

Talking Newspaper Online Services Guide

Author: Gavin Smalley – Witney Talking News

Versioning

1.0

13th June 2020

Gavin Smalley

First Release

Contents

Versioning.....	1
Introduction.....	3
Skills Required.....	3
Background.....	5
Copyright.....	5
Working Backwards.....	6
Goals.....	6
Pathways & Costs.....	7
Instructions.....	8
Personal Support.....	8
Credits.....	8
Further development.....	9
Conclusion.....	9
Annex 1: Voice Assistant and Podcast Guide.....	10
Introduction.....	10
Requirements.....	10
General.....	10
Google Assistant.....	10
Apple Siri.....	11
Amazon Alexa.....	12
Options.....	13
Instructions.....	14
General.....	14
Dedicated Podcast Providers.....	14
WordPress Feed Adaptations.....	15
Code.....	15
Validation.....	16
Conclusion.....	16
Annex 2: Website and Integrations.....	17
Introduction.....	17
Website.....	17
Requirements and Instructions.....	17
Further Details.....	19
WordPress Themes.....	19
WordPress Plugins.....	20
Site Content.....	22
Ongoing Process.....	23
Integration and Automation.....	24
Introduction.....	24
What does it actually do?.....	24
Assumptions.....	25
Requirements.....	26
Instructions.....	27
Setup.....	27
Ongoing Operations.....	30
BWBF Sonata+ & Talking Newspaper App.....	31
Conclusion.....	31
Annex 3: Telephone Dial-In Service.....	32
Introduction.....	32
Options.....	32
Architecture.....	32
Instructions.....	33
Phone Number.....	33
VPS (Virtual Private Server).....	33
PBX (Switchboard).....	34
VOIP Trunk.....	36
Conclusion.....	36

Introduction

This guide provides a basic framework of instructions on how to deliver TN content on the web, via voice assistants or smart speakers (such as Amazon Alexa, Google Assistant and Apple Siri) and over the telephone in-house within a constrained budget.

Firstly let it be said, right at the beginning, that a certain level of technical expertise is required to achieve this “in-house” and to do so in a way that is secure. More details of the skills required can be found in the relevant section below. If there is no way of securing such skills then there are commercial services available (albeit for a fee) and in the long run it may be better to engage those if sufficient funds are available. It is often difficult as a small charity to commit to relatively large sums of money, when a cheaper alternative is available but it must be remembered that there is more to “good value” than the bottom line – if you are not confident about being able to deliver or maintain a service in-house then it may be better value to buy it in, even for a greater outlay as what matters at the end of the day is that listeners get the content reliably.

I think it also needs to be made quite clear here that the purpose of this document is not to denigrate in any way the value or importance of the commercial services available in this field; they have their place as much as the in-house approach does.

All that being said, please don't let it discourage you from trying! There is no intention to create a sense of “gate-keeping” but equally do not feel you have failed if you decide the best way is to have someone else do it. Every TN is different and there is no one-size-fits-all solution. Indeed several hybrid solutions are discussed in this document which may provide a way forward.

This guide describes six pathways. They vary in cost, complexity and in what they deliver. Pathway 1 would be the recommended route for a TN to place content on to voice assistants where there are no skilled IT workers available. It isn't the cheapest option (~£80 per annum) but it is by far the easiest. Pathway 6 is the most complex option requiring a relatively broad range of IT and programming skills. The remainder fall in-between.

Please note: All costings in this document are believed to be correct at the time of going to press.

Skills Required

The skills required, whilst not perhaps always as deep and complex as might be imagined, will vary according to the pathway followed. They are broken down below:

- For any pathway
 - The ability to create a single MP3 file of each edition
 - The ability to use a webpage to upload a file

- A basic understanding of what a podcast is
- For the voice assistants (Alexa etc)
 - For pathway 1 (the simplest way of setting this up) you need nothing but the above.
 - For the more complex version hosted through your own website then everything in the website section below.
- For creating and hosting your website
 - Initially you will need to use cPanel (or whatever alternative your host of choice offers) to set up WordPress. Your web host may be able to do this for you
 - You will also need to use the same system to set up security certificates (Lets Encrypt is the simplest as it is needed only once). Again your host may be able to assist you
 - A basic understanding of uploading files, typing in text boxes and desire to frequently check the updates page and apply them in good time.
 - To enable email delivery you will need a basic understanding of mailman (or a suitable WordPress plugin). This can be a tricky area as you need to ensure that you can send emails to those who subscribe but also create restrictions such that others cannot send to the same lists.
 - If you also wish to host your own playlists or your own upload service you will also need the skills listed under the automation section.
- For the telephone dial in service (the most complex area to deliver):
 - Operating System installation on a remote computer.
 - A basic understanding of the Linux command line (updates, directory handling, permissions, symlinks).
 - A basic understanding of a modern PBX and call flows.
- For automating the interactions:
 - Basic knowledge of PHP. The code for the initial use case (see background section) is provided but it may need to be adapted to individual TN requirements
 - Basic shell scripting (for the same reasons as above)

That covers the bare minimum for each service. For those more highly skilled there are more options, albeit outside the scope of this guide.

Even if there is no-one immediately known in your TN with these skills, it may be worth asking volunteers – who knows what they, their partners, adult children, friends or neighbours may do for a living! Alternatively it may be worth approaching a local college –

creating your website could become a student's A-level project. Or maybe a local company who specialise in web design – they may be prepared to work pro-bono or for a reduced fee in return for a credit on the page.

The telephony solution has fewer options. Sensibly there are two choices: go it alone (which requires a modest skill level) or use the one commercial provider available.

Background

This document began life in a discussion on the TNTALK mailing list around costs of digital content delivery with a passing reference to the fact that I had delivered a solution for the Witney Talking News that provided playlists, web access, email delivery, a podcast and Alexa access for the sum of £7.20 per annum. Subsequently increased to £95 per annum to include the telephone access system.

The subject has come up several times since and each time it does I have been asked how it was done. This document is an attempt to set out what is required to achieve a similar result for a similar budget.

It is still quite possible to deliver all but the telephone solution for under £10/yr with a relatively modest skill set. It just takes time, imagination and a little compromise.

The compromise is important to note. The favoured commercial solution offers a service through Alexa which allows people to say “*Alexa, play my Talking Newspaper*” and then allows them to skip forwards and backwards through the tracks as if they were using a traditional MP3 playback unit. This solution doesn't work quite like that. The invocation is different and the user may only skip forward and back a few seconds at a time, not in whole tracks. That said this solution also works with Google and Apple too. The same applies to the telephone service described in this document – it is not a direct replacement for the commercial offering, it has the same restriction in that it plays one single track per edition and skipping is only a number of seconds forwards/backwards not by story.

Finally, I am not, nor have I ever been, a professional web developer and I claim no formal expertise and accept no liability for any loss or damage to any person or organisation through caused by any content (or omission thereof) in this document. What I have done is simply document my own journey in a way that attempts to help others set off on their own.

Copyright

Before proceeding I think it is also worth mentioning the copyright implications of following the rest of this guide.

Please note I am not a legal professional and what follows is not legal advice, just points to consider.

Most TNs have specific agreements with the papers they read from that they may make audio copies for closed distribution. There are also some copyright exemptions that TNs may claim for the closed distribution of certain other works.

Whilst it is possible (though not necessarily desirable) to password protect both website content and podcasts, it is not possible to play such podcasts through any of the voice assistants. For telephone delivery it is also possible to protect it with a PIN (or even by recognising specific callers' phone numbers) but this is extra work and extra administration and so not necessarily desirable.

The result of all of that is that as soon as you implement the goals of this guide your content becomes publicly available and those agreements and exemptions may no longer apply. Before any of this starts then it is a good idea to write to or email each of the publications used to ensure they are happy for their content to be freely available on the internet in audio format. If in doubt seek formal advice from TNF or appropriate legal counsel.

