



Articles » General Programming » Ultrabooks » App Innovation Contest Entries

YtoX

By **Prabu Arumugam**, 3 Dec 2012

We're sorry, but the article you are trying to view has been deleted.

Please note

This article is an entry in our [AppInnovation Contest](#). Articles in this sub-section are not required to be full articles so care should be taken when voting.

Introduction

Whether it is a weekday or weekend, imagine a life with your Ultrabook™ always 'on' your desk or your backpack. YtoX lets you control and extend the capabilities of your Ultrabook™ using interesting recipes. You can customize the recipes to do interesting things for you, to make your personal and work life easier.

Background

What YtoX aims to do to Windows (particularly Ultrabook™) is similar to what [Microsoft on{X}](#) did to Android. But unlike on{X} where you have to write JavaScript code to create recipes, YtoX allows you to create recipes using simple expressions that can be created in GUI using regular operations like touch, swipe, drag & drop, etc. There will be a sharing platform too, so that users can download and install the recipes created/customized by other users.

YtoX leverages the sensors (like compass, accelerometer, gyroscope, ambient light sensor, GPS) available in the Ultrabook™, to make it intelligent and for better use of the device.

How does it work?

The general form a recipe is "If X then Y", where X is a trigger and Y is an action.

A *Trigger* is a set of conditions, bound together by ANDs and ORs. A condition is a combination of left expression, right expression and a comparison operator (like less than, equal to, near to, etc). An expression is a valid combination of predefined tokens. A token can be a predefined parameter (like location, time, app, battery power, etc.) or an operator or a routine/function.

An *Action* is a task which is executed whenever the conditions of the corresponding trigger match. Actions can be simple like showing a toast-notification (or) advanced like video recording.

Eventhough the recipes sound more technical, the GUI for creating/customizing recipes will be quite user friendly.

The Recipes

YtoX comes with a predefined set of recipes. Some of them are:

- Remind me of my first appointment tomorrow when I turn on the device after 10pm.
- Remind me to keep the umbrella and raincoat in my backpack if it is going to rain the next day.
- When I leave from work, remind me to buy milk.
- Automatically reply "I'm in a meeting, will ping you back", when a friend starts a chat during a meeting.
- Notify me if I'm not traveling in one of my usual routes.
- Remind me to leave the bike/car for service, if I haven't visited the showroom for 3 months.
- Post the location to *Facebook* (or *Twitter*) when I'm in my favorite restaurant for more than 20 minutes.
- Remind me to visit the gym if haven't been there for 3 days.
- Show me the *Scorpio* horoscope everyday at 8am.
- Email me all latest news related to *Steven Sinofsky*.
- Launch the Photos app if I shake the device.
- Start playback if a wired headset is connected.
- Mute the volume when a meeting starts, and revert back when the meeting ends.
- Record the audio for all meetings or a particular meeting scheduled in Outlook.
- Capture photos when my device is used in new place.
- Reply "Battery is draining, will call you on mobile", when my wife starts a chat and battery is < 20%.
- Adjust the brightness if I'm reading for more than 2 minutes.
- Open Google if I draw 'S' gesture on the screen.
- When I save an excel document in "My Documents" folder, copy it to my Dropbox.

YtoX Configuration

After installing YtoX, you have to teach YtoX something about your world, like:

- Known Locations (like home, work, parents home, canteen, etc.)
- Special People (like spouse, sister, boss, etc.)
- Known Gestures (like diagonal swipe, S shape, etc.)

These configurations can be done using *Control Panel* component of YtoX. Apart from these, YtoX will maintain a known set of locations like Eiffel Tower, Taj Mahal, Sydney, Newark Airport, etc., so that you can create recipes like "When I leave Taj Mahal, upload today's photos from My Pictures folder to Flickr" (or) "When I enter Newark Airport, message my boss about my arrival via Facebook".

Building Recipes

Data Types

As discussed earlier, a token can be a constant or variable or routine or operator. Unlike typical mathematical expressions, the YtoX tokens will be of advanced data-types.

The following table throw some light on data-types of tokens:

Data Type	Constants	Variables
Location	Home, Work, Eiffel Tower, etc.	Current Location
Time	Today, Yesterday, Tomorrow	Current Time
Device Orientation	Portrait, Landscape, Upside Down, etc.	Current Orientation
Weather	Sunny, Rainy, Cloudy, etc.	Current weather
Duration (Timespan)		

Routines

Routines are functions/methods written to operate on tokens to get some meaningful information. The following table lists some of the routines in YtoX:

Routine	Arguments	Return Type	Purpose	Scope
lastVisit	location	timespan	Returns time since last visit to a given location	
past	time	bool	Determines whether time has passed the given time	Day
getWeather	time	weather	Returns weather at given time	
meetingInProgress		bool	To find whether any meeting is in progress	
knownRoute		bool	Determines whether current route is a known route	
durationAt	location, scope	timespan	Returns time spent at a given location	Day, Month, Overall
batteryRemaining		integer	Returns the remaining battery power in percentage	

Scope is used to apply temporal limits on a routine.

Events

YtoX exposes some predefined events and recipes can be hooked to these events. When that event occurs, the associated recipes will be executed. All the other recipes which are not associated with any event are executed by *Engine* at regular intervals. Some of the events in YtoX are:

- Device rotated to landscape
- Device rotated to portrait
- Device is shaken
- Device power on
- New chat

- Location changed
- Email received
- New news article (captured via RSS or push notification)
- New device connected
- Meeting started
- Meeting ended
- Left from a known location
- Entered into a known location

Each event will be fired with some parameters. The values of these parameters can be used in expressions. For example, 'New chat' event will contain 'who' and 'channel' as parameters, 'who' is the name of the person who initiated the chat and 'channel' is the chat client (like Gtalk, Facebook, etc.). 'New device connected' will contain 'device-type' as parameter which can be 'microphone', 'camera', etc.

Activities

Some of the activities that can be integrated with recipes are listed:

Activity	Type
Show a toast notification	Local
Open an app	Local
Open a URL	Local
Start Music	Local
Mute the volume	Local
Unmute the volume	Local
Record audio	Local
Photo capture	Local
Video capture	Local
Set Brightness	Local
Copy to Dropbox	Local
Send Email	Network
Upload photos to Flickr	Social
Upload photos to FB	Social
Send message to a contact at FB	Social
Reply to chat	Social
Post to FB	Social
Post a Tweet	Social

Recipes

Having understood the above concepts, building recipes is simpler.

Recipe: Remind me to keep the umbrella and raincoat in my backpack if it is going to rain the next day.

Trigger: IF getWeather(TOMORROW) == RAINY

Action: NOTIFY("Wanna soak in rain? If no, pack your umbrella and raincoat now.")

Recipe: Remind me to leave the bike/car for service, if I haven't visited the showroom for 3 months.

Trigger: IF totalDays(lastVisit(SHOWROOM)) >= 90

Action: NOTIFY("There is no service indicator in your bike? Take your bike to the showroom today before it ruins a day.")

Recipe: Start playback if a wired headset is connected.

Event: ON new device connected

Trigger: IF device-type == MICROPHONE

Action: Launch Music app

Recipe: Reply "Battery is draining, will call you on mobile", when my wife starts a chat and battery is < 20%.

Event: ON new chat

Trigger: IF from == WIFE and getBatteryRemaining() < 20

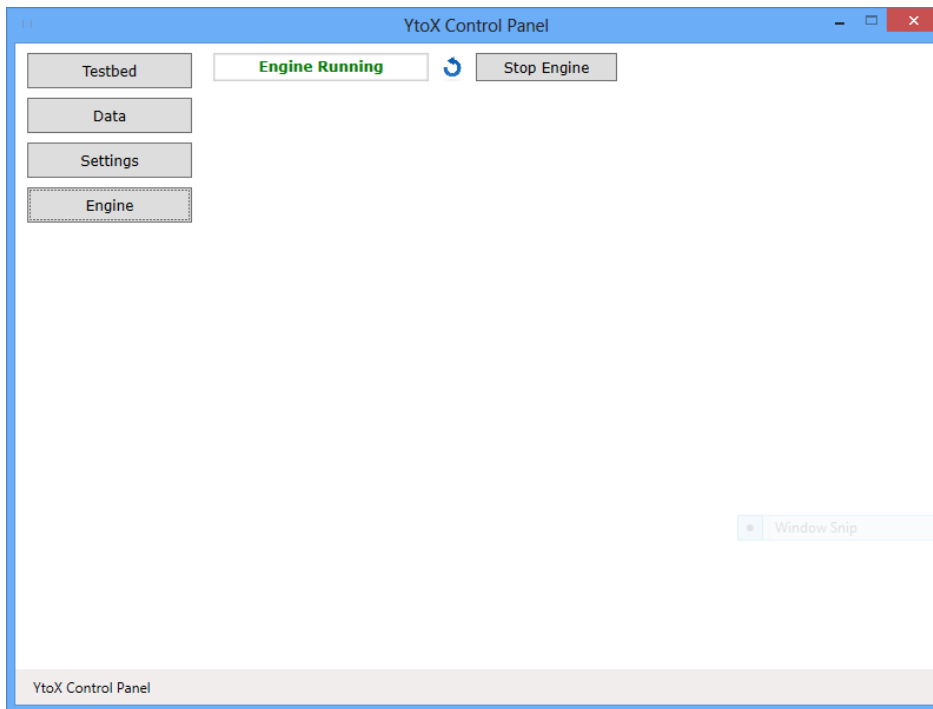
Action: ReplyToChat("Will call you shortly")

Architecture

YtoX contains three components, namely *Control Panel*, *Recipes* and *Engine*.

Control Panel

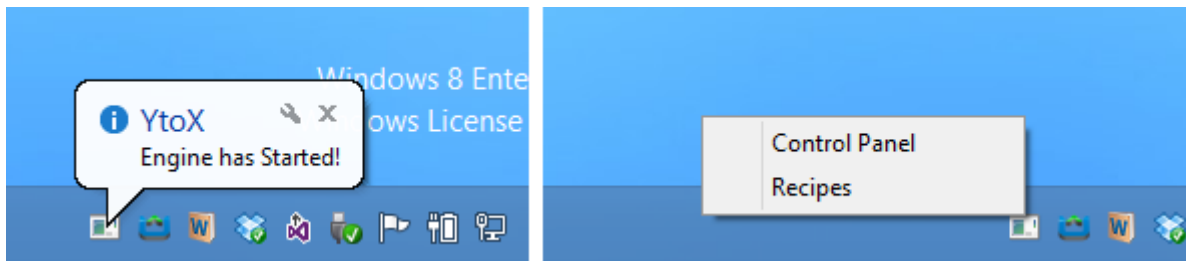
The *Control Panel* has options to let YtoX know about your world (as explained above in the section 'YtoX Configuration'). You can even start/stop the *Engine* from the *Control Panel*. Refer this screenshot:



Recipes

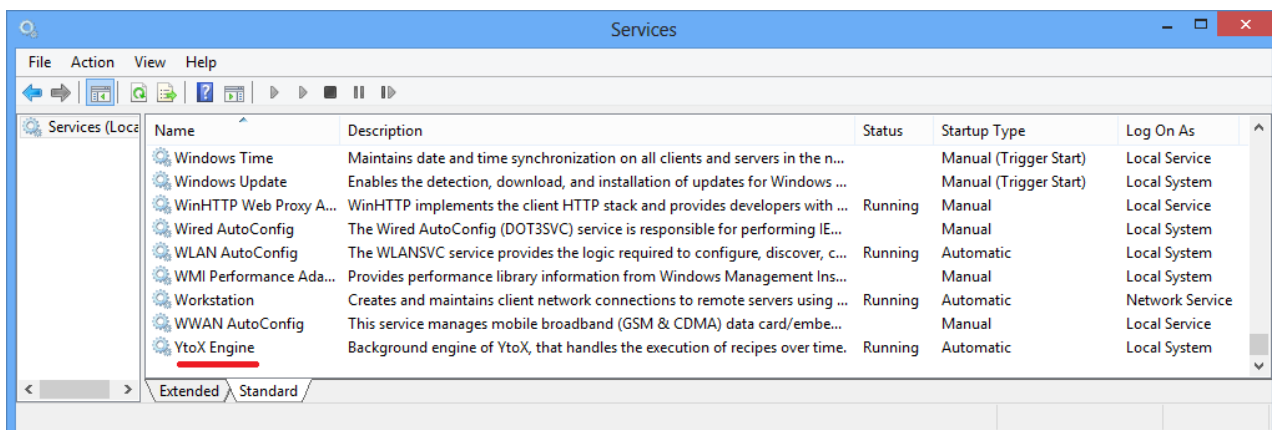
The *Recipes* component allows you to create new recipes or customize available recipes. *Recipes* is a fullscreen WPF desktop app with metro-like tiles. Tiles approach will be easy to use for scenarios like 'touch to select and enable/disable the recipe from bottom bar'.

The *Control Panel* and *Recipes* can be launched from taskbar also.



Engine

The *Engine* is a *Windows Service* that runs in the background to execute the actions of matching recipes over time. *Engine* will be responsible for executing both event-based recipes and interval-based recipes. *Engine* will be power aware, i.e. it will stop executing the recipes if it receives a 'critical battery-low' event, and the execution will resume once the device is connected to a power source.



Platform: YtoX is a traditional Windows Desktop app (with reference to WinRT APIs) and fits well into 'Productivity' category. It is being developed with "Visual Studio Ultimate 2012" with .NET 4.5 as target framework. Persistency (saving and loading of data) is achieved via JSON files.

External Dependencies (so far): [NAudio](#), [AForge.NET](#), [MongoDB BSON Library](#).

Why the name "YtoX"

Well, "X tends to Y" is the usual mathematical paradigm, where X is a trigger and Y is an action. But most of the time, we know what action we require before we sit to determine the conditions that trigger it. So, "Y tends to X" will be more appropriate for our day-to-day tasks isn't?

History

- 22 Oct 2012 - Initial Draft
- 23 Oct 2012 - Experimental work with WinRT APIs
- 24 Oct 2012 - YtoX Phase1 source

Do watch this article for source code and screenshot updates.

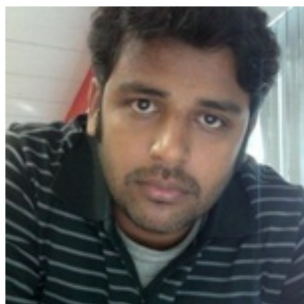
As a personal note to AppInnovation Contest organizers, a fully functional YtoX will be ready by Round 2 deadline 21 Nov 2012. Most of the recipes listed in this article will be available, at the time of release. In fact, the power of YtoX lies in supporting more & more data-types, routines, events and activities. And given the completely customizable nature of YtoX, more and more recipes can be added after the release.

Please use 'Browse Code' option at the left to see the amount of work done so far.

License

This article, along with any associated source code and files, is licensed under [The Code Project Open License \(CPOL\)](#)

About the Author



Prabu Arumugam


Technical Lead Bally Technologies
India Pvt. Ltd.
India 🇮🇳

Hello World!! I am an ardent programmer with passion for computers and gadgets. Developing utilities, games, libraries and custom controls are my favorites. I love algorithms and solving them programmatically is my favorite pastime. I write articles, most of them are in my personal website

<http://www.codeding.com>.

Follow on  Twitter

Comments and Discussions

 Visit <http://www.codeproject.com/Articles/481328/YtoX> to post and view comments on this article, or click [here](#) to get a print view with messages.

[Permalink](#) | [Advertise](#) | [Privacy](#) | [Mobile](#)
Web03 | 2.7.131119.1 | Last Updated 3 Dec 2012

Article Copyright 2012 by Prabu Arumugam
Everything else Copyright © [CodeProject](#), 1999-2013
[Terms of Use](#)