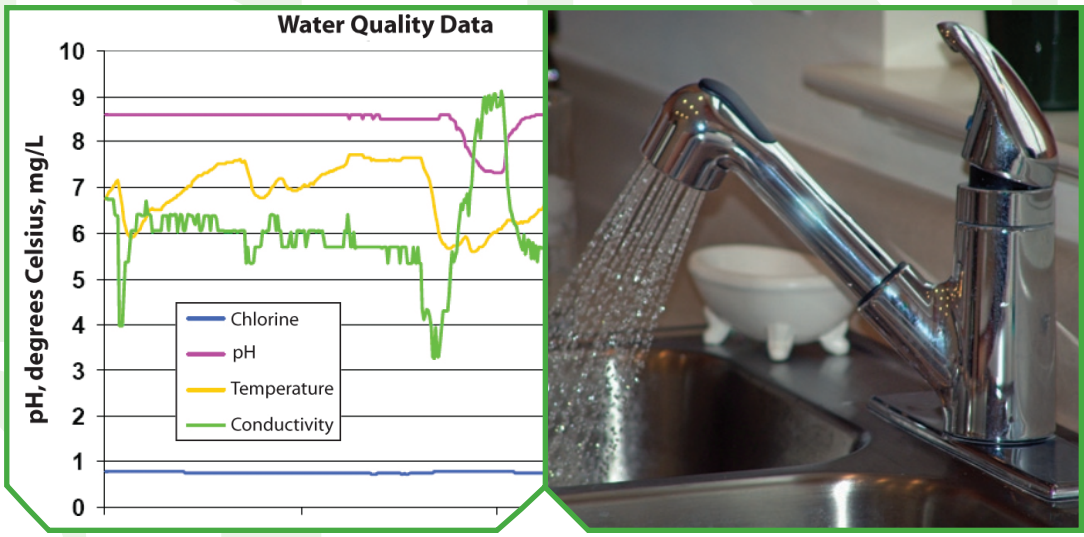
EPA 600/R-08/040A | March 2009 | www.epa.gov/ord

United States
Environmental Protection
Agency

CANARY User's Manual

VERSION 4.1



Office of Research and Development
National Homeland Security Research Center

EPA 600/R-08/040A | March 2009 | www.epa.gov/ord

CANARY User's Manual Version 4.1

by
D.B. Hart and S.A. McKenna
National Security Applications Dept.
Sandia National Laboratories
Albuquerque, NM 87185-0735

CANARY
D.B. Hart, K.A. Klise, S.A. McKenna, M.P. Wilson
Sandia National Laboratories
Albuquerque, NM 87185-0735

Project Officer:
Regan Murray
National Homeland Security Research Center
Office of Research and Development
U.S. Environmental Protection Agency
Cincinnati, OH 45256

March 2009 Revision

Office of Research and Development
National Homeland Security Research Center, Water Infrastructure Protection Division

Preamble

The U.S. Environmental Protection Agency (EPA) through its Office of Research and Development funded and collaborated in the research described here under an Inter-Agency Agreement with the Department of Energy's Sandia National Laboratories (IAG # DW8992192801). This document has been subjected to the Agency's review, and has been approved for publication as an EPA document. EPA does not endorse the purchase or sale of any commercial products or services.

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under Contract DE-AC04-94-AL85000.

Questions concerning this document or its application should be addressed to:

Regan Murray
USEPA/NHSRC (NG 16)
26 W Martin Luther King Drive
Cincinnati OH 45268
(513) 569-7031
Murray.Regan@epa.gov

Forward

In July of 1970, the White House and Congress worked together to establish the United States Environmental Protection Agency (EPA) in response to the growing public demand for cleaner water, air, and land. The Agency was assigned the daunting task of repairing the damage already done to the natural environment and establishing new criteria to guide Americans in making a cleaner environment a reality. Since 1970, EPA has worked with federal, state, tribal, and local partners to advance its mission to protect human health and the environment.

EPA leads the nation's environmental science, research, education and assessment efforts. With more than 17,000 employees across the country, EPA works to research, develop and enforce regulations that implement environmental laws enacted by Congress. In recent years, between 40 and 50 percent of EPA's enacted budgets have provided direct support through grants to State environmental programs. At laboratories located throughout the nation, the Agency works to assess environmental conditions and to identify, understand, and solve current and future environmental problems. The Agency works through its headquarters and regional offices with over 10,000 industries, businesses, nonprofit organizations, and state and local governments, on over 40 voluntary pollution prevention programs and energy conservation efforts.

Under existing laws and recent Homeland Security Presidential Directives, EPA has been called upon to play a vital role in helping to secure the nation against foreign and domestic enemies. The National Homeland Security Research Center (NHSRC) was formed in 2002 to conduct research in support of EPA's role in homeland security. NHSRC research efforts focus on five areas: water infrastructure protection, threat and consequence assessment, decontamination and consequence management, response capability enhancement, and homeland security technology testing and evaluation. EPA is the lead federal agency for drinking water and wastewater systems and the NHSRC is working to reduce system vulnerabilities, prevent and prepare for terrorist attacks, minimize public health impacts and infrastructure damage, and enhance recovery efforts.

This User's Manual for the CANARY software package is published and made available by EPA's Office of Research and Development to assist the water community by improving the security of our Nation's drinking water.

Jonathan Herrmann, Director

National Homeland Security Research Center
Office of Research and Development
U. S. Environmental Protection Agency

Acknowledgements

The national Homeland Security Research Center would like to acknowledge the following organizations and individuals for their support in the development of the CANARY User's Manual and/or in the development and testing of the CANARY Software.

Office of Research and Development – National Homeland Security Research Center

John Hall
Robert Janke
Regan Murray
Terra Haxton

Office of Water – Water Security Division

Steve Allgeier
Mike Henrie
Katie Umberg

Sandia National Laboratories

David Hart
William Hart
Katherine Klise
Shawn Martin
Sean McKenna
Eric Vugrin
Mark Wilson
Laura Cutler
Marguerite Sorensen
Mark Wunsch

American Water Works Association Utility Users Group

Kevin Morley (AWWA)
David Hartman (Greater Cincinnati Water Works)
Yeongho Lee (Greater Cincinnati Water Works)
Dan Quintanar (Tucson Water)
Zia Bukhari (New Jersey American Water)

Shaw Group

Srinivas Panguluri (Working at EPA T&E Facility)

Contents

Table of Figures

List of Tables

1 Introduction

Contamination warning systems (CWSs) have been proposed as a promising approach for reducing the risks associated with contamination of drinking water. In order to maximize detection likelihoods, CWSs can incorporate multiple detection technologies, such as online continuous water-quality monitoring, public health surveillance, physical security monitoring, and customer complaint surveillance. The goal of a CWS is to detect contamination incidents in drinking water systems rapidly enough to allow for the effective mitigation of adverse public health and economic impacts. In 2006-2010, the U.S. Environmental Protection Agency (EPA) is deploying and evaluating CWSs at a series of drinking water utilities.

With current technology, the online monitoring component of a CWS is based on available water quality sensors that measure, for example, free chlorine, total organic carbon, electrical conductivity, oxidation-reduction potential, and pH. Recent research has shown that many contaminants of concern will cause detectable changes in these water quality parameters (Hall et al., 2007), (US EPA, 2005). However, these parameters are known to vary considerably over time in water distribution systems due to normal changes in the operations of tanks, pumps, and valves, and daily and seasonal changes in the source and finished water quality, as well as fluctuations in demands.

Data analysis tools are needed to distinguish between normal variations in water quality and changes in water quality triggered by the presence of contaminants. Often referred to as event detection systems (EDS), such data analysis tools can read in SCADA data (water quality signals, operations data, etc.), perform an analysis in near real-time, and then return the probability of a water quality event occurring at the current time step. A water quality event is defined as the period in time within which water of unexpected characteristics occurs. The CANARY software described here provides a continuous measure of the probability of an event to a water utility operator.

The goal of CANARY is to take standard water-quality data and use statistical and mathematical algorithms to identify the onset of periods of anomalous water quality, while at the same time, limiting the number of false alarms that occur. The working definition of “anomalous” can be set by the user by selecting the configuration parameters. These parameters may be configured differently from one utility to the next and may even need to vary across monitoring stations within a single utility. CANARY can be set up to receive data from a SCADA database, and return alarms to the SCADA system. In addition, it can be run on historical data to help set the configuration parameters in order to provide the desired balance between event detection sensitivity and false alarm rates.

CANARY is designed to be extensible, allowing outside researchers to develop new algorithms that can added to CANARY. In this version, there are several change detection algorithms within CANARY, including: a linear filter, a multivariate nearest-neighbor algorithm, and a set-point proximity algorithm. These algorithms identify a background water quality signature for each water quality sensor and compare each new water quality measurement to that background to determine if the new measurement is an outlier (anomalous) or not.

The definition of the water quality background is updated continuously as new data become available. A binomial event discriminator (BED) examines multiple outliers within a prescribed time window to determine the onset of either an anomalous event or a change in the water quality baseline. Information on the development of the three event detection algorithms can be found in: (McKenna, Klise & Wilson, 2006) (time series increments and linear filter), (Klise & McKenna, 2006a), (Klise & McKenna, 2006b) (multivariate nearest neighbor), and (Hart et al., 2007), (McKenna et al., 2007), (binomial event discriminator).

A new addition to CANARY is a water quality pattern matching capability. Changes in utility operations often create changes in the water quality within the distribution network and these changes can produce false alarms in EDS tools. In many cases, these water quality changes occur on a daily basis, although they do not occur at exactly the same time or produce exactly the pattern from one day to the next. Pattern matching in CANARY is designed to work with historical data from a monitoring station and identify recurring patterns in those data. CANARY allows the user to select the water quality changes that should be included in a pattern library. The stored patterns are multivariate including all water quality monitoring signals and a clustering algorithm is used to summarize the various changes in water quality with a reduced parameter set. The changes in the various water quality signals are conceptualized as a series of points linked in time that define a trajectory and a trajectory clustering approach (Gaffney, 2004) employing fuzzy c-means clustering (Dunn, 1973) and (Bezdek, 1981) is used within CANARY. Identification of recurring patterns in data sets can significantly reduce the number of false alarms.

To be useful to water utilities, event detection systems must have low false positive rates, high likelihood of detecting true events, and be sufficiently reliable. The CANARY software is being released to the public in order to promote widespread development and testing of event detection algorithms among researchers, consultants, vendors, and utilities. In this way, water utilities will be provided with high performing tools that can be trusted as part of their daily operations.

The rest of this manual is organized as follows:

- Quick Start – the installation process and startup guide.
- Design – the design section discusses modes of operation and the different algorithms that can be used in event detection.
- Configuration – this section describes how to configure CANARY for a specific need.
- Inputs and Outputs – these chapters describe how data should be formatted for use by CANARY, and what kind of information will be provided in return.
- Appendices – licenses, checklists, references and other miscellaneous information.

A tutorial is available online at the download site: http://www.epa.gov/nhsrc/water/teva.html

2 Quick Start

This chapter presents a step-by-step guide to installing and running CANARY on a PC running Microsoft® Windows™ operating systems.

2.1 Installation

The CANARY installer is available as an executable setup. For information on how to obtain the installer, visit the website http://www.epa.gov/nhsrc/water/teva.html. Documentation in the downloadable files also contains links to request access to the bug-reporting facility and tutorials, documents, and other information.

To install CANARY for the first time, download the "setup.exe" file. To update a previous version of CANARY, download the "update.exe" file from the website. The following instructions apply to both methods.

1. Double-click the "setup.exe" or "update.exe" file that was just saved to disk. A dialog box will pop up. Click "Next" (see Figure 1).
   The following screen will present the license agreement. You must accept the license agreement in order to install and run CANARY

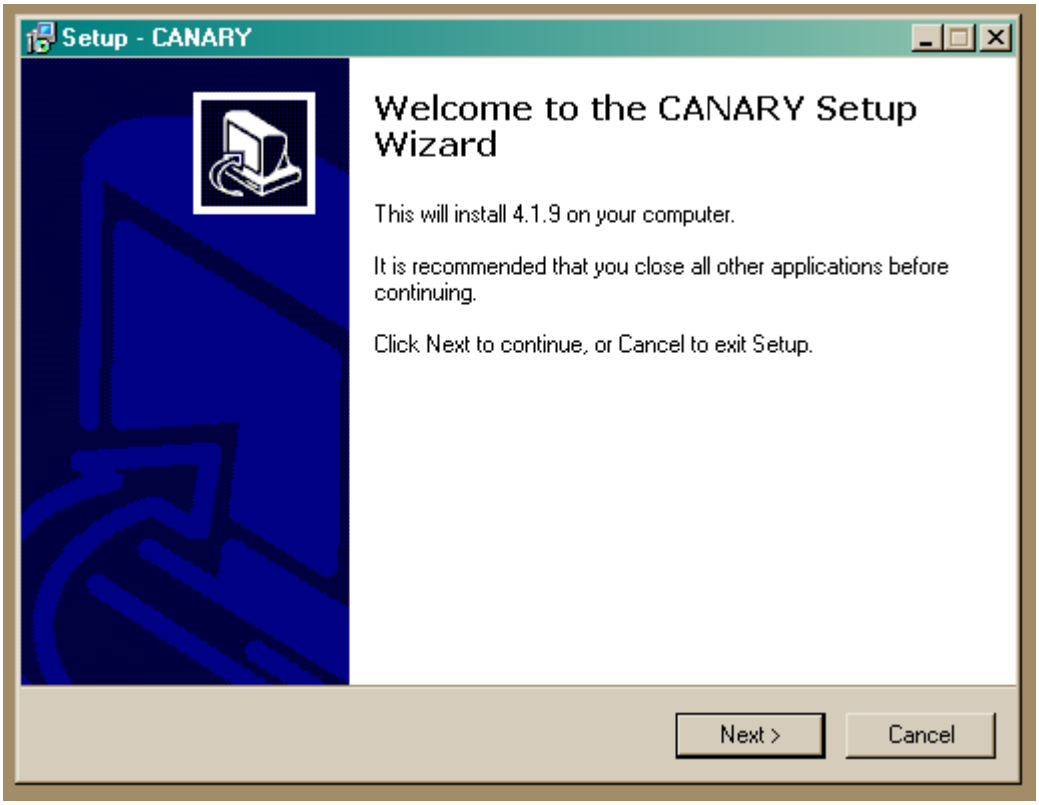


Figure 1 - Installation startup dialog.

