this video comfortably using the general figures above (actually for ProCasts over 98% of my visitors can see the video!).

As you can see below the 772x579 version of the video almost completely fills my MacBook's Safari window. If a second tab was opened then the remaining spare space disappears. If a video with depth greater than 579 pixels is shown then the controller bar disappears - so you can't pause or control the video (but with this player you can click in the grey regions to the side to make it close). Do think about your audience's computer screens when you're choosing your display size!
If you have a specific audience - Mac users for example - you may be tempted to use a larger format as most Mac users have large desktop screens. Note of course that laptops have smaller screens - my MacBook is 1280x800 (see the screen shot above). Many laptops now have a widescreen format and these often limit the depth of the video to 579 pixels or less.

**Recommended export resolution**

A safe export resolution to work with is 640x480 (using a 4:3 ratio) - it will work on almost everyone's computer screen even if they have overloaded browser windows and other things that fill their screen.

A larger alternative would be 772x579 (again using a 4:3 ratio) - this should fit onto all modern laptop screens even with a browser's toolbar and navigation bar open.

**Video examples of different export resolutions (including SD, HQ and HD)**

It is much easier to watch a video that's been recorded and uploaded at different resolutions if you want to see how different the resolutions can be. For the table below I've recorded a set of four videos (one is widescreen and the others are normal) and uploaded them to YouTube and Vimeo.

The recording shows the playback of a "real world" Vimeo HD video that was shot in my city of Brighton. It shows the Vimeo interface, video playback and then full screen playback. If you watch the highest resolution video and then the lowest one you can see the difference
in the captured framerates, using a screencasting tool to capture full motion video really pushes the software to its limits.

The videos were recorded using Camtasia Studio 6 on Windows XP, in each case I changed the computer's resolution and recorded the full desktop. For all of the videos I recorded at 30 frames per second, for the second widescreen example I exported the video again at 15 frames per second - both YouTube and Vimeo still interpret it as an HD video and you can barely see any different between the two.

There are several things to note:

1. If you watch the widescreen video in YouTube and Vimeo, enable HD in YouTube and then go full screen with both you'll see that they both are very high quality. I'd argue that the Vimeo video was slightly better defined (it is generally accepted that Vimeo use more bits to store their data so their video is clearer) but YouTube's is very good.
2. You can easily embed YouTube HD into your site with their embed code, you have to buy Vimeo's Plus account to embed their video in HD.
3. Both 15 and 30 fps video gets shown as HD in both sites.
4. The widescreen (1280x720) videos are very hard to read if they're not at full screen - bare this in mind when you're choosing your recording resolution.
5. The video recorded at 1024x768 is easy to read when it is shown inside YouTube and Vimeo (they display the video at roughly 640x360).

You'll find the clips below in the External links column, open these in a web browser. You can see the links directly in the Handbook's accounts at YouTube and Vimeo:

- http://www.youtube.com/user/ScreencastingHandbk
- http://vimeo.com/user2879256

<table>
<thead>
<tr>
<th>Resolution</th>
<th>Frames per second (fps)</th>
<th>External links</th>
<th>YouTube HD?</th>
<th>Vimeo HD?</th>
<th>YouTube's Max. embed resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>1280x720 (16:9)</td>
<td>30</td>
<td>YouTube, Vimeo</td>
<td>Yes</td>
<td>Yes</td>
<td>853x505</td>
</tr>
<tr>
<td>1280x720 (16:9)</td>
<td>15</td>
<td>YouTube, Vimeo</td>
<td>Yes</td>
<td>Yes</td>
<td>853x505</td>
</tr>
<tr>
<td>1280x960 (4:3)</td>
<td>30</td>
<td>YouTube, Vimeo</td>
<td>Yes (with black borders)</td>
<td>No</td>
<td>640x505</td>
</tr>
<tr>
<td>1024x768 (4:3)</td>
<td>30</td>
<td>YouTube, Vimeo</td>
<td>Yes (with black borders)</td>
<td>No</td>
<td>640x505</td>
</tr>
<tr>
<td>800x600 (4:3)</td>
<td>30</td>
<td>YouTube, Vimeo</td>
<td>No - HQ</td>
<td>No</td>
<td>640x505</td>
</tr>
</tbody>
</table>

The videos were recorded on Windows XP and exported using Camtasia Studio 6, I’ve listed the export details below - see the suggested export settings sections that follow for your own guidelines.

Note in YouTube if you watch the HD videos without switching to the full screen view then you'll see black bars to each side of the video. This is an artefact of their player - the videos are actually being shown at 16:9 (I've measured the pixel dimensions to confirm this) with additional black bars surrounding the player.
Export notes for the widescreen (16:9) video

The 30 fps video was exported using Camtasia Studio 6's "HD" setting, this preserves the output frame rate (the video was recorded at 30 fps). The 15 fps video was exported in the same way but with the output frame rate manually set to 15 fps.

- Screen resolution -1280x720 (via File|Project Settings)
- Video settings configured as "HD" via File|Produce Video As..., the following settings are the defaults, they were investigated by doing "Add/Edit Preset" for "HD":
  ◦ file type: MP4 (the option is listed as "MP4/FLV/SWF")
  ◦ format: MP4 (via Flash Options|Video)
  ◦ frame rate: automatic - this preserves the recording frame rate (via Flash Options|Video)
  ◦ key frames: one every five seconds (via Flash Options|Video)
  ◦ encoding method: "Quality" (via Flash Options|Video)
  ◦ "Quality": set to 55% (via Flash Options|Video)
- Audio settings:
  ◦ bitrate: 96 kbits (via Flash Options|Audio)

Export notes for the normal aspect ratio (4:3) video

For the regular aspect ratio videos I created a "Custom" export type which uses:
- Screen resolution - set to 800x600, 1024x768 or 1280x960 (via File|Project Settings)
- Video settings configured as a "Custom Production Setting" via File|Produce Video As...:
  ◦ file type: MP4 (the option is listed as "MP4/FLV/SWF")
  ◦ format: MP4 (via Flash Options|Video)
  ◦ frame rate: 30 fps (via Flash Options|Video)
  ◦ key frames: one every five seconds (via Flash Options|Video)
  ◦ encoding method: "Quality" (via Flash Options|Video)
  ◦ "Quality": set to 55% (via Flash Options|Video)
- Audio settings:
  ◦ bitrate: 96 kbits (via Flash Options|Audio)

Goal - what do you want to achieve with your video?

The goal of your video may dictate how you think about the export settings. The main choice is between whether your video wants to be seen by as many people as possible or whether the video will live in your own site under your own control.

Goal - widest possible distribution

Wikipedia lists 57 video hosting websites, if you want to distribute your video to the majority of these then you will need a video that will work well for them all. Looking at Wikipedia's comparison of video services there is a mix of resolutions from 320x240 (4:3) to 1280x720 (16:9). All of the sites accept an MP4 file upload.

800x600 would be a sensible choice for your export size. The video will either be downsampled (to e.g. 480x360, 768x576 or a similar resolution) or stretched or cropped to fit a widescreen resolution and it will be legible in all of the sites.

Using MP4 you can keep a high video quality, since each site will transcode the video to their own preferred format it won't matter if your video is larger than it could be when you upload it. Note that many of the video distribution sites have limitations like 100MB maximum file sizes, you'll need to respect these limits.

**Goal - widespread distribution in the most common video sites**

The most common English-language video distribution sites are probably YouTube, Vimeo and Blip.tv. All three support widescreen and normal formats so you could use widescreen to show wider applications. Don't forget that the widescreen displays will shrink down to smaller sizes (e.g. Blip.tv's upload page talks about 640x360) so make sure that your text is still readable!

As for "Goal - widest possible distribution" you'll want to export using MP4 at the highest quality level as these sites will transcode the video to their own formats and sizes.

**Goal - embedding in your own site and some distribution**

Building upon "Goal - widespread distribution in the most common video sites" you could use YouTube, Vimeo and Blip.tv for exposure and embed the video in your own site using one of their embed options. This means that you don't have to worry about using your own bandwidth.

Don't forget that some sites don't like commercial videos - Vimeo for example won't let you show videos that are commercial (but tutorials and support videos should be fine). You'll find details further below in "Distributing your screencast".

When you embed the video into your own site do remember to use an option that will play clearly on a screen resolution of only 1024x768 - you don't want to lose viewers.

**Goal - hosting in your own site**

If you are hosting the video in your own site then generally I recommend exporting the video using the FLV format (see below) rather than MP4. You can use MP4 but you won't know if all of your viewers have a recent enough version of Adobe Flash to decode MP4 in their browser. All the big sites (e.g. YouTube, Vimeo) generally use FLV but they encourage you to upload using MP4 so they have the highest quality material to convert from.

Having analysed ShowMeDo's visitor logs I know that about 2% of visitors don't have an MP4-compatible version of Adobe Flash, this number will vary depending on your audience. If your audience is web-savy then it is possible that all of them will have a recent version of Adobe Flash which includes MP4 support.

You'll have the option of exporting your video in either regular or widescreen dimensions. Inside ProCasts we generally export videos for clients at 640x480 (4:3) using FLV since we know that these will fit into the majority of laptop and netbook computer screens.
DESTINATIONS - WHERE WILL YOUR VIDEO LIVE?

Depending on where you want your video to live you'll want to choose the right export settings for your video. The main choice will be whether the video lives in a 3rd party site like YouTube, in your own site or on a mobile phone. Suitable settings are discussed below.

Destination - YouTube

YouTube has an [optimising your video uploads](http://www.google.com/support/youtube/bin/answer.py?answer=132460&topic=16612&hl=en-GB) help page which lists the requirements for HD and normal uploads, I've summarised the details using my own experience below. See "Video examples of different export resolutions (including SD, HQ and HD)" earlier in this chapter for examples of successful YouTube HD exports and the settings I used.

**YouTube HD settings**

**General export settings**

- Screen resolution -1280x720 or similar with a 16:9 ratio (note if you upload at 640x360 (16:9) then the video is displayed widescreen but without the HD logo - that's because the video is smaller than YouTube's player)
- Video settings (you might already have an "HD" export option in your program with settings like these pre-configured):
  - file type: MP4 (to create a .mp4 file)
  - format/codec: H.264 or MP4 (H.264 is the codec name that you want, MP4 is the shorthand that some tools might use)
  - frame rate: automatic (YouTube doesn't seem to care about low framerates, I've used 15fps and 30fps successfully for HD)
  - key frames: one every five seconds (the more key frames you have the higher your visual quality and the bigger your file size, it won't affect HD encoding but it can improve the final visual quality)
  - encoding type: some times you get the option for "two pass" or "best" encoding - always choose these settings for screencasts
  - encoding method: "Quality" with a "high" quality level (in Camtasia Studio I've used "55%", this is their default for "HD" export)
    - OR
  - encoding method: "bit rate" this entirely depends on the complexity of the visual elements of the video and the resolution, a figure of 4,000 kbps is probably sensible. Below 2,000 kbps you'll probably see artefacts (blocks or smudges of colour that look bad). Above 6,000 kbps you probably won't see any difference. Bigger values make for bigger files.
- Audio settings:
  - audio type: AAC if available, MP3 is second best (AAC produces higher quality audio than MP3)
  - bitrate: 96 kbits for mono audio, 256 kbits if you have a really fancy microphone and you really care about the audio definition
  - sample rate: 44.1 kHz (also known as 44 kHz)
Camtasia Studio 7 (Windows) export settings

Camtasia Studio 7 introduces a "Share to YouTube" option via "File|Produce and share...", previously you had to export a MP4 video and then upload it via YouTube's web interface.

If you want to manually upload your video (perhaps you want to use another editor to make some final edits to the video) then use these settings:

- Editing dimensions -1280x720 (via View|Project|Editing dimensions...)
- Video settings configured as "HD" via File|Produce and share..., the following settings are the defaults, they were investigated by doing "Add/Edit Preset" for "HD":
  - file type: MP4 (the option is listed as "MP4/FLV/SWF")
  - format: MP4 (via Flash Options|Video)
  - frame rate: automatic - this preserves the recording framerate (via Flash Options|Video)
  - key frames: one every five seconds (via Flash Options|Video)
  - encoding method: "Quality" (via Flash Options|Video)
  - "Quality": set to 55% (via Flash Options|Video)
- Audio settings:
  - bitrate: 96 kbits (via Flash Options|Audio)

Camtasia 1.1 (Mac) export settings

Camtasia for Mac is a new tool and the output options aren't as flexible as they should be in later versions. Getting an HD output when you don't have the right aspect ratio is a bit of a pain - my MacBook has a 1280x800 screen (16:10) and the YouTube direct upload options making it fiddly to get an HD video into YouTube without cropping.

