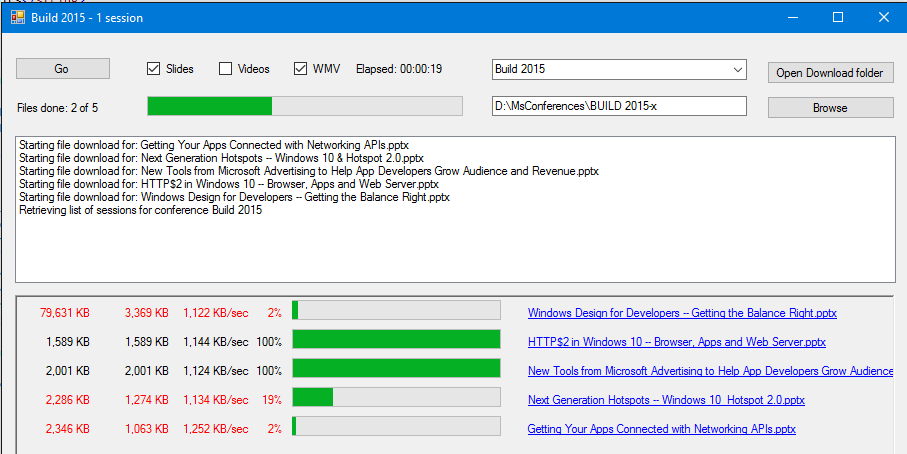
Download MS Conferences

DownloadMsConferences.exe is a program to download all videos and any associated PowerPoint slides for a specified Microsoft conference (e.g. //Build2016/) that has an RSS .xml file that describes the sessions. Note that by design the program will always download the highest resolution video file available. Make sure you have enough disk space. Major conferences such as Build can easily consume over 100GB of disk space.

For example, the URL <http://s.ch9.ms/events/build/2016/rss/mp4> describes the sessions for //Build2016/.

Note: This program was compiled with .Net 4.6.1 and requires it to run. It uses a *Config* file called DownloadMicrosoftConferences.Config.xml, which must be in the same directory as the .exe file. See the section *Format of DownloadMicrosoftConferences.Config.xml File* later in this document for more information.

The User Interface – Top Part



The combo box near the upper-right lists all the conferences known to the program, drawn from the *Config* file.

The text box below lets you specify the directory in which the video and slide files will be saved. It will be filled in automatically with a default value specified in the *Config* file. Or you can edit the text box yourself. Or you can use the Browse button to select a directory.

The Open Download Folder button will bring up the specified download directory in File Explorer.

Once you’ve selected the conference and the download directory, use the check boxes near the top left to select whether you want just the slides for the sessions downloaded, or just the videos, or (normally) both. The WMV checkbox is to support conferences that have videos in both .mp4 and .wmv formats; if checked it gives preference to downloading .wmv flies over .mp4 files.

When all of the above options have been set, click the Go button. It will read the RSS file specified in the *Config* file and extract information about the sessions. It will then proceed to download them.

Immediately below and to the right of the Go button we see the number of session files (video / slides / both) started so far, the number that have finished so far, and a progress bar indicating the overall download progress. Note that this is in terms of the number of files to be downloaded, and not in terms of the number of bytes to be downloaded.

Under that we have a listbox that simply shows the progress of the overall downloads, mostly in terms of “Starting the download for: <Filename>”. However there may also be messages about errors that occur during processing, and also messages that say that a file has been skipped and won’t be downloaded. More on these messages later.

The User Interface – Bottom Part

The bottom pane has the most information. Each entry has the following information:

* The size of the file being downloaded
* The number of KB downloaded so far
* The instantaneous download speed
  + Note: I added this calculation recently and I’m not 100% sure it’s accurate. Still, it doesn’t really matter if it is. At least it shows that data is being downloaded.
* The percentage download complete for that file
* A progress bar
* The name of the file being downloaded.

For example, the top line in the pane shows a file named *Windows Design for Developers – Getting the Balance Right.pptx* being downloaded. It is 79,631 KB in size. But at this point the program has only downloaded 3,369 KB, which is 2% of the file. At the point of this snapshot it was downloading at a rate of 1,122 KB/sec.

A few notes about this pane…

* Active downloads have their file sizes etc shown in red
* Files that have finished their download are shown in black
* Files that have not yet started their download have a blank file sizes section. But they do have their filename shown
* The program is written to download a maximum of 10 files simultaneously.
* Session titles that have invalid filename characters in them (e.g. *HTTP/2 in Windows 10: Browser, Apps and Web Server.pptx*) will have the invalid characters replaced by valid ones. So for this session name, the “/” has been replaced by “$”, and the “:” by “ –“.
* If the file download aborts for any reason (e.g. a timeout), the file sizes and filename will have a cyan background, and an error message will be shown in the listbox above
* When a file has finished downloading successfully, another file download will automatically start, up to the maximum of 10 concurrent downloads.
  + If the download aborts for any reason, sometimes a new download will *not* start. I suspect this is a bug in the .Net framework, perhaps a race condition.
* Hovering the mouse over a filename will show the description of the session from the MSDN web page for it
* Clicking on a filename that has finished its download will launch the file. Clicking a filename that hasn’t finished downloading will take you to the MSDN page for that session.
* In general there will be too many sessions to show at once. Scroll the pane to the right to see the other sessions.
* This program is driven by the RSS .xml file associated with the conference, and if this format changes appreciably, the program may fail. It also does web scraping and may well misbehave if Microsoft changes aspects of its MSDN pages.

The Disk Files Produced

The program is designed to be restartable. Closing the program will not harm anything. Any files that have been fully downloaded will remain. When the program is run again, it will skip over the fully downloaded ones. However for the rest of the files it will start the download at the beginning. It will not, for example, download just the remaining 2% of a file that had previously been 98% complete when the program closed.

As described in the *Config* section, rather than dumping everything into the same directory, files are placed in directories based on keywords found in the session titles. For example, using the default set of keywords (called *Topics*), the .mp4 and/or .pptx files for the session named *Innovation in C++ Code Generation* would show up in D:\MSConferences\*Conference Name*\C++. You’ll also see links to the MSDN page describing the session.

Some sessions have more than one topic in them. A session named *Building C++ Universal Apps with Visual Studio 2015* could go into

* The C++ directory
* The UWP directory for Windows 10 Universal apps (topic keyword “Universal”)
* The Visual Studio directory

Based upon the order of topics in the *Config* file, C++ comes first, so the physical file(s) (.mp4 / .pptx) would go into the C++ directory. However links to those files would also go into the UWP and Visual Studio directories.

If you were to look at the MSDN page for a session, you’d notice that the session description might contain additional information. For example, for the Build 2016 session *Async Programming Improvements for C++ and UWP* (at <https://channel9.msdn.com/Events/Build/2016/P489>), you’d see that this session is categorized with

* Tags of *Universal Windows Platform (UWP)* and *C++*
* Level *200 – Intermediate*
* Session Type *Pre-recorded*

This information is kept in 3 subdirectories directly under the target download directory named *$Level, $Session Type* and *$Tags*. Unfortunately these directories only have links to the session MSDN web pages. It wouldn’t be hard to provide links to the files themselves, but I didn’t have time to add these.

Enjoy!

Source code available on request. It does need Visual Studio 2015 (even just the free Community edition) to compile.

Format of the DownloadMicrosoftConferences.Config.xml File

This XML file is divided into two parts: <Conferences> and <Topics>

<Conferences>

Each <Conference> within the <Conferences> group has three elements.

<Conference>

<Name>Build 2015</Name>

<TargetDir>D:\Downloads\VideoTest\BUILD 2015</TargetDir>

<RssUrl>http://s.ch9.ms/events/build/2015/rss/mp4</RssUrl>

</Conference>

|  |  |
| --- | --- |
| **Name** | The name of the conference that you want to download session videos and/or slides from. This shows up in the combo box that allows you to select the conference. |
| **TargetDir** | The directory where you want the session data to be download to. |
| **RssUrl** | The URL of the RSS file (in .xml format) that defines the information about the sessions for the specified conference. |
| **Topics** | An optional section in the same format as the <Topics> data. This allows you to override (i.e. replace) the default Topics on a conference-by-conference basis. |

Topics

With some conferences having literally hundreds of sessions, we didn’t want to dump everything into a single directory. One advantage of this is to be able to shunt certain session data to, say, an Azure sub-directory of the <TargetDir>, and others to, say, a Windows 10 sub-directory. Now the user can, for example, ignore everything in the Azure sub-directory (if that’s what s/he wants) and concentrate on sessions about Windows 10.

The Topics are processed in order against the Title of the session. Given the list of Topics below (current as of January 12, 2016 but has been updated since), a given session title would be searched for the string “Xamarin” (in a case-insensitive fashion). If “Xamarin” were found anywhere in the Title, the session file(s) (video and/or slides) would be placed in the “Xamarin” sub-directory. If “Xamarin” were *not* found in the Title, the Title would be checked against each subsequent string in turn. If none of the Topics were found, the file(s) would be placed in the <TargetDir> directory for the conference.

Some of the entries have a backslash in the text. This supports a type of aliasing. For example, <string>Cloud\Azure</string>. This says to search the Title for “Cloud”, but if that text is found, don’t create a “Cloud” sub-directory and place the file(s) there. Rather put them in the Azure sub-directory.

The order of the Topics can be important. During development, the “.Net” string was placed before the “ASP.NET” entry. Because the program takes the first entry it finds, *all* ASP.NET articles were placed in the .Net sub-directory and none in the ASP.NET sub-directory. So the entry for ASP.NET was moved to be *before* the entry for .Net.

Thus there is the question of where to put a Title of, say, “Using C# to develop Windows Store, Android and IOS apps using Xamarin”. So does this get filed into the “C#” sub-directory, or “Windows Store”, or “Android”, or “IOS” or “Xamarin”? So even though “Xamarin” is the last word in the Title, since it is first in the list below, the session data will be placed in the “Xamarin” sub-directory. The remaining topics (“C#”, “Windows Store”, “Android”, “iOS”) are called secondary topics. Directories for these are created, but the downloaded files are *not* placed in the directories. Instead there will merely be link(s) to the files in the primary (Xamarin) directory. So you’ll find the data (or, at least, a pointer to it) in all relevant directories, but only one real file.

Sub-directories are created “on-demand”. So if, say, there are no Titles with, say, IOS in the Title, then no IOS sub-directory will be created.

<Topics>

<string>Xamarin</string>

<string>Universal\Windows 10</string>

<string>Windows 10</string>

<string>Windows 8</string>

<string>Windows Store</string>

<string>Windows Phone</string>

<string>WP8\Windows Phone</string>

<string>Azure</string>

<string>Cloud\Azure</string>

<string>Android</string>

<string>iOS</string>

<string>Linux</string>

<string>Entity Framework</string>

<string>Windows Server</string>

<string>DevOps</string>

<string>Graphics</string>

<string>C#</string>

<string>HTML</string>

<string>ASP.NET</string>

<string>.Net</string>

<string>XAML</string>

<string>Visual Studio</string>

<string>C++</string>

<string>TypeScript</string>

<string>JavaScript</string>

<string>Office</string>

<string>XBox</string>

</Topics>

Topics on a Per-Conference Basis

The high-level <Topics> tag is used by all conferences that don’t define their own <Topics>. But if there is one it will be used instead. For example, …

<Conference>

<Name>Build 2015</Name>

<TargetDir>D:\Downloads\VideoTest\BUILD 2015</TargetDir>

<RssUrl>[http://s.ch9.ms/events/build/2015/rss/mp4</RssUrl](http://s.ch9.ms/events/build/2015/rss/mp4%3c/RssUrl)>

<Topics>

<string>Windows Band</string>

<string>Python</string>

</Topics>

</Conference>

Then only sessions with a title containing “Windows Band” or “Python” in it will show up in its own directory.