2. Choose where you would like to install CANARY (see Figure 2)

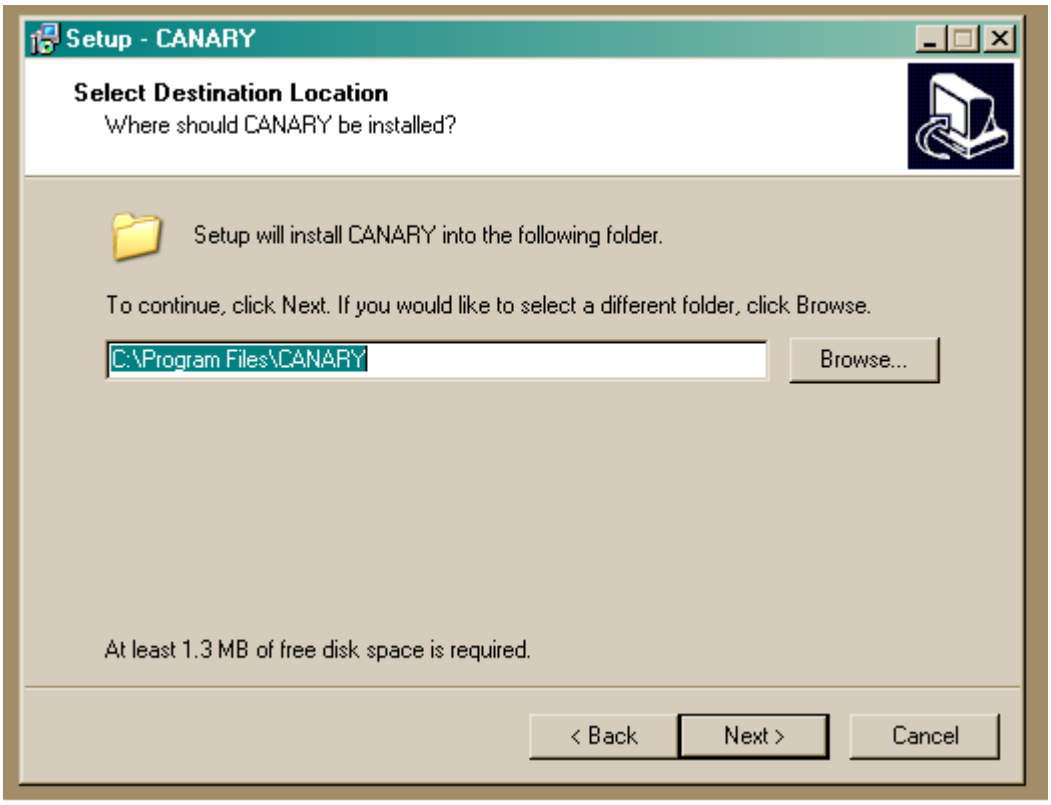


Figure 2 - Choose the installation location.

3. Choose where you would like the Start Menu program group (see Figure 3)

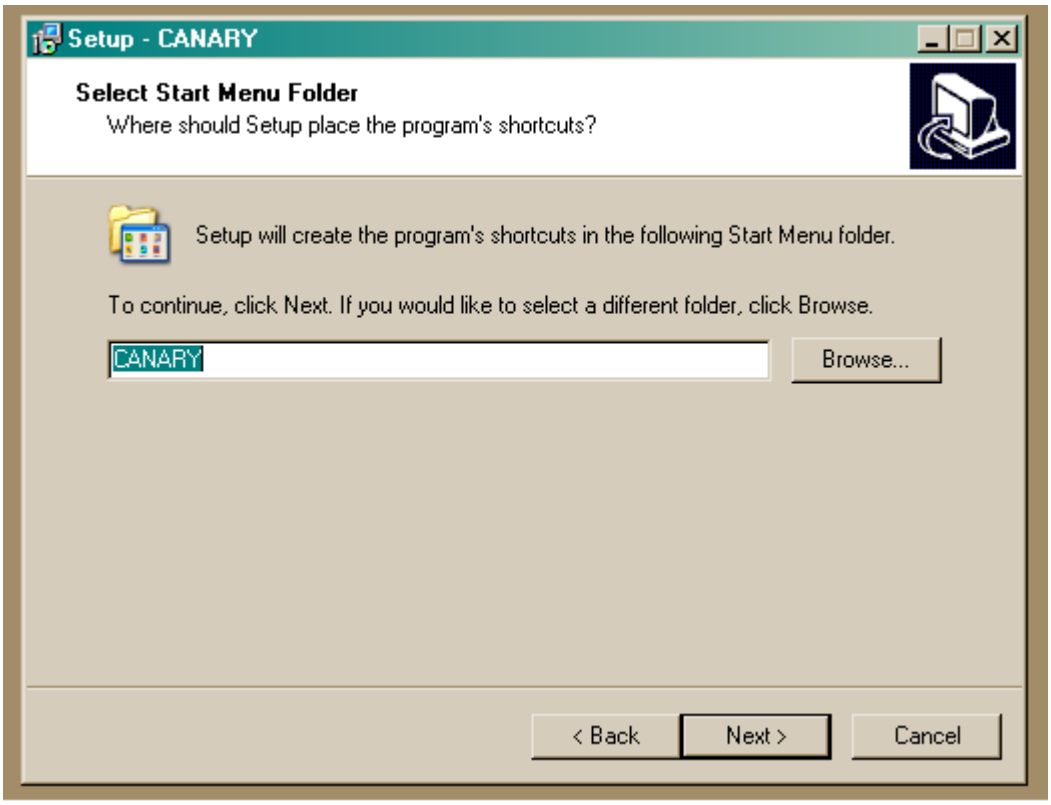


Figure 3 - Select start menu folder dialog box.

4. The next question will verify the information is correct (see Figure 4). If the "MATLAB Runtime Component Library" has not been installed, it will be installed next, otherwise, the progress bar will be shown, such as in Figure 5.

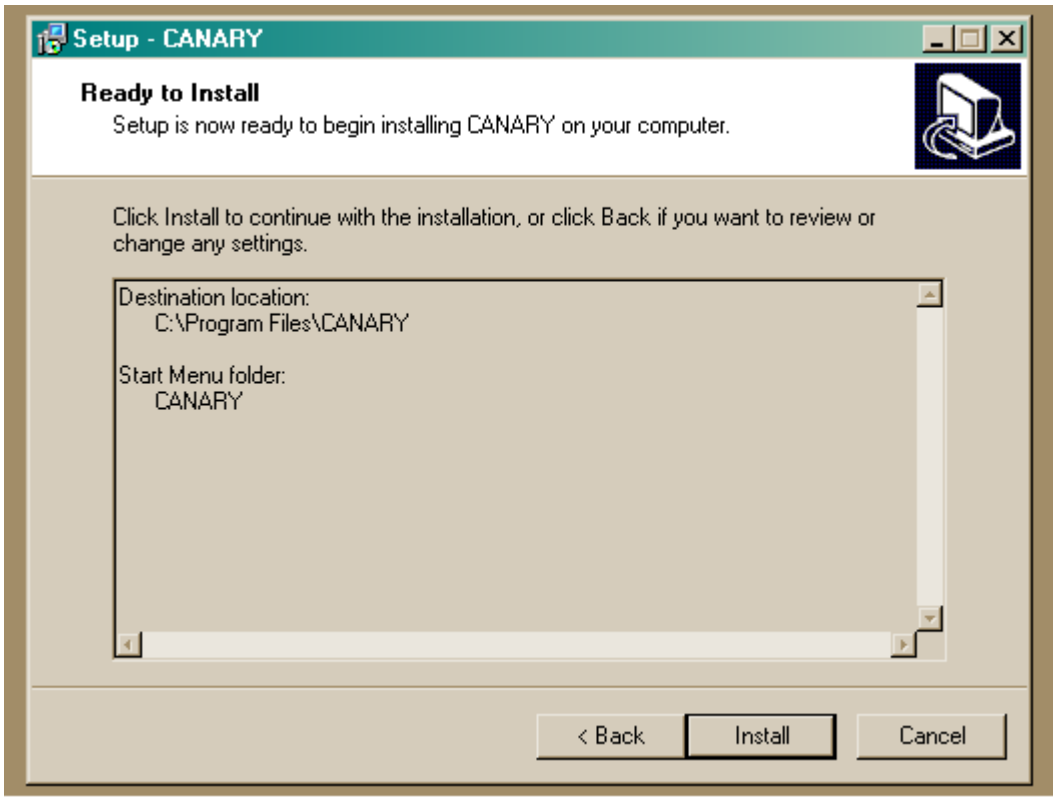


Figure 4 - Installation options verification page.

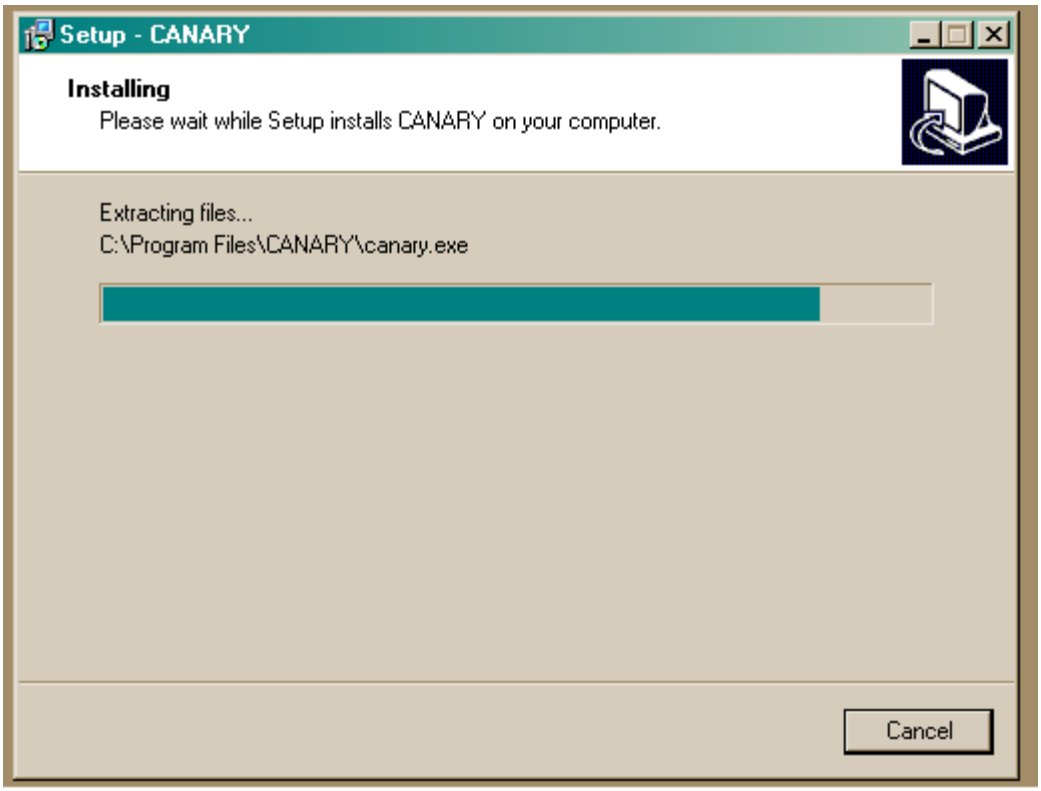


Figure 5 - Installation progress dialog.

5. Everything is now complete!

2.1.1 Files and Associations

There are four types of files that will become associated with CANARY after installation:

- *.EDSX – This file extension indicates that it is a configuration and settings file for CANARY. Double click this type of file to run CANARY, or right-click and choose "Edit" to open the configuration editor (see Chapter 6).
- *.EDSD – This is a CANARY data file (output). This type of file will be graphed when double-clicked.
- *.EDSC – This is a CANARY clustering-pattern library file.
- *.EDSL – This is a log file from CANARY; it will open in "Word Pad" if double-clicked.

2.2 Running CANARY

Once CANARY is installed, it can be run from the "Start" menu, by going to "All Programs" → "CANARY" → "CANARY". This will bring up a window that will look like Figure 6.

To run CANARY, simply open one of the configuration files (*.EDSX; see Chapter 6).

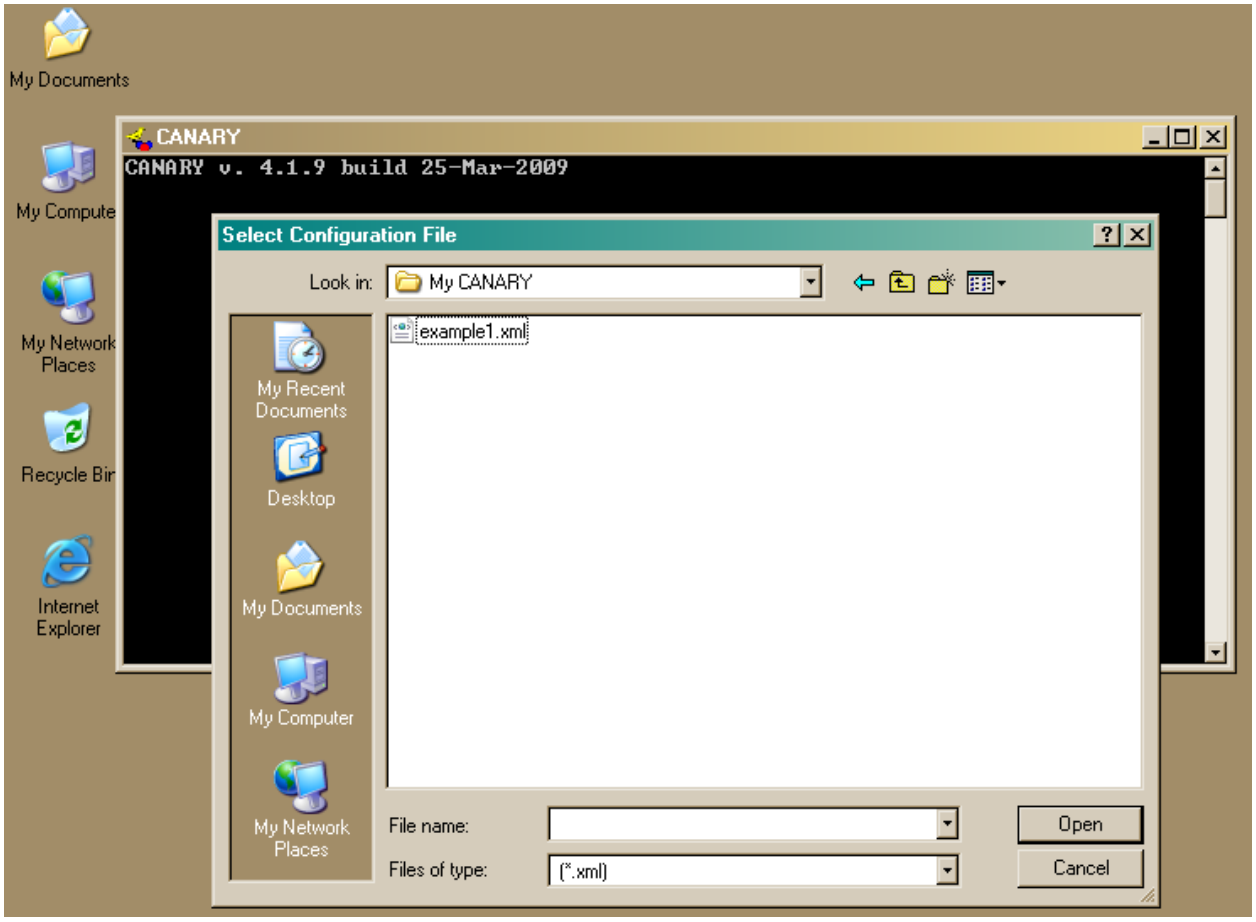


Figure 6 - Configuration file selection after running CANARY from the "Start" menu.

There are two other options in the programs menu that may be of significant use: "Edit Configuration" and "Graph Outputs". The configuration editor will be discussed in detail in Chapter 6. The graph tool will create graphics files in the PNG format that can be used to visualize the water quality data and discovered events in off-line mode (see 7.1). Both the “Edit Configuration” and “Graph Output” selections will open separate programs that serve as pre- and post-processors to CANARY, respectively.

3 Design

CANARY moves one step at a time through a data file, or through data provided from an online connection to a SCADA system (Figure 7) , and determines whether or not the observed data at the current time step is consistent with the expected data values. This determination is made by classification of the residual – the difference between the observed and expected data values – into either the normal (background) class or the abnormal (outlier) class. The expected values are derived from a choice of one or more algorithms that use linear combinations of previously observed data values to estimate the next observed value. By examining the results of the residual classification over multiple consecutive time steps, the probability of a water quality event occurring at the current time step is calculated. If the probability of an event exceeds a user-defined probability threshold, CANARY declares that an event is occurring. Changes in water quality that occur with some regularity, such as those caused by daily changes in utility operations, can be recorded and stored in a pattern library for future reference.

CANARY has been designed for two different modes of operation: on-line and off-line. The off-line mode uses historical data to analyze algorithm performance given different parameter settings. On-line mode uses real-time data, typically through connection to a SCADA database, to do on-line analysis of water quality in a system. While the algorithms operate identically in both cases, CANARY's outer control structures are significantly different. Diagrams showing how CANARY is organized are shown in Figure 8 and Figure 9.

Parameters, controls, and input and output options for CANARY are specified in a configuration or "config" file. The config file is written in Extensible Markup Language (XML) and different tag names are used to divide the file into relevant sections. A detailed description of the configuration file is presented in Chapter 6 of this User's Manual.

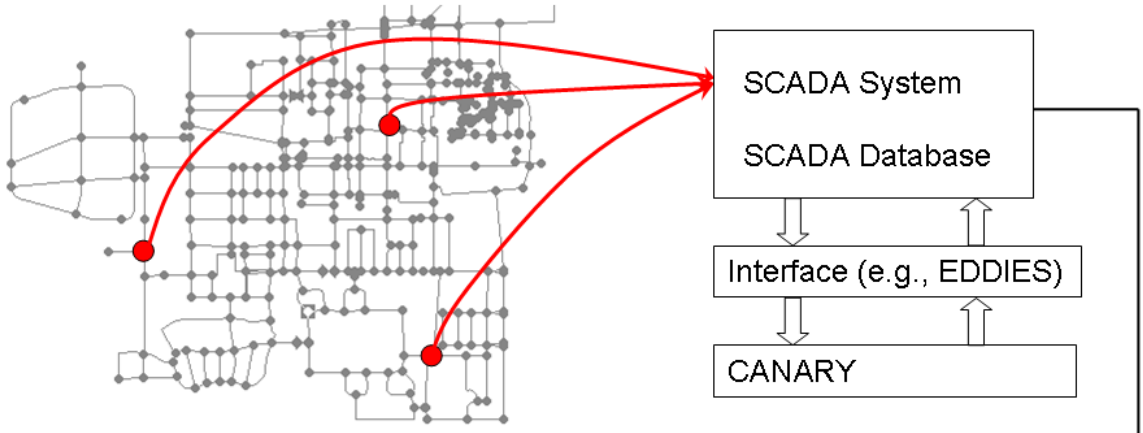


Figure 7 - CANARY's interaction with a SCADA system.

3.1 Off-line Mode

CANARY was originally designed to test the feasibility of using different algorithms to detect anomalous events on large sets of historical water quality data. These data sets were generally presented in the form of a spreadsheet or text file, but occasionally in simple databases. Off-line mode reads in the entire data set and then processes it one step at a time, as if the data were coming into the system sequentially. It does this without any delay between time steps and tracks the time interval that would have occurred internally. Results are presented on the screen as a series of text messages as they are calculated, and they are also saved in output files. The boxes highlighted in blue in Figure 8 indicate the quantities written to output files and saved in offline mode.

Off-line mode also can calculate performance metrics of different algorithms, if “real” event information is provided with the historical data. This event data can be real world tracer tests performed in the field, data from laboratory studies, or simulated events added into background water quality. If records are available, water main breaks, treatment plant changes, or other occurrences that may affect water quality can be added in and used as events.

There are three main motivations to use CANARY in off-line mode:

1. Configure algorithm parameters for a monitoring station based on historical data that does not contain any known events. This mode is used to identify the algorithm parameters that will both create the best estimations of the observed water quality values (minimization of the residuals between estimated and observed water quality values) and reduce the false positive event alarms. Currently, this is the most common off-line use of CANARY as it is necessary to define the correct parameter set for each monitoring station and most historical data sets do not contain known events.
2. Set algorithm parameters for a monitoring station based on historical data that contains known events. In order to determine the ability of the algorithm parameters to minimize the number of missed detection (false negative alarms), it is necessary to have some known events in the historical data set. Few historical data sets contain known events, and if they do there are not enough of them to provide the statistical number necessary for setting algorithm parameters. In most cases, the known events must be simulated and added to the observed water quality data.
3. Identify recurring water quality patterns. CANARY contains a pattern recognition approach to construct a pattern library from historical data. The patterns represent changes in water quality that could trigger an alarm under normal operation of CANARY. Once stored in the library in off-line mode, these patterns can then be compared to any observed water quality that may trigger an alarm during on-line operation.

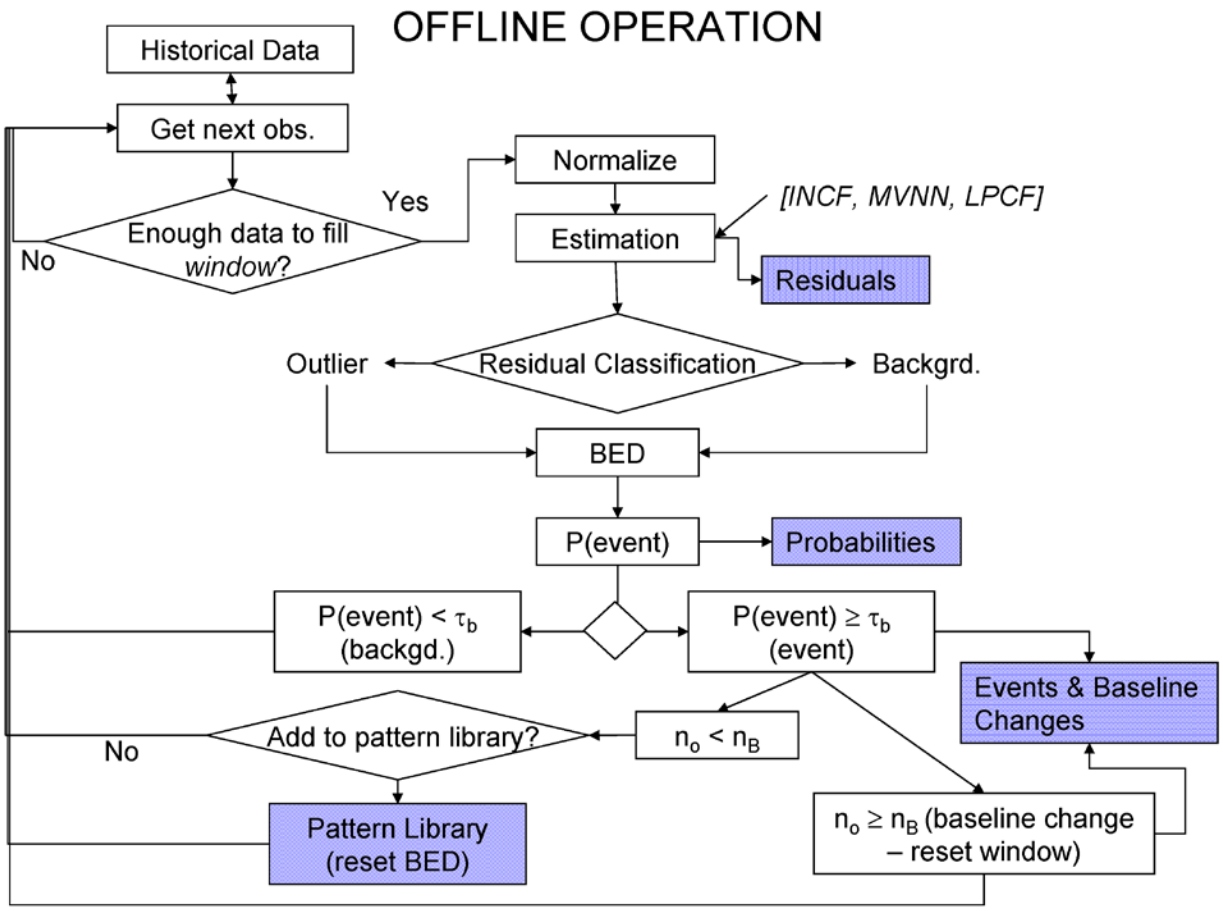


Figure 8 - CANARY's off-line operation.

3.2 On-line Mode

While event detection on historical data is useful for research and development, water utilities need to be able to detect water quality changes in real time. CANARY's on-line mode provides this functionality. Figure 8 demonstrates the process flow for running CANARY in online mode, while Figure 7 shows how CANARY fits into an overall contaminant warning system. Multiple stations can be monitored simultaneously, and results can be output both to the screen and back to a SCADA system. Because connecting an external program to a SCADA system can be challenging, there are checklists for a utility's SCADA manager or contractor in Chapter 8. One option for connecting CANARY to SCADA is the EDDIES software, which is being developed by CSC and CH2M Hill for the US EPA Water Security (WS) Initiative pilot program. CANARY has direct interface capability with the EDDIES database for use in WS pilots, and the details are discussed in the EDDIES.ReadMe file included in the distribution of the CANARY software.

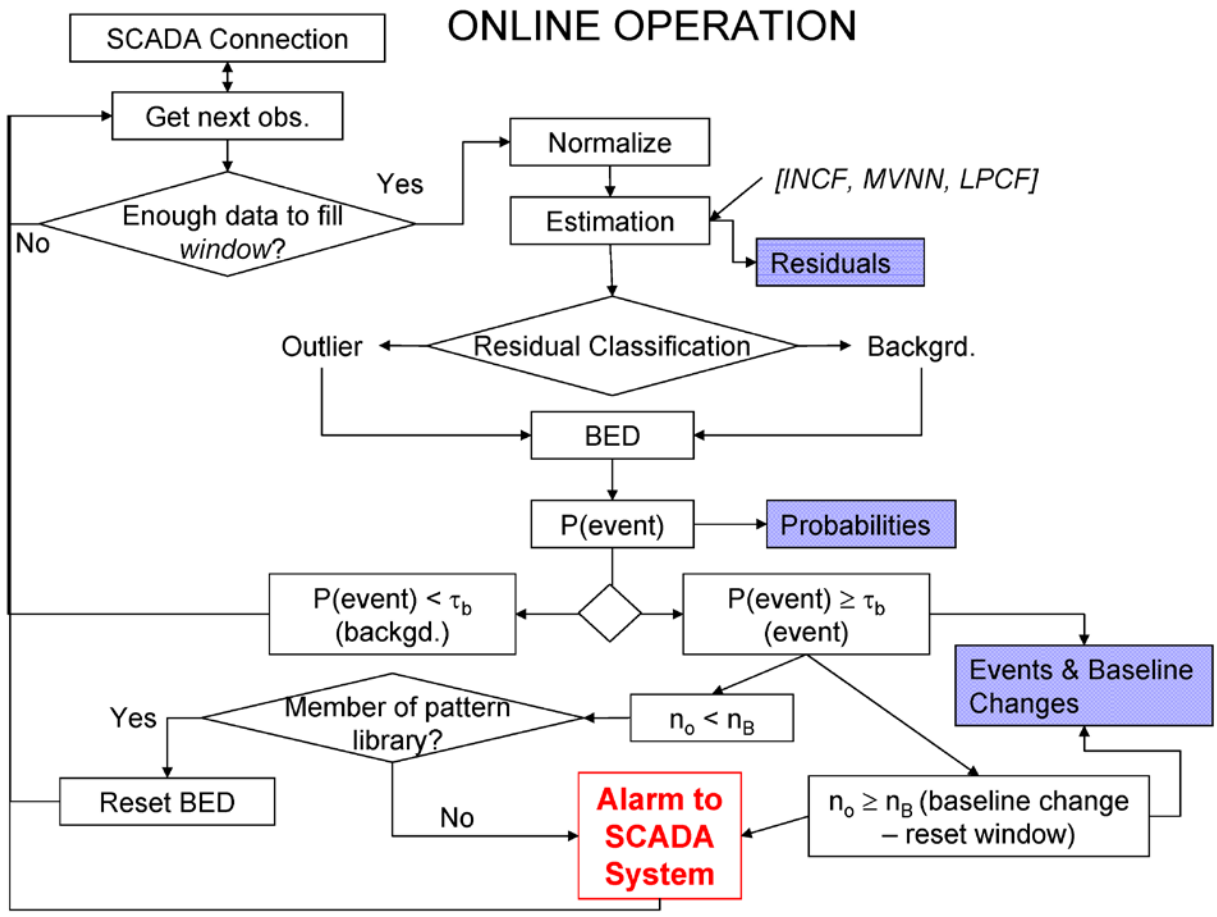


Figure 9 - CANARY's on-line operation mode.

3.3 Event Detection Algorithms

This section aims to help the user understand the algorithms well enough to choose acceptable starting values for the different algorithm parameters. It is not intended to give full descriptions of the algorithms, or of their implementations, but rather to give a broad overview of each algorithm and how the different parameters may impact performance. Each section will include at least one reference to relevant peer-reviewed publications that can provide additional background on the algorithms. Additionally, the document: Water Quality Event Detection Systems: Development, Testing and Application of CANARY (McKenna et al, in review) provides more detail on the motivation, development and application of CANARY.

Because event detection systems have an inherent trade-off between high sensitivity and the number of false alarms, each user must decide what the best parameters will be for a specific use of CANARY. For example, a researcher may have more tolerance for false alarms while testing a new algorithm or piece of sensor hardware, while a utility operator may decide to use settings that provide less sensitivity in order to decrease the number of alarms that require further investigation and response.

3.3.1 Terminology

Because there are multiple, differing, opinions in literature as to what constitutes an event, it will be helpful to clarify certain terminology that will be used in this document. A graphic showing some of the terms is provided in Figure 10 - Terminology shown graphically.

- **signal** – Every data stream is considered a signal. Usually, the data will come from a SCADA system, and will be composed of water quality (WQ) data (e.g., chlorine level) and operations (OP) data (e.g., tank levels, flow rates). Certain hardware may also provide error data to the SCADA controller, in which case this information can be used by CANARY as alarm (ALM) signal.
- **data-interval –** CANARY works on time-series data. These data are discretized according to the data-interval (sampling interval). For example, a common data interval is two-minutes. If data are provided more often than the data interval, then only the most recent values will be used. If data are only provided every few data intervals (for example, total-organic carbon sensors tend to take several minutes to process a single measurement and reset), the hardware will generally provide the same value until it changes, in which case the same value would be sent to CANARY for multiple data-intervals.
- **precision –** The hardware that reports back to a SCADA system typically has a limit in the precision it can provide, dividing an otherwise continuous range of possible values into discrete bins. The SCADA system, however, will report data back to CANARY to a default number of decimal places that is typically much larger than warranted by the precision of the sensor. By specifying the precision of a signal, CANARY recognizes that what may look like a significant change (e.g., conductivity going from 310 to 320) is in fact a move to the next available data point. Setting this precision value properly can help reduce false alarms on very stable signals.
- **monitoring station – A** monitoring station defines a set of signals that are used together for event detection. These signals are generally from hardware that is physically all at the same

geographical location, but it doesn't have to be. Generally a monitoring station will always include some water quality signals, but will not always have operations or alarm signals.

- **sub-station –** A sub-station is defined as a monitoring station that is physically located at the same place as other monitoring stations. For example, a single tank may have multiple outlets that are monitored separately in CANARY, but which are located in the same building. In this case, each outlet becomes a monitoring station in CANARY and these can be differentiated by referencing them as different sub-stations.
- **residual –** The difference between the estimated and observed water quality values at a single time step is the residual. The size of the residual is compared to the threshold, τ_a (see below for definition), in order to determine if the observed water quality is indicative of background water quality conditions or of anomalous conditions. Residuals can be positive or negative, indicating under- or over-estimation of the observed water quality; however, only the absolute values of the residuals are used for comparison to the threshold.
- **threshold –** The threshold value is the key parameter for residual classification. The threshold τ_a is set in units of standard deviation, σ, not in the native units of the data. This allows data of differing ranges and units to use the same threshold. The standard deviation that is used is calculated from the background data included in the normalization window (see below). Setting the threshold as a relative measure in terms of standard deviations allows the absolute value of the threshold to increase and decrease depending on the variability of the observed data. Periods of higher background variation will be compared to a higher absolute value of the threshold, and vice versa for periods of lower background variability.
- **normalization window –** The most recent background data observed for this signal is contained in a moving window. The data values within this moving window are normalized to have a mean of zero and standard deviation of 1.0. The length of this window is set by the user, with consideration of site characteristics. For example, a site below a tank that fills and drains on a daily schedule may need a window size of slightly longer than one day, while an in-network location may only need three to six hours of history. How to choose an appropriate window length and threshold are discussed further in Chapter 8.
- **outlier –** An outlier is defined as a data value at a single time step that is considered anomalous relative to the background or expected behavior for that time step. This definition is made by the residual at that time step exceeding the threshold value. Any time-step when one or more signals deviate from the expected value by more than the threshold is classified as an outlier.
- **event –** An event is a period of sustained abnormal activity, where many outliers occur in a short period of time. The binomial event discriminator (BED) is a statistical algorithm that is used to decide how many outliers are needed in a given time period to create an event. The BED parameters are entered by the user to make this determination.
- **probability threshold –** The probability threshold, τ_B, defines the probability of an event that must be exceeded before an event is signaled to the by CANARY.
- **baseline-change** – When an event has continued for a user-defined period of time, a baseline-change is said to have occurred. This means that the event has sounded a warning for long enough that two things have happened:

  - the operators have listened to the alarm long enough to either identify the cause, or they have initiated action to find the cause, and,
  - it Is been long enough that CANARY should start looking for changes from this "new" background water-quality.

  Baseline-changes are significant events, because they mean that CANARY will stop signaling that an anomalous water quality event is occurring. A baseline-change can only occur after an event, but not all events will be of a long enough duration to become baseline-changes.
- **water quality pattern –** A recurring trend in one or more WQ signals that would normally be of significant magnitude to cause an alarm, but which is a "normal event." Such patterns may be caused by daily demand changes, pumps or plants turning on and off, or water treatment activities. If the pattern is regular enough, it can be identified, stored in a library and used as a recognized pattern to help eliminate false alarms from normal activities.
- **clustering –** The process of identifying water quality patterns is called clustering. Cluster files are used to keep a library of patterns that can be used to identify "normal events" and decrease false alarms. Typically, there will be one or more operations (OP) signals that will provide useful information to the clustering algorithm.

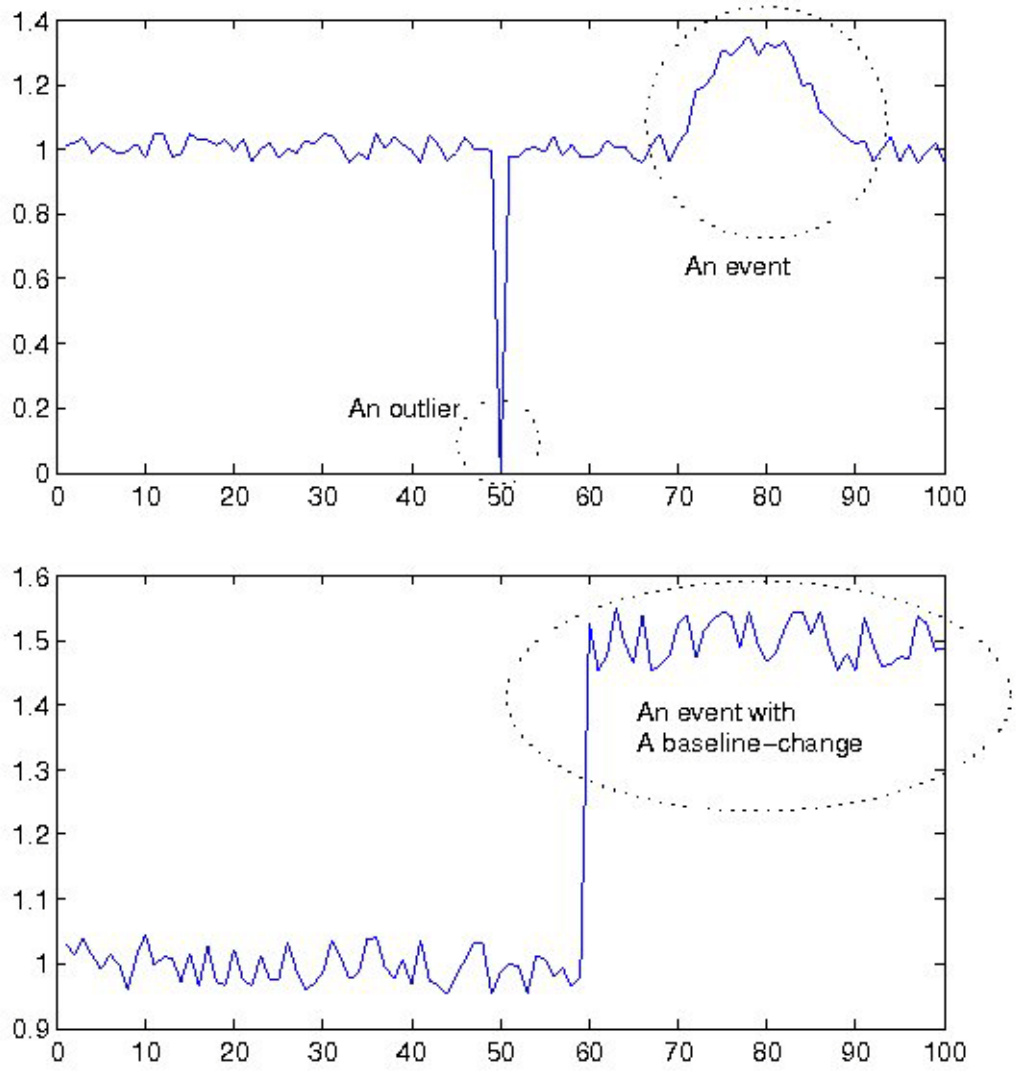


Figure 10 - Terminology shown graphically

3.4 Water Quality Estimation and Residual Classification

This section describes how CANARY uses water quality signals to estimate values for the next time step and calculate and classify residuals.

3.4.1 Normalization Window

Because the signals that CANARY operates on are of different magnitudes and measured in different units (e.g., mg/L, NTU, pH, etc.), it is necessary to normalize the data prior to making comparisons. A window of time steps, specified by the parameter n_h, determines the number of data points included in the normalization of the data. The historical data, X_h, are then normalized by the mean and standard deviation to obtain the normalized data, X_S. This operation is shown in Equation 1.

Equation 1 – Calculation of normalized data within normalization window

$$X_S = \frac{X_h - \bar{x}}{\sigma_x}$$

The result is a data set of length n_h, with a mean near 0.0 and a standard deviation, σ, near 1.0. This does not mean that the data constitutes a normal distribution (it almost certainly will not). As CANARY steps through each new time step, the oldest data point is removed from the window, the most recent data point is added to the window and the data in the new window is normalized again to create a new mean and standard deviation.

3.4.2 Linear Prediction Filter (LPCF)

The LPCF algorithm uses digital filtering along with linear coefficient estimation to predict what the newest observation should be based on the trends of the data in the historical window. The estimation of the new water quality value is a weighted average of the previous values contained in X_S. The linear coefficients provide the weight for each previous value and are updated at every time step to adapt to changing background water quality. The coefficient calculations are formulated to provide estimates that are unbiased and have minimum variance. The algorithm uses the **filter** and **lpc** functions from MATLAB, a description of which can be found in (The MathWorks, 2002), or in (McKenna, Klise & Wilson, 2006). The threshold, τ_a, refers to the maximum error allowed, in units of standard deviation, between the estimated data values for the new time step and the observed data values. Error values, or residuals, greater than τ_a are considered outliers. Typical values for τ_a may range from as low as 0.5σ at stations with very stable water quality, up to 1.5σ or higher at stations where water is mixed from different sources. This algorithm processes each signal separately, and then compares the largest absolute error to the threshold.

3.4.3 Multivariate Nearest Neighbor (MVNN)

This algorithm compares the newest observed water quality data against all the data that are in the history window. The normalized water quality data are mapped in an m-dimensional multivariate space where m is the number of signals defining that space. In a two signal example, which is the easiest to visualize, the data create a series of points in 2-D space (e.g., residual chlorine vs. pH). Then, the newest observation is plotted against the others. The residual is the distance from the new point to the closest historical data point. The threshold, τ_a, is the maximum allowable distance the new data point can be

from its nearest neighbor, in units of standard deviations. The MVNN algorithm works for any number of signals, and, in contrast to the LPCF algorithm, only one residual value is calculated to be compared against the threshold, no matter how many signals are used. Additional information is available on the MVNN algorithm in (Klise & McKenna, 2006a) and (Klise & McKenna, 2006b). Typical values for τ_σ range from 0.5σ to 3σ.

3.4.4 Set-point Proximity Algorithms (SPPB and SPPE)

These algorithms are new to CANARY version 4.1, and have not yet been published in other literature. The goal of these algorithms is to provide a ramped warning.increasing value of the probability of an event as any water quality signal approaches one of the set point limits. The threshold, τ_a, takes on a slightly different meaning for these algorithms, helping define a distance, Δ_R, away from either set point value over which the probability will increase. The two different options, SPPB and SPPE, indicate which ramp probability distribution to use to define the increase in probability of an event as a function of the normalized distance away from a set point limit. The options available are a Beta distribution (SPPB) or an exponential distribution (SPPE). The beta distribution increases probabilities more quickly than the exponential distribution.

3.4.5 Homegrown Algorithms (JAVA)

Because CANARY was developed with algorithm testing in mind, external, compiled Java functions can be used to calculate the residual. The API is described in the DummyAlgortihm.java file included in the software distribution.

3.4.6 Consensus Algorithms (CAVE and CMAX)

In real-time SCADA operations, CANARY may be limited to reporting output from a single algorithm. If there are two or more algorithms that the user wants to combine in a single report, the consensus algorithms "CAVE" and "CMAX" are one methods of accomplishing this task. These consensus algorithms combine the outputs of more than one event detection algorithm used to analyze the data. The "CAVE" algorithm takes the probability of an event from all the input algorithms and reports the average probability of an event as calculated across those algorithms. The "CMAX" algorithm reports the maximum probability of an event as calculated across those algorithms. The analyst may want to have duplicate outputs (such as to a text file) to aid in identifying which algorithm caused an alarm, but this method allows for interaction with minimalistic SCADA connections.

3.5 Water Quality Event Determination

The estimation and residual classification algorithms described above provide a binary determination at each time step: outlier or background. These results are used as input to two additional pieces of the water quality event identification: calculation of the probability of a water quality event at each time step and determination if an observed outlier time step is part of a previously recognized water quality pattern. Both of these additional processing steps are designed to reduce the number of false positive alarms produced by CANARY. It is not necessary to use either of these additional steps if only a binary outlier/background indication of water quality is required; however, we have found that determination of the probability of an event at each time step provides a much richer picture of water quality changes within a distribution network and the integration of results across time steps reduces false alarms. For

monitoring stations with water quality changes induced by operations changes (e.g., monitoring stations near a tank), the pattern matching capability can provide a significant reduction in the number of false positives.

3.5.1 Binomial Event Discriminator (BED) and Event Time-out (ETO)

The binomial event discriminator is not an event detection algorithm itself, but an algorithm designed to integrate results over multiple time steps to provide the probability of an event and to help limit false positives. As described in (McKenna et al., 2007), the BED uses a separate, smaller history window to track the number of recent outliers. This window is n_B time steps long. The probability of n_O outliers occurring in n_B trials, when the probability of an outlier occurring at any given time step is P_B, is defined as P_E, where P_E is the output of the **binocdf**(n_O, n_B, P_B) probability calculation. If P_E exceeds the probability threshold, τ_B, then an event alarm is output by CANARY. At each successive time step, the newest outlier calculation is added to the window, and the oldest is removed, and P_E is recalculated. If the event continues for the number of time steps that determines a baseline-change, n_{ETO}, then a baseline-change is noted, and all the algorithms are reset.

Unlike the estimation algorithms, which provide a Boolean "above threshold" (outlier) or "below threshold" (background) indicator, the BED provides the continuous probability that an event has occurred at each time step.

The values of the BED window, n_B, and baseline-change ETO window, n_{ETO}, are typically chosen for a specific class of events. For example, if anomalies lasting less than four time steps are not of interest, n_B should be set larger than four, allowing short anomalies to avoid sounding an event alarm. If operations managers believe that one hour is long enough to evaluate or start action on an alarm, then n_{ETO} should be set to the number of time steps that occur in one hour. Because of this, there are no "typical" values for the BED or ETO parameters. Also note that the ETO window can be set regardless of whether the BED is used.

3.5.2 Water Quality Pattern Matching

The water quality pattern matching tool uses the idea of trajectory clustering to represent a time-series of water quality data with a relatively low-order polynomial. Regression models and clustering of the regression coefficients are used to construct a library of common water quality patterns from historical data. The length of the time series in the pattern is generally limited to less than 100 time steps prior to and including the indication of a water quality event by CANARY. For example, three signals are being examined and at some point Cl decreases, pH increases and Conductivity remains constant. The changes in Cl and/or pH are enough to create an event in CANARY due to P(event) exceeding τ_B.

Starting at the current time step and looking back n_C time steps, a low-order (e.g., 3rd order) regression model is fit to each signal. The fits are done independently on each signal and in the example case the coefficients will be quite different (decreasing vs. increasing vs. flat) but the polynomial model (e.g., 3rd-order) applied to all signals must be constant. This process is repeated for all events that the user deems representative of common water quality patterns. The library of patterns is stored within a matrix that has one row for each event that was identified by CANARY. The total number of columns is the order of

the polynomial plus one times the number of water quality signals examined (an nth-order polynomial has n+1 coefficients). For example, use of a third-order polynomial for analysis of three water quality signals will result in 12 columns in the matrix.

Online water quality pattern matching uses the information in the pattern library to discard events that are indicative of previously identified water quality patterns. In this case, if the pattern closely matches a known event (within the 20th percentile, for example), then a message is submitted that a known event has occurred, and no other alarm is sounded. If no known cluster is similar enough to the current data, processing continues, and an alarm may be produced.

4 CANARY Inputs

Inputs and outputs to CANARY come from one or more *data-source.* In this chapter, the format for an input-type data-source is discussed.

Each data value must be associated with four different pieces of data: the date and time of the measurement, the sensor it came from, the monitoring station it belongs to, and the data value itself. The date and time are converted into a discrete time index, k; the sensor identifier is used to identify which data column the value belongs to, with an index j. The location specifies which data set to access.

There is additional information that may be provided on a real-time basis, such as alarm flags, quality flags, comments, etc., but the four data pieces described above are the only required elements of the input data. A table showing the different entry values for the *input-type* parameter in the config file is provided as Table 1. There are two primary ways these data are organized: column-based (field-based) and row-based (record-based).

Table 1 - Acceptable values for the input types parameter.

Input Type	Brief Description
CSV	A text spreadsheet file, in comma-separated-values (CSV) format, described in Section 4.1.1
DB	A field-based database format, for direct SCADA-CANARY connections, described in Section 4.1.2
XML	A record-based XML-formatted message to pass information to a third-party SCADA integration system. Discussed in Section 4.2
EDDIES	A record-based database format for integration with the EDDIES tool. Discussed in Section 4.2
MAT	Re-use binary output from a previous CANARY run. Described in Section 4.3.

4.1 Column-based Inputs

In a column-based input, each row in the file or database corresponds to a single time-index, and all the data values for that time step are reported in columns on that line. The primary advantage of this format is that very few data are repeated, with column headers and date indices specified only one time each. However, this format does not make it easy to associate multiple pieces of information (such as SCADA quality tags) with a date/signal-value pairing. Nonetheless, this is probably the most common and popular format for storing large quantities of information.

4.1.1 CSV Type Input Data-sources

Most users of CANARY will be familiar with spreadsheet applications. One of the most common formats used to transfer data from one application to another is comma-separated values (CSV) format. In this format, each column is separated by the "," delimiter and a new-line indicates the end of a record; generally, the first row defines the column labels. When CANARY uses CSV files as input, it follows this convention.

The first column of a CSV CANARY file should contain the date and time of measurement, and be titled "TIMESTEP" or something similar. The exact name should be indicated in the *timestep-field* parameter in the configuration file settings for the data-source. Each additional column should contain the SCADA Tag Name for a signal defined in the config file. These tags should be unique and often are chosen to denote the monitoring station and sub-station as well as the type of signal “WQ” or “OP”. An example CSV file (as it would be displayed in a spreadsheet application) is shown in Table 2. CSV files are most useful when examining historical data off-line, though if a SCADA system is set up to print outputs to a CSV file every time step, it could theoretically be used in on-line mode.

Table 2 - A sample CSV style spreadsheet view

TIME_STEP,	TEST1_CL2,	TEST1_COND,	TEST1_PH,	TEST1_TEMP,	TEST1_TOC,	TEST1_TURB,
1/1/2008 0:00,	0.9,	308,	8.78,	10.1,	1.01,	0.04,
1/1/2008 0:02,	0.9,	308,	8.78,	10,	1.01,	0.04,
1/1/2008 0:04,	0.9,	309,	8.78,	10,	1.01,	0.04,
1/1/2008 0:06,	0.9,	308,	8.78,	10,	1,	0.04,
1/1/2008 0:08,	0.9,	309,	8.78,	10,	1,	0.04,

4.1.2 DB Type Input Data-sources

The same format described above for files is also very common for databases. In this case, each table has fields that correspond to the names in the first row of the file. CANARY will query the database and read in a record that corresponds to a certain time step. This can be done in on-line or off-line modes of CANARY operation. The name of the field that contains the time step data should be provided in the *timestep-field* parameter in the configuration file, and should be of "DateTime" type in the database. The data presented as Table 2 adequately represents a database as well as a CSV data file.

4.2 Record-based Inputs

Record-based inputs tend to be defined by middle-ware applications that shield the SCADA system from the event detection system that is providing analysis, and which reformat and screen information prior to passing information back to the SCADA system. These formats tend to combine the control elements and input/output elements into a single package, defining how CANARY should communicate with the system. There are two systems that CANARY has been designed to integrate with: one is a proprietary XML message passing scheme, which is indicated by using an *input-type* of XML; the other is the EDDIES format used by the US EPA Office of Water (OW) Water Security Initiative.

4.3 Outputs from Previous CANARY Runs

It is possible to re-use the output from a previous CANARY run as input to a new run. In this case, the input-type is "MAT." These files are HDF5 binary formatted data files that are produced by MATLAB. These are typically smaller than the associated database or CSV text file would be, and load much more quickly. However, they can only be used in off-line modes of operation.

5 CANARY Outputs

If CANARY is to function as an event detection tool, then it must provide some indication that an event has occurred. This is accomplished in several different ways. The primary output is to the screen where CANARY is running; a message is displayed saying that an event has been detected, and the when and where information are provided as the date and time and the monitoring station name. This information is also copied to the log file. It is likely, however, that CANARY will be running in the background on its own computer, one which does not have someone watching it all the time. In this case, CANARY can respond back to the SCADA database and provide notification that an event has occurred.

For off-line operation, real-time alerts are not as essential. In this case, CANARY provides CSV file output that lists the date and time of events and which algorithm detected the event. It can also provide probability of an event at every time-step. Regardless of the mode of operation, CANARY will always create a `.mat` binary file that contains a record of all the results that have been calculated during a particular CANARY run. This file can be loaded directly into MATLAB by those who have it, or it can be reused as an input to CANARY. This output `.mat` file is also used by the "Graph Outputs" program to create plots of the data and the event detection results. The output options are described in Table 3.

Table 3 - Output types for CANARY data-sources.

Input Type	Brief Description
FILES	Text spreadsheet files, in comma-separated-values (CSV) format, described in Section 5.2
DB	A field-based database format, for direct SCADA-CANARY connections, described in Section 5.3
XML	A record-based XML-formatted message to pass information to a third-party SCADA integration system. Discussed in Section 5.4
EDDIES	A record-based database format for integration with the EDDIES tool. Discussed in Section 5.4

5.1 Console Output

CANARY does not provide console (screen) output for every time step that is processed. Providing this output would significantly increase the processing time needed. However, in order to indicate that CANARY is functioning properly, CANARY provides a message after every day of processing to let the user know that processing is proceeding properly. This message looks like:

```
<Msg From="EDS" To="info" >
  <Cont>07/05/2008 00:00</Cont>
  <Subj>Continuing to process...</Subj>
</Msg>
```

When an event or other warning occurs, this information is also printed to screen. An event message looks something like the following:

```
<Msg To="CWS" From="CANARY" >
   <Subj>An event was detected with 93% Probability at SITE1</Subj>
   <Cont>2009-08-14 14:04:00</Cont>
</Msg>
```

By using an XML message format, CANARY's output log can be monitored and parsed by third party applications for automatic reporting.

5.2 FILES Output Created

Using the *output-type* of FILES means that several different outputs will be created in CSV format. These files can then be easily imported into spreadsheet applications for additional analysis or annotation. The files are organized by monitoring station and have multiple columns for multiple algorithms. All the information about the columns is listed in the first row as header data. The different files are:

- *.raw.csv – This file contains the raw data collected by CANARY during operation. This is useful for grabbing data from a database for archival or analysis purposes.
- *.res.csv – This file contains the actual residuals (positive and negative values) for each signal and algorithm at a given location.
- *.prb.csv – This file contains the probability of an event that was calculated for each time step.
- *.evt.csv – This file contains the Boolean event status for each time step and algorithm.

In off-line analysis mode, the large amount of data that has to be written at each time step when the *output-type* of FILES is selected slows down CANARY somewhat. However, in on-line modes, this delay tends to be small compared to the data interval. Because the data are written immediately, this mode can be very useful for debugging connections between SCADA and CANARY, since the last data provided and its analysis are available to view immediately.

5.3 Database Output

Using the DB output-type means that CANARY should try to write to a database. This *does not* have to be the same database that is being used for input, which means a buffer can be used to separate CANARY from the SCADA database. If used, the user-ID given to CANARY must have write privileges for the specified *output-table* defined in the configuration file.

CANARY will create, if necessary, and update the table specified by assigning fields named after the different monitoring stations, and all data for a given time step will be provided on a single row (record).

5.4 Other Outputs

Both EDDIES and XML formatted data-sources will be used for both input and output, as described in Section 4.3. Additionally, a MAT type file will be created every time CANARY exits, or when the "Save" button is pressed on the CANARY user interface. If running in "Training" mode, cluster files may be created following CANARY termination, and these will be discussed in Chapter 7.2.

If debugging is active, two additional files, "debug.xml" and "debug.sql" may be created. The XML file will contain lots of debugging information regarding the internal workings of CANARY. The SQL file will

contain all the SQL queries and/or updates that CANARY sends to the database. Both these files may become very large, and debugging should be turned off for deployment use.

6 Configuration Details

Configuration for the CANARY software is done through a configuration file. This file is written using Extensible Markup Language (XML) formatting. An XML file uses "tags" to define different options that control CANARY's operation. XML files are text files, and can be edited using text editors or special XML editing programs. Generally, if the user double-clicks an XML file, it will open in an internet browser, producing text that looks like the following:

<tag-name parameter1="value1" >*TextValues or <other-tags/>* … </tag-name>…

A new feature contained in CANARY is a configuration editor interface, so editing the XML configuration file by hand is no longer absolutely necessary. The configuration editor main screen is presented in Figure 7. Some default values may be presented when starting a new file from scratch. Others require the user to input all the options. The figures presented as part of this chapter will not have real values in the appropriate fields, but will instead reference the section number where a particular option is discussed. To load an existing file, select "Load File." To save the file, select "Save File."

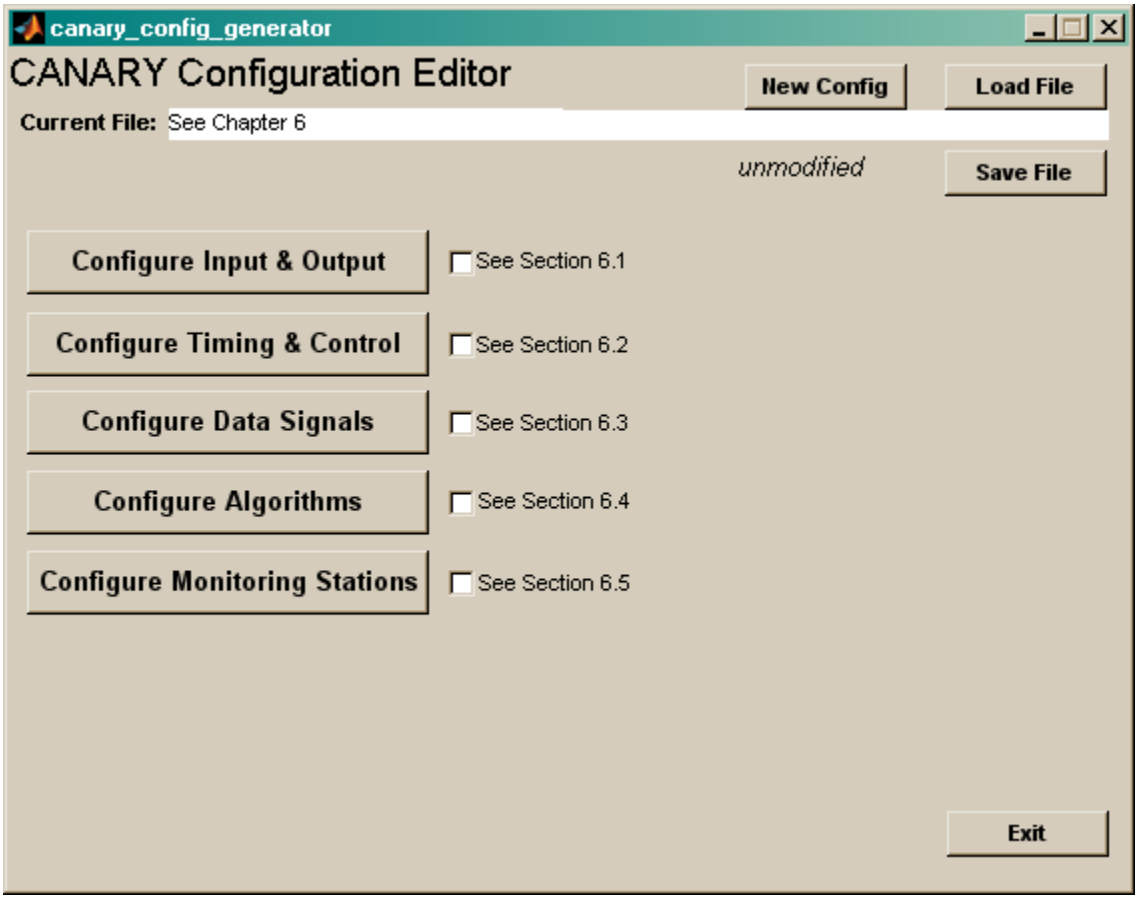


Figure 11 - Main configuration editor screen.

6.1 Data-sources (Inputs and Outputs)

A "data-source" is any input or output that is used by CANARY, as discussed in Chapters 4 and 5. The configuration editor screen is shown in Figure 8. There are three main sections, the standard options, database login information, and additional details.

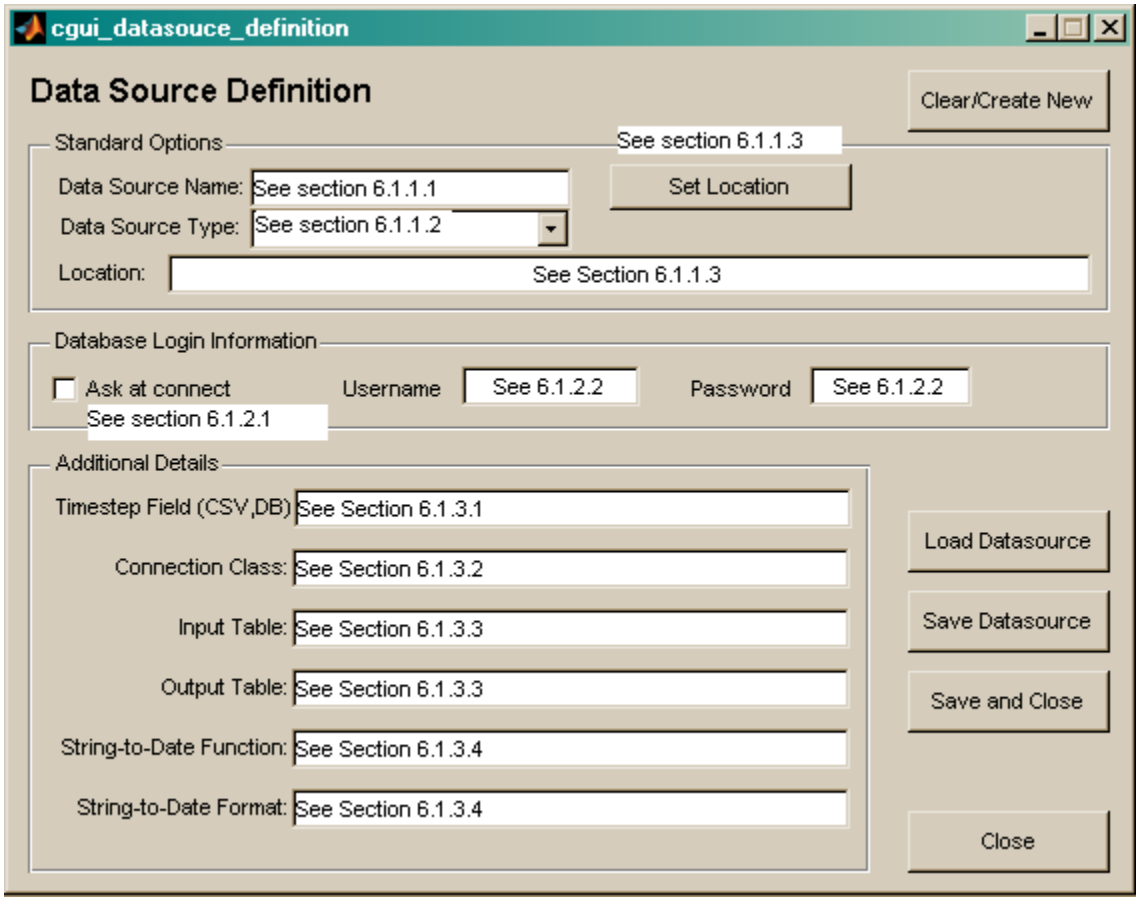


Figure 12 - Configuring input and output data-sources.

6.1.1 Standard Options

These options are required for all data-sources.

6.1.1.1 Data Source Name

This parameter is always required. The **data-source-name** defines the name by which this data-source will be referenced throughout the configuration file. It must start with a letter, is case-sensitive, and cannot contain spaces. Some examples of possible short-ids: FILE1, Water_Works, D8thStreet.

6.1.1.2 Data Source Type

The "type" parameter is also required. The type must be one of the options detailed in Table 1 and Table 3: XML, DB, CSV, FILES, MAT, or EDDIES.

6.1.1.3 Location

The **location** parameter can be several different possible values. It can contain a part of a data-source location, or it can be a complete internet location. Use the "**Set Location**" button to create a location from different parts, or to select a file. The location field can also be edited directly.

- For files – the "location" parameter can specify:
  - a file located in the same directory as the configuration file, "data2.csv"
  - a filename stub, "data2"
  - a complete file and path name, such as "C:\My Data\data2.csv"
- For databases or networked locations, "location" can specify:
  - the complete internet URL, such as "jdbc:oracle:thin:@//localhost:1521/xe"
  - the address and port of a connection, such as "192.168.21.29:9403",

6.1.2 Database Login Information

These options are only required for databases connections.

6.1.2.1 Ask at connect

If the **username** and **password** used to connect to the database need added security beyond directory Access Control Lists (ACLs), click on the "**Ask at connect**" box, and do not enter a **username** or **password** in the following boxes. CANARY will prompt for the **username** and **password** the first time it tries to connect to the database when it is launched. It will save this information internally for as long as it is left running, but will not save the clear-text (non-encrypted) passwords.

6.1.2.2 Username and Password

When connecting to a database, most users will be required by the database administrator to log in with a **username** and **password** that will grant specific read and write privileges. For security reasons, the configuration file should be kept in a safe location, since the **username** and **password** are saved in clear-text (non-encrypted). Talk to the database administrator if there are further questions regarding security or to set up a limited account to use that will protect critical data.

6.1.3 Additional Details

These options apply both to databases and to files.

6.1.3.1 Timestep Field

The **timestep-field** is used for both files and database data-sources. In CSV files, this defines the header name used for the column with date and time information. In databases, this tag defines the field in the table where date and time information is stored. The default value, if this tag is omitted, is "TIME_STEP".

6.1.3.2 Connection Class

The **connection-class** is part of a database driver for Java. This information will be located at the vendor website, usually under "Java Connector" or a similar heading. The vendor should provide a ".jar" file, which is an executable Java file that can be used on the desired system. Among the documentation will be the name of the "Data-source" connection class. The name of this class will be entered in the **connection-class** field. Some examples are:

- com.mysql.jdbc.jdbc2.optional.MysqlDataSource
- oracle.jdbc.pool.OracleDataSource

Talk to the database administrator about how to obtain these files and class names.

6.1.3.3 Input Table and Output Table

When using databases, the **input-table** and **output-table** define where to read and write data. If data is stored in more than one table in the database, define multiple data-sources, one for each table in the database. The tables must be formatted correctly, as discussed in Sections 4.1.2 and 5.3.

6.1.3.4 String-to-date Function and String-to-date Format

Every database has its own method of converting text strings into date numbers. To facilitate this, the **string-to-date-function** and **string-to-date-format** options must be set according to the database that will be used. If the date format is inconsistent across files, the **string-to-date-format** field can be used to define a specific format for use in a single CSV file. In this case, the format string is the same style described in Section 6.2.2.4 regarding timing options.

6.2 Timing and Control Settings

This section describes the most general settings for CANARY, the timing options and the mode of operation. The configuration editor screen is presented in Figure 9. There are three sections: control settings, timing settings, and driver files.

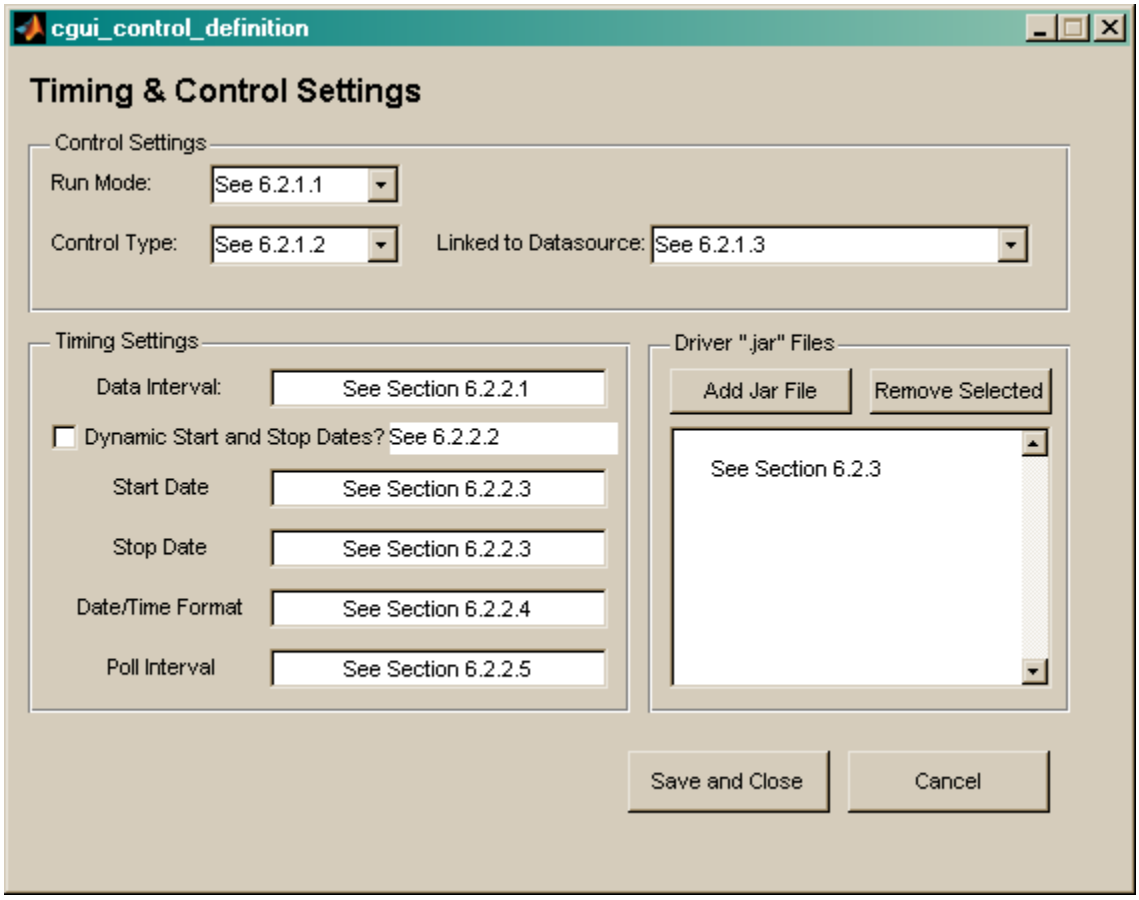


Figure 13 - Timing and control settings configuration screen.

6.2.1 Control Settings

The control settings are pop-down menus that allow the user to choose the setting from a list.

6.2.1.1 Run Mode

The **run-mode** defines the mode of operation for CANARY. This tag must occur exactly once in each configuration file. There are five different values that can be used:

- **SaveOnly** – This mode does not process data, but instead loads data and saves it to a binary output file. This is most useful when getting data out of a database for off-line analysis.
- **Batch** – This is the typical off-line mode of execution that is used for analysis.
- **Training** – This mode is identical to "Batch" mode, but includes the cluster creation subroutine that creates the pattern library after the run is complete.
- **RealTIme** – The "RealTime" mode of execution is used when CANARY is put into a deployment situation using an on-line database or XML connection.

- **EDDIES** – When using the "EDDIES" software, this is the correct mode to use.

6.2.1.2 Control Type

The **control-type** defines how CANARY interacts with the SCADA system. In off-line mode, this will almost always be "**INTERNAL**." Internal mode directs CANARY to use its own timing clock for analysis. In "**EXTERNAL**" mode, CANARY takes direction from an XML-type server regarding when to process new data, and when to send results. In "**EDDIES**" mode, all control is handed over to the EDDIES software. For **EXTERNAL** and **EDDIES** control, a **data-source** must have been defined to handle the control.

6.2.1.3 Linked to Data-source

If necessary, this drop-down menu will become available to link to an **EDDIES** or **XML** typed **data-source**. This allows CANARY to date direction from SCADA middleware programs.

6.2.2 Timing Settings

Every CANARY configuration must include information regarding the timing of data that is coming in as input. Additionally, off-line analyses need start- and stop-dates to know which data to process. The timing settings section resolves these issues.

6.2.2.1 Data Interval

The **data-interval** is specified in the format HH:MM:SS. Only one data interval can be specified per instance of CANARY.

6.2.2.2 Dynamic Start and Stop Dates

If running in real-time mode, this checkbox can be selected, which will allow the current date and time, or the SCADA middleware provided date and time to be used as starting and stopping dates.

6.2.2.3 Start Date and Stop Date

The date and time of the first and last data point to be used in the analysis. If running in EDDIES or REALTIME modes, these define how much extra data will be included prior to starting analysis.

6.2.2.4 Date/Time Format

The start and stop dates defined in Section 6.2.2.3 must be in the format specified here. The codes to use are presented below, along with some examples. This format must match the format provided in DATABASE type data-sources, but can be overridden for files by using the field described in section 6.1.3.4.

- mm – the code for a two-digit month, such as "02" for February.
- mmm – the code for a three-letter month, such as "Feb" for February.
- dd – the code for a two-digit day, such as "03" for the third day of the month.
- yy – the two digit year, such as "08" for 2008.
- yyyy – the four digit year, such as "2008."
- HH – the code for the hour of the day. 12- or 24-hour format depending on whether AM is present

- AM – used to indicate that "AM/PM" should be used, and 12-hour time formats. Omit for 24-hour time formats.
- MM – the code for the minutes of the hour.
- SS – the code for seconds.

The following are some common date/time formats that are used by different countries and applications.

- yyyy-mm-dd HH:MM:SS – this is the standard format that is returned by Java when converting a database DateTime formatted field to a string.
- yyyymmddTHHMMSS – this is the international standards organization (ISO) standard format for dates and times.
- mm/dd/yyyy HH:MM AM – this is a common format used in the United States.
- dd/mm/yyyy HH:MM:SS – this is the common format used in Europe and Asia.

Of course, any format can be specified by using a combination of the above, and when using CSV type data-sources, it is possible to define a specific format for use when reading data from the file using these same codes. Note that the format specified here must match the format used in the CSV file that is being read.

6.2.2.5 Poll Interval

The **poll-interval** defines how often CANARY will attempt to contact a database or network source for new data. This is independent of the **data-interval**. This interval can be zero, but for most database connections, it is better to keep this value at several seconds.

6.2.3 Driver ".jar" Files

All database connections and external algorithms will require an entry to the ".jar" files list. To add a file, click "Add," and then navigate and select "Open" to load the file into CANARY.

6.3 SCADA Signals and Data

Most of the configuration work required of the user will be addressed in this section. Every data signal that CANARY can use must be defined before CANARY can work with that signal. Figure 14 shows the screen that will be used. Once these signals are all defined, the configuration file can be copied and edited, so that the signals do not need to be entered twice.

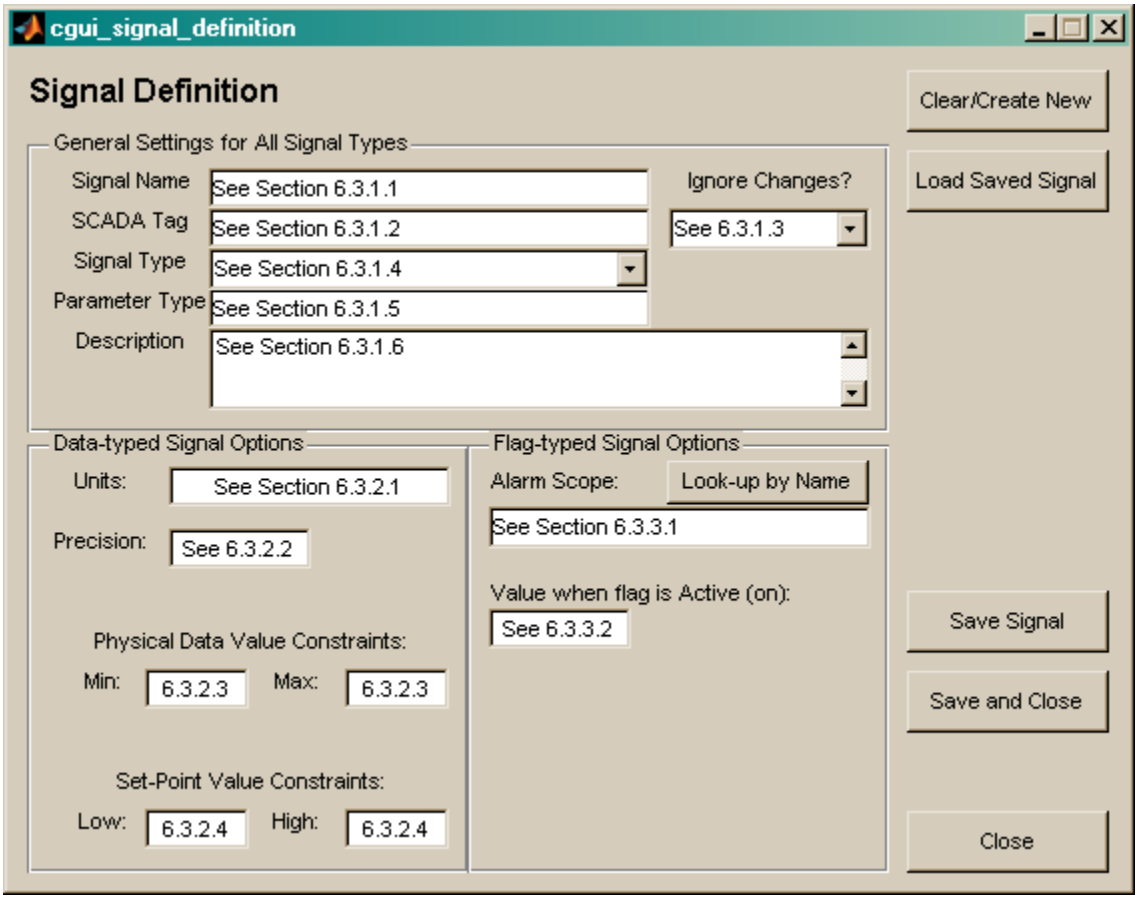


Figure 14 - Configuration screen for data (water-quality and operations) and flag (alarm and calibration) signals.

6.3.1 General Settings for All Signal Types

Every signal will require this section to be filled in completely.

6.3.1.1 Signal Name

The **signal-name** is used to reference the signal in other parts of the configuration program. This name is case-sensitive, cannot contain spaces or symbols (except "_"), and must start with a letter.

6.3.1.2 SCADA Tag

The **scada-tag** is the name that is used by the SCADA system to identify a particular hardware data value. These may be descriptive, such as "STATION1_PUMP1_STATUS," non-descriptive, such as "PUMP1," or completely obfuscated, such as "_P0x038A84F2E_223." The authors have had experience with all types, and the user will likely need assistance from the SCADA administrator to find a comprehensive list of the signals used in their system.

6.3.1.3 Enable Signal

This field changes based on the signal type (discussed in 6.3.1.4). For most signals, this field is either "Yes" or "No" to respectively enable or disable the signal from use. For operations signals, however, this field has additional options. The title is changed to "Ops triggers recalib," meaning: does a change in this operations-type signal mean that the algorithm should do a short-term recalibration? For example, when a pump comes online, it may be necessary to allow a short time period for water-quality parameters to re-stabilize. Setting this field to "On Increases," "On Decreases," or "On Changes" will activate this feature for operations type signals.

If one of the activating values is selected, the trigger threshold is based on the slope of the data across the past five time steps. If the slope is greater than – or less than the negative of – one unit of precision (see 6.3.2.2), then no alarms will occur because the operations signal has triggered a short-term recalibration. To eliminate this feature (which is the default behavior), simply select "Never" for this field.

6.3.1.4 Signal Type

This is one of the following types and should be chosen from the drop-down list.

- Water Quality
- Operations
- Hardware Error Flag (Alarm)
- Calibration Flag

6.3.1.5 Parameter Type

This is a user entered value that helps identify what this signal measures. For example, "CL2" may be an appropriate parameter type for a free-chlorine sensor.

6.3.1.6 Description

A free-form description for information purposes only.

6.3.2 Data-typed Signal Options

Water quality and operations data are considered data-typed signals. These signals need some additional options described in the sections below.

6.3.2.1 Units

A text string that can include LaTeX type formatting that describes the units of measurement.

6.3.2.2 Precision

As described in Chapter 3, the precision is used to discretize the analog data into digital increments. This can be defined by the hardware manufacturer, by a setting on the hardware, or by using common-sense. Changes that are less than this amount will never trigger an alarm.

6.3.2.3 Physical Data Value Constraints

This section allows a hard minimum and maximum data value to be specified for a particular signal. For example, the chlorine level can never be negative, only zero. By contrast, temperature is technically

infinite, which means a minimum of "-Inf" should be used. If no minimum or maximum is desired, use "-Inf" and "Inf" but do not leave these blank. Data values that exceed the minimum or maximum value entered here will not be processed by CANARY

6.3.2.4 Set-Point Value Constraints

This section defines set-point values for use in the change detection algorithms. Only a low and a high value are provided, and "Inf" and "-Inf" should be used instead of blanks. No matter what event detection algorithm is being used, a water quality value that exceeds either set point limit will be classified as an outlier. If multiple signals exceed the respective set point limits at a given time step, only one outlier will be reported for that time step.

6.3.3 Flag-typed Signal Options

Any signal that returns a binary value, such as 0/1, "on" and "off", etc., is a flag-typed signal. Typically, sensor hardware alarms are of this type of signal.

6.3.3.1 Alarm-Scope

This defines what signal a given alarm or calibration signal applies to. This scope provides the necessary connection between a signal and the hardware alarm for the sensor that produces that signal. To select a value, either type the **scada-tag** into the field, or click on "Look-up by Name," which will provide a list and fill in the box automatically. "Calibration" type signals do not need an **alarm-scope**, they apply to an entire monitoring station (see Section 6.5.4).

6.3.3.2 Value when flag is Active (on)

Use this field to indicate what value is used to indicate that a flag is active, or in abnormal status.

6.4 Algorithm Definitions

In this section, the configuration parameters required to select algorithms are described.

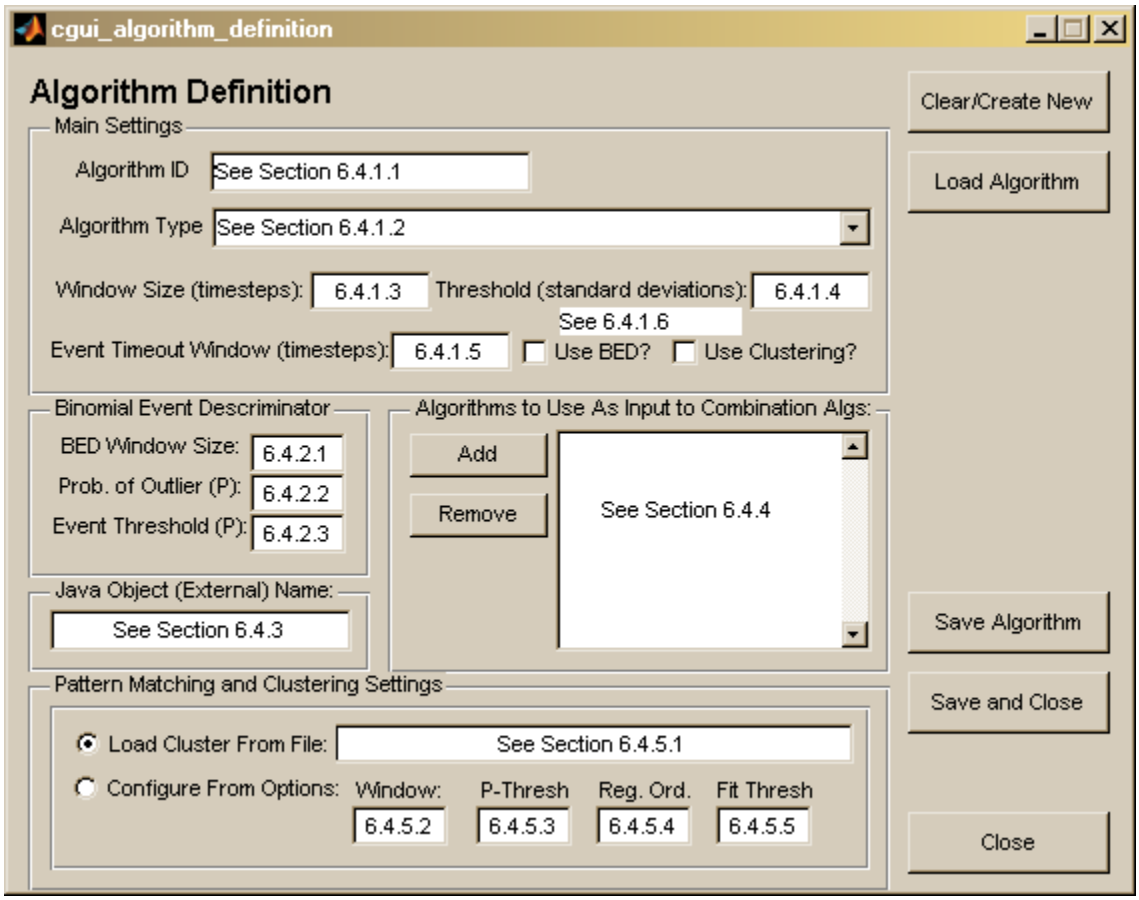


Figure 15 - Configuration screen for defining event detection algorithm settings.

6.4.1 Main Settings

These settings are required for all algorithm definitions.

6.4.1.1 Algorithm ID

This is the short name that is used to distinguish one algorithm configuration from another.

6.4.1.2 Algorithm Type

This is one of the algorithms described in Chapter 3. Select the desired algorithm type from the drop-down menu.

6.4.1.3 Window Size

The **window-size** is the number of time-steps to use in the normalization and prediction window. See Chapters 3 and 7 for more information on how to choose the best value for this setting.

6.4.1.4 Threshold

The **threshold** is the error in the prediction, in units of standard deviations, which must be met or exceeded before an outlier is declared.

6.4.1.5 Event Timeout Window

The **event-timeout** window size is the number of time-steps that will elapse before an alarm is automatically silenced.

6.4.1.6 Use BED and Use Clustering

These two check-boxes activate the **binomial-event-discriminator** and **pattern-matching** algorithms, respectively. The options activated by these check-boxes are described in Sections 6.4.2 and 6.4.5.

6.4.2 Binomial Event Discriminator Settings

The **binomial-event-discriminator** provides a probability of an event based on the number of outliers that occur in a certain time frame.

6.4.2.1 BED Window Size

The **bed-window** is the number of time-steps used in the calculation of the binomial event probability. In terms of a binomial distribution, this is the number of trials. This value must be less than or equal to the **window-size** parameter described in Section 6.4.1.3.

6.4.2.2 Probability of an Outlier

This value defaults to 0.5, or a 50% chance of an outlier occurring at any given point in time. This value changes the shape of the probability curve used by the BED, but does not necessarily need to be an actual outlier occurrence rate – the default value in combination with the appropriate choice of **bed-window** and **probability-threshold** has worked fairly well for typical use.

6.4.2.3 Event Threshold

This is the **probability-threshold** that must be exceeded before a series of outliers becomes an event. This value is also used by the set-point proximity algorithms, which do not use the BED.

6.4.3 Java Object Name

For algorithms of "JAVA" type, this is the class name of the algorithm to use.

6.4.4 Algorithms to use as Input to Combination Algorithms

The consensus algorithms use the outputs of other algorithms as inputs. To add an algorithm as input, use the "Add" button to select a previously defined algorithm. Select an algorithm and click "Remove" to delete an algorithm as input.

6.4.5 Pattern Matching and Clustering Settings

The pattern matching and clustering settings are defined in this section.

6.4.5.1 Load Cluster from File

The "training" mode of CANARY automatically runs the clusterization algorithm upon batch-mode termination and creates the pattern library. The output file, or the file output after running "canary --clusterize," can be entered here to enable pattern matching.

6.4.5.2 Configure from Options: Window

To specify the length of the pattern matching window during training, use this setting. This setting must be less than or equal to **window-size**.

6.4.5.3 Configure from Options: P-Thresh

This is the probability threshold that must be exceeded before the clustering algorithm is activated This is independent of the probability-threshold value set in the BED settings, but performs the same function, choosing what probability constitutes an event and what is not. It is used to determine when to add to the library (during creation) and when to check a possible event against the library (during normal execution).

6.4.5.4 Configure from Options: Regression Order

This is the polynomial regression order that is used to create clusters.

6.4.5.5 Configure from Options: Fit Threshold

This defines how similar two patterns must be to be considered "the same." Lower numbers will produce more clusters in the library, higher numbers will produce fewer but looser clustering. This value is considered a probability and therefore ranges from zero to 1.0.

6.5 Monitoring Station Definitions

The final step in configuring CANARY for a given system is to define monitoring stations/locations. The definition screen is presented in Figure 16. At least one location must be created before CANARY will be able to work.

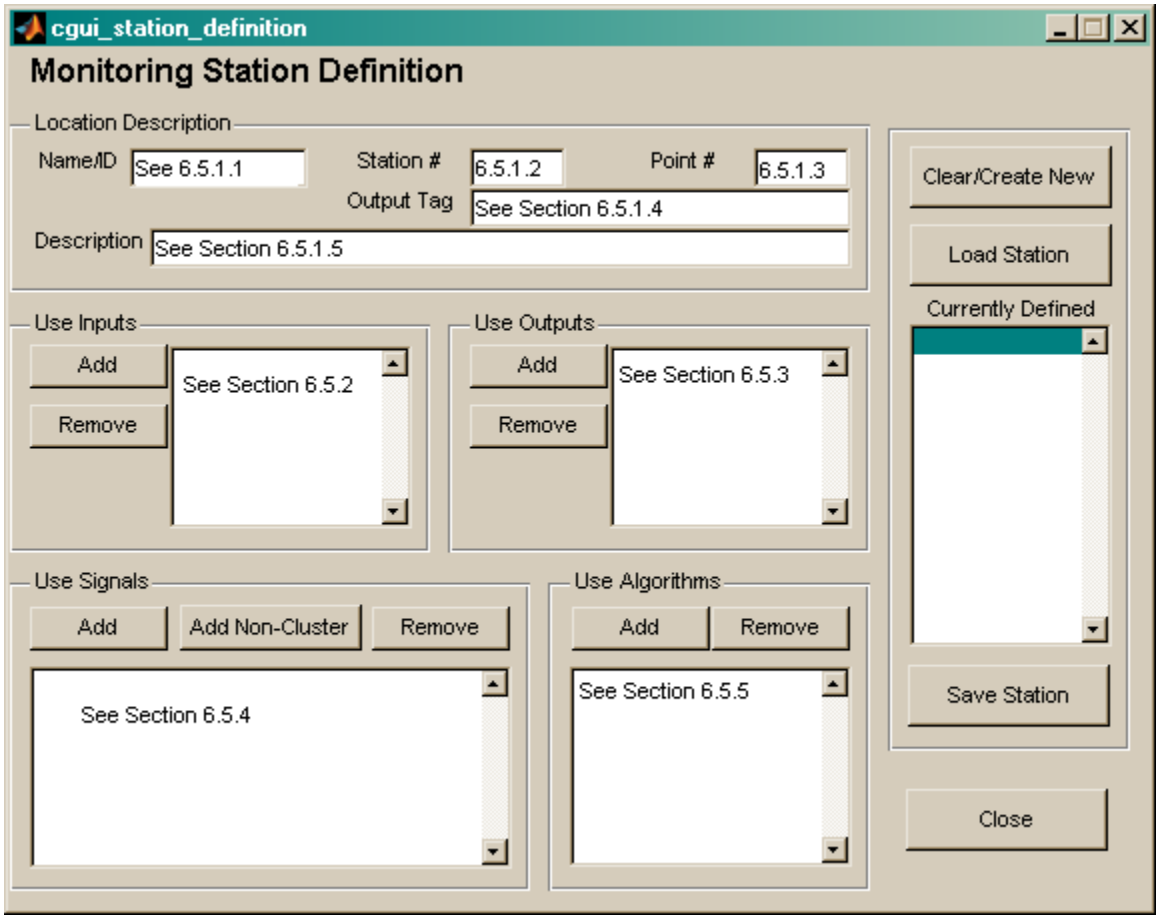


Figure 16 - Configuration screen to define monitoring stations.

6.5.1 Location Description

Basic information regarding a monitoring station is input in the **location-description**, which is illustrative that most monitoring stations are tied to a particular geographical location. However, not every signal need come from the same physical location.

6.5.1.1 Name/SCADA ID

This is the **monitoring-station-name**, and it is used in parsing other options throughout the CANARY configuration and output processes.

6.5.1.2 Station Number

This is an XML specific option that assigns an integer ID to a given **monitoring-station-name**.

6.5.1.3 Point Number
This is an XML specific option that assigns a specific integer ID to a given **output-tag**. This is typically needed when the output of CANARY is sent to a database or back into a SCADA system.

6.5.1.4 Output Tag
The name of the field that should be used when outputting XML data. Not currently used by EDDIES, but may be incorporated into generic database output at a future date.

6.5.1.5 Description
Free-format text.

6.5.2 Use Inputs
Select the data-source definitions to use as **input** to this monitoring station.

6.5.3 Use Outputs
Select the data-source definitions to use as **output** from this monitoring station.

6.5.4 Use Signals
Select the signals to use as input to the change detection algorithms employed at this station. A generic "Add" button adds calibration-type signals and data-typed signals that are **clustering-active**. To add a signal that should not be used in the clustering algorithms, use the "Add Non-Cluster" button instead. It is not necessary to add "Hardware-alarm" typed signals, since they will automatically be included when their **alarm-scope** signal is added to a monitoring station.

6.5.5 Use Algorithms
Select the change detection algorithms that should be employed at this site. The first algorithm selected will be the algorithm that is reported in **EDDIES** or **XML** typed outputs. It is not necessary to select the input algorithms used by the **consensus**-typed algorithms – they will be included automatically.

7 Advanced Command-line Options

CANARY has two built-in utilities that can help in the evaluation process. The first of these is a graphing utility that will take CANARY output and produce time-series plots of the data and EDS probability. The second is the clusterization tool that is executed automatically at the end of the "Training" run-mode.

7.1 Graphing Tool

The CANARY graphing tool can be started by typing the following command from the CANARY programs directory, or by running the "Graph Tool" batch file in the "Start" menu group for CANARY:

```
canary   --graph
```

This will bring up a window that asks which output file from CANARY to use. A second dialog box will appear asking which of the locations inside the output file should be graphed. Finally, a dialog box will ask which timing option to use. These are described in Table 3, below. The outputs of the graphing tool will be PNG formatted files, each of which is 8" wide by 10" tall, for easy printing by any graphics viewer or tool. The PNG files can easily be imported into MS Word or MS PowerPoint files. The file names of the PNG output are of the format: "*OutputFileName_Location_Mode_Date*.png".

Table 4 - Graphing mode options

All	Graph the entire data set on a single sheet
Quarterly	Graph 3-months on a single sheet
Monthly	Graph 30-days on each sheet
Weekly	Graph 7-days on each sheet
Daily	Graph each 24-hour period on a separate sheet

7.2 Cluster Generation Tool

The cluster generation tool is used to create patterns from "events" that exceeded a certain threshold during a previous CANARY run. This is especially useful when trying to eliminate false alarms due to periodic, similar operational changes. The format of the command is:

```
canary   --clusterize   INPUTFILE.mat   OutputSuffix   WindowSize   Threshold
```

The "WindowSize" parameter is the size of the patter matching window, and should not be larger than the algorithm window that was used to create "INPUTFILE". The threshold is the **probability** of an event that triggers a pattern matching request.

Once started, a series of graphs will be presented, one for each possible event to include in the clusterization algorithm. Selecting "Auto" will choose all possible events. Once the events are selected, the cluster file is created as "*LOCATION.*OuputSuffix". This process is identical to the one used in the "Training" run-mode, except that "Training" mode always uses the suffix of "cluster."

8 Training CANARY and Choosing Parameter Settings

This chapter provides a process by which an water quality analyst or network operator can select parameters that are appropriate for a given monitoring station. While it may be that certain stations are similar enough to share the same settings, in general, each station will need to be "trained" prior to using CANARY with that station's data.

While this chapter describes the process for training CANARY, more general information on event detection systems, false alarms vs. false negatives, and other significant issues are discussed in a comprehensive report available from the EPA web site: http://www.epa.gov/nhsrc/water/teva.html. While this user's manual describes how to pick various parameters, the implications of those choices are better discussed in the report, and the authors encourage any user who will be deploying CANARY in real-time to review the information provided in that document.

This chapter will have some "fill-in" areas to aid in the configuration of CANARY. The user is advised to use these to their best advantage, writing on paper or filling in the data electronically.

8.1 What is "Training?"

Training is the process by which algorithm parameters are selected that produces the most desirable results on a given data set. This process also involves identifying patterns that may trigger false alarms, but which are not "events" that CANARY should look for. While initial training may take a significant level of effort, it should not be necessary to re-train CANARY unless there is a major shift in the way operations are performed. It is possible that a utility will need more than one set of parameters to use at different times of the year to compensate for different water sources prevailing at different locations at different times of the year (e.g. a “winter” and a “summer” set of parameters. Also, once the first station is configured, training the other stations goes much more quickly.

8.2 Getting Ready to Train CANARY

First of all, historical data is needed to train CANARY accurately. During development and testing with a pilot utility over the course of two years, it was found that there were frequent problems in data stability and other hardware issues that required fixing. The user can expect that, like any new hardware installation, installing sensors and getting those sensors to report reliably and accurately to the SCADA system will take some adjustments. The user does not want to train CANARY on these initial data acquired during the “prove-out” of the sensor hardware.

In general, the user will want three to six months of stable water quality data to use for training. Stable means that hardware issues seem to have been resolved (or at least, identified as such) and any abnormal operations, such as a once-yearly cleaning, are excluded from the timeframe. These data are best provided through a SCADA historian database or output to CSV files. The data interval is generally already configured in the SCADA system. Table 5 is available for the user to record this information.

Table 5 - Find out the data interval

Data Interval (HH:MM:SS):	

If there are strikingly different operations during different seasons (winter vs. summer, wet vs. dry), the user will want to train CANARY separately on those data sets, providing one set of parameters for each season. This information can be recorded in Table 5

Table 6 - Preparing training data files

Season "A"	Data File Name:	
Station:	Starting Date and Time:	
	Ending Date and Time:	
Season "B"	Data File Name:	
Station:	Starting Date and Time:	
	Ending Date and Time:	

8.3 Choosing the Right Windows

The authors have found that window sizes between half of a day and one and a half to two days in length tend to increase the accuracy of the algorithms in predicting water quality. In general, shorter window sizes are more sensitive to changes in water quality and longer window sizes have fewer alarms. The authors recommend using the shortest reasonable window size to start with. Some reasons to start by using a window of one-day or longer include: lots of mixing of different water types at the monitoring location, daily operations that impact water quality, high changes in demand across the day at this location. Once chosen, calculate the number of time steps in this window using Table 6:

Table 7 - Calculate the window size

Window Size:	3600 minutes per day / Data Interval (in min.)	3600 . ( )	

8.4 Minimizing False Alarms

Minimizing false alarms without sacrificing the ability of the system to catch possible contamination events is the most difficult aspect of the training process. The best way to find an appropriate threshold value is to take a set of historical data, run one algorithm with several thresholds at once (just type space-separated number in the "threshold" box in the configuration manager) and then examine the results visually. Make allowances that a certain number of false alarms are inevitable, and find some "event," either operations or a simulated event added to the data, that must be caught regardless of the threshold.

Because of the ability of the clustering and pattern matching to use operations data to help limit false alarms, try not including the operations signals to start (set the "ignore changes" flag to "all"). Then, run CANARY with a set of thresholds, and look for the highest threshold that still catches the operational changes that are causing alarms. Once this value has been found, add the operations signals back into

the equation, and run the clustering algorithm on the results. This will allow CANARY to omit these typical events while maintaining sensitivity.

9 References

Bezdek, J 1981, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York.

Dunn, JC 1973, "A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters", *Journal of Cybernetics*, vol 3, pp. 32-57.

Gaffney, SJ 2004, "Probabilistic Curve-Aligned Clustering and Prediction with Regression Mixture Models", PhD Dissertation, Department of Information and Computer Science, University of California, Irvine.

Hall, J, Zaffiro, AD, Marx, RB, Kefauver, PC, Krishnan, ER, Haught, RC & