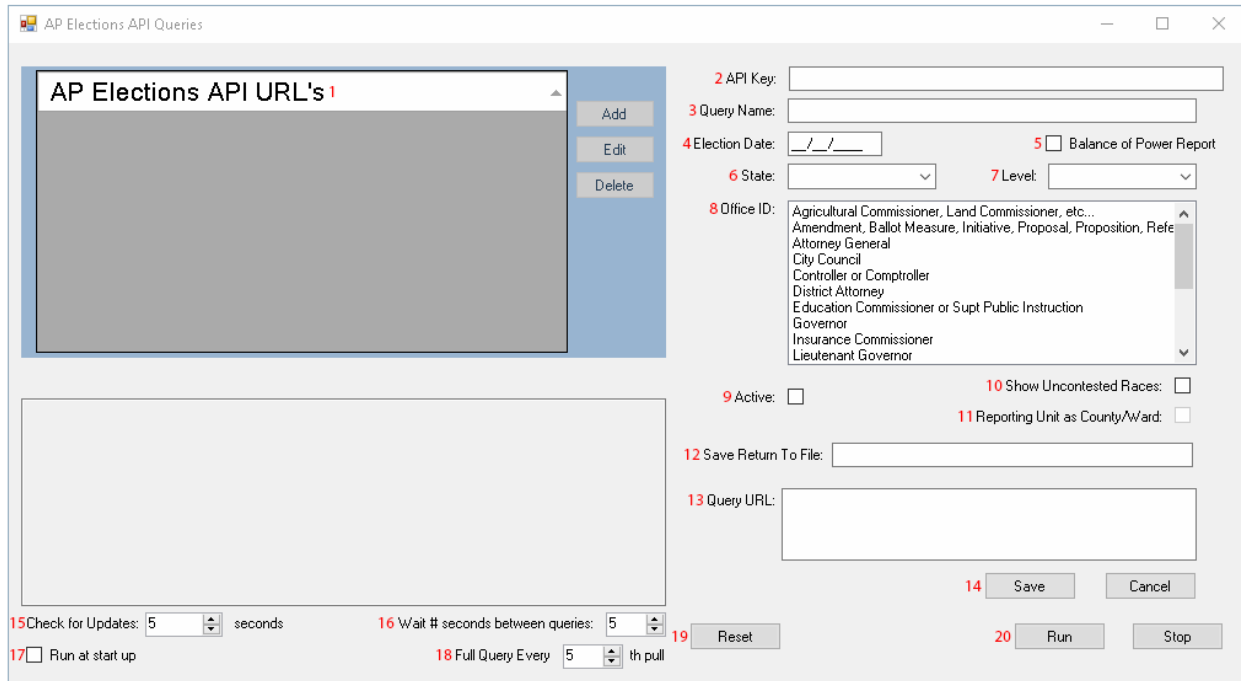




AttendantSolutions™

API User Guide

The AP API is a data source from AP that BTI uses to bring in voting results data. This is a separate application from Attendant Solutions. It is located only on the main BTI server and cannot be accessed from a workstation. Once opened, a window with several settings and options will open.



1. AP Election API URL's

This is where you will see a list of your created queries. It allows you to Add, Edit, or Delete a selected query. Click the Add button to create a query. Once your queries have been created, click Edit to make any changes to them. Any number of queries may be created, however there is a limit to how often you can send a request to AP's servers. This is indicated in your welcome letter from AP and is usually 10x per minute. Given the settings that we recommend later in this guide, we recommend using up to 5 queries at a time.

2. API Key

This is your unique API key received from AP. This will give you access to your state and other state's voting results depending on your AP contract and is usually given out once per year. You can have multiple API keys such as the case for a National key. This key will automatically transfer to each new query when created.

3. Query Name

This is the name you give your query and it can be titled anything that makes sense to the user. This will also be the name of the story containing the races in the Load Races window of Election Voting. Ex. Statewide Results.