Working Backwards

Usually in describing a how to achieve something one would start at the beginning but this document has been built the other way around. It has been done this way because everyone is working to the same goal but from different places. It is impossible to document every possible starting point and so I have taken the most likely desirable outcomes and worked backwards from each to explain what needs to be in place for each to work. That way there is no requirement for a specific starting point.

This may seem counter-intuitive, but I have found in many years of professional problem solving that the easiest way to derive a path to the solution is to start from the desired solution. Once a clear goal is in place it is often much easier to see a path back towards the current position.

For this guide then we start by considering each desired goal and its requirements.

Goals

This document addresses the following goals:

- Delivery of content via voice assistants (Alexa, Google Home, Siri)
 - I have listed these all as one goal, because once the work has been done for one the others are each most of the way there. However you may, of course, pick which you choose to support.
- Delivery of content via a podcast
 - This is a pre-requisite for the above

- Delivery of content via the web (and optionally email)
- Delivery of content via the telephone

It is not necessary to pursue all of these. You are, for the most part, free to pick and choose which you want to follow. However some are dependant on each other and this will be explained in due course.

The first step on this journey is to decide which of these goals are required. This is an important step and thorough consideration should be given to it as getting this right at the start, and picking the correct pathway below, can save a lot of repeated effort later.

Pathways & Costs

Depending which combination of goals you choose there are a number of possible ways forward. Each pathway has its own merits, compromises and costs. I've listed what I believe are the most likely scenarios here:

1. Only need voice assistants
 - The easiest way to achieve this is to follow annex 1 using a hosted service. This requires a manual upload for each publication but supports all three voice assistants with very little technical knowledge required. That comes at a cost though (~£80 per annum using Audioboom).
2. Need both a website and voice assistants
 - Follow annex 1 and the first half of annex 2 using the WordPress solution for both. This is the cheapest option (~£7.20 per annum) but requires a little more skill than the option above. Also requires someone to create a post each week.
3. As above but fully integrated
 - As above but also follow the integration and upload steps in the second half of annex 2. No cost increase and slightly easier to manage (as post creation is automatic) but a little more complex to set up.
4. Only need the phone service
 - This is (or at least can be) a standalone solution, simply follow the instructions in annex 3. Whatever you do this will cost a minimum of £95 per annum.
5. Require voice assistants and telephone but with neither integrated
 - The obvious way to set this up is to follow annex 1 using a hosted service and then annex 3. This is the most costly option at ~£175 per annum
6. Require the phone service integrated with a website

- Begin by following annexes 1 and 2 until you have both working and then follow annex 3.
- Then all that is required is to automate the transfer of the audio to the PBX machine from your website host and a script to update the symlink(s) to make it play the current file. This syncing is left as an exercise for the reader, but scp or rsync over ssh and then a small shell script would appear to be the way forward.

Instructions

Specific instructions for each goal may be found in the respective annexes.

Throughout the document I have deliberately avoided providing detailed step-by-step instructions. Primarily because if there is a need for such involved detail there has to be consideration given to whether the skills detailed above have been met, but also because exactly what to click when is likely to change over time. Instead I have outlined what needs to be achieved at each step.

Personal Support

As much as I would like to offer my personal support to every TN in the country there really isn't time in the all the world for me to do that on an ongoing basis and neither can I offer training in any of the skills required.

For TNs with specific needs or with very limited funds where in-house delivery is the only way forward I am willing to consider requests for more involved help. This would, of course, be on a best efforts basis.

I am, though, more than happy to answer specific questions from any TN embarking (or considering embarking) on this or a similar project on the basis that any such answers are provided as-is with no guarantee. Also, it is worth repeating here, that I am not, nor have I ever been, a professional web developer and I claim no formal expertise and accept no liability for any loss or damage to any person or organisation through caused by any content (or omission thereof) in this document.

I can be contacted either via the Witney Talking News website (linked in the footer) or on the "Issues" page of the GitHub repository that contains this guide (<https://github.com/yknivag/TN-Online-Service/issues>).

Credits

Alongside the author, credit must be given to Mr Paul Ramsden from TNF for his encouragement to begin and complete this guide and to Mr David Hancock from Crewe and Nantwich Talking Newspaper for providing vital information relating to Google

Assistant, the text-to-speech engine mentioned in the website section and for reviewing the draft.

I'd also like to thank Witney Talking News, the TN for which I volunteer, for their patience and support whilst I created the project on which this guide is built.

Further development

These items are points worth considering but outside the scope of this guide.

- On-line Donations
 - Many TNs are reliant, at least in part, on donations from listeners to fund their services. Postal listeners may donate easily through the pouches they return each week but online listeners do not have that regular free post contact. You may wish, therefore, to provide a way for them to make donations through your website. This may be done directly on your website (see annex 2) or by registering for the PayPal Giving Fund and providing a link to that page.
- Google My Business
 - Good performance in search results is absolutely vital to keep people coming to your website and in facilitating people finding your website. One of the biggest things you can do to make this happen is to claim your "Google My Business" page and update it with as much information as you can. Specifics are out-of-scope for this guide but there is some guidance available through the help settings in the "Google My Business" portal.

Conclusion

There are many routes to deliver any project, many balances between monetary cost, time invested and skills required. Hopefully this guide has provided you with a way to assess the best way forward for your organisation in terms of your non-postal delivery journey as you weigh the three sided balance of time, knowledge and money.

Annex 1: Voice Assistant and Podcast Guide

Introduction

Voice Assistants (such as Amazon Alexa, Google Assistant and Apple Siri) are a revolution in technological services for VIPs. Whilst it is true that they alienate as many people as they help (particularly those who are hard of hearing or have difficulties with clear speech) for most people living with sight loss they are fantastic tool. They are readily affordable and easy to interact with, requiring very little learning of commands which makes them accessible to young and old alike.

One often mentioned barrier to take up for these services is the lack of an internet connection. It should be remembered that Alexa and Google Assistant are both available on Android smartphones and that Siri is available on iPhones. All of these will work without a home internet connection through the data plan and several networks now offer generous data plans for modest prices.

That said, it should be expected that take up will be relatively slow as people are often reluctant to try new technologies, however enabling, and also due to substantial privacy concerns. It is for this reason that care should be taken not to overspend on this, at least until demand is proven.

Requirements

General

You may be wondering why voice assistants and podcasts have been placed in one section. Essentially it is because offering your service via all three voice assistants is as simple as pointing them at your podcast feed. The only trick is knowing what the specific podcast requirements are for each assistant.

Whilst the requirements for each voice assistant are all different they are all tolerant of the quirks of the others, and so it is possible to create a single feed which works with all three by following all the sets of requirements at once.

The instructions in this section, therefore, refer to how to set up a podcast and how to ensure it meets the relative requirements.

Google Assistant

This service is also variously known as Google Home, Google Nest, Google HomePod and “OK Google”. It works through a variety of dedicated devices and on all current Android phones and tablets.

Google Assistant will play any podcast listed in the Google Podcasts directory.

For a podcast to be listed in that directory it must meet the following criteria:

- It must be valid (see validation below).
- It must contain a link to your webpage.
- That webpage must contain a hidden link to your podcast feed in the format below.
`<link type="application/rss+xml" rel="alternate" href="RSS_URL" />`
That link must be the only one on the page that has the `rel="alternate"` attribute.
- Once your podcast feed and webpage are set up like that, then you can either wait for Google to index your feed or you can use <https://pubsubhubbub.appspot.com/publish> to “Publish” your feed (or “Topic URL” as Google calls it). Note that nothing will happen when you press the “Publish” button, this is normal and due to the odd way in which the form is designed.