If you have a widescreen (16:9) resolution already (e.g. 1280x720) then you don't need to change anything as the full screen recording will have the right ratio. All you need to do is use "Share | YouTube...".

When I first tried the built-in "Share | YouTube..." option it uploaded the 1280x800 video, when the YouTube link was ready I discovered it had uploaded a 1280x720 video with automatic cropping of the top and bottom of the screen. The result of this is that 40 pixels from the top and bottom of the video are deleted - this is enough to lose the app menu bar at the top and the dock at the bottom. There appears to be no way to not crop the upload. By investigating "Edit | Adjust Canvas..." I discovered that it was set for "YouTube HD (1280x720)".

To avoid cropping I use "Edit | Adjust Canvas..." and choose "Fit to Recording Size" (on my MacBook this is 1280x800). When I "Share | YouTube..." the video uploads as a 1280x800 video (no cropping) and YouTube shows it in an HD widescreen player with black letterbox bars.

ScreenFlow 2.1 (Mac) export settings

ScreenFlow has a "Publish to YouTube" option and you can tick a box to generate an HD video of the export - note that this stretches your screen if you didn't record in 16:9.

An alternative is to select "Export..." and "Scale to a custom size" in the 16:9 format (e.g.
1280x720) and tick the "Letterbox Content" which adds black bars to the left and right sides so your video isn't stretched.

You could record in a 16:9 screen format if your monitor allows it (my MacBook is 16:10 and doesn't offer 16:9 aspect ratios) and then you won't need to stretch or letterbox the output. This means you'll have to change your screen's resolution to e.g. 1280x720.

**READER: If your program has different configuration options to the ones listed above then send me an email (or better - post into the Google Group) and I'll help you out, when we've settled on the right figures then I'll update this section of the book.**

### YouTube SD/HQ settings

The main difference between Standard Definition (SD) or High Quality (HQ) and High Definition (HD) is that HD uses the widescreen format and SD/HQ use the normal 4:3 aspect ratio. If you upload in a 4:3 resolution (or indeed any non-16:9 resolution) then YouTube will attempt to convert the video to a form of HD and it'll add black bars to the left and right sides of the video to fill in the gap.

If you upload a video at 640x480 or above then you should always get the HQ option (which is crisper than the older SD). Above in "Video examples of different export resolutions (including SD, HQ and HD)" you can see that an 800x600 upload was converted to HQ and 1024x768 and 1280x960 were both converted to HD with black bars.

### Destination - Vimeo

Vimeo has a [compression guidelines](http://vimeo.com/help/compression) chapter that details the settings you need for successful uploads. See "Video examples of different export resolutions (including SD, HQ and HD)" earlier in this chapter for examples of successful Vimeo HD exports and the settings I used.

### Vimeo HD settings

**General export settings**

These settings are almost identical to the YouTube HD settings. Vimeo states that you need a high framerate (e.g. 30 or 25 fps) but my experiments with 30 fps and 15 fps both produced HD video in Vimeo.

- Screen resolution -1280x720 (other 16:9 resolutions might work, 640x360 does not give you an HD option even though it is 16:9)
- Video settings (you might already have an "HD" export option in your program with settings like these pre-configured):
  - file type: MP4 (to create a .mp4 file)
  - format/codec: H.264 or MP4 (H.264 is the codec name that you want, MP4 is the shorthand that some tools might use)
  - frame rate: 30 (15 fps has worked for me and Vimeo state that 25 fps works, other values may also work fine)
• key frames: one every five seconds (the more key frames you have the higher your visual quality and the bigger your file size, it won't affect HD encoding but it can improve the final visual quality)
• encoding type: some times you get the option for "two pass" or "best" encoding - always choose these settings for screencasts
• encoding method: "Quality" with a "high" quality level (in Camtasia Studio I've used "55%", this is their default for "HD" export)
  OR
• encoding method: "bit rate" this entirely depends on the complexity of the visual elements of the video and the resolution, a figure of 4,000 kbps is probably sensible. Below 2,000 kbps you'll probably see artefacts (blocks or smudges of colour that look bad). Above 6,000 kbps you probably won't see any difference. Bigger values make for bigger files.
• Audio settings:
  ◦ audio type: AAC if available, MP3 is second best (AAC produces higher quality audio than MP3)
  ◦ bitrate: 96 kbits for mono audio, 256 kbits if you have a really fancy microphone and you really care about the audio definition
  ◦ sample rate: 44.1 kHz (also known as 44 kHz)

Camtasia Studio 7 (Windows) export settings

Camtasia Studio 7 has no direct support for Vimeo. You'll have to use the following settings to export a MP4 video and then upload it via YouTube's web interface.

• Editing dimensions -1280x720 (via View|Project|Editing dimensions...)
• Video settings configured as "HD" via File|Produce and share..., the following settings are the defaults, they were investigated by doing "Add/Edit Preset" for "HD":
  ◦ file type: MP4 (the option is listed as "MP4/FLV/SWF")
  ◦ format: MP4 (via Flash Options|Video)
  ◦ frame rate: 30 (via Flash Options|Video - in my experiments 15 fps worked fine and Vimeo states that 25 fps works too)
  ◦ key frames: one every five seconds (via Flash Options|Video)
  ◦ encoding method: "Quality" (via Flash Options|Video)
  ◦ "Quality": set to 55% (via Flash Options|Video)
• Audio settings:
  ◦ bitrate: 96 kbits (via Flash Options|Audio)

Camtasia 1.1 (Mac) export settings

Camtasia for Mac is a new tool and the output options aren't as flexible as they should be in later versions. Getting an HD output when you don't have the right aspect ratio is a bit of a pain - my MacBook has a 1280x800 screen (16:10) and the YouTube direct upload options making it fiddly to get an HD video into YouTube without cropping.

If you have a widescreen (16:9) resolution already (e.g. 1280x720) then you don't need to change anything as the full screen recording will have the right ratio. All you need to do is use "Share | Export...", to make sure you have the right canvas size use "Edit | Adjust Canvas..." and choose "Fit to Recording Size".

If you've recorded at 1280x800 (MacBook default resolution) you can use "Edit | Adjust
Canvas..." and choose "Fit to Recording Size" (on my MacBook this is 1280x800). Next use "Share | Export..." and you'll get a 1280x800 .MP4 video.

Upload the video to Vimeo and it will be accepted as HD. It shows the entire video (no cropping) so it must stretch the video (although this isn't documented in Vimeo's help text). The video quality is very high.

**ScreenFlow v2.1 (Mac) export settings**

You will need to export your video to an .MP4 file and upload it to Vimeo. Choose "Export..." and "Scale to a custom size" in the 16:9 format (e.g. 1280x720). If your output video is stretched it means it wasn't recorded in a 16:9 aspect ratio - one solution is to tick the "Letterbox Content" box which adds black bars to the left and right sides so your video isn't stretched.

You could record in a 16:9 screen format if your monitor allows it (my MacBook is 16:10 and doesn't offer 16:9 aspect ratios) and then you won't need to stretch or letterbox the output. This means you'll have to change your screen's resolution to e.g. 1280x720.

*READER: If your program has different configuration options to the ones listed above then post into the Google Group and I'll help you out, when we've settled on the right figures then I'll update this section of the book*

**Vimeo normal (4:3) settings**

As noted above for "YouTube SD/HQ settings" the main difference will be whether you upload with a 16:9 widescreen aspect ratio or with a 4:3 (or other) ratio.

**Destination - ShowMeDo**

ShowMeDo presents videos at 776x582 (4:3), the general advice is to upload at 800x600, 1024x768 or above (using the 4:3 ratio). If you use the "YouTube HD settings" shown above for the rest of your export options then your ShowMeDo videos will be very crisp and clear. ShowMeDo has add video instructions.

**Destination - your site**

If you're sure that your viewers will have an up to date version of Flash (Mac users can be fairly sure of this, Windows and Linux users can't be sure) then you could simply opt to use .MP4 files. If you're not sure and you don't want any of your viewers to be unable to watch your video then default to .FLV videos. As of December 2009 the number of general Flash users who can't watch .MP4/H.264 video because their version of Flash is too old is probably less than 2%.

For both situations you'll want to read the "Common formats and a guide to their settings" section below to find the right export settings.

**Destination - mobile phones**

Current I'm only covering the iPhone. If you want me to cover a different phone - post into the Google Group and we'll see what can be arranged.
iPhone

The iPhone's resolution is 320x480 in portrait mode, if you turn it on its side then it will be 480x320. The aspect ratio is an unusual 3:2, this is part-way between the normal 4:3 and the 16:9 widescreen ratios.

If the phone had a 4:3 aspect ratio then a comparable resolution would be 480x360. For a 16:9 aspect ratio a comparable resolution would be 480x270.

Inside ProCasts when I'm exporting videos for the iPhone I export the usual version at 640x480 (for the website) and then export a second version with these settings:

- **Video settings**
  - resolution: 480x320
  - file type: M4V (important - it must be M4V and not MP4!)
  - format/codec: H.264
  - frame rate: 15 fps
  - key frames: one every five seconds
  - encoding type: (probably you don't have an option for this but "best" is always a sensible choice if it is available)
  - encoding method: 700 kbps (this could be raised to 2,000 kbps if the lower value makes the video unclear)

- **Audio settings**
  - audio type: AAC
  - bitrate: 128 kbps
  - sample rate: 44.1 kHz

**Settings in common tools:**

- Camtasia Studio 6 has an export setting for the iPhone
- Camtasia for Mac has an export setting for iTunes which works on the iPhone
- ScreenFlow has an iPhone export option

**COMMON FORMATS AND A GUIDE TO THEIR SETTINGS**

Your exporting format will determine how easy it is for someone else to watch your video and how easy it is to edit the video in a video editor. The simplest choice of format is .MP4 - it plays on over 90% of web-browsers through Flash and it also plays on most modern computers using built-in video players.

The older .FLV format is more widely supported but is a bit trickier to configure if you want small file sizes with a clean, crisp image. The Windows-specific .WMV format is very clear for small file sizes but tends to only play on modern Windows machines (so you might lose Mac or Linux viewers). Adobe's .SWF format is not a video format, instead it is an animation format, this makes it very hard to edit (unless you have Adobe's toolkit) and it doesn't support longer videos.
ArsTechnica has a nice [history of video compression technology] if you'd like further background.

A common point of confusion is between the codec and the file format. Codecs are used to compress a video into a stream of bits, these bits are held inside a file format. Below I list the common codecs and file formats along with useful settings.

**Codecs**

The name codec is the shortened form of "coder-decoder" (or, less accurately "compressor-decompressor"). It refers to the algorithm ([wikipedia]) that converts a video from one format into another.

Often you start with a huge bit-stream for the raw, uncompressed and very high quality video and you *encode* (or *compress*) it down to a smaller, lower quality version. Later you *decode* (*decompress*) the smaller version and convert it to a form that can be displayed on the screen for playback.

As time passes we see new codecs being created - these occur because people discover better ways in which we can compress certain types of video and because hardware works faster (e.g. faster CPUs, GPUs and dedicated video chips). Video for mobile phones for example needs to be easy to decompress but it can be CPU-intensive to compress because that process is often performed on a desktop computer rather than the phone.

Desktop computers are generally now fast enough to play high quality video at full screen on high resolution monitors if they have a dedicated, mid-range graphics card. In the future we can expect CPU power to outstrip the growth in screen sizes (unless we get a crazy new technology direction like 3D!) and we'll probably see less innovation in codecs.

If you want to see what codec a video is encoded in you'll need a tool - see the "Screencasting tools" chapter.

**H.264 or AVC (mainly used in .MP4 and .M4V files)**

H.264 ([wikipedia]) is almost a dominant codec for Internet-based video and in 1-2 years it is likely to be the default choice for all applications, Adobe's Flash drives the adoption. The On2 VP6 codec is more prevalent simply because it came first (from Flash 8) and H.264 was only released to Flash users from version 9.0.115.

AVC (Advanced Video Coding) is the alternate name for the H.264 codec standard.

If you are creating .MP4 files then the following settings are almost certainly the right ones for you. If you are making a .M4V for an iPhone then see the "Destination - mobile phones" section above.

- Video settings
  - file type: MP4

---

format/codec: H.264 or MP4 (H.264 is the codec name that you want, MP4 is the shorthand that some tools might use)
frame rate: automatic (this will preserve your recording frame rate)
key frames: one every five seconds (the more key frames you have the higher your visual quality and the bigger your file size - this can improve the final visual quality)
encoding type: some times you get the option for "two pass" or "multi-pass" or "best" encoding - always choose these settings for screencasts
encoding method: "Quality" with a "high" quality level (in Camtasia Studio I've used "55%", this is their default for "HD" export)
OR
encoding method: "bit rate" this entirely depends on the complexity of the visual elements of the video and the resolution, a figure of 4,000 kbps is probably sensible. Below 2,000 kbps you'll probably see artefacts (blocks or smudges of colour that look bad). Above 6,000 kbps you probably won't see any difference. Bigger values make for bigger files.