4. Election Date

This is the actual date of the election. Do not insert any testing dates. A query must have the correct election date in order to pull in testing data sent from AP.

5. Balance of Power

Place a checkbox here to pull in the Balance of Power for an election. This can only be used for General and Presidential elections. The application will automatically fill in all of the required settings so you will just need to click Save.

6. State

This is the state associated with the API key. The option "National" is used for a National API key.

7. Level

This setting will determine if your data will come into Attendant Solution's Election Voting by race or by county. The options are State (by race) and RU which stands for Reporting Unit (by county). For States that do not report by county by rather by district, District can be selected.

8. Office ID

This is a list of possible offices that an election could cover. These can be individually selected so as to only bring in results for certain offices that a station is interested in. If nothing is selected in the Office ID, it will automatically bring in data for every office that is being covered in the election.

9. Active

This setting is the on and off switch for a query. When it is checked, the query is active and will bring in results. If it is not active, the application will skip over that query and will only run queries that are active. This must be done for each query as there is not an option to make all queries active.

10. Show Uncontested Races

This option will allow the API application to bring in any uncontested races in an

election. It is advised that a separate query be created for uncontested races.

11. Reporting Unit as County/Ward

This setting when checked, will bring in results for each race in the query with a reporting unit for each county that is a part of the race. The results that come in will be by “county” or “ward” and require a county data source in Election Voting in order to link and load races.

12. Save Return To File:

A file path with a file name may be placed here to have the application write the actual response that AP sends back when a query goes out to pull data. This can be used to confirm that you are receiving the proper data.

13. Query URL

This is the URL for the query that is created based on the chosen settings. If highlighted and copied into a browser, it will take the user to the file where the query is pulling its data. It is helpful for testing as well because if the query is not pulling any data, it will usually tell you what is wrong in a message. An example of this would be “Must Supply API Key”. This message would tell the user that their API key has expired or that they are using the wrong API key.

14. Save/Cancel

This is where to save your query after it is created and any changes have been made.

15. Check for Updates

This setting controls how often the API application will go out and search for results. If no change has been made in the election files, it will not bring anything back and will not update. The recommended setting is 30 seconds.

16. Wait # Seconds between queries

This setting controls how long the application will wait before cycling the next active query. The recommended setting is 5 seconds. This means that the application will cycle through the first active query and once it is finished, it will wait 5 seconds before moving to the next active query.

17. Run at Start Up

This setting controls whether the application will start running queries when opened. If it is checked, the application will start cycling through active queries as soon as it is

opened. When not selected, the user must click the [Run] button to start the query cycles.

18. Full Query Every _ th pull

This setting determines when the API will reset and do a full pull for each of your active queries. A Full pull meaning that it will go and pull data back for every race in your query, not just the races that have received updates. Please note that this setting cannot be set to any number lower than 5.