Once your podcast is listed in the Google Podcasts directory, you may play it from any Google Assistant device by saying “OK, Google. Play <name of podcast> podcast”.

Also once your podcast is listed it will be available in the Google Podcasts smartphone app and may show up against your listing in Google search results too.

Some limited listening stats may be obtained through the Google Podcast Manager.

Apple Siri

The story is much the same for Siri, although it is iTunes podcasts that she is looking at and so the goal becomes to get your podcast in the iTunes directory.

This is rather more complex than the Google version as Apple has a number of very specific requirements for the construction of the podcast feed. They have some specific tags they require which no-one else uses and also a requirement for a very large image to be linked in the feed.

The requirements are as follows:

- The feed must validate (see validation below) after all the extras below are added
- Inside the `<rss>` tag at the start you must add the following to define the `itunes` namespace:
`xmlns:itunes="http://www.itunes.com/dtds/podcast-1.0.dtd"`
- After the `<channel>` tag you must add the following:
`<itunes:category text="News"/>`
`<itunes:subtitle>Your sub-title here</itunes:subtitle>`
`<itunes:summary>Summary of your feed</itunes:summary>`
`<itunes:author>TN Name</itunes:author>`
`<itunes:owner>`
 `<itunes:name>TN Name</itunes:name>`
 `<itunes:email>TN Email</itunes:email>`
`</itunes:owner>`

```
<itunes:explicit>no</itunes:explicit>  
<itunes:image href="https://link.to/LargeLogo.png" />  
<copyright>https://link.to/podcast-copyright-statement/</copyright>
```

You must, of course update the entries for your circumstances.

The large logo must be at least 1400x1400 pixels square. Note that whilst it should be large in physical size the file still needs to be as small as possible (to facilitate rapid loading) and so a good level of compression is needed.

- Within each `<item></item>` tag pair you must include:
`<itunes:explicit>no</itunes:explicit>`

Details of how to achieve all this are in the instructions below.

Once you have met the conditions, you must then create an iCloud account, download iTunes and verify that account in the iTunes store. Then you may use <https://podcastsconnect.apple.com/> to notify Apple of your podcast. After a few days you should receive an email advising that your podcast has been approved and is available in iTunes.

It is worth mentioning here that you may also use PodcastsConnect to verify that your podcast meets Apple's standard.

Finally once you have been approved and listed in iTunes listeners may hear it by saying "Siri, play *<name of podcast> podcast*".

Some listening stats may be obtained going forward through the PodcastsConnect page.

Amazon Alexa

Alexa is the most logistically difficult to deliver content to as it doesn't maintain its own directory or index of podcasts. Instead it plays podcasts from any number of third party directories, each of which have their advantages and disadvantages. More than one may be offered, but bear in mind that offering too many may generate a bewildering array of options which may confuse more potential listeners than it helps.

When I started this journey the service I chose was "AnyPod". It is still the service used by Witney Talking News but it is not currently accepting new registrations and so an alternative must be sought.

"TuneIn" is an obvious choice as they are free, but in my experience their service is less than reliable and the support worse. Spotify is a reliable option (and also free) but requires the listener to have a Spotify account which isn't always free and is an extra level of setup for them.

A newcomer to this field is Pocket Casts – I haven't tested the service though, so use at your own risk. However it appears to be very similar to AnyPod.

All that said, there is nothing special required for Alexa, she will happily play any podcast so long as the feed is valid (see validation below).

All you do is submit the link to your podcast to your chosen compatible directory and it will be available on Alexa usually within a couple of days. The invocation though will be dependant on the directory you choose. For example:

- For AnyPod, you'd say *"Alexa, ask AnyPod to play <name of podcast>"*.
- For Spotify you'd say *"Alexa, ask Spotify to play <name of podcast>"*.
- For TuneIn: *"Alexa, ask TuneIn to play <name of podcast>"*.
- And for Pocket Casts: *"Alexa, ask Pocket Casts to play <name of podcast>"*.

The same goes for other services.

Options

To deliver this the need boils down to a podcast which is constructed in a specific way and subscribed to a specific set of directories/indexes.

There are then a number of options as to how this may be achieved and which one you follow depends only on where you start from (as they all lead to the same end result).

Those options are as follows:

- Use a podcasting service (pathway 1 or 5):
 - This is the easiest way to get started as there is little to do.
 - Most of them support automatic submission to the appropriate directories to allow playing through all the smart services.
 - They will always generate a feed which is compliant with all services.
 - They offer a way to host the audio too.
 - They are relatively affordable.
 - The downside is that they require payment.
- Deliver it through your WordPress website (pathway 2, 3 or 6):
 - If you already have a WordPress website then this is the logical way forward.
 - You need to decide if you are going to host the audio yourself or somewhere remotely. This will depend on what is allowed by your hosting provider (see annex 2 for more details).
- Create a way yourself of delivering it:
 - If you are not using WordPress and you don't want to use a podcasting service then you will need to:
 - Host the audio.

- Create the XML with a script.
- The “how” for this option is out-of-scope for this guide.

Instructions

General

There are three ways to proceed:

- With a dedicated provider (pathway 1 or 5):
 - It requires very little technical knowledge to set up and maintain.
 - However there is a running cost and a requirement to manually add each publication.
- Using your WordPress website (pathway 2, 3 or 6):
 - There is no added cost here above the cost of the website, however there is some level of skill required.
 - A new post will need to be added manually to your website for each publication unless you complete the integration steps in annex 2.
 - For the completely integrated solution you will need to avoid a podcast plug-in (or provide your own updates to stage 2 of the integration code).
- Using an integration to your own site that you write yourself.
 - This is outside the scope of this guide but for a web developer it ought not to be a taxing challenge.

Dedicated Podcast Providers

There are many companies that provide this service. Some better than others. Most of them support all the voice assistants. The rule of thumb appears to be that if it supports Apple then a service will likely support all three. Apple maintain a list of “approved” suppliers at <https://itunespartner.apple.com/en/podcasts/partnersearch>

Whilst I haven’t used any of them and so cannot make a formal recommendation, Audioboom and Blubrry appear to be good contenders. The former offering a plan which would suit almost any TN for just \$99US (~£80) per annum and very easy to set up.

With these providers each time you issue a new publication then you use their website to upload a single MP3 file containing the entire audio, add some descriptive text, press a button and that’s it. Nothing technical beyond creating the single MP3 and uploading it to a website.

WordPress Feed Adaptations

If you are using WordPress for your website (as is the recommendation in the website section) and if you create a post for each edition then if you apply a specific category to those posts then you may create a feed from that category.

This will give you a podcast with very little extra work. The exact URL will depend on your WordPress settings but it will be something like

https://attn.org.uk/category/<category_name>/feed/

That feed should work with Alexa as is (once added to an appropriate directory). It may work with Google too, but that depends on your WordPress theme - view the source code of your home page and check for `<link>` tags containing `rel="alternate"`. If there is only one and it is your feed that's great, otherwise you will need to edit your theme to make it so. Specific instructions for that are not possible here as there are too many themes available – it may be necessary to discuss with the author of theme if you can't find the appropriate part to edit.

For Apple's Siri there are some more changes required. These are two ways of doing this; one is to use a podcast plugin and the other is to add the code below to your theme's `functions.php` file.

Using a plugin will change the feed URL so it is best to settle on one before you register your feed with the directories. Changing the registered feed isn't always easy.

Adding code to the `functions.php` file can be done in the WordPress "Theme Editor" but note that any update to the theme thereafter will wipe out the changes so it is worth considering creating a child theme which will hold just your changes and preserve them.