Audio settings:
- audio type: AAC if available, MP3 is second best (AAC produces higher quality audio than MP3)
- bitrate: 96 kbits for mono audio, 256 kbits if you have a really fancy microphone and you really care about the audio definition
- sample rate: 44.1 kHz (also known as 44 kHz)

On2 VP6 (mainly used in newer .FLV files)

The On2 VP6 codec is the standard inside FLV files and has been since Flash 8.

There are several key settings to bear in mind:
- data rate - higher data rates mean sharper video and lower data rates mean fuzzy video
- key frames - with infrequent key frames your video will get increasingly smudgy and then it will sharpen and then the cycle will repeat, with frequent key frames you won't see the smudging
- resolution - as the resolution increases you will need to increase the data rate otherwise you will get fuzzy video
- audio - use 96 kbps data rate with 22 kHz mono MP3, always use mono if you have only recorded one channel of music (else the second channel adds to the file size but adds no sound!)

For a 640x480 FLV video using the On2 VP6 format you will find these settings to be useful:
- data rate - start with 2,000 kbps, increase this rate if text isn't very sharp, you'll have the correct rate when "everything feels sharp enough" (this is very much down to your perception!)
- key frames - a starting point is one key frame every 30 seconds, if the video gets increasingly fuzzy and then becomes sharp at the 30 second boundary then you can increase the data rate (see below for the reason)
- the above settings should result in 2-4MB per one minute of video so a 3 minute video will be 6-12MB in size - doubling the data rate can double the size of the video

The "right" settings for an FLV video are very much an exercise in experimentation. For safety you can set the data rate to be very high (e.g. 4,000 kbps for a 640x480 video) but then your video can become very large and so it is slow to download.

The settings above are the default transcoding settings inside ShowMeDo, they keep most screencasts crystal clear at 2-4MB per minute. Inside ProCasts we have created videos which have more intense visual sections (e.g. lots of fine animation) - in these cases I have increased the data rate incrementally and decreased the key frame interval until an appropriate compromise was found which results in a high visual quality without too big a file size.

The key frames can be made infrequent (using the suggested one per 30 seconds of video) unless the user is likely to skip back and forth (the key frames define the points you can jump to in an FLV video). Normally the key frames help make a "real world" video sharp every few seconds, with a screencast we can set the data rate to be high (2,000 kbps normally is a good setting) and then we never get fuzzy video, so we don't need many key frames.

If you're having trouble with the right settings then post some details into the Google Group, depending on your needs one of us can help you choose the right numbers.

H.263 and Sorenson Spark (mainly used in older .FLV files)

The H.263 codec was released during the early 1990s, it was designed for teleconferencing with low bit rates. The Sorenson Spark codec was introduced for Internet video in Adobe Flash 6, this codec is an incomplete implementation of the H.263 standard.

Whilst this format was popular before 2008 it has generally been replaced by the On2 VP6 codec (described above). If you need to configure the Sorenson Spark codec then the settings for the On2 VP6 codec should be a useful guide.

Theora (mainly used in .OGV and .OGG files)

The Theora codec is an open-source implementation of the older On2 VP3 codec. It is very popular on open-source operating systems and it can be read by many video sharing sites (e.g. YouTube and Vimeo support files containing the Theora codec).

Despite the age of the underlying VP3 codec the team responsible have addressed many of the early quality concerns and now Theora encoded files are seen as superior to H.263 videos and comparable to H.264. For screencasting the Theora codec creates clear video, the codec is used by default in the open-source RecordMyDesktop screencasting tool.

Reader - I don't know of good settings for the Theora codec - if you can share useful resources in the Google Group then I will update this section of the Handbook.

72. http://recordmydesktop.sf.net/
DivX and Xvid (mainly used in .DIVX and other files)

DivX\(^{73}\) is a commercial codec, Xvid\(^{74}\) is the open-source alternative. These were very popular formats but with the widespread adoption of H.264 they are of little use to screencasters. Users need to install one of these codecs on their computer to watch the video.

Animation (great for lossless exporting but results in big files)

The animation codec is a lossless format, this means that the video won't lose any definition but it will have a large file size. This can be useful for archive purposes and sharing early versions of a video before exporting for general usage.

- Video settings
  - file type: MOV
  - format/codec: Animation
  - frame rate: automatic (it will export at the same rate as the recording frame rate so nothing is lost)
  - key frames: automatic
  - encoding type: set "Quality" to "Best"
- Audio settings:
  - audio type: PCM (this is a lossless audio codec)
  - bitrate: 16 bits (this can be higher if you have a good microphone setup, 32 bit might be a sensible larger number)
  - sample rate: 48 kHz (also known as 44 kHz)

Dirac

The Dirac\(^{75}\) codec is a patent-free free codec developed by the BBC. The goal is to create a free codec that is comparable to or exceeds the capabilities of H.264. Currently a research version of the codec is available but it is not widely used.

H.265/H.264+ (still a draft specification, not yet in the real world)

The H.265\(^{76}\) codec is a future version of H.264, it is still in the discussion stage. It is possible that it will not be a completely new codec, if that is the case then it may be named "H.264+". The goal is to create videos that look the same as those from H.264 but with a 25%-50% file size saving.

Containers

Containers are files that hold the stream of bits created by a codec. Some containers only take one type of codec, others are general purpose and take many types of codec. AVI, MOV and MKV are three well-known container formats that can hold many different codecs.

If you want to learn about the file format that a video is stored in you’ll need a tool - see the "Screencasting tools" chapter.

**.MP4**

For Internet video, .MP4 files are the new king and within 2 years they are likely to be the standard. Currently .MP4 videos are the easiest way to get very high quality video with relatively small files sizes into the web-browser on user's machines without requiring them to install new codecs. The driver behind this trend is Adobe Flash 10.

.MP4 files can contain various codecs but usually they contain H.264 encoded video.

Approximately 2% of a general Flash userbase will have an older version of Flash (pre 9.0.115) that can't decode the H.264 codec inside .MP4 videos, if you don't mind about these users then use .MP4 to host your video on the Internet.

If you're uploading your video into a site like YouTube or Vimeo then .MP4 files are perfect. .MP4 files can be played on the desktop on most modern operating systems.

See the discussion on the H.264 codec above to get the right settings for screencasts in .MP4 videos.

**.FLV**

.FLV files are the most common type of video file on the Internet. They weren't designed to be played on the desktop (but players do exist - see "Screencasting tools" in "Screencasting software") and generally they're played via the web browser. .FLV is a container format that can hold several kinds of video but commonly it now uses the On2 VP6 codec.

If you want to reach the largest possible audience of viewers on the Internet then you still need to use .FLV videos with an On2 VP6 codec.

.FLV is the default file type for YouTube. Commonly an .FLV file contains an On2 VP6 encoded video (see above), generally it is harder to configure the VP6 encoding for perfect results than it is to configure H.264 for .MP4 files.

**.MOV**

.MOV files are QuickTime Movies, normally these are created by QuickTime and they are often associated with a Mac. In the past a .MOV file might have contained a Sorenson 3 video (a codec similar to Sorenson Spark and H.263) or an Animation video.

You are more likely to deal with .MP4 files now as QuickTime has supported the H.264 codec for some time. All Mac screencasting tools can export .MP4 files.

---

The .AVI format is perhaps the best known video format from the last decade. It is a flexible container that can hold many types of codec. With the rise in popularity of .MP4 files it is unlikely that you'll need to deal with .AVI video. If you do work with .AVI video be aware that the format can hold many types of codec and the best settings for each codec can be hard to derive without a lot of experimentation.

The .WMV (Windows Media Video) is a Windows-specific video codec. .WMV files can encode using the VC-1 codec which is a proprietary and is reasonably similar to H.264. It also supports the Windows Media Video Screen codec which is for screencasts. This screencast codec can produce very small files but it is only seen on relatively new Windows machines.

If you are only working with new (Windows XP and newer) Windows users then the .WMV format might be useful. It did require a Pro purchase but that may not be the case now.

The .SWF (Small Wave Format / ShockWave Flash) files are a proprietary standard that can be played back by Adobe Flash in a web browser. They were popular because they created small video files, now they are generally replaced by .FLV or .MP4 video.

.SWF files are hard to edit unless you have the right Adobe package or commercial editing tool. Normally a .SWF files contains scenes of animation or images rather than video as held in the other containers. It also has a limit of 16,000 frames making it unsuitable for long videos (at 15 frames per second this equates to a maximum length of 18 minutes of video).

This format might be useful if piracy is a concern as editing it is very tricky, otherwise I don't know of any good reason to consider this as a useful video format.

The .OGV and .OGG files are a free, open container format (.OGG is the older name). This format normally holds a Theora video (see above).

.MKV is the Matroska file format, it is an open standard that can contain many types of codec. You are unlikely to come across .MKV files when screencasting.

.DivX

.DivX is the DivX corporation's file format. You are unlikely to come across .DivX files when screencasting.

.CAMREC and .CMREC

.CAMREC files are produced by TechSmith's Camtasia Studio on Windows, it is a proprietary lossless format. .CMREC is the Camtasia for Mac equivalent.

.FBR

BBFlashBack from BBSoftware stores its videos in a proprietary lossless format inside a .FBR file.

**Distributing your screencast**

You'll have to decide on where you want to store your screencast so it gets distributed to the right people. Is it ok for the screencast to be public? Does it need to be kept private? Here are some options:

- **Public** - upload it to a site like YouTube or Vimeo
- **Public view but private storage** - store it on your own web server and embed it on a public web page, this makes it harder for a viewer to embed the video onto their site (and you can do things like block external IP addresses if you know how to configure your web-server correctly)
- **Private view with private storage** - store it on your own web server and place it behind password-protected or obfuscated URLs so an outsider won't find it

The different hosting options will also consume different amounts of your time. YouTube is super easy, you can have a video embedded in 15 minutes in most sites. If you decide to host your own files and use your own player then you might need a day of experimentation to make it work reliably, especially if you need to test the player in multiple browsers and operating systems.

**Distribution and embedding (with their player)**

Below I list some of the most common video sites for hosting screencasts. If you want ideas for other sites then see TubeMogul, they offer a service that lets you upload your video automatically to 28 partner sites (including those listed below).

**YouTube**

*YouTube* is the easiest option for both distribution and embedding onto your site. Historically it offered only a low video quality but in early 2009 it offered High Def (HD) video. Personally I use it for a lot of ProCasts works, it has no commercial-use restrictions (so you can show commercial videos) and it has a huge audience of passers-by for great exposure.

Most common video formats are accepted including .OGV. Embedding a YouTube video is super-easy - if you use YouTube it is quite likely that a viewer will embed your video into their site because *it is just so easy*. Your video will also be viewable on an iPhone, this might be useful if your audience includes mobile viewers.

See my [ProCasts channel](http://www.youtube.com/user/IanProCastsCoUk) on YouTube - I host many of our client videos there along with a few for the Handbook.

Limitations: up to 10 minutes only, up to 2GB upload size.

**Screencast.com**

*Screencast.com* is offered by TechSmith. Their offering is distinct in that it hosts exactly the video you uploaded without conversion to a standard format. If you upload e.g. an MP4 video and the viewer has a Flash player that *isn’t* MP4 compatible then it won’t work (e.g.

87. http://www.youtube.com/user/IanProCastsCoUk
versions of Flash before 9.0.115). Videos hosted in screencast.com aren't exposed to the world via public directories so you can upload semi-private content, you can also host commercial material.

They offer a free basic account and a $10/mo. premium account (I use this for ProCasts and this Handbook). Screencast.com will host any video content, it doesn't have to have been created with a TechSmith tool.

See this example video[^88] hosted at Screencast.com, it is used in my Make a screencast in 30 minutes[^89] blog post.

Most common video formats are accepted including .OGV but the format that is displayed is exactly the format you uploaded (so if e.g. you upload an OGV you'll have to hope your viewer has an OGV player!). Embedding is easy.

**Vimeo**

Vimeo host very high quality videos but their Terms and Conditions forbid commercial usage so you can't show product demos or host tech-support material for commercial software. You can demo open-source tools. If you violate their T&C do expect to have your video deleted without warning or, possibly, even having your account suspended (but they seem to be friendly in reactivating the account if a genuine mistake was made, they just hate spammers stealing their bandwidth).

Hosting is free for their basic account and they have a pretty large audience of passers-by if you want exposure to new people.

