Toronto Police Service

Know Your Neighbourhood (KYN) Application

Implementation Guide: Arcade Expressions

Produced By:

Dale Langford[a], Eric McNeill[a], Eric Millan[b]

[a] GIS Cartographic Specialist, Fleming College
[b] GIS Applications Specialist, Fleming College

Client Contacts:

Debbie Verduga • Senior Crime Analyst, Analytics and Innovation, Toronto Police Service
Tyler Munn • Senior GIS & Data Integration Specialist, Toronto Police Service
Derek Cooper • Crime Analyst, Analytics and Innovation, Toronto Police Service
Gayathri Ganesan • Crime Analyst, Analytics and Innovation, Toronto Police Service

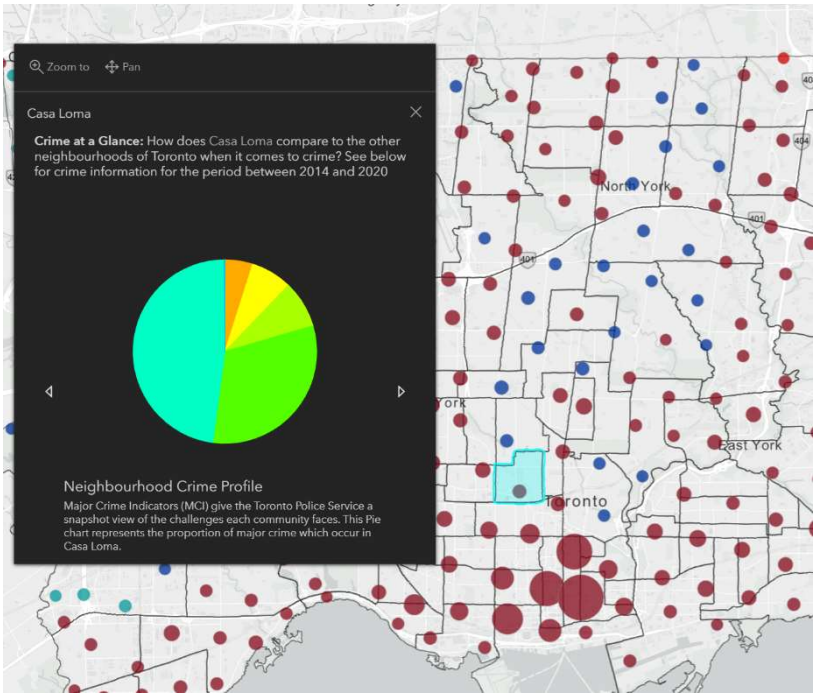
Project Advisor:

Shawn Morgan • Professor and Program Coordinator (GIS Post-Graduate Programs), Fleming College

This guide was prepared by students at Fleming College in collaboration with the Toronto Police Service.

Arcade Expressions – Custom Pop-up Designs

One of the most powerful tools within ArcGIS Online Maps is the ability to customize the information visible to users as they explore the features in a web map. At a basic level, pop-ups are widgets which show attributes relating to the object selected. In the case of an individual crime, for example, we can show the type of crime and the date of occurrence. Arcade, ESRI's in-house expression language, allows for more advanced information to be displayed. This guide will demonstrate how to configure the pop-ups with advanced information about neighbourhood crime. This dataset was prepared in the first chapter of this guide. Open the crime layer as a web map in ArcGIS Online to begin.

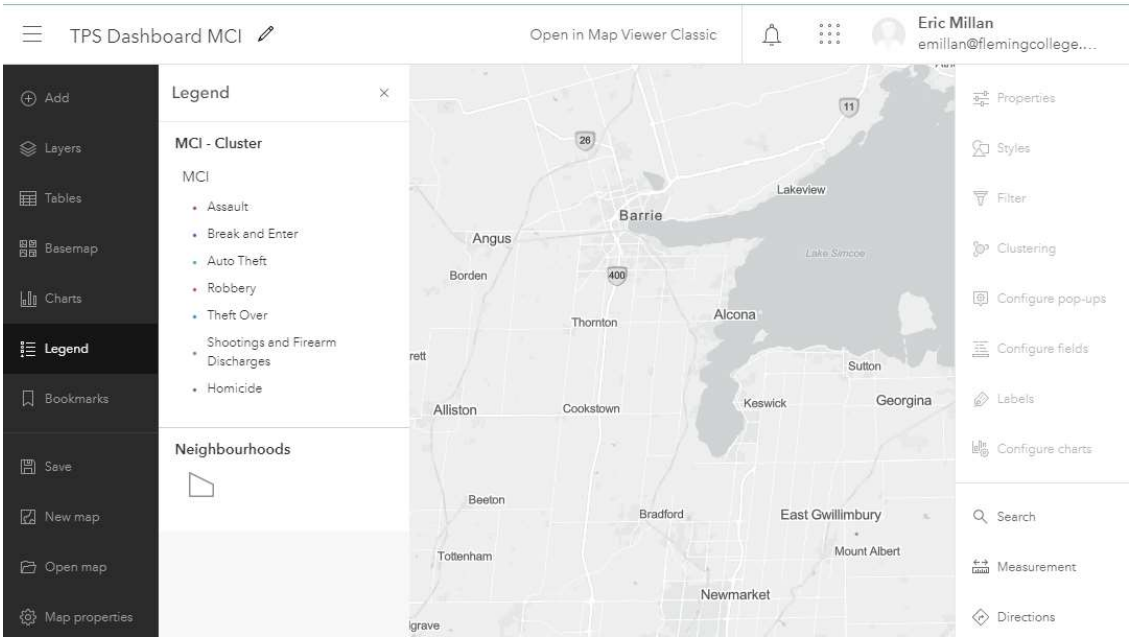


An example of a Neighbourhood Pop-up from the Know Your Neighbourhood Application.

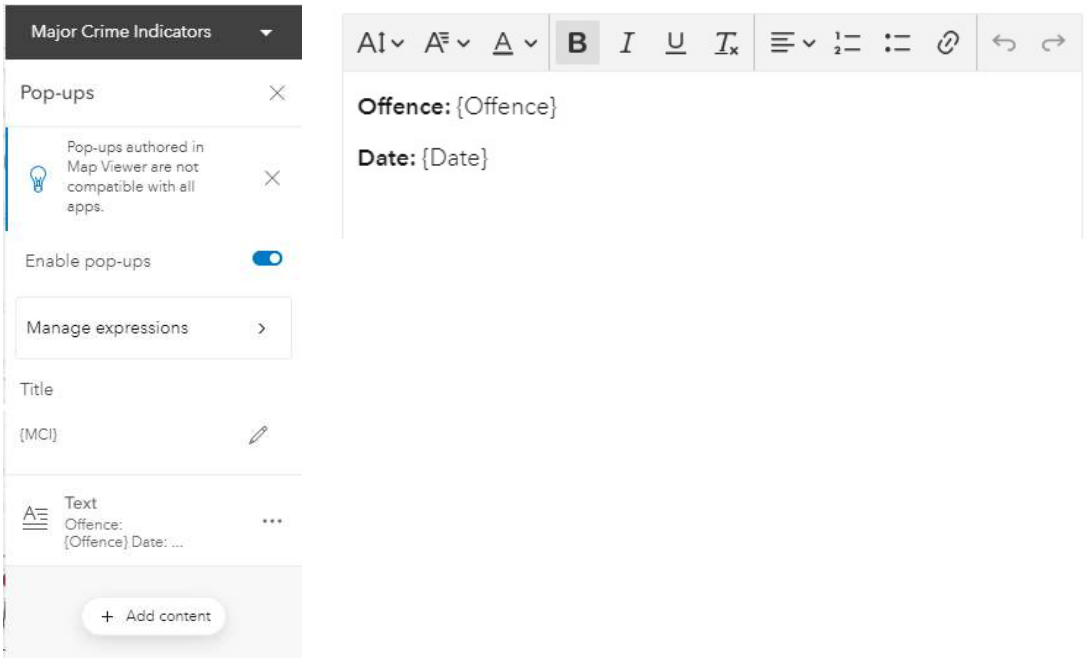
1) **Navigate to the Pop-up Configuration Screen:** Pop-ups are configured within web maps. In the case of the KYN Application, there are 4 web maps which need to be configured and the pop-ups need to be replicated in each one. This guide will demonstrate how to configure a pop-up in a single Map (The TPS Dashboard MCI Map).