Code

This should be added to the end of your theme's `functions.php` file:

```
/* Change RSS feed for iTunes compatibility */

function attn_rss_feed_ns_changes() {
    echo 'xmlns:itunes="http://www.itunes.com/dtds/podcast-1.0.dtd"';
}
add_action( 'rss2_ns', 'attn_rss_feed_ns_changes', 10, 0 );

function attn_rss_feed_changes() {
    echo "<itunes:category text=\"News\"/>\n";
    echo "<itunes:subtitle>" . get_bloginfo( "description" ) . "</itunes:subtitle>\n";
    echo "<itunes:summary>" . get_bloginfo( "description" ) . "</itunes:summary>\n";
    echo "<itunes:author>" . get_bloginfo( "name" ) . "</itunes:author>\n";
    echo "<itunes:owner>\n";
    echo "<itunes:name>" . get_bloginfo( "name" ) . "</itunes:name>\n";
    echo "<itunes:email>" . get_bloginfo( "admin_email" ) . "</itunes:email>\n";
    echo "</itunes:owner>\n";
    echo "<itunes:explicit>no</itunes:explicit>\n";
    echo "<itunes:image href=\"https://link.to/LargeLogo.png\" />\n";
}
```

```

        echo "<copyright>https://link.to/podcast-copyright-statement/</copyright>\n";
    }
    add_action( 'rss2_head', 'attn_rss_feed_changes', 10, 0 );

    function attn_rss_feed_item_changes () {
        echo "<itunes:explicit>no</itunes:explicit>\n";
        echo "<itunes:summary><![CDATA[";
        echo the_excerpt_rss();
        echo "]]></itunes:summary>\n";
    }
    add_action( 'rss2_item', 'attn_rss_feed_item_changes', 10, 0 );

```

Validation

A podcast is really just an XML document that follows (for our purposes here) the RSS specification and contains an audio track as an enclosure. Both XML and the RSS specification are quite strict and not particularly “human friendly” and so the best way to validate that the feed you have created meets the required syntax is to use an online validator.

There are a number of podcast validators on the internet but they don’t all contain the specific Apple requirements. I have found the most universal to be <https://podba.se/validate/> though many people also use <https://castfeedvalidator.com/>

Both tools will give an indication of whether each aspect of the feed is valid and also hints on what to change if it isn’t.

Conclusion

Delivering via these services is little different to delivering a podcast. There is no “secret sauce” required to make your service available over a voice assistant any more than to make it available to older online technologies. Hopefully the instructions here have demystified the process somewhat.

As at the time of writing there are two other voice assistant services available (Microsoft Cortana and the BBC’s “Beeb”). MS Cortana has not been covered here because you need a PC to use it (it isn’t available as a standalone device). The “Beeb” is not covered here because, at the time of writing at least, it will only supply BBC content.

Annex 2: Website and Integrations

Introduction

This section provides a high level guide to re-create the solution I created for Witney Talking News. There will likely be areas you need to adapt to meet the specific needs of your organisation. For example, the solution I created requires only that the recording technician upload the individual tracks at the end of the recording session and set the edition number and date. All the rest is then automated, including provision via BWBF hardware and app.

You could choose to avoid the integration step, and create a new blog post for each edition manually. That will still give you a website and access to the voice assistants for under £10 per annum but without a lot of the coding. However, if you wish then to use BWBF's devices and app then you would also have to upload to them separately.

I have written this guide using WordPress as a foundation as that is what I used. Other options are available but I have not included them here for the sake of brevity. WordPress, being at its heart a blog, seemed the perfect starting point.

From here on, the guide is split into two sections, the first that covers the basic setup and the second that covers the integration and automation.

Website

Requirements and Instructions

You will need at least the following:

- A domain name
 - For a charity or community organisation in the UK the best practice is to take a domain name ending .org.uk – this both clarifies that you are a UK organisation and also that you are not a company trading for profit. It isn't mandatory, but it is advisable.
 - What comes before the .org.uk is your choice so long as nobody already owns it. However consider the following points:
 - The chosen name should be readily associated with your organisation.
 - It should be as short as possible to be easily memorable.
 - Whilst domain names can contain underscores and hyphens they should be avoided if possible. Think about how it will sound when you read it out and which is easier for a visually impaired person to remember.

- Consider, for example “Any Town Talking Newspaper”. Ideal would be something like “attn.org.uk” which you would say quite easily as “Ay Tee Tee En dot org dot uk”. “any-town-talking-newspaper.org.uk” on the other hand, whilst possibly more relatable, is rather long winded to announce, “any hyphen town hyphen talking hyphen newspaper dot org dot uk”.
- This is also the case with email addresses. It is much easier to say “enquiries at ay tee tee en dot org dot uk” than “enquiries at any hyphen town hyphen talking hyphen newspaper dot org dot uk”. Not to mention the possibilities of someone with poor sight mistyping an address so long.
- For the rest of the guide the domain name you choose will be referred to as [attn.org.uk](https://www.kualo.com/free-charity-hosting) (please substitute the correct domain wherever that is referenced).
- A web host that offers a cPanel interface
 - There are many such hosts around. At Witney Talking News we use Kualo (<https://www.kualo.com/free-charity-hosting>) as they have a free plan for charities. They also supply the domain name, and it is this that makes up the cost of £7.20 per year.
 - This will not work for TNs which are not registered charities unfortunately. If your organisation is in that position then expect to pay in the region of £100 per annum for an account with a company like Kualo. There are many cheaper, but downtime and slow loading can be common place at the lower end of the market.
 - It is quite possible to achieve all that follows with a host that uses Plesk or has no control panel at all. However the former I have no experience of and the latter is too complex to document here. Thus the rest of the guide assumes you have a cPanel host.
- SSL certificates
 - This used to be an expensive and complex process. It is as easy now as logging into cPanel, scrolling down to “Security” and clicking “Lets Encrypt SSL”. Follow the instructions and you will have https enabled within a few minutes free-of-charge.
 - This is vitally important, not because the text on your page needs to be encrypted, but some sites which do not support https are no longer showing up in search results in Google (and possibly other search engines).
 - If you are considering a “contact us” form or taking donations through your website, https is vital – it is what creates the little padlock on the address bar that we are all told to look for before entering any information into a website.

- A WordPress website on that host
 - Do this after you set up the certificates as otherwise it will create a lot of work.
 - Scroll to “Softaculous” at the bottom of the Control Panel
 - Select WordPress, choose https and leave the installation directory blank.
 - Fill in the details about your organisation, choose an admin password and install it.
- One disadvantage of the Kuali charity plan is that it doesn’t allow you to use the unlimited storage space for a permanent archive. If you wish to create such an archive you will need a library account at archive.org
 - This will allow you to place your publications into their permanent record. You may reference these recordings through WordPress using the IAML plugin mentioned below.
 - Not every TN will wish to do this (or be able to gain copyright approval to do so). It was something I was very keen to implement though to make the TN the audio permanent local record in the same way as local libraries retain the printed paper in microfiche as a visual permanent record.
 - Initially at archive.org you will need to upload your content to the “General Audio” collection. Once you have uploaded more than 50 items you may apply for your own collection and then move your content to that collection. You may then upload any future content there. This makes your content easier to find and marks it out as yours.

Further Details

Beyond the points to consider above I am not going to detail how to install and configure WordPress – there are comprehensive instructions at <https://wordpress.org/support/category/getting-started/>. Where there are two versions of something for wordpress.com and wordpress.org you will need the version for wordpress.org.

WordPress Themes

There are in excess of 10,000 themes available, over 3,500 of those free of charge. The theme effectively governs what your website looks like to sighted people and, to a much lesser extent, what it sounds like to those with a screen reader.

There are two specific things to bear in mind when choosing a theme. It must be “responsive” and it must be “accessible”. The former is important to ensure that viewing the site on a mobile phone is a pleasant experience – this is far more important than you may think, over 50% of website views are from a mobile device! The latter is perhaps the

most important feature as a truly accessible theme will ensure that those using assistive technologies (screen readers, special input devices etc) are properly catered for.

Most themes can be customised to at least some degree to take into account corporate colours etc.

The one I chose was “Tiny Framework” by Tomas Mackevicius, purely because it was simple, clean, responsive and accessible. However the choice of theme is left as an exercise for the reader, but it is probably wisest to choose one that may be installed directly from within WordPress as these have been submitted for approval.

The page at <https://wordpress.org/themes/tags/accessibility-ready+custom-logo/> should provide a reasonable list of those themes which claim to be accessible.

WordPress Plugins

Plugins are used in WordPress to extend the basic functionality. There are currently around 54,000 available in the official directory and many more besides that. It is very easy to start installing many plugins but they should be used as sparingly as possible, as they can introduce security flaws if written badly. Which you use will vary depending on what you are trying to achieve with your website but here is a list of most of the ones I used:

- For handling media
 - Internet Archive Media Manager - <https://wordpress.org/plugins/media-library-internet-archive-content/>
 - This allows the audio for our archive to be stored at archive.org rather than on our host’s server as they do not permit large quantities of media to be stored. If you are not storing an archive or your host permits local storage this is not required.
 - This is a plugin I created myself specifically for Witney Talking News. I have tried to make sure it is self explanatory, if not, please contact me or use the support tool on the plugin detail page.
- For accessibility
 - ResponsiveVoice Text To Speech - <https://wordpress.org/plugins/responsivevoice-text-to-speech/>
 - NB: This plugin doesn’t read page or post titles out of the box. Should you require that functionality now, you can find a patched version at <https://github.com/yknivag/ResponsiveVoice>.
- For a contact form

- Contact Form 7: Accessible Defaults - <https://wordpress.org/plugins/contact-form-7-accessible-defaults/>
 - This must be installed before Contact Form 7. It ensures that appropriate screen reader prompts are added to the forms.
- Contact Form 7 - <https://wordpress.org/plugins/contact-form-7/>
- Contact Form 7 – reCAPTCHA - <https://wordpress.org/plugins/contact-form-7-recaptcha/>
 - You need to create a free Google account for this, full instructions are included in the plugin.
 - It is vital though to prevent endless amounts of SPAM messages.
- For GDPR Compliance
 - GDPR Cookie Consent Banner - <https://wordpress.org/plugins/uk-cookie-consent/>
- For social sharing and SEO
 - Google Analytics – The plugin Witney Talking News are using is no longer available, but there are many others to choose from.
 - You'll need to register with Google Analytics but this can give a really good insight into how people are using your site.
 - Google XML Sitemaps - <https://wordpress.org/plugins/google-sitemap-generator/>
 - This allows Google (and other search engines) to better “map out” your website and improve your search rankings.
 - Open Graph and Twitter Card Tags - <https://wordpress.org/plugins/wonderm00ns-simple-facebook-open-graph-tags/>
 - This gives a richer experience when viewers share your pages or posts on Facebook or Twitter.
 - Share Buttons by AddThis - <https://wordpress.org/plugins/addthis/>
 - You need a free AddThis account, directions can be found in the plugin.
 - It adds “sharing icons” to every page and post to encourage viewers to spread the word through Facebook, Twitter and any number of other such services.
- If you wish to take donations on your website (see the discussion in the general body of the document) then I would recommend the plugin I wrote specifically for Witney Talking News:

- Simple PayPal Recurring Donations - <https://wordpress.org/plugins/simple-paypal-recurring-donations/>
 - This allows you to collect one time payments from debit or credit cards or via PayPal. It also allows continuous monthly donations for things like “sponsor a listener for £5/month” style campaigns.
 - You will require a PayPal account. Please do set this up properly as a charity account – it is true that this process is not easy or quick, but doing so not only makes the process compliant but also leads to lower payment processing fees.

Site Content

At this stage you have a blank website (save for some sample content). What you put on the site is up to you, of course, but there are a number of things you must consider:

- By law any website of a charity or business must include that organisations “imprint”. That is the full registered name, address and company/charity number. Kualo’s charity plan also requires a link back to the charity hosting page. I have found the best option is to include these in the site footer.
- For GDPR, you need a privacy page to explain what you do and don’t do with people’s browsing data and how you handle cookies. WordPress helpfully includes a template but you will need to modify it.
- When you upload pictures be sure to describe them as best you can in the “Alt Text” field. When someone with a screen reader accesses your website this description will be read to them in place of the picture. Make the experience the best you can for them.
- Consider a contact page, and a donate page. Maybe also a page about volunteering.
- Also consider carefully how you categorise your posts:
 - I have found then that the easiest way to catalogue and display each publication is to create a post for each in the blog and put these posts in a specific category (separate from, say press release type posts). This way that category’s “feed” can become your podcast (as detailed in Annex 1).
 - At Witney Talking News we publish weekly editions and bi-annual magazines. I created a category called “Publications” and then created two child categories called “Editions” and “Magazines”. Each post is catalogued in either “Editions” or “Magazines” only but the feed from the parent category (“Publications”) forms our podcast as it contains all the contents of both its children.

- If you are wishing to create an archive of all the recordings you have stored then you will need to create a post for each and manually set the date back in the post editor to the date of the publication. NB: If you are using the integration and automation code below this is unnecessary as that code can process archive publications alongside the current ones.

You are free to browse the public facing site I create for Witney Talking News (<https://wtm.org.uk/>) for ideas, but please don't slavishly copy the content there. It is always best when content is written in the "local voice" adopting the tone and turn of phrase your listeners are used to hearing from you.

One last point is to consider who the site you are creating is for? Is it just for listeners and potential listeners? Or is it also there for the family members, friends, carers and agencies who care for and support potential listeners? Bear in mind the sort of information that these people may be seeking also.

Ongoing Process

Once you have the basic structure you will need to create a post each time you make a publication (weekly for most TNs). If you are utilising the Integration and Automation code you may ignore this as that code does this for you. For those not wishing to utilise that package, you will need to do this each time:

- Create a single MP3 of your entire publication end to end.
 - Remember to strip off any content you are not allowed to publish online (like "In Touch").
- If you are hosting these yourself, then upload the file using the WordPress Media Library. Otherwise upload it to archive.org and then add to the WordPress Media Library using the IAML plugin.
- Create a post in WordPress and include an audio element pointing at your MP3.
- Be sure to add the post to the correct category and then publish it.
- If you want to get ahead you can create several such posts with the bare details and keep them as drafts. You can also add extra text to the page such as a list of stories or readers names (with their permission) to make them more feature rich.

That's it! Your page will be live, your podcast and all the voice assistants will update automatically (so long as you have them set up as per Annex 1).

If you are not using archive.org then you will need to routinely remove older posts and the audio files to meet the requirements of your host. Your podcast will, by default, contain only the latest 10 so it would be reasonable to remove anything beyond say 3 months old.

Integration and Automation

Introduction

So, by now everything is in place, you have delivery by web, podcast and all three voice assistants. However after each publication you must upload individual files to BWBF, make a single file, upload that to archive.org, add it to WordPress with IAML, create a blog post attaching that single file and send out emails. You also need to agree a consistent template for the posts and emails. It doesn't look very professional if one is entitled "7/6/20" and the next week "14th June 2020".

This section takes care of that repetitive work, enforces consistency and brings everything together.

It is entirely optional – there is nothing wrong with doing this manually. Consider though what happens if you take a week off, who else can do it all?

What does it actually do?

This code, once installed and configured (see below) allows you to simply take the individual files from your USB stick, go to a webpage, enter an edition number, drop the files onto the box and all the rest will take care of itself. A single MP3 will be generated automatically, uploaded to archive.org, added to WordPress, a post written and published and playlists created. All whilst you sleep. It cleans up old editions from the playlists automatically to prevent too many files being stored on the server and correctly sets the metadata in the MP3 files so that they are attributable to your organisation.

This overnight functionality takes place in a series of three scripts called, imaginatively, `stage1.php`, `stage2.php` and `stage3.php`. The first updates the metadata on the individual files, transfers those amended files to the appropriate directory for the playlist generator and initiates the process at Transloadit (see below) for generating a single MP3. The second takes that single MP3, updates the metadata, uploads it to archive.org and adds both a media library entry and blog post to WordPress thus updating the podcast. The third script simply sends an email to all those who have subscribed – this is separate to allow it to be set to an appropriate time of day without impacting the other scripts.

It will also allow you to process archived versions in the same way. Just set the correct date of publication during upload and it will be backdated automatically in the background. There is a limit to how many you can process at any one time though, as there are usage limits at Transloadit – I'd suggest no more than 2 archive ones per week along side your normal one.

Separately, there are scripts in the directory `httpdocs` called `playlists.php` (which dynamically generated `pls`, `m3u`, `m3u8`, `asx`, `wpl`, `ram` & `json` playlists) and `recentlinks.php` which allows you to have links like

<https://media.attn.org.uk/web/links/edition/latest.mp3> which will always play the latest edition (you may replace the `mp3` with `html` or `htm` also to make a link to the blog post). This means you can have static links in your WordPress side bar or menu which automatically update. All the same permutations can be used in this format as can be used for the playlists (see <https://wtm.org.uk/listen-online/streaming-playlist-guide/> to see the possible formats). You can replace `web` with anything you wish – it can then be used to track, for example, where traffic is coming from. For example in the emails the links replace `web` with `email` – this way in your viewing stats you can see how many clicked the link from the email and how many from the website. The same applies to the playlist links also. This behaviour is managed jointly by those two scripts and the `.htaccess` file. Both also support reference by edition or magazine number, eg.

<http://media.attn.org.uk/web/playlists/edition/1234.pls>.

It is important to note that these `pls`, `m3u`, `m3u8`, `asx`, `wpl`, `ram`, `json`, `mp3`, `html` & `htm` files never actually exist anywhere on the server. They are generated virtually on-the-fly by these two PHP scripts. When for example

<https://media.attn.org.uk/web/playlists/edition/latest.m3u> is typed into the browser, the `.htaccess` file re-writes that request in the background to become [https://media.attn.org.uk/playlists.php?](https://media.attn.org.uk/playlists.php?source=web&type=edition&ref=latest&format=m3u)

`source=web&type=edition&ref=latest&format=m3u`, the `playlists.php` file then generates output that looks like an `m3u` file.

Take a look at <https://wtm.org.uk/listen/> and the pages in that menu for the full range of ways of listening produced by this code.

Lastly, the system generates comprehensive log files. These are stored in the directory `subdomain_root/uploads/_logs` each named to match a folder in `subdomain_root/uploads` (which is where the uploaded files are initially put). If anything goes wrong with processing a particular publication this should be your first port of call. However please do not delete any log files which relate to publications still in the uploads directory as the three stages use the presence or otherwise of these log files to determine what has or hasn't already happened and you could easily cause unwanted re-processing. If something does go wrong most things may be re-processed safely, except the upload to archive.org. If an upload has happened but needs replacing then this must be done manually at archive.org.

Assumptions

It is important to remember that this code was written specifically for Witney Talking News and their specific use case. There are, therefore, a number of underlying assumptions and hard-coded ways of working. Essentially the code assumes that you have two types of publication, editions and magazines. Editions are given a 4 digit edition number, whereas magazines are given a three digit number prefaced with the letter M. Magazines also allow for the input of a name and theme (and each of these should be used as the metadata is rather unpredictable otherwise – this should be fixed in a future release).

This may be directly suitable for many TNs (there is no need to use the magazine functionality if, for example, you only have regular editions). With minor tweaks it could probably be made to fit the publication regime of most TNs.

Requirements

There are several requirements for this to work:

- A subdomain (the rest of this guide will refer to that as media.attn.org.uk, substitute the one you create throughout the rest of the guide).
 - You should create this in cPanel and add a Lets Encrypt certificate to it.
 - Make a note of the root directory for this subdomain. This guide will refer to that directory as [subdomain_root](#) for the remainder of the instructions. Wherever you see that substitute the correct directory name.
- An account at Transloadit (<https://transloadit.com>) for automated audio transcoding.
 - Create a standard free account and then email Transloadit with details of your organisation and charity number. For registered charities they will then, at their discretion, move you to their first tier paid plan for free.
- An account at archive.org (see website instructions above).
- An email mailing list for those who wish to receive by email to subscribe to.
 - Mailman comes included with cPanel hosting, basic instructions for this are below. You may use any system to provide this so long as there is a single email address to send the message to.
 - Alternatively you may choose not to use [stage3.php](#) and instead use the WordPress subscribe feature to enable listeners to sign up to email of your new posts.
- The integration and automation code I wrote from GitHub (<https://github.com/yknivag/TN-Online-Service>).
 - This code is supplied free-of-charge, as-is with no warranty or explicit support. It is supplied subject to the LGPL license which means you are free to use it for whatever purpose you wish, but if you change or improve it you must share those changes freely with others under the same license. If you wish to change it, please fork the repository and keep it public. If you feel there can be non-breaking improvements, please also create a pull request.

Instructions

Setup

When you have those requirements set up:

- Download the code from GitHub as a .zip file. Use the File Manager in cPanel to place that zip file in the root of the subdomain you created and then to unzip it
- Create an FTP user in cPanel for Transloadit and point the “Directory” setting to `subdomain_root/uploads`. Note down the username and password you created.
- Create an email account in cPanel from which to send automated emails and take a note of that account’s credentials.
- At Transloadit now:
 - In Credentials from the main menu:
 - Make a note of the auth key and auth secret, you’ll need them later
 - Click to add a new set of Third-party Credentials:
 - Add the FTP username and password you created above.
 - In Templates:
 - Copy and paste the code below:

```
{
  "steps": {
    "imported": {
      "robot": "/ftp/import",
      "credentials": "xyz_ftp_credentials",
      "path": "0000_01-01-1970/"
    },
    "concatenated": {
      "robot": "/audio/concat",
      "use": {
        "steps": [
          "imported"
        ],
        "bundle_steps": true
      },
      "preset": "mp3",
      "bitrate": "112000",
      "ffmpeg_stack": "v3.3.3"
    },
    "exported": {
      "robot": "/ftp/store",
      "use": [
        "concatenated"
      ],
    },
  },
}
```

```

        "credentials": "xyz_ftp_credentials",
        "path": "_processed_files/0000_01-01-1970.mp3"
    }
}
}

```

- Changing `xyz_ftp_credentials` to the name you gave the credentials in the previous step.
- There may be other things you wish to change later but for now save the template and make a note of the Template ID (at the top of the page).
- At archive.org generate your API keys.
 - This is not a process which is in anyway obvious, even to those who have done it before, but it is reasonably well documented at <https://archive.org/services/docs/api/ias3.html>
 - Make a note of both keys
- Set up the mailing distribution list.
 - From cPanel in the Email section, choose “Mailing Lists”, give it a sensible name and a strong password, set it to public and create it.
 - Now click to “Manage” the list. Most of the options are OK at the defaults, though you may want to change the description, info, welcome message etc to be more relevant to your organisation. However there are a few specific things that need to be changed for security:
 - In `General Options`, ensure `reply_goes_to_list` is set to `Poster`
 - Under `Subscription rules` ensure `private_roster` is set to `List admin only`
 - Under `Sender filters`, ensure `accept_these_nonmembers` is set to only the address you will use in the script to send the automated email. You may also want to add a committee person’s email here too so that in the event of a problem that person can email the list to explain the situation.
 - In the same section, ensure that `default_member_moderation` is `Yes`, and that `member_moderation_action` is set to `Reject` (this stops subscribers being able to message each other. You may want to update `member_moderation_notice` to something that reflects your organisation.
 - In `Digest options`, set `digestible` to `No` (there is no point for this particular purpose) and set `digest_is_default` to `Regular`
 - There are many other settings that you may wish to take a look at also but these cover the basics.