See my Vimeo channel[^90] - I host some of our older ProCasts screencasts there along with various real-world videos (including some on screencasting).

Most common video formats are accepted including .OGV. Embedding is super-easy.

Limitations: Non-commercial videos only.

**Blip.tv**

Blip.tv is focused on letting people create shows with an offer of share of revenue from advertisers. You might want to consider this if you're building a channel of content that could be popular and so could attract advertiser's interest. Blip.tv videos can also be automatically syndicated to alternative major platforms including YouTube and the Internet Archive.

Wikipedia has some further background[^91]. Although the site focuses on real-world videos a quick search on 'screencast'[^92] shows that the hosting has a high quality.

[^88]: http://www.screencast.com/users/IanProCastsCoUk/folders/Jing/media/502e865f-5cf4-4f5c-8011-11f1d2636aee
[^90]: http://vimeo.com/user707645/videos
[^91]: http://en.wikipedia.org/wiki/Blip.tv
[^92]: http://blip.tv/search?q=screencast
Hosting is free and they claim a very large viewership. Embedding looks very easy.

Limitations: They're only interested in content where you own the copyright (i.e. no pirated material) and their focus is on series rather than individual videos.

**ShowMeDo**

ShowMeDo.com was created in 2005 by Kyran Dale and this very author. The focus is on tutorials for open-source software. The site has a strong open-source audience (especially for Python, OpenOffice and screencasting) and a friendly community in the Google Group.

Most common video formats are accepted including .OGV. Embedding requires a bit of manual code creation but overall is pretty easy.

I've authored over 140 tutorial videos at ShowMeDo, see my homepage\(^93\) for the full list. Note that many are for the Club which can only be accessed by paying subscribers.

Limitations: Non-commercial videos only, preferably focused on open-source software. Videos must be tutorials (everything is viewed by an admin before publishing).

**Embedding with your own player**

If you want full control over who holds your video and how the player looks then you'll want to host your own video and use your own player. Camtasia Studio on Windows comes with its own player, two other popular solutions are FlowPlayer and the JW Player.

For hosting you'll have the option of simple FTP storage (a tried and trusted method), Amazon's web services and content delivery networks. These options are discussed here.

**Camtasia Studio's Player (only for Camtasia Studio for Windows)**

Camtasia Studio lets you export an MP4 or FLV video with supporting HTML and .SWF player. You can upload the entire directory to the Internet (including your video and all the support files) and if you visit the URL in a web browser then the video will appear online. This is a convenient way to get your videos online without having to learn how another media player works.

TechSmith provide many tutorial videos\(^94\), they're all embedded using this technique. See their Produce in Camtasia Studio 6\(^95\) video for a perfect example of how their player embeds the video into a webpage.

**FlowPlayer**

FlowPlayer (flowplayer.org) is a relatively new player that is gaining a lot of popularity. These notes are for Flowplayer 3.1 on April 2010. ShowMeDo.com used to use the JW Media Player, Kyran Dale switched to FlowPlayer in 2010 to ease debugging problems and to open more options for complex use.

---

Pros:

• works with FLV and MP4 videos
• has a plug-in framework
• can be skinned
• has the option of serving different video based on user's bandwidth

Cons:

• fewer examples than the JW Media Player
• may have fewer plugins for sites like WordPress, Drupal, MediaWiki

Cost

• Free for non commercial and commercial use and the FlowPlayer logo is shown
• $95USD perpetual license allows you to remove their logo and replace it with your own

Documentation

flowplayer.org/documentation has full guides for installation, configuration and using the API.

Examples of use

To get started you should use the official documentation - the required JavaScript tends to change with each new release and any fixed example here would be out of date with the version of the player you use. For deployed examples see these three:

• ShowMeDo's videos (all videos)
• Butterfield Fulcrum Group's homepage
• Mockup Screen's video demo

JW Player from LongTail Media

The JW Player\(^{96}\) is a product from LongTail Media, it used to be called the JW Media Player and before that the FLVPlayer (until Adobe asked them to change their name). It has been around for years, it is very configurable, it is also very frustrating to configure. The help forums are very unstructured so searching for help can also eat your time and sanity.

ShowMeDo.com used to use the JW Media Player, Kyran Dale switched to FlowPlayer in 2010 out of frustration with low-level debugging and cross-platform support problems.

Pros:

• works with FLV and MP4 videos
• has a plug-in framework
• can be skinned
• has the option of serving different video based on user's bandwidth
• long history, lots of deployed examples and use case
• lots of questions answered in forum

Cons:

• forum format changes frequently and caches get lost
• forums don't tag which version of the player they're discussing so answers are often confusing when you're trying to fix a bug in a specific version
• API can change frequently in subtle and it-will-take-hours-to-debug ways
• it is generally impossible to find older documentation to match older versions of the player

Cost

• Free for non-commercial use
• 59 Euros will get you a commercial license for 1 site

Documentation

The JW Player wiki\(^97\) has full documentation and examples.

Examples of use

To get started you should use the official documentation - the required JavaScript tends to change with each new release and any fixed example here would be out of date with the version of the player you use. For deployed examples see these two:

• StartupToDo's homepage
• BrandWatch's homepage

LightBox, ThickBox and FancyBox for pop-up players

Videos take up a lot of screen real-estate, generally it is better to embed a thumbnail for a video into your page and to pop up a full-sized player only when the thumbnail is clicked. The toolkits of this effect are generally known as LightBoxes after the LightBox\(^98\) JavaScript library.

For examples of the effect see the example website links for FlowPlayer and the JW Player above. Anything can be embedded in these boxes including YouTube videos, generally they allow the video to auto-play and disappear when the user clicks away from the video. This example is from BFGL.com, it shows the homepage demo video playing over the homepage, behind the video is a small thumbnail image that activates the pop-up:

Do expect to lose half a day or more when investigating these libraries, cross-platform and cross-browser compatibility is a big time drain. Make sure you have the video working in your player in a piece of reference code *before* you try a second experiment with a Box effect! This will save you from debugging many libraries at once.

ThickBox[^99] has many demos and is easy to start with, LightBox[^100] is older, LightBox 2[^101] is newer and FancyBox[^102] is another option.

**HTML5**

The latest version of HTML is HTML 5, it includes a new `video` element that is designed to make it easy for content producers to embed video into webpages. Rather than force you to use Adobe Flash (with one of the above players), the aim is for your browser to understand several standard types of video format and to automatically play them without extra libraries and plug-ins.

Currently there is no consensus as to which video format is best supported - the battle is between the patent-encumbered H.264 format and the non-patentend but not necessarily patent-free Ogg Theora. [Wikipedia][^103] has all the details, see also a discussion about [Ogg Theora for `<video>`][^104].

Some sites (e.g. Daily Motion, YouTube) are experimenting with the `<video>` attribute but you'll need a recent browser. Since there are a large number of older browsers in use on a daily basis it will be some time before we see `<video>` appearing in normal web pages.

[^99]: http://jquery.com/demo/thickbox/
[^100]: http://leandrovieira.com/projects/jquery/lightbox/
[^102]: http://fancybox.net/
[^103]: http://en.wikipedia.org/wiki/HTML5_video
Unless you specifically want to test the new element I’d suggest sticking with Adobe Flash and a good video player.

Hosting

If you want full control over how your video is shared with the world then you'll want to choose your own hosting option. Your choice of host will determine where your video is physically stored and how it is served over the Internet to viewers.

The simplest options is using FTP hosting - this is a time-honoured way of serving static data files on the Internet. Amazon's Web Services offer two interesting services that can provide you with a low-cost content delivery network where you pay for only as much data as your serve. Larger content delivery networks exist if you have larger delivery requirements.

Generally if you're serving to a small number of users (e.g. you expect less than 1,000 full views per day) then you can probably serve your video reliably using a good FTP service or Amazon's S3. Don't fret about occasional spikes in usage, just plan for the average case.

If you're serving to millions of people per day then you'll want a content delivery network as a partner.

If you're somewhere inbetween the two then some experimentation may be required. For ShowMeDo.com I note below that we served 100,000 videos a month using an FTP service (that works out at roughly 3,000 full plays per day). To go much beyond this number we'd have needed to consider parallel servers and load-balancing.

You may be surprised at how few views you get per day for your new video. Often clients of my ProCasts expected to be hit by 1,000s of views during the release of a new video, only to discover that they had 100s of views per day at peak times. It is hard to get people to watch videos all the way through - so you may find your estimates of a user's true interest in your video to be too high. Experimentation will give you good feedback.

Your own FTP hosting

The easiest hosting to arrange would be your own FTP space. Commodity hosting sites like GoDaddy offer unlimited space and data-transfer from $15 per month (with tiered cheaper plans down to $5/month). These FTP hosts are perfectly fine for serving screencasts though the rate at which the data reaches the viewer may be less than optimal (it'll all depend on where the host is, where the viewer is and the data-rate offered by the host).

For the first 3 years of ShowMeDo.com's life we used GoDaddy to serve 100,000 screencasts per month to an international audience. Many people wondered how we achieved this feat given our limited resources - the fact was that we used $20/month hosting (as it was priced back then) for 100Gb/month of data transfer.

Generally we encoded videos at 4mb of file per minute of video (so a 3 minute video was roughly 12mb in size), these videos served fast enough that they streamed ahead of the viewing position throughout the US and Europe.

ShowMeDo.com continues to use commodity FTP hosting but now it serves via a selection of
providers. You may well find that your existing hosting for your current website works fast enough and has large enough limits - do some experimenting.

**Amazon's S3 and CloudFront as a cheap Content Delivery Network**

Amazon offer a Simple Storage Service (S3) whereby they will hold data files 'in the cloud'. You get a public URL to your video, you can then embed this URL into the HTML for your player. You won't know exactly where your video is being stored, you'll just know that it reliably gets served using Amazon's infrastructure.

To a viewer this will look like a standard FTP solution. It is more interesting for you because rather than paying a fixed monthly fee, you'll only pay for the bandwidth that you use. Note that S3 is likely to be more expensive than commodity FTP hosting.

An additional service from Amazon is CloudFront - this is a content delivery service that sits in front of S3. The goal of CloudFront is to serve your video from a data server that is physically close to your viewer's computer so your content is served faster. You pay only when the service is used, CloudFront is great for small data volumes that need to be served quickly to viewers.

If you need to serve your data quickly to both the USA and Europe then consider investigating CloudFront as a cost-effective alternative to a larger content delivery network.


**Larger content delivery networks like Akamai**

Akamai is an example of a large-scale content delivery network (CDN). They can guarantee any level of service (both up-time and delivery speed) that you require for a global audience. You will pay more for this level of service that for Amazon's alternative. Note that a CDN can offer 100% availability even in the face of large-scale network disruption due to multiple redundancy across many geographical locations.

This Wikipedia page has a long list of content delivery networks.

**Streaming media**

Streaming media solutions are useful if you serve a lot of data and if you want to control whether a user can store your video.

If you serve a lot of data then a streaming server can help you efficiently transmit just the data that a user needs to watch, and no more - this saves you in bandwidth costs. A streaming server also often comes with digital rights management which can make it very hard (though never impossible) for a competent user to store the stream of data.

It is less common for companies to use streaming media servers now unless they're broadcasting huge volumes of data. Generally it is very easy and reliable to simply transmit files using one of the other services listed above.

COMMON WORKFLOWS

A common question in the Handbook's Google Group is about workflow - what's the most efficient way to create screencasts? Below I list several common workflows.

See the Google Group for this good discussion of people's workflows.

SHOWMeDo - 'GOOD ENOUGH IS GOOD ENOUGH' TUTORIALS

ShowMeDo.com contains peer-produced tutorial screencasts made by the users and the founders. Generally speaking the tutorials are not produced with a professional team, instead they're produced by an individual with as much (or as little) time as they have to bring to the subject. If you're worried about whether people will take your videos seriously, remember that a tutorial that "does the job" is far better than a perfect production that never gets produced.

When I produce tutorials for ShowMeDo.com I generally go through this process:

1. Write down the aims of the screencast series and the intended audience
2. Break down the videos into a series of 3-10 episodes, each about 5 minutes long - at this point I just have titles and goals for each video
3. Quickly sketch the main steps for each video - here I confirm that I'm meeting the needs of my viewers
4. Run through each sketch making notes to confirm that I can do it all within 5 minutes
5. Record each episode using Camtasia Studio with live narration
6. Edit the episodes to remove dead sections
7. Add a title clip and a closing credit
8. Export and upload to ShowMeDo.com

The goal of the above steps is to keep creeping closer to the state of having a published, live video. I'd aim to record and produce each 5 minute video in 2-3 hours, on top of this there's as much practice time as required.

PROCasts - PROFESSIONAL SOFTWARE TOURS

Inside ProCasts we have a much longer process, I'll break it down here.

Understand the needs of the viewer:

1. Who is the viewer?
2. What are their needs?
3. Which messages will they see in competing websites?
4. Talk to existing users to confirm that we've identified the needs that drove them to try this product

Generally we'll work with a client on many sheets of paper - we have a questionnaire and we carry lots of post-it notes. The goal is to extract lots of information from the client's head so we really understand the space around their product and the needs of the viewer.

106. http://groups.google.com/group/thescreencastinghandbook/browse_thread/thread/f527249fd156cd5f
Storyboard:
1. Brainstorm around the needs of the user to identify story elements
2. Come up with a variety of big stories and short story segments
3. Pull out one good story that covers all the key points that we can tell within 3 minutes

Next we fall back to even more post-it notes. We write ideas for story segments onto the post-its, we'll also write characters and ideas and stick them over a table. We can order the post-its as we want and then photograph some of the results:

Create screencast:
1. Create drafts of each animation, iterate and refine with the client
2. Create draft screencast recordings with rough voice-overs, iterate and refine with the client
3. Assemble a rough screencast with a raw narration track with all assets in the correct order
4. Create the final narration, cut it into sentences and add each sentence to the right part of the screencast
5. Apply fades, zooms and on-screen annotations
6. Cut dead segments of the video to tighten the production
7. Add opening and closing titles
8. Add music, start the music loud, make it quiet for the bulk of the screencast, fade it out over the final few seconds

The final process is the one that people recognise, this is probably 1/3 to 1/2 of the total work for the project. The goal is to stay as light-weight as possible during the entire process with plenty of back-ups of earlier versions so you can step back as required and use earlier assets.

Recording Meetings

I use ScreenFlow on my MacBook to record meetings - the camera and mic are great for recording the discussion and the screencast element captures any websites we look at during the meeting. Obviously this is all done with the client's permission!

I make sure I record the full screen including the system clock. In my written notes I
annotate topics we discuss with the current time - that way I can tie up sections of my
written notes with the right section of the recording (very useful if you go back to the
recording a month later!).

**SCREENCASTING A SCREENCASTING TOOL**

A specialist case when screencasting is to use one screencasting tool to record another -
this is useful when you're teaching the viewer how to use a certain product. The obvious
thought is to run two copies of the same tool - but popular programs like Camtasia Studio
only allow one instance to run.

On Windows when demonstrating Camtasia Studio I use either Jing (for quick demos) or
BBFlashBack (for high-quality recordings) as my recorder. In the past I've also used
HyperCam. BBFlashBack Express is free and exports an H.264 video which can be imported
back into Camtasia Studio for editing.

On the Mac I've used Camtasia to record ScreenFlow and vice versa. On Snow Leopard I'm
sure that QuickTime X will record both and the resulting H.264 video could be imported
back into your preferred editor for production.

On Linux recordMyDesktop can make a recording of another instance of recordMyDesktop.
Alan Pope was kind enough to provide two demo movies as confirmation in this Google
Group thread[^107], the first shows his (large!) desktop, the second is a recording of a smaller
window within the desktop.

[^107]: http://groups.google.com/group/thescreencastinghandbook/browse_thread/thread/
a63d773f4b94e934
SCREENCASTING SOFTWARE

WINDOWS

Camtasia Studio 7

One liner: The premiere Windows screencasting package
URL: http://www.techsmith.com/camtasia.asp
Price: $299USD
My experience: Heavy use since Camtasia 3

Camtasia Studio 7 is the best screencasting package on Windows and has been for years.

Positively:
It has many features including a fully featured editor. You'll be recording, editing and exporting in the hour. It will take a bit longer to learn all of the editing features but TechSmith provide many training videos on their website so you'll quickly learn how to solve whatever task you have in mind. The audio clean-up tools are handy, they save you from having to export to Audacity. Camtasia Studio also has e-learning capabilities so quizzes are possible.

Negatively:
It has suffered from growth in certain areas and neglect in others (like the subtitle editor), it can also crash on occasion which will lose you your recent edits. If you remember to save to a new filename every 30 minutes then you'll never lose much work (it has happened to me on occasion), even with this gripe it is still the best editor on Windows.

I use it for both ProCasts and ShowMeDo productions.

BBFlashBack 2.7

One liner: The second-best screencasting package for Windows
URL: http://www.bbsoftware.co.uk/bbflashback.aspx
Price: £109GBP for Professional, Free for Express
Review: BBFlashBack Pro 108 v2.6.6
My experience: Light use with occasional heavy use since early 2.x

BBFlashBack is the second best screencasting package for Windows, it lacks some significant features like title screens but it comes at a lower price.

Positively:
It is easy to learn and you'll quickly be able to record, edit and export. There are no training videos but the documentation is good enough and with some experimentation you'll quickly see what's available. Email support is also very good.

Negatively:
The lack of a title screen editor is a pain if you want titles - you can import static images but you then have to make your title screens in a separate application. The editor also lacks an

108. http://thescreencastinghandbook.com/blog/bbflashback-pro-2-6-6-review/
automatic de-noising routine so you have to export a WAV, clean it in a tool like Audacity and then import it back in.

Jing

One liner: The easiest way to create screencasts on Windows
URL: http://www.jingproject.com/
Price: Free or $15USD/year
My experience: Light over several years

Positively:
This is about the simplest screencasting tool to get started with - it installs in a few minutes on Windows or Mac and then 'it just works'. If you just want a quick tool for recording or you want to introduce someone to the art of screencasting then this is a great place to start.

Negatively:
Annoyingly Jing is limited to 5 minute recordings. By design it does not feature an editor. The free version uploads the result as an SWF to screencast.com, the paid version allows you to upload to YouTube and download an H.264 version for local editing.

HyperCam 3

One liner: Low cost screencasting and editing package
URL: http://www.solveigmm.com/?Products&id=HyperCam
Price: 30 Euros
My experience: Early use of HyperCam 2 inside ShowMeDo, no use of HyperCam 3
HyperCam has been around for a long time, recently Solveigmm worked on a much improved 'v3' which adds a simple editor. Years back I used to use HyperCam 2 (the version without the editor) to record ShowMeDo tutorials.

Positively:
HyperCam has been around for ages, it works on all versions of Windows and it can be used to record live video and games (the framerate isn't perfect but I believe it is better than Camtasia and BBFlashBack). The editor is a welcome introduction, it only allows for simple trims and joins but for simple productions these represent the bulk of the required edits.

Negatively:
It uses whatever screencasting codec is on your machine rather than providing its own, in the past I've used the MS RLE 2 encoder which is built-in. Recording using Camtasia and BBFlashBack is more reliable as they include appropriate codecs rather than requiring you to see what you have installed and fussing with configuration options (that said, you probably have an encoder that'll work out of the box, I always did). As a workaround you can install the CamStudio LossLess Codec (see below), this is more reliable than the MS RLE 2 codec.

**CamStudio**

One liner: GPL screencasting system
URL: [http://camstudio.org/](http://camstudio.org/)
Price: Free
My experience: Heavy use of CamStudio for ShowMeDo productions

CamStudio is an open-source screencast recorder (there is no editor), earlier versions were commercial and later the authors released them as free software. The software has had no significant updates for several years and may not run smoothly with Vista and Windows 7, it also suffers from audio/video desynchronisation with some soundcards (you'll only know if it affects you by running some tests).

Positively:
It is free and it comes with a screencast codec (you can use this with other screencast recorders like HyperCam too). It is very simple so you'll quickly learn what it can do.

Negatively:
Unsupported, a touch buggy, may not work on modern OSs and audio/video can become desynchronised.

**Adobe Captivate**

One liner: Adobe's long-established screencasting and e-learning tool
Price: $800 USD
My experience: None

Positively:
This has long been regarded as one of the serious screencasting and e-learning tools. If you commonly work with Adobe products then the interface is probably going to feel familiar.

Negatively:
Very expensive, it has a learning curve that's more involved than with tools like Camtasia Studio.

**Screencam**

One liner: Another established screencasting tool  
URL: [http://www.smartguyz.com/](http://www.smartguyz.com/)  
Price: $199USD  
My experience: None

Screencam has been established since 2004, apparently it includes an editor. I have no experience with the tool at all.

**FRAPS**

One liner: Best screencast recorder for games  
URL: [http://www.fraps.com/](http://www.fraps.com/)  
Price: Free with watermark, $37USD for commercial use  
My experience: Almost none

Years back I tried FRAPS for some ShowMeDo games demos. It bills itself as the best screencast recorder to capture high frame-rate games, it seems to do the job admirably. I have no other experience with FRAPS.

**Mac**

**ScreenFlow 2.1**

One liner: Possibly the best screencasting tool on the Mac  
URL: [http://www.telestream.net/screen-flow/overview.htm](http://www.telestream.net/screen-flow/overview.htm)  
Price: $99USD  
My experience: Usage since its release

Positively:
ScreenFlow has been the leading screencast recording and editing package on the Mac for a couple of years. It is easy to use, works smoothly and has a ton of effects. TeleStream's website has many tutorial videos.

Negatively:
It lacks a de-noiser for the narration track and sometimes the editor does quirky things - placing edit points and selecting between them to add new effects and remembering to edit the camera's display on the narration track is a bit odd. You remember to get past these problems, they're just a bit frustrating at times.
Camtasia Mac 1.1

One liner: Almost the best screencasting tool on the Mac (a pretty close thing vs ScreenFlow)
URL: http://www.techsmith.com/camtasiamac/
Price: $99USD
My experience: Light usage since its release

At its release it was hoped that Camtasia Mac would surpass ScreenFlow, instead both programs had similar sets of features and Camtasia Mac felt less refined (as befitted its v1.0 status). The real benefit for us, the users, is increased competition on the Mac. Both Camtasia Mac and ScreenFlow will improve faster as a result.

Positively:
It is relatively easy to learn though the interface takes some getting used to. If you like Camtasia Studio on Windows then you'll be surprised that the interface is quite different - though you'll learn it quickly and you can trust that TechSmith will offer great tech support.

Negatively:
Strange user-interface that takes a while to learn.
Jing

One liner: The easiest way to create screencasts on a Mac
URL: http://www.jingproject.com/
Price: Free or $15USD/year
My experience: Light over several years

Positively:
This is about the simplest screencasting tool to get started with - it installs in a few minutes on Windows or Mac and then 'it just works'. If you just want a quick tool for recording or you want to introduce someone to the art of screencasting then this is a great place to start.

Negatively:
Annoyingly Jing is limited to 5 minute recordings. By design it does not feature an editor. The free version uploads the result as an SWF to screencast.com, the paid version allows you to upload to YouTube and download an H.264 version for local editing.

iShowU

One liner: A reliable screencast recorder that's been around for a while
URL: http://www.shinywhitebox.com/home/home.html
Price: $30-$60 USD
My experience: None

SnapzPro

One liner: Another reliable and long-existing screencast recorder
URL: http://www.ambrosiasw.com/utilities/snapzprox/
Price: $69USD
My experience: None
**Linux**

**RecordMyDesktop**

One liner: The best known screencast recorder for Linux
URL: [http://recordmydesktop.sourceforge.net/about.php](http://recordmydesktop.sourceforge.net/about.php)
Price: Free (open source)
My experience: Light over several years

Positively:
Generally ‘it just works’ and produces .OGV videos on Linux systems.

Negatively:
It only produces .OGV videos, you'll need to find a compatible video editor like Pitivi to edit the results.

**xVidCap**

One liner: Another well-established screencast recorder for Linux
Price: Free (open source)
My experience: None

**WEB-APPS**

**ScreenToaster**

One liner: Another simple online screencasting tool
Price: Free
My experience: Light

**Screenr (Screencasting for Twitter)**

One liner: A simple online screencasting tool with Twitter integration
URL: [http://screenr.com/](http://screenr.com/)
Price: Free
My experience: None

**Screencast-O-Matic**

One liner: Another simple online screencasting tool
Price: Free with Pro accounts
My experience: Light

**ScreenCastle**

One liner: Another simple online screencasting tool
URL: [http://screencastle.com/](http://screencastle.com/)
Price: Free
My experience: Light

**EDITING SOFTWARE**

**WINDOWS**

**Camtasia Studio 7**

Camtasia Studio has evolved over many years. The learning curve isn't too steep and TechSmith's site has many video tutorials, the software also has good help text. The editor can be fiddly and various cruft has built up in the program over the years (the subtitle editor is particularly tough to use) but overall the experience is very positive.

Do remember to save your work frequently and often to new filenames - very occasionally Camtasia Studio will crash and you'll lose all of your recent (and not so recent) edits. This has happened to me on a handful of occasions over several years and the result has generally been painful.

Multiple scenes allowed? YES
Text and graphic callouts? YES
Easily create title screens? YES
Export audio track for editing elsewhere and import again? YES
Automatic audio clean-up tools for denoising the narration? YES


**BBFlashBack 2.7**

BBFlashBack Pro's editor works in a similar way to Camtasia Studio's. In my first experience I was cutting video, adding text annotation and exporting a .MP4 video in a few hours.

The editor isn't as fully featured as Camtasia Studio's, in particular the lack of multiple scenes and title pages is a pain. To add a title page you have to remember to record a section where nothing happens at the start of the video and then import a pre-prepared image (as a callout) that fills the screen during this section.

One minor annoyance is that you can't zoom out with the timeline so you have to pan left and right to make your edits. Sometimes it is useful to see the entire video on screen so you can cut large sections - most editors provide this feature but BBFlashBack doesn't. You can still make big edits, you just have to pan left and right on the timeline until the right part of the video is on screen which is fiddly.

Multiple scenes allowed? NO
Text and graphic callouts? YES
Easily create title screens? NO
Export audio track for editing elsewhere and import again? YES
Automatic audio clean-up tools for denoising the narration? NO

Official documentation: [http://www.bbsoftware.co.uk/bbflashback/](http://www.bbsoftware.co.uk/bbflashback/)

TheScreencastingHandbook.com © Ian Ozsvald 2010
Starter tutorial: Use the movies in the link above

Adobe Premiere

Personally I haven't used Adobe Premiere, it was used extensively by my video editor in ProCasts for all of our client work. Typically I’d record screencast segments in Camtasia Studio and then export them for serious editing in Adobe Premiere. My video editor chap would add hand-created Flash animations and Adobe Premiere's editing features made the process far more efficient.

There is a steep learning curve and a high financial price so you'll only want to learn this if you'll spend a lot of time editing videos. Remember also that Adobe Premiere is very strong in the world of professional real-world video editors so much of the terminology is geared towards real-world video production, this adds some frustration to the learning curve.

Multiple scenes allowed? YES
Text and graphic callouts? YES
Easily create title screens? YES
Export audio track for editing elsewhere and import again? YES
Automatic audio clean-up tools for denoising the narration? UNKNOWN


Sony Vegas

I have no direct experience with Sony Vegas. I’ve read reports from other screencasters that Sony Vegas is a good Windows editor, I’d welcome notes from a reader to expand this section - you'd be fully acknowledged.

Blender

I have no direct experience with Blender. I've been told by Linux screencasters that Blender is a very nice editing package, I'd welcome notes from a reader to expand this section - you'd be fully acknowledged.

Official documentation: http://wiki.blender.org/index.php/Main_Page

VirtualDub

VirtualDub is a powerful and free video editor. For simple cropping and cut/paste edits it works well. The interface is a bit clunky and it lacks many features of modern screencast editors like nice text effects and zooming.

Multiple scenes allowed? YES
Text and graphic callouts? NO
Easily create title screens? NO
Export audio track for editing elsewhere and import again? YES
Automatic audio clean-up tools for denoising the narration? NO
MAC

ScreenFlow 2.1

ScreenFlow has a simple learning curve, the editor can be a bit idiosyncratic at times. I've found myself forgetting to place my cursor on the correct side of an edit point and then finding that my edits in the toolbox (for things like text animations and zooms) apply to the wrong piece of the timeline. You'll get used to this soon enough.

One annoyance in ScreenFlow is that you have to record your full screen and then apply zooms later to focus on your application. If your desktop doesn't have a true HD aspect ratio (my MacBook doesn't, it is 16:10 rather than 16:9) then you have to be careful how the final video is exported if you want a 16:9 ratio for sites like YouTube.

Multiple scenes allowed? YES
Text and graphic callouts? YES
Easily create title screens? NO
Export audio track for editing elsewhere and import again? NO (but if you export a movie, clean the audio in another tool and export an audio track you can import the new audio track and replace your original - but this is a time consuming process!)
Automatic audio clean-up tools for denoising the narration? NO (this is frustrating if your microphone is noisy - my MacBook's built-in microphone makes a continuous high pitched noise that I can't easily clean so I have to use an external microphone)

Camtasia Mac 1.1

As noted in the previous Screencasting Software section Camtasia Mac is a young piece of software. I found the interface confusing for the first 30 minutes, after that I began to get used to it. ScreenFlow has the advantage here and it still feels as though ScreenFlow is the superior product.

I suspect that there will be a continual battle between Camtasia Mac and ScreenFlow and that both of their interfaces will rapidly improve.

Multiple scenes allowed? YES
Text and graphic callouts? YES
Title screens? NO (but you can import images and overlay them at the start of the video)
Export audio track for editing elsewhere and import again? NO (the documentation says it is possible but it appears to be fiddly and I couldn't get a .WAV exported file - I'd suggest that this is beyond the ability of a normal user in this version of Camtasia Mac)
Automatic audio clean-up tools for denoising the narration? YES

Official documentation: http://www.telestream.net/screen-flow/literature.htm
Adobe Premiere

See the notes for Adobe Premiere above in the Windows section.

Blender

See the notes for Blender above in the Windows section.

LINUX

PiTiVi

I have no direct experience with PiTiVi. I've been told by Linux screencasters that PiTiVi is a very nice editing package, I'd welcome notes from a reader to expand this section - you'd be fully acknowledged.

Official documentation: http://www.pitivi.org/?go=documentation
Starter tutorial: http://www.pitivi.org/manual/

Blender

See the notes for Blender above in the Windows section.
**SCREENCASTING TOOLS**

**VIEWING SCREENCASTS**

Generally a screencast video will run in a normal media player. The exception is .FLV videos - these won't play in Windows Media Player or QuickTime, to view these on your desktop you'll need to get another player (several are free and very good).

**Video Lan Client (AKA VLC - Windows, Mac, Linux)**

http://www.videolan.org/ is an open source video player that works on many systems, in particular it can also play .FLV videos. I use it on Windows, Mac and Linux:

![IE6 Update](image)

**MPlayer (Windows, Mac, Linux)**

http://www.mplayerhq.hu is another open source video player, it works everywhere that VLC works and it also plays .FLV videos. You'll want to try both to see which interface you prefer, I use it on Windows, Mac and Linux.

The following shows the SMplayer interface for mplayer on Windows:
SCREEN RESOLUTION TOOLS

Sizer (Windows)

http://www.brianapps.net/sizer.html is a simple application that tells you the size of your window and lets you move a window to a preset location and size. If you're commonly recording the same application to make a set of videos you can ensure a consistent visual theme if the windows always look the same:

MouseZoom (Windows)

http://www.neuber.com/free/mousezoom/index.html gives you a small window that lists the current mouse co-ordinates and the colour under the cursor. I’ve used this to record and reposition the mouse between recording sessions. It also zooms the view under the cursor - but you can minimize the window to show the mouse co-ordinates so just the title bar is visible:
Mouse Position Menu (Mac)

http://alphaomega.software.free.fr/mousepositionmenu/Mouse%20Position%20Menu.html is a utility that tells you exactly where your mouse is on the Mac desktop. If you need to take multi-part recordings and you want your mouse to be positioned accurately between each recording, note the end position of the mouse with this app and reset your mouse for the next recording.

**AUDIO EDITING**

**Audacity (Open source - Windows, Mac, Linux)**

audacity.sourceforge.net is an excellent open source audio editor, it has everything you need to clean a narration recording. For my use either I import a noisy recording from a tool like Camtasia Studio for clean-up or I record my narration directly in Audacity for 100% control (I do this for all ProCasts jobs).

Inside Audacity you can split your longer narration into shorter files for easier handling. Steps I advise you to follow include:

- De-noising using the Noise Removal option in the Effect menu - first you sample a short period of noise (e.g. several seconds worth), then you apply the tool a second time to the entire recording - and then the background hum disappears!
- Dynamic range compression - use Compressor and stick with the defaults, it'll quieten any loud sounds and amplify the quieter ones so the recordings sound more even
- Amplify quiet sections - sometimes you move away from the mic and you need to boost the volume in selective regions
- Silence - sometimes you record unhelpful noises (for me: seagulls and passing aeroplanes!) - the Silence tool lets you remove these sections

For a full screencast tutorial that shows you how to edit audio from a screencasting tool (Camtasia Studio is used in this example) using Audacity, see Editing Audio from Camtasia with Audacity on the ProCasts Blog.

---

The Levelator (Open source - Windows, Mac, Linux)

Audacity has an option to even out the audio levels in a recording but The Levelator ([www.conversationsnetwork.org/levelator](http://www.conversationsnetwork.org/levelator)) does a far better job. If you have two audio sources (e.g. two speakers in a narration) then this tool becomes even more useful.

You run the tool, drag in an audio file, let it process and then it writes out a new audio file where the speakers sound as though a professional audio producers was balancing the recording levels during the performance. If ever you feel that some of your speech is too loud and some of it is too quiet then do think about using The Levelator, it really does perform some automatic wonders to even out a recording.

FOR CODEC AND FILE FORMAT INFORMATION

There will be times when you've been given a video file and it won't run or won't load in your editor. You will need to figure out which codec was used to produce the video so you can find an appropriate way of converting this format into another, more useful format. The right approach is to use a codec inspector to discover which codecs are used in the video file, the tools below are generally all free and will give you the information you need.

**Video Lan Client (AKA VLC - Windows, Mac, Linux)**

[http://www.videolan.org/](http://www.videolan.org/) lets you view Codec Details for a video including for .FLV videos. Since it works across the major platforms this makes it very useful when investigating a video that won't play to learn which codec decoder you're missing:
QuickTime (Mac, Windows)

QuickTime on Mac and Windows also lets you see a simple view of codec information. You need QuickTime Pro on Windows or Mac (or QuickTime X on Snow Leopard which includes Pro for free) to see detailed information. Here’s what you get in the free QuickTime:
GSpot (Windows)

http://www.headbands.com/gspot/ provides a very detailed account of all the media information in a video file. It only runs on Windows but if you're on Windows - this is the tool for you:

MediaInfo (Windows, Mac, Linux)

http://mediainfo.sourceforge.net/en is another free cross-platform tool, it provides detailed information like GSpot:
Transcoding

Transcoding is the act of converting a video from one format to another format. Perhaps you have a .FLV video and you want a .MP4 version, or you have a .WMV version and you want it to play on any computer rather than just Windows machines. The following two tools will do the conversion for you.

Your existing screencasting tools like Camtasia and ScreenFlow can probably load an existing video and export it in the format you need, experiment with these tools first. Note that many screencast editing tools can export .FLV videos but cannot import them.

Flash Video MX Standard from Moyea (Windows)

I've used Moyea's Flash Video MX Standard, it is a relatively cheap tool for converting AVI videos (and other filetypes) to .FLV versions.

ffmpeg (Open source - Windows, Mac, Linux)

ffmpeg.org is the home for ffmpeg, it is one of the oldest well-supported video tools for Windows, Mac and Linux. It is command-line only (but GUIs exist) because it has a million options - if you're comfortable with command-line usage then you'll find ffmpeg fairly easy to use.

Normally you build your own version of the software from the provided source. If you hunt around you'll find pre-built binaries, ffmpeg for Windows is useful on Windows. If you'd like a GUI then take a look at AVANTI and SUPER. Whilst old on the Mac ffmpegX works well. Wikipedia's page will give you some more links.

111. http://www.ffmpeg.org/
mencoder (Open source - Windows, Mac, Linux)

mencoder is a part of mplayerhq.hu, it is the command line transcoding part of the package. It is very similar to ffmpeg, wikipedia has some background material.
MICROPHONES

Microphone choice is a very important topic. A cheap microphone (or worse - one bundled for free with your machine!) will have a very poor frequency response so your voice will be poorly recorded. It is likely that you'll sound flat, distant and hissy. More expensive mics will improve the recording of your voice (often you get a great jump in quality with only a small jump in price if you're starting out).

For some background read The Effect of Sound on Users\(^{119}\) - as noted:

> "... audio has even been shown to affect the audience perception of the quality of a presentation more than the visuals"

How do you know which mic to buy? There are so many to choose from! I went through three over two years and have now settled on a relatively expensive sE 2200A (with phantom power and a FastTrack USB interface - Windows/Mac only annoyingly). The best thing for you is to listen to how other people sound using a variety of mics, I've gathered a set of examples here.

The main differences you need to know about are analogue vs digital (USB) and dynamic vs condenser.

CHOOSING A MICROPHONE

Analogue vs Digital (USB)

Analogue mics usually connect via a 3.5mm jack plug, you might remember them from the days when every Creative sound-card came with a desk-standing mic in light cream. These always came with 3.5mm jack plugs and their quality was awful. They were dynamic mics (see below) and their long cable picked up electrical noise which then entered the computer via the 3.5mm jack plug. The computer's analogue-to-digital converter would then usually do a bad job of converting the signal (because the chips would be cheap since the hardware was bundled on the motherboard) so you'd end up with a rough, noisy digital output inside the computer.

Digital connections via USB mean that the electrical signal is converted into a digital signal inside the mic so only a clean digitalised signal is sent to the computer. No further noise gets added so you get a much better signal. You'll hear the difference in the list below (see the 'Altec Lansing 3.5mm' example).

---

Dynamic vs Condenser

Dynamic mics for our uses are typically cheaper, condensers are normally the more expensive. If you have a cheap (or free) bundled mic with your computer then it is probably a dynamic mic.

A dynamic mic works a bit like a loud speaker in reverse - current is created by the movement of a coil, the coil moves because you're blowing air onto it.

A condenser mic is based on the design of a capacitor. The diaphragm acts as one plate of the capacitor and the vibrations caused by air movement change the distances between the two plates. The capacitor is set with a charge and the change in capacitance is measured and converted into an electrical signal.

How they both work doesn't really matter, you will come across both terms though and you need to know that typically (for our use) dynamic mics can be cheap-and-nasty through to perfectly-good and condenser mics tend to be ok through to perfectly-good.

XLR

When you buy more expensive mics you'll often find that they use an XLR connector. To connect an XLR cable to the USB input on your computer you'll need an interface box like the FastTrack USB (Windows+Mac, not Linux) device. Some of the better mics have a choice of XLR or USB - go with USB if you're just going to plug the mic directly into your PC and you're not going to expand your setup into a full recording studio (this can get expensive!).

Common microphones

Below I've collected a lot of examples of mics in use in screencasts. You might also find this short thread inside the ShowMeDo Google Group to be useful.

Samson C03U, Lapel Mic, QuickCam Communicate Deluxe, Blue Snowball (USB), Logitech Headset, AT2020 (USB)

Betsy and Matt Pierce at TechSmith recorded six mics in action for their Microphone Round-up:

- Samson C03U - $210.00
- Lapel Microphone (available at TechSmith.com) - $49.95
- Logitech QuickCam Communicate Deluxe Webcam - $79.99
- Blue Snowball - $99.00
- Logitech Headset - $39.99
- Audio-Technica AT2020USB - $249.00

121. http://en.wikipedia.org/wiki/Condenser_mic#Condenser._2C_capacitor_or_electrostatic_microphone
The comment thread on the above link is great, you'll see a lot of opinions from others.

**Altec Lansing (3.5mm), Microsoft LifeChat LX-3000 (USB), Samson C0U1 (USB)**

Tim Bower gives us a lovely example of three low-cost mics for screencasting:\(^{126}\):

- **Altec Lansing** - $20
- **Microsoft LifeChat LX-3000 headset** - $25
- **Samson C0U1** - $70 - works with Linux

His demo is very clear, he uses pictures of each mic and shows the waveforms in Audacity. The use of the Altec Lansing is useful - you can really hear the audio noise that is added to the recording from both background noise and electrical noise.

### Cheap 3.5mm mic

Most of my early ShowMeDo screencasts were recorded with a £15 headset mic from Maplin (like Fry's or RadioShack). This example\(^ {127} \) is pretty representative - the audio is harsh and a bit rough. This is why you should probably avoid the cheap mics!

### SM58 (XLR, needs USB interface)

The Shure SM58 is a very well-known dynamic mic - it is one of the world's best selling and was introduced way back in 1966. It is used on stage by singers and is widely recognised as being 'a great mic'. The following example by Alan Pope on Installing Ubuntu isn't great because the audio has had a lot of compression (you can hear some odd high-pitched wavering noises, amongst others) but you'll hear that his voice is full, clear and well represented (i.e. not tinny of bassy). This example came via this thread in the Google Group\(^ {128} \).

### Plantronics 550 (USB)

Alan Pope recorded this off-the-cuff screencast on the Ubuntu Live Installer - this recording is pretty special because he did no audio clean-up - what you hear is exactly as he recorded. The Pantronics 550\(^ {129} \) works on Windows, Mac and Linux and 'just works' - if you want hassle-free recordings from a headset mic then this might be your simplest choice.

You can also hear some other examples here recorded before and after denoising\(^ {130} \) using the 550. These examples came via this discussion in the Google Group\(^ {131} \).

---

128. http://groups.google.com/group/thescreencastinghandbook/browse_thread/thread/a78f9bd91873245
130. http://popey.com/%7Ealan/plantronics
131. http://groups.google.com/group/thescreencastinghandbook/browse_thread/thread/a78f9bd91873245
Dexun Pro 58

In Gasto's screencast about popfilters\(^{132}\) (see below) he uses a Dexun Pro 58\(^{133}\).

sE2200A (XLR)

My preferred mic is now the sE2200A\(^{134}\). My OrchestrateApp tour\(^{135}\) screencast was recorded using this mic. Note that its sensitivity is very directional, if you have it upside down or backwards then the quality of the recorded sound varies dramatically - I aim to create a screencast that demonstrates this clearly.

ATM73a (mini XLR)

The ATM73a\(^{136}\) is one of my older condenser mics, you'll see a screencast example here\(^{137}\). This is a headset mic, it needs phantom power (supplied - it has a beltpack that is quite light) and is used by people who jump around on stage a lot. It wasn't so great for screencasting for me - it seemed to have a very narrow frequency response range so my voice sounds quite flat.

Mic Technique

With little experience it is easy to produce noisy, hissy, low quality audio recordings that simply sound awful. The good news is that there's a lot you can do to quickly improve the situation - and most of it has no financial cost. I'm assuming that you've chosen a reasonable USB microphone. Whilst these techniques will help if you have a low quality microphone you will see the best improvement if you have a microphone that can record the full range of your voice without introducing much extra noise.

An overview of some of the following techniques are demonstrated in this blog post: http://thescreencastinghandbook.com/blog/good-microphone-technique

Removing noise sources

Noise is a constant problem when recording. It comes in many forms - external sounds like cars, silent interference from electrical devices, bad microphone technique and even loud breathing. Generally speaking the noise can wreck your narration recording. Below I list the main sources of noise and explain how to make them go away.

It is very easy to become paranoid about various noises and the quality of your audio - remember to get a 3rd party opinion. Often you'll find that other people can't hear the problem that you've worked up about.

---

Electrical noise

Electrical noise is caused by electrical interference - generally you won't hear it when recording but you will hear it on playback. It is incredibly frustrating to get to the end of a good recording to find beeps, clicks and whistles that are louder than your voice. They're impossible to remove if they interfere with your narration and you simply have to start again.

The electrical noise generates electromagnetic waves which you won't hear (they aren't audible) but which will induce a signal into your microphone, this signal is then treated by the computer as a source of sound.

These are the main causes of electrical noise:
- Mobile phones (think of the "beep ba beeeeeeep beeeeep" that repeats for 10 seconds that you might hear if your mobile phone is left close to a speaker)
- Relay switches in fridges and freezers (these carry a big current, when the electrical relay switches over it causes a spike of electrical noise that is brief but very loud)
- Motherboard noise - this is only a problem if you use a 3.5mm audio jack rather than a USB connector (the audio jack is analogue and is subject to buzzing noises generated by variable quality components used in most computers)

The easiest way to deal with electrical noise is to turn off everything that might interfere. Turn off:
- Mobile phones
- Fridges and freezers if they're within 4 metres
- Electrical fans or other equipment that use motors or switch on intermittently

Your experience will vary depending on your recording environment, I've found mobile phones to be the biggest source of undesirable interference.

Audible noise

This is the kind of noise that you can hear - if you can hear it then so can your microphone:
- Cars driving by
- Builders and workmen crashing and banging
- Aeroplanes flying overhead
- Phones ringing
- Background chatter of people in the office
- Doors slamming shut

Depending on the microphone you might find that some of these noise sources are attenuated (attenuation means that the volume is turned down low - maybe so low that it isn't even recorded). If your microphone has good rejection for sounds behind it with good reproduction of sounds in front of it then distant background sounds like cars and aeroplanes might be rejected. Some microphones are good at recording exactly what you can hear in which case background noises are likely to be recorded as well.

Your goal should be to isolate yourself as much as possible from the outside environment so external noises aren't audible. If you can't hear it then your microphone can't hear it. It is very hard (normally impossible) to edit out the background noises that interfere with your narration so isolating yourself is the best approach.

The low cost solution is to find a quiet room at a quiet time of day. At home I've had great
success by waiting for the evening so the neighbour's builders have finished and road noise (and seagulls here in Brighton!) is minimal. In offices it is best to find a quiet room at the other end of the building, maybe early in the day or after everyone has left.

A more professional solution is to hire a recording booth. Generally these are designed to isolate you from the outside world. You might have to take your own equipment (e.g. a laptop and a microphone) or they might provide a recording engineer with microphone in which case you'll get audio files as a delivery.

The following steps will help you to remove as much audible noise from your recording environment as possible:

- Find a quiet recording location away from people at a quiet time of day
- Close all windows and doors in your room and connecting rooms
- Turn off other PCs (their fans add to the background hum)
- Turn off other electrical equipment - fridges turn on infrequently, telephones ring at the worst possible time

**Lip smack**

If you have a nice microphone the first thing you'll notice is how sensitive it is - I was shocked at first to discover how many times my lips made a "smacking" sound when I first opened my mouth to start a new sentence.

To reduce lip smack you can apply lip salve, generally you need to open your mouth slowly and pause for a moment before speaking. If you pause then you have a region of audio that can be edited to reduce the lip smack.

**Removing echo**

If you record in an environment with lots of reflective surfaces then you're likely to experience echo. Think of the echo that occurs in a church, this happens in any room that reflects sound. An easy solution is to have lots of soft furnishings - I've found that recording in a living room is easier than an office as a living room has sofas, curtains and carpet which all absorb sound rather than reflect it. Solid walls, solid floors and a lack of soft furnishings will encourage reflected sound.

Don't get paranoid about this - if you can't hear any obvious echo then you have nothing to worry about. If you can hear obvious echo then you need to adjust your recording environment, it might be the case that you are talking in the corner of a room with many reflective surfaces and if you move to the centre of the room (away from the reflective surfaces) then the problem disappears. It might also be the case that a few blankets draped over surfaces helps (but usually they don't do a great job).

Simple things to try include moving around the room and changing your direction, you have to think about how far the sound of your voice travels before it is reflected and whether those reflections are strong and come straight back at the microphone.

**Avoiding high-energy "esses" (generally known as sibilants)**

Sibilants are strong "ess" sounds you get when you say a word with "s" in it. Try this phrase: "sammy the slippery snake slide down the slope". For some people (including me) the "esses" are very strong - in an audio recorder you can see the "s" parts of the narration
as they have a loud volume.

This is particularly a problem if the microphone's most sensitive part is directly in the path of the sound that comes out of your mouth. A simple way to alleviate this problem is to position the microphone elsewhere. For me the microphone works well if it is positioned above my head, pointing down towards my nose (a few inches above my mouth). The microphone is sensitive enough to record everything I say very clearly but the high energy "ess" sound isn't directed straight at the microphone, so it isn't recorded so loudly.

If sibilants are a particular problem for you then remember that in a good audio editor (e.g. Audacity) you can reduce the volume just for the loud part of the sibilant. If you reduce it in steps of a few decibels at a time and check that the surrounding audio still sounds natural (big changes will result in loud-quiet-loud transitions in your audio that sound really odd) then you'll be able to manually attenuate the annoying sibilants fairly quickly. This is a manual, time consuming step. It is better to adjust the microphone in a way that makes it less sensitive to your sibilants in the first place.

**Avoiding booms and bangs (generally known as plosives) and using a pop-filters**

Plosives are the loud wind-rushing sounds you get if you say "boom, pah, bop, bang". You mouth sounds out a rush of air and when this hits the microphone, it records a noise that's like the sound of the blowing wind.

There are three ways to reduce the sound of plosives if they're a problem for you. First - you can modify how you say plosive-laden words. Rather than taking a big breath and booming the word out, instead you could choose to say it more softly.

Second - you could turn your head away from the microphone by a few degrees each time you say a plosive-laden word, this way the main wind that you generate won't rush over the microphone.

Both of the above techniques require practice. A third solution is to use a pop filter. A pop-filter can be built for $10, it slows down the blast of wind so the air doesn't hit the microphone - this helps if your breathing (both inhaling and exhaling) is being recorded. For an example see this screencast on pop-filters.

A pop filter is just a fine mesh of material stretched around a frame and placed between you and the microphone. Personally I've found that a pop filter makes very little difference but some people swear by them. Experiment if plosives are a problem for you.

**Avoid breathing into the microphone**

Breaths - which are normally silent to us as we're so used to ignoring them in other people - tend to be recorded clearly with a good microphone. This is a problem because you don't want your narration to be punctuated by deep breaths all the time.

Your microphone may not be very sensitive to breathes so this may be less of an issue for you.

---

139. http://www.youtube.com/watch?v=k4GHNPaw4qc
If breathes are recorded clearly then there are a couple of solutions:

- Breathe softly - don't take huge gulps of air but take more frequent, shallower breaths
- Don't breathe out directly at the microphone - angle your lips or turn your head so the flow of air is away from the sensitive part of the microphone

**Gathering your thoughts and smiling**

Always remember - you're not in a rush and you can take your time. If you rush then your recording will sound hurried. If you remember to slow down then you can also remember to smile - your voice will sound warmer and more friendly. Obviously a hurried voice isn't a source of noise - but I figure it is worth reminding you of this point as it will help you make recordings that you're proud of!
EXTRA HARDWARE

MOUSE VS TRACKBALL?

Generally I prefer to use a mouse in my dominant hand when recording screencasts. I've learned over the last 10 years to be ambidextrous thanks to some rather debilitating RSI back around the late 1990s.

Typically I use a trackpad (the Logitech Marble Mouse - great to help RSI!) in either hand but I'm not so great at fluid movements - when recording I find that using a mouse in my right hand gives me the smoothest mouse movement.

WEBCAM

A high quality webcam is great for recording the face of the presenter. They generally work well if you're passionate about your presentation and they can make static presentations more engaging. They're also very useful for meetings where you want to record as much context as possible.

Personally I've used a Logitech Webcam Pro 9000, it can record at a high resolution with a good framerate and the drivers are good at adjusting the image so it looks good regardless of lighting conditions. The webcam in my MacBook is also very good. Old or cheap webcams are likely to give poor framerates and washed out colours, do test your webcam to see how you look.

Demo videos

The webcam can be a great way to show your viewers that you are the author and that you're passionate about your product. On The Screencasting Handbook's blog I have a friendly Critique of Gibraltar Software's Tour Screencast - Jay Cincotta uses the webcam at the start and end of his demo video during a set of presentation slides. During the body of the screencast when he shows working software he removes the webcam image.

This is a great way to show your face and keep the presentation sections interesting. Below you can see a part of Jay's opening:

The difficulty is that if you're recording your face then you have difficulty with later edits - every cut becomes obvious (in fact you can see a few in Jay's video). In particular this means that taking a breath and gathering your thoughts is tricky - you have to keep your head still so the cut isn't a problem.

Generally I'd recommend using the webcam only if you're really passionate and comfortable with being on camera. There's no point using the webcam to record someone else - it'll just come across as though they're a hired speaker.

Remember to position your camera at head height so you're not looming down into the camera and pay attention to the lighting. You want even, natural light so your face is well lit. In the screenshot above you can easily see Jay's face and the minor shadow isn't a problem. If you have strong side lighting (e.g. if you're near a window or recording at night with a desk lamp) then you'll have half your face in shadow - this will look very odd.

Remember also that natural lighting conditions will change during the day - if it is a cloudy day then you might get sunny periods which brighten the room momentarily. This tends to look really odd, so record when the conditions are stable or use a good set of natural indoor lights.

Meetings

As mentioned at the start in "What's the value of screencasting?" a screencasting tool is a great way to record a meeting with a client. The web cam gives context to the audio stream and the on-screen activity, just make sure that your webcam is pointing at towards you and your client rather than at the wall.

An hour-long meeting is easily recorded and exported as a .MP4, when you scan through the video later you might find that the webcam gives you the necessary visual information to help you locate what you're looking for.

Copy holder

At one point I tried using a copy-holder (this Fellowes one) to hold my script while I dictated. Generally I record my audio separately from the video and having my script held in line with the mic seemed like a good idea.
Annoyingly my mic (an sE2200A) is too good - it picks up the reflected sound from the copy-holder and you get a light echo on the recording. I've since found that holding the script mostly-flat (so any of my voice that hits it bounces away from the mic) is just as easy as having a copy-holder and it doesn't introduce any echo. Also the Fellowes was a bit rubbish - it looked solid but it was lightweight plastic that rocked and felt flimsy!

**HOW SCREENCASING WORKS**

Broadly speaking there are two ways to record a screencast, you either capture screenshots of what the viewer is shown or you call into the Operating System's interface and extract low-level representations of what is being displayed.

**Capturing a screencast**

Capturing screenshots is the easiest and generally the worst technique. To capture a screenshot you're taking a copy of what's displayed on screen, compressing it as a single image and then storing it. This is a slow process so you generally get a very low framerate. The slowest part of the process is typically asking the Operating System for a screenshot so it doesn't matter if you're not compressing the screenshot or if you're storing it in RAM (which is fast compared to writing it to a disk).

The Operating System will have a number of programming interfaces, generally there are high-level interface which are easy to use through to low-level interfaces where you need a lot of background knowledge to understand what's going on. To capture low-level graphic events quickly you need to burrow into these programming interfaces.

The process is complicated with each update to an Operating System - more programming interfaces are created and more techniques are required to efficiently capture all the screen updates so the recording is an accurate representation of the screen.

A final complication will arise if you want to record live video streams. Often the Operating System will delegate parts of the video rendering process to the graphics card, so the graphics card inserts elements of the video before the video is output to your monitor. This is known as a [video overlay](http://en.wikipedia.org/wiki/Video_overlay). This means that your Operating System doesn't know what's inside your video player - if your recording techniques aren't sophisticated enough then you'll get a black box in place of a movie player's contents!

Generally most screencasting applications aren't very good at capturing these very low level processes - on Windows FRAPS is the best known, it will capture all the levels of output with a very good framerate.

**Storing the screencast**

Having capture the video stream the application will then need to store it. Years ago this was a difficult process - CPUs were slow so compressing the video stream was hard but disks were also slow - so streaming a large uncompressed volume of data to the disk was also hard. Often this resulted in the final video having missed frames.

As CPUs and disks became faster the process became easier, now it is common for a

---

screencasting program to use a lossless format to store everything in a file without any loss. Since compression can occur during recording it means that the files on disk aren't too huge - once upon a time a 10 minute screencast might have taken gigabytes of storage, now it might take hundreds of megabytes. This means that the entire video can be stored in memory during the editing process, so editing is faster and more convenient.

**OTHER RESOURCES**

**THE SCREENCASTING HANDBOOK’S GOOGLE GROUP**

Our [Google Group](http://groups.google.com/group/thescreencastinghandbook) is for readers of this Handbook, it is a place where you can give me feedback and we can ask screencasting questions and seek answers amongst our peers.

I'd be very happy to offer friendly critiques of public screencasts if you have one to share.

**PROCASTS TUTORIALS AND CRITIQUES**

My main business in ProCasts, there we create professional screencasts that explain our clients' software to their website visitors. As I built our reputation I created a 9-part [screencast tutorial](http://blog.procasts.co.uk/2009/03/why-screencast-if-a-picture-is-worth-a-1000-words/) series. You'll find that various links are used here in the Handbook but you'll probably find extra information there, do take a look.

I also have a set of five [screencast critiques](http://blog.procasts.co.uk/category/screencast-critique/) where I explain problems I've seen and ideas for addressing them. If you're new to screencasting you'll probably find lots of useful ideas (along with feedback for some from the original authors) that you can integrate into your own productions.

**SHOWMEO.com**

Back in December 2005 I co-founded [ShowMeDo.com](http://showmedo.com) with Kyran Dale. We both had academic backgrounds and had spotted the opportunity for teaching via screencasts - the site we wanted didn't exist so we scratched our own itch and built our learning community.

Today 50-100,000 people each month visit the site to watch over 1,000 open-source tutorial screencasts that have been created by over 100 authors. Some of the examples used in this book have come from ShowMeDo (notably by Horst, Gasto and Dai).

If you're interested in sharing your knowledge to teach others about open-source topics then do take a look at ShowMeDo. It has a very friendly community and lots of visitors, topics like Python, OpenOffice, GIMP, Ruby, Perl and more are covered.

**PEEPCODE.COM**

Geoffrey Grosenbach created [PeepCode.com](http://peepcode.com) back in 2006 to teach the world about Ruby programming, since then he's expanded the scope to cover Javascript, Unix and more and has experimented with pdf guides too.

143. http://groups.google.com/group/thescreencastinghandbook
Some of the examples are free and they have a very high production quality, if you're curious to see what's possible with straight screencast tuition then Geoffrey's examples are a good source of inspiration.

**Railscasts.com**

Ryan Bates created Railcasts in 2007 (a while after Peepcode) to show short Ruby and Rails tips in screencasts, he gained some publicity simply because his opening video sequence was loud and fun.

**Scrast.net**

John Basille created Scrast.net to list screencasting news. The industry is still growing so there isn't a huge amount of news but John's coverage tends to carry useful detail.

**ScreenCastsOnline**

Don McAllister established ScreenCastsOnline to teach Mac users how to use all the various tools that were available. He grew his site until it took over as his full-time occupation, he can be seen at various Mac conferences too. He details his setup and workflow on his site. Earlier in 2009 I interviewed him over at ProCasts.

**Lynda.com**

Lynda.com (http://en.wikipedia.org/wiki/Lynda.com) is the oldest running screencast-based tutorial site. The screencasts are mostly about commercial software (it is sort-of opposite to ShowMeDo in that respect) and are recorded in a studio. Topics covered include Photoshop, MS Office, AfterEffects and Flash.

**JQueryForDesigners.com**

Remy Sharp (a Brighton local and fellow pub attender) shares free JQuery tutorial screencasts on JQueryForDesigners.com. He's built a loyal, large audience who love his relaxed presentation style and authoritative knowledge. One of his posts explains how he screencasts on Mac.

146. http://scrast.net/
Books

"Camtasia Studio 6: The Definitive Guide" by Daniel Park is published by Wordware Publishing, it is 600 pages and is packed with information. I have a copy here, Daniel has written a fine manual for Camtasia Studio, he’s been covering Camtasia Studio since at least version 3.

"Screen Recording - A Practical Guide with Case Studies" by Stefan Schnabl is self-published through Lulu as a softback book. I’ve not read it and Amazon doesn’t have reviews, it was first published in February 2010. Note that this book is sold through Lulu.com where you can find a free PDF version.

**Tell others about The Screencasting Handbook**

I have almost no marketing budget for this book and I'm spending all of my time writing it, researching your needs and delivering the answers that will teach you new skills. In return I'd absolutely love publicity from you - a simple tweet will help me to spread the word, a blog review of the chapters and the community would be wonderfully useful!

Tell me about any review or forum mention and I'll make a note of it on the blog.

I'm happy to provide quotes if you want to write a blog entry, just get in contact.

**Professional Screencasting and Demo Videos with ProCasts.co.uk**

Do you need professional help with screencasting? I'm writing this Handbook because of my experience building ProCasts and earlier experience co-founding ShowMeDo, for both I've had people ask me to help them create their own videos for sales, marketing, tuition and support.

In ProCasts we create professional training, tutorial, sales and support screencasts and demo videos. We help all companies (small or large) and we can both help you create your own screencasts or take on the full job ourselves. Take a look at the ProCasts Examples and then get in contact (Contact@ProCasts.co.uk).

**Acknowledgements**

Thanks to Colleen McGuire, Horst Jens, Ghislain Fabre, Fred Grover, Dai Butt, Jerol Harrington, Tim Bower, Bob Walsh, Gasto (Gabriel Hasbun), Alan Pope, Phil Shapiro, Silvio Grosso and John S. Richards for valuable feedback, links and resources. Also I thank Andy White for acting as my accountability buddy, he's working on his Podcasting Unleashed book as I'm writing this book.

Thanks also to Cafe Delice in Brighton's North Laines for opening at 8.30am and serving excellent coffee and pastries whilst I worked on this book.

If you want more detail on my book-writing process see my blog entry: How I'm writing The Screencasting Handbook (November 2009).

**Do you have the latest version of The Screencasting Handbook?**

To see if you have the latest version of this Handbook visit http://thescreencastinghandbook.com/ and compare it to the version number you'll find in this Handbook on page 2.

If you don't have the latest copy then you're probably not on the Updates mailing list. You

156. http://procasts.co.uk/
should have been added to the Updates mailing list when you bought the book but if you weren't (we might not have mailed you or maybe you didn't receive the invitation - sorry!) you need to get hold of me. Contact me on ian@thescreencastinghandbook.com.