   Open the desired map in the map viewer, Select Layers on the left-hand tab and select the Major Crime Indicators Layer. This layer is the point data representing crime events. With this layer selected, click on the Right-Hand Tab “Configure Pop-ups”

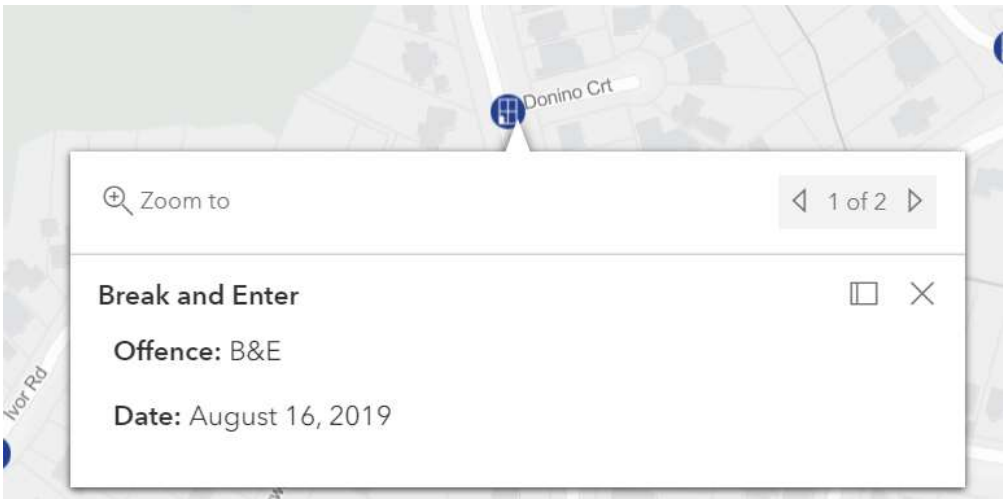
   There are several types of content which can be included in a pop-up window including tables, charts, images and text. The default is usually a table called “Fields List” which shows all the attributes of a given selected feature. While information rich, this data is not very consumable for users. We will be focusing on packaging information into charts which are more accessible.



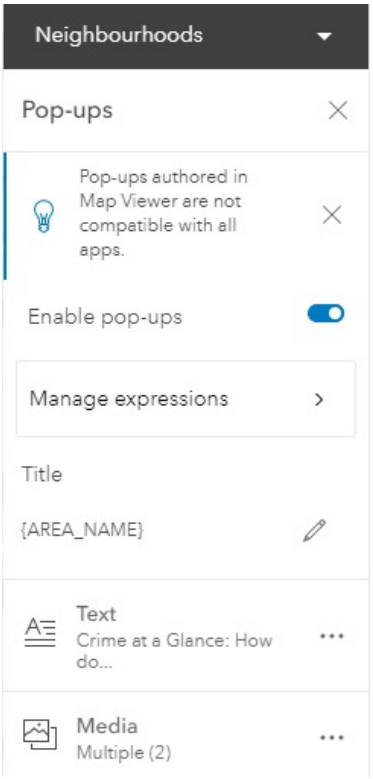
2) **Create Simple Pop-up for Major Crime Indicators:** While point data is important for this application, neighbourhoods are the focus rather than individual crimes. For this reason, most of the attributes have been obfuscated from the data set. For this reason, a simple text based pop-up has been configured. Set the title to {MCI} and select “Add Content” ▯ “Text”, to create a text-based object. In the window which opens, add the text indicated below. Note that whenever “Squirrely” Brackets are used within Arcade, it is a stand in for either a field or expression related to the feature. The Title expression “{MCI}” will return the attribute of the MCI field for the item in question.

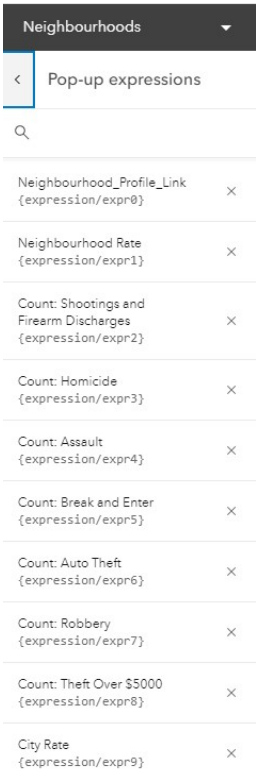


3) **Check Pop-up Content:** Save the pop-up expression, navigate to the map and click on a feature. You should be able to see the results of the code you have written.



4) **Create an Expression:** Now we will move onto the more interesting process of configuring the pop-ups for the Neighbourhood Layer. As before, select the Neighbourhood Layer and then to the Configure Pop-up Tab. Set the Title to {AREA_NAME} and Click on “Manage Expressions.

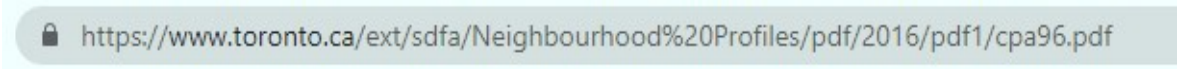




Expressions are code within ArcGIS Online which allows you to perform live calculations within the Webapp. They are made with ESRI’s proprietary coding language, Arcade.

Using Arcade Statements, our first expression will be to create a dynamic link for our neighbourhood layers.

5) **Create a Dynamic Link:** Using Arcade Statements, we can create links to external resources, provided there are normalized naming structures. In the case of neighborhoods, the City of Toronto has published detailed demographic reports known as Neighbourhood Profiles. These web-pdfs show detailed information about the neighbourhoods including population, race, income, language and much more. These topics are outside the scope of our project, but we can easily help users find this information by building a dynamic link.

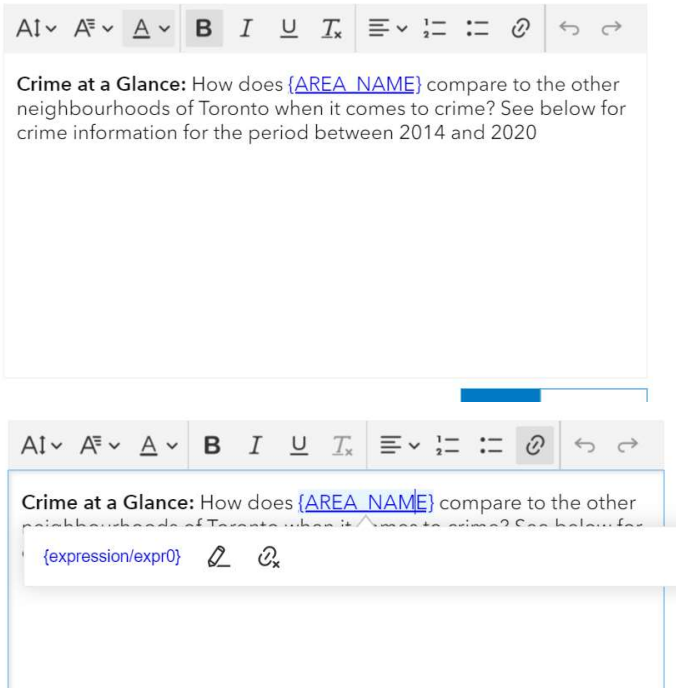


By examining the address of the community profile report for Casa Loma, we note the url ends in “cpa96.pdf”. We know that the neighborhood’s ID number is 96 and that this information is included in our neighbourhood ID dataset. Navigate to Our Expressions, click add expression and copy the address into the window (As our expression needs to return a string value, make sure that it is wrapped in quotations).

Next, delete the number “96” and insert the variable {$Feature.AREA_LONG_CODE}. This arcade expression will return the attribute “Area_Long_Code” for the feature which has been selected. In the case of Casa Loma, this means it will return the number 96.

Upon troubleshooting, it was discovered that the naming scheme was not standardized by neighbourhood numbers alone, but that single digit neighbourhoods only also had a zero as a prefix. To handle this, a simple expression was added which checks to see if the area code is less than 10, and if it is, to add an extra ‘0’ as a string value.

6) **Test Dynamic Link:** Save the expression above and navigate back to the pop-up configuration menu. Add a text feature and when complete, highlight the text which you would like users to click on to navigate to your web resource. In our case, we want users to click on the area name in the text feature to navigate to the neighbourhood profile. Once highlighted, click the chain symbol and insert the arcade expression's name. In our case, it is our first expression and is labelled {expression/expr0}.



Now complete, click on a neighbourhood in your web application, and the subsequent pop-up should have a dynamic link which will take you to the appropriate neighbourhood profile.

This ability to easily navigate to exterior resources is a powerful tool for the Toronto Police Force. As long as web addresses are created with a standardized naming scheme relating to the neighbourhood IDs, many resources could be linked from these pop-ups. This could include information about public bulletins or meetings, contact information for community police offices, or detailed maps published as .pdfs.

7) **Configure Dynamic Graph Expressions: The** are many ways to summarize the data available into useful information. On the main dashboard page, the metrics of “crime change over time” is displayed as both a net number, as well as by a trendline and bar graph. These are displayed based on the viewable extent of the web map. What we want to do in the pop-up window is create interesting graphs which summarize the crime profiles of the neighbourhoods themselves. Because we have population data available for the city, we will build expressions which show crime on a per capita basis in neighbourhoods, as well as for the rest of the city.

   To Begin, we want to build an expression which calculates the amount of crime per capita for the entire city. To do this, create an expression matching the one shown below.

   This expression counts the total number of crimes in the web layer, divides it by 7 (because there are seven years’ worth of data available). It then counts to total population of all the neighbourhoods and divides by seven. This Calculation results in the Number of Major Crimes Per 1000 Residents for the City of Toronto. Because we can see the result and it is not going to change, you can comment out the equation and simply type the rate as “return 12.760”. Once Completed, open another expression window to calculate the rate for just a single neighbourhood.

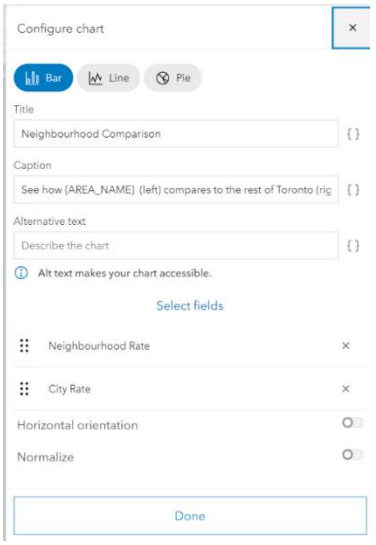
Neighbourhood Rate Edit

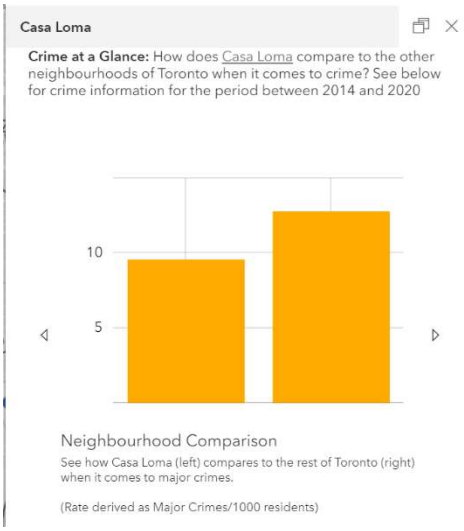
Expression Test

```
1  var allcrimes = FeatureSetByName($map,"MCI - Cluster")
2  var neighbourhoodCrimes = intersects(allcrimes, $feature)
3
4
5  var countCrimes = Count(neighbourhoodCrimes)
6
7  // Divide Number of Crimes in Neighbourhood by Years being studied
8  var yearAverage = countCrimes / 7
9
10 // Get Population in 1000s
11 var pop1000 = ($feature.Population/1000)
12
13
14 // return left(((yearAverage / $feature.Population)*100), 6)
15
16
17 return Round((yearAverage / pop1000), 3)
```

This expression is similar to the previous one for calculating the crime rate for the city, however it makes use of the intersects function in arcade to select only the crimes which occurred in the selected neighbourhood (line 2). Note the ability as well to truncate integer values in line 17. With two expressions built, we can now create our dynamic pop-up graphs.

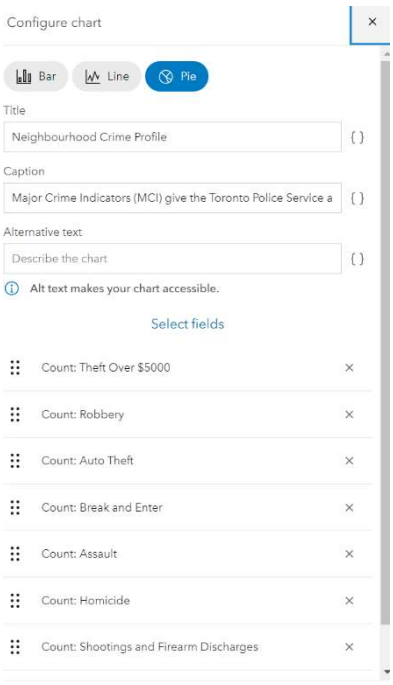
8) **Configure Graph in Pop-Up:** Navigate back to the Configure Pop-up window (for the neighbourhood layer), click on "+ Add Content" and select "Chart", then "Bar". Enter a descriptive title and description for the chart, then click "select fields". This will take you to a list of all the attributes of the feature, as well as the two new expressions just created. Select them in order you want them to appear in the final graph. In this case, we want the neighbourhood crime rate on the left, and the city rate on the right. Click "Done" and check your po-up for your new dynamic graph. Now users have an easy to see view of the amount of crime in their neighbourhood and how it rates to the city average.

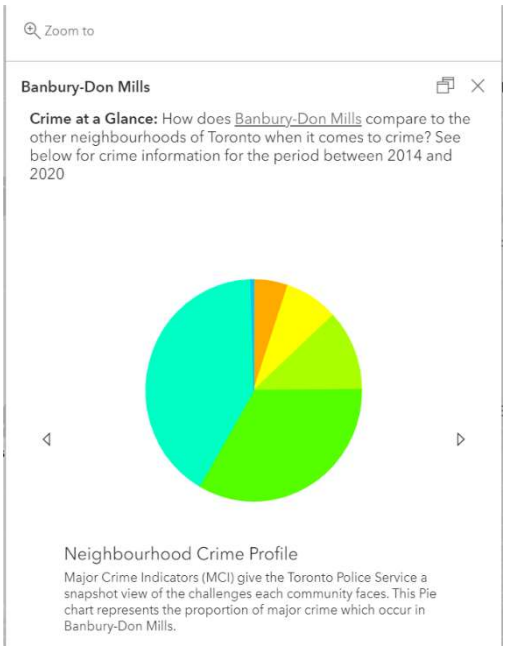




9) **Create Crime Pie Charts:** Repeating the process of creating an expression and adding it into the pop-ups, we will now create counts of the amount of crime which occur in the selected neighbourhood. Navigate to the expression builder, and create an expression as below:

This expression iterates through each of the point features within the selected neighbourhood and counts it as long as the MCI category is “Assault”. Replicate this expression for each of the MCI categories, changing the text in the “for loop” to match the text meant to be counted(eg. for the homicide layer, change the expression to crimetype.mci == “homicide”). When each of these expressions are created, add them to a new pie chart.





Concluding Notes:

The arcade expressions we have built here are basic ways of exploring the datasets and introduced the idea of dynamic graphs which make use of spatial tools, and dynamic links which make use of the information formatted in data preprocessing. It was one of the objectives of the KYN app to allow for the creation of printable reports within the web app. One way to achieve this goal would be to create the types of web maps desired, store them online as .pdfs, and link to those through dynamic links with arcade expressions