- Now, in the cPanel file manager in the `subdomain_root/scripts` directory edit the emails that will be sent.
 - These are in the files `email_template_html.dat` and `email_template_plain.dat`
 - The files are plain text files and need to be changed to match your organisation. Just be sure to leave all the `###NAME###` type tags as they are. These are replaced in the code to make the emails unique each week.
- Still in the cPanel file and the `subdomain_root/scripts` make a copy of `config-sample.ini` and call it `config-live.ini`
 - Edit this file and update all the settings to match the details you noted down above and those specific to your Kualo account. Pretty much every setting will need to be changed. This is where you will utilise all the credentials you have been noting down.
 - Lastly at the end of this file, set a `username` that you or other volunteers in your organisation will use to access the upload tool and then use a service like <https://www.md5hashgenerator.com/> to create a hash of the password you would like to use and add this in the `passhash` setting.
- Again in the file manager in the same directory. Edit the file `cleanup.sh` and change the values of `UPLOAD_DIR`, `PROCESSED_DIR` and `MEDIA_DIR` to reflect your Kualo username and the server root directory for your subdomain.
- Next the automation needs to be setup. This is done using cron (which is a job scheduling tool).
 - In cPanel find “Cron Jobs” under “Advanced”. Set the Cron Email to one you will receive – any errors will be sent here.
 - Then you need to add four jobs:


```
04 02 * * 5,6 usr/bin/php home/username/subdomain_root/scripts/stage1.php
04 04 * * 5,6 usr/bin/php home/username/subdomain_root/scripts/stage2.php
30 08 * * 5,6 usr/bin/php home/username/subdomain_root/scripts/stage3.php
01 02 * * 5   usr/bin/php home/username/subdomain_root/scripts/cleanup.sh
```
 - This is not the place for a primer on the structure of a cron job, the important thing here is the 5th column (the weekday setting). In the example above they are set for every Friday and Saturday morning (except cleanup). This is because at Witney Talking News we record and upload on a Thursday evening. The Friday run processes everything that’s been uploaded that week. The Saturday re-run is there in case the recording engineer had a problem with connection and couldn’t upload until Friday. Obviously you will need to set these to run on which ever days you need. Unless you have a specific need leave the timings alone – specifically the gap between stages 1 and 2 as this allows Transloadit time to process the file. Their processing is asynchronous, and your

job will join a queue of unknown length and so you cannot assume it will always return in 5 minutes (although it will most of the time). Stage 3 controls the time of day the email is sent and may be freely adjusted (so long as it is after stage 2 and at a reasonable hour).

- Finally, you need to change the directory that media.attn.org.uk points to. This may be a confusing step but it is absolutely vital for the security of this service and also to make it work.
 - In cPanel, access Subdomains page, find the media.attn.org.uk subdomain in the list, and click the pencil in the “Document Root” column.
 - Add the text [httpdocs](#) to the end of the path and click “Change”. It should change from something like [subdomain_root](#) to [subdomain_root/httpdocs](#)
 - This protects the scripts and upload directories and ensures that they are not accessible via the website. It also prevents anyone reading the config file and discovering the keys in it.
 - This has now technically changed the document root we created earlier and have inserted into all those config files, but those remain the same, don’t update those to this new value!
- To make the service look your own, back in the cPanel file manager browse to the [subdomain_root/httpdocs/css](#) directory and edit [app.css](#). Set your corporate colours in here to personalise the uploader. Similarly you may replace the logo file in the [subdomain_root/httpdocs/images](#) directory.

Ongoing Operations

For normal operation, once a master has been created it is simply a matter of uploading the content each week as below:

- Begin by browsing to <https://media.attn.org.uk/uploader> (the subdomain you picked during setup).
- Login using the username and the password you hashed in the setup step (this system supports only one shared username and password).
- Then simply follow the instructions on the first screen:
 - For a regular edition enter the latest four digit number and ensure the correct date is set – this should be the date of publication.
 - For a magazine enter a three figure number (the M prefix will be added automatically), the publication date, the name and theme.
 - For both you will see the name building at the end of the form as you type, along with some examples of what the name should look like. It is important to get this

right at this stage as the eventual metadata, blog post and podcast entry will all be derived from these entered details.

- Press “Submit”.
- On the second screen:
 - Drag and drop your files on to the dashed area, you can select all of them at once and do this, or one at a time in any order. They **must** however be numbered sequentially in the format `001_anything.mp3`, `002_something-else.mp3`. If you are using KARDS this is the default output, if you are using something else you will need to arrange for the files to be in this format.
 - They will appear in the file list on the right and you can monitor each upload.
 - If any fail, drop those files on again. It doesn't matter how many times you drop files on only one copy will end up on the server.
 - Once all the files have uploaded successfully, press “Logout”.
- The automated scripts you set up in cron during the setup will now process these files according the schedule you set behind the scenes.

BWBF Sonata+ & Talking Newspaper App

There are two options here. You may continue using the BWBF file upload service to place your content onto their services, and you must do this if you want to retain access to their statistics dashboard. Or you can ask them to use the playlists generated by this tool. If you do this you should supply them with the following list (replacing [attn.org.uk](https://media.attn.org.uk) of course):

```
Latest Edition:  https://media.attn.org.uk/bwbf/playlists/edition/latest.m3u
1st Prior Edition: https://media.attn.org.uk/bwbf/playlists/edition/prior1.m3u
2nd Prior Edition: https://media.attn.org.uk/bwbf/playlists/edition/prior2.m3u
3rd Prior Edition: https://media.attn.org.uk/bwbf/playlists/edition/prior3.m3u
4th Prior Edition: https://media.attn.org.uk/bwbf/playlists/edition/prior4.m3u
5th Prior Edition: https://media.attn.org.uk/bwbf/playlists/edition/prior5.m3u
```

and optionally (if you have a magazine)

```
Latest Magazine: https://media.attn.org.uk/bwbf/playlists/magazine/latest.m3u
```

This gives you the maximum permitted current plus 5 prior editions. You may, of course, decide any lower number of priors in conjunction with BWBF. At Witney Talking News we have found that people often listen to the current and up to three prior.

Conclusion

Whether you follow just the website part, or both the website and integration guides, you now have a way of distributing your content through the web, email, podcast and (if you also follow Annex 1) all three major voice assistants. All for the sum of £7.20 per annum.

Annex 3: Telephone Dial-In Service

Introduction

Be under no illusion, this is a complex task. It is quite possible for someone with a technical background but it is not something one can pick up in a weekend. As with the rest of the document the instructions here are quite “high level” and deliberately so. They should provide sufficient prompts for someone with a basic Linux sysadmin background.

Similarly I make no apologies for the number of abbreviations here. Whilst not all them may be known immediately to a sysadmin, a person with sufficient experience to complete this task successfully ought to have no problem looking them up.

Options

There are only two viable options; one is to pay for the only available commercial solution, the other is to follow the instructions below.

There is a third way, using Twilio, but this is billed per minute per incoming call and very quickly mounts up to an unsustainable variable cost. For this reason I haven't covered it here.

Architecture

The solution is made up of several levels. Initially a telephone number is required. This will take calls from the telephone networks and route them to your PBX.

The PBX itself is made up of a hosted virtual server running a Linux based OS and the FreePBX software. That software takes the call, plays the menu, interprets the key presses and subsequently plays the audio.

It is important to note that voice over IP (VOIP) is particularly sensitive to network configuration and a server with a dedicated public address is required as the calls would likely be interrupted by NAT.

Exactly how the menu system is set up will depend on the requirements of your TN. The solution we use at Witney Talking News offers the following key press menu:

1. Press 1 to hear the current edition
2. Press 2 to hear previous editions
 1. Press 1 to hear the edition from one week prior
 2. Press 2 to hear the edition from two weeks prior
 3. Press 3 to hear to edition from three weeks prior

3. Press 3 to hear the latest magazine

Each menu also has press * to hear the options again, # to return to the main menu or 0 to hear instructions on how to skip forwards and backwards during playback.

If no input is provided at all the system times out, plays the latest edition automatically and then automatically terminates the call. This is important as some listeners may struggle to find the keys on their telephone.

Depending on your specific requirements the structure may vary. You could, for example, set it up with no menu at all and go straight to the recording.

Instructions

Phone Number

There are many providers but none as reliable or as affordable as Andrews & Arnold (<https://www.aa.net.uk/voice-and-mobile/voip-information/>). Of particular benefit here is there is no minimum term for the number, it runs only month to month.

There is a cost/benefit trade-off here from 0800 on one hand where the organisation pays for the calls and listener doesn't, through to 0870 where the organisation can actually make money from the calls. The first can work out very expensive and the latter is unlikely to be welcomed by listeners.

That leaves a local number (01 or 02 depending where you are) or a charity 0300 number. All are charged at the same rate to both organisation and listener. My recommendation would always be one in the same dialling code area that your TN is based in. It's easier to remember then and some people are still wary of the 03 numbers.

Set up is simply a case of choosing a suitable number on the A&A website and setting up an account. Alternatively you may telephone them. They are well equipped to deal with direct debits requiring multiple signatories and remarkably helpful.

At the time of writing an active 01/02 number can be had for as little as £1.20 per month with no cost to the organisation for incoming calls.

VPS (Virtual Private Server)

There are many possible options here, but my recommendation is <https://vultr.com/> for two reasons. Their \$5US (plus VAT) per month plan is sufficient to run FreePBX for the amount of load that a TN would place on it and also because they offer a one-click installation process for FreePBX – the only provider I could find that did so.

The plan you need is under “Cloud Compute” - “SSD Cloud Instances” and is the 25GB one.

Sign up for an account, deploy a new server and, after choosing your location, under “Server Type” choose “ISO Library” and then “FreePBX”. Give the machine a sensible host name and deploy it.

Once you power it on, load the console and follow the OS installation. Once it is installed, go back to the control panel and remove the FreePBX ISO. Now restart your server from the console. Run `yum update` and `yum install` to make sure any updates are performed. Once those complete, restart again and you can then use the web GUI to configure the PBX.

I found it useful to setup a DNS entry in the Kualo control panel for the server’s public IP back so I access it via a subdomain rather than having to remember the IP address.

PBX (Switchboard)

It really helps to think about exactly how you want to do this before you start and draw out your menus and destinations. Then you can tick them off on your list as you create them.

There are many tutorials and much documentation around for how to best configure FreePBX and I’m not going to detail every step here. Instead I will set out a few points to consider and some detail around the audio playback as this is the primary feature that sets the solution apart from a standard PBX and I found it to be quite poorly documented for this particular use case.

Initially ensure that all FreePBX updates are installed. Then create your audio playback endpoints:

- In the Admin menu choose “Config Edit” and select the file “extensions_custom.conf”.
- Exactly what you paste here will depend on what options you are offering in your menu. For the menu outlined above you will need the following content:

```
[edition_latest]
exten => s,1,Set(VOLUME(TX)=10)
exten =>
s,n,ControlPlayback(custom/publications/edition_latest,10000,3,1,5,2,*)
exten => s,n,Return()

[edition_prior1]
exten => s,1,Set(VOLUME(TX)=10)
exten =>
s,n,ControlPlayback(custom/publications/edition_prior1,10000,3,1,5,2,*)
exten => s,n,Return()

[edition_prior2]
exten => s,1,Set(VOLUME(TX)=10)
exten =>
s,n,ControlPlayback(custom/publications/edition_prior2,10000,3,1,5,2,*)
exten => s,n,Return()
```

```
[edition_prior3]
exten => s,1,Set(VOLUME(TX)=10)
exten =>
s,n,ControlPlayback(custom/publications/edition_prior3,10000,3,1,5,2,*)
exten => s,n,Return()

[magazine_latest]
exten => s,1,Set(VOLUME(TX)=10)
exten =>
s,n,ControlPlayback(custom/publications/magazine_latest,10000,3,1,5,2,*)
exten => s,n,Return()
```

- Save the file.
- Note that there is no file extension on the file name – this is important!
- The numbers after the file name relate to the key controls during playback, in this instance 3 for skip forwards, 1 for skip backwards, 5 to restart, 2 to pause/continue and * to return to the menu. The 10000 tells the system to skip 10 seconds at a time in either direction.
- Now from the console, navigate to `/var/lib/asterisk/sounds/en/custom/` and create a directory called publications. In that directory insert (by FTP or similar) the following files (or an adjusted list to match your config above):

```
edition_latest.mp3
edition_prior1.mp3
edition_prior2.mp3
edition_prior3.mp3
latest_magazine.mp3
```
- I actually created a directory inside the publications one for the files themselves and set the names above to be symlinks. That meant if I needed to I could move the locations of the files at any time without changing the FreePBX configuration.
- Apply the changes to the configuration on the web front end by clicking the big red button in the top right hand corner.
- You now have “targets” that will play your files. The next step is to create “Custom Destinations” that point at those targets.
- Again in the “Admin” menu choose “Custom Destinations” and then “+ Add Destination”. You need one for each of the above and the target for each should be `name,s,1` where `name` is the bit in the square brackets in the extract above. So for the first it would be `edition_latest,s,1`
- The “Return” can be used to go back to the menu when the audio is finished or set it to “Terminate Call” “Hangup” to have the system put the phone down automatically at the end.

- You can have more of these destinations than you have end points if you want the return to be different in different circumstances. For example I created two for the latest edition, one which returns to the menu after playing (which is called by pressing 1 at the main menu) and one that hangs up the call which is where the main menu times out to.
- Before you create your menus (IVR) you need to record them, you need clean audio files of someone reading out the options (one track per menu). Once you have the recordings you need to get them to the correct format and you can do this using <https://g711.org/> - choose “Standard Definition WAV (8kHz, Mono, 16-Bit PCM)” as the output format. Upload each recording and save the outputs.
- You can add them to the system using the “System Recordings” option in the “Admin” menu. These are now available to be chosen when you configure your IVR at the next stage.
- Once you have these custom destinations you can create your IVR as per the normal documentation, just choose your custom destinations for the digit presses.
- Remember to “Apply Changes” once you have finished configuring.

VOIP Trunk

Assuming you have chosen Andrews & Arnold as your number provider and FreePBX as your content delivery solution then you can simply follow this guide to make one point to the other - https://support.aa.net.uk/VoIP_Phones_-_FreePBX

For this to work as is, you will need to enable IPv6 in the Vultr control panel and assign an IPv6 address to the server.

For editing the configuration files you may use the same “Admin” - “Config Edit” option as you did to add the custom endpoints above.

Conclusion

Under normal conditions this is not a service I would ever have envisaged creating. I have never understood why anyone would want to listen over the telephone as it is so impractical – there is, for example, no way to continue from where you left off if you are disturbed. However when it became apparent at the start of the COVID-19 crisis that postal delivery would present a serious challenge to both the organisation and for listeners (many of whom were in shielding groups) something had to be done for those without access to the internet. On the basis that most people have access to some form of telephone this seemed the most reasonable option. However it is not something I would consider during a time where postal delivery is an option as I cannot imagine demand would be sufficient to justify the cost.