19. Reset

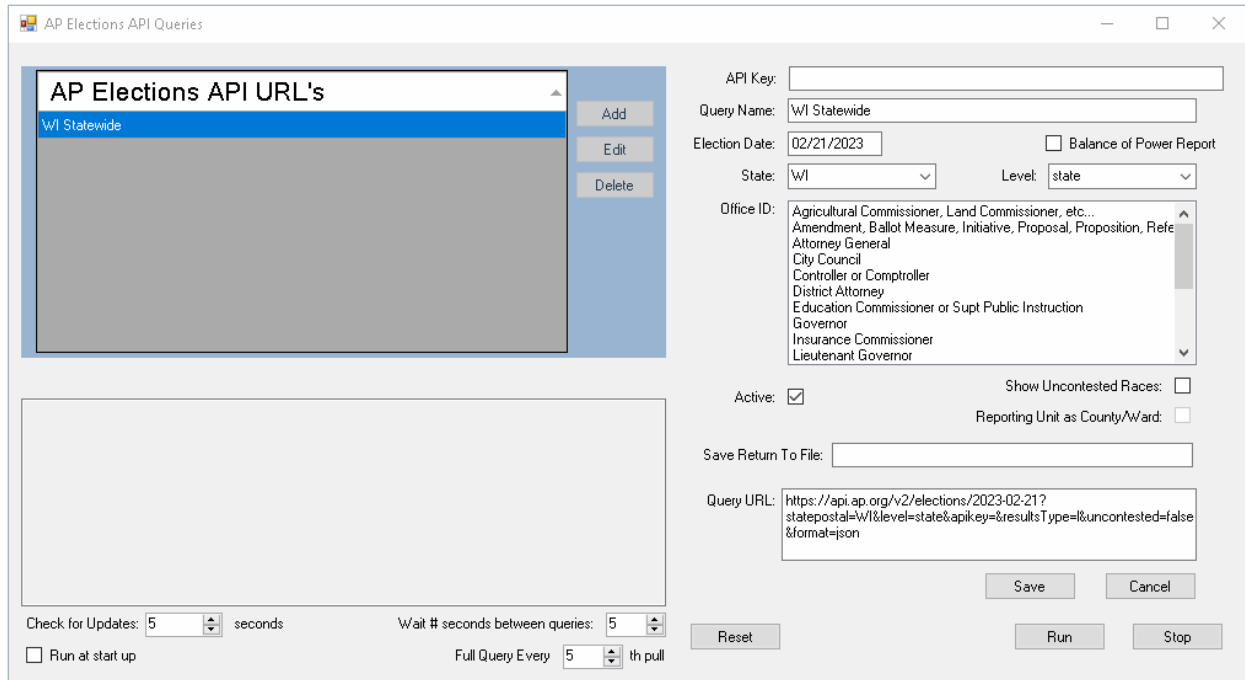
This will reset all of the queries in the API, thereby clearing out all results data that the API has acquired. This reset feature only affects the data in the API and is separate from the reset feature in Attendant Solution's Election Voting. Please note that the API will also reset when the 'Run' button is hit.

20. [Run]/[Stop]

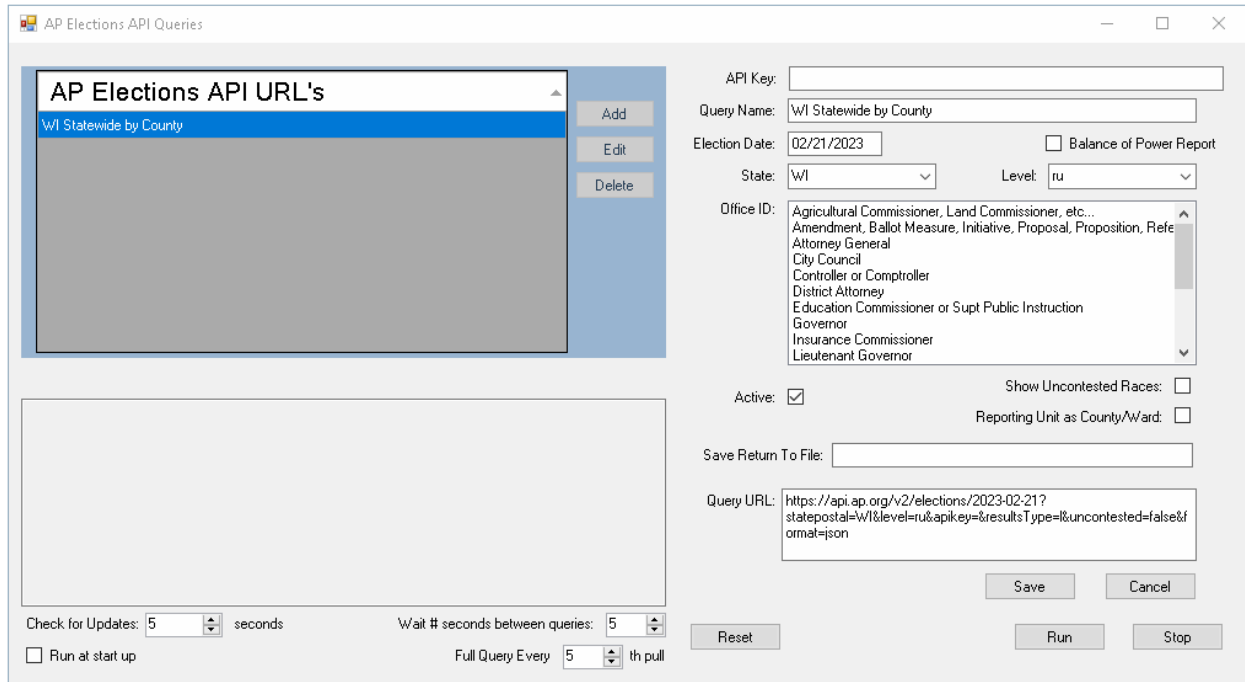
These buttons are used to start and stop the API application's cycling of queries. When the 'Run' button is hit, the API will do a full reset of all queries before starting to cycle through the active queries.

List of the three most commonly used queries in the API.

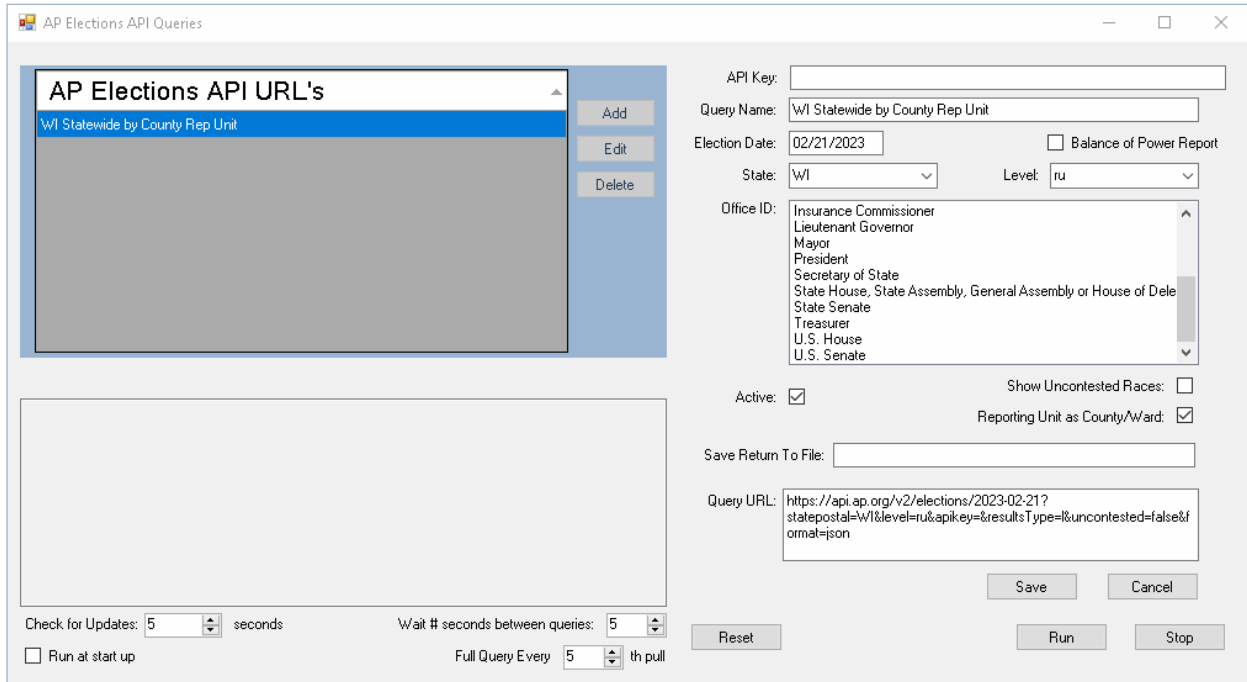
Statewide - This is the most commonly used query and will pull in all possible data by race that is in the election. This query will pull in results on the state level (by Race) and uses the "by race" data source. This query is set up by selecting your state, selecting "state" as the Level, and not choosing anything in the Office ID section (will default to select all offices). Example below.



By County – This query will pull in races on the state level (by race) and will use the “by race” data source. It will break down each race by county but **without a reporting unit** because it is still considered “by race”. It is used to show how each county votes on a race. Since it is still a “by race” query, precincts reported and totals for the counties will not be returned with this query. It is created by selecting your state and choosing “ru” for the Level. With this query it is recommended that the user select which offices they want to pull in results for, and to spread multiple offices over multiple queries because of the large amount of data that can come in. Example below.



By County With Reporting Unit – This query will pull in races on the county level **with a reporting unit**. The data will contain a race’s precincts reported and totals for each county. It is created by selecting your state, choosing “ru” for the Level, and selecting the “Reporting Unit as County/Ward” option. Like the previously discussed query, it is recommended that the user select which offices to pull in results for because of the large amount of data that can come in. Example below.



Once you have created your query, you may have to create a new data source in Election Voting to access the data and be able to link/load to the races. Click the [New] button in the Data Sources tab. Select “AP API Service AI” from the Source Type drop down list. In the description, create a name for the data source (Ex. AP-API Service by Race CA). If the created data source will be by race, then leave “US United States” in the State option and “Race” in the Reports By option. If the data source is to be by county, then change the State option to the state that will be used in the data source and change the Reports By option to “State”. Click the [Save] button to save the data source.

API Source by Race:

AP-API Service by Race | Applicable Races | Latest Votes | Update Log

- AP FTP Service
- AP FTP Top Races Only
- AP-API Service
- AP-API Service by Race**
- County Manual Data Entry
- Manual Data Entry
- MN SOS Election Bridge



New Delete Show All

14 Sources, 7 Enabled

Editing 'AP-API Service by Race'

Source Type

Description

State

Reports By

- Set Race Winner if Declared by this Source
- Replace the Precincts Override with the Total Precincts provided
- Increment Vote Counts for Testing this Data Source

Notes **Enabled**

Disable Source

All 2 Races Active

Load Races from **or from**

A Data Source provides Race Results for the race as a whole or based on a Reporting Unit such as County, Ward or Parish. In order to match the incoming data, each Reporting Unit must be identified. If Results are provided by County, then the highest number of votes for a County from any Source is used to accumulate the Total Votes for the Race. This Race Total is then compared against any Data Sources that are providing Results 'By Race' to determine the latest Vote Totals for the Race.

API Source by County:

Maintain Election Details - Data Sources

AP-API Service by County | Applicable Races | Latest Votes | Reporting Units | Update Log

AP FTP Service
AP FTP Top Races Only
AP-API Service
AP-API Service by County
AP-API Service by Race
County Manual Data Entry
Manual Data Entry
MN SOS Election Bridge

Editing 'AP-API Service by County'

Source Type: AP-API Service Al
Description: AP-API Service by County
State: WI Wisconsin
Reports By: County

Set Race Winner if Declared by this Source
 Replace the Precincts Override with the Total Precincts provided
 Increment Vote Counts for Testing this Data Source

Notes: **Enabled**

Disable Source

Save Cancel

All 2 Races Active
Load Races from Data **or from** Spreadsheet
72 County Units, with 0 Precincts and 0 Registered Voters

15 Sources, 8 Enabled

New Delete Show All

A Data Source provides Race Results for the race as a whole or based on a Reporting Unit such as County, Ward or Parish. In order to match the incoming data, each Reporting Unit must be identified. If Results are provided by County, then the highest number of votes for a County from any Source is used to accumulate the Total Votes for the Race. This Race Total is then compared against any Data Sources that are providing Results 'By Race' to determine the latest Vote Totals for the Race.

Close

Once you have created your AP API data source, click the Load Races from [Data] button. The Load races window appears. Click the [Search] button. This will refresh the window and you will see a list of story references. Select the story reference that has the name of the active query you created in the AP API application. From here you link/load your races as you would with previous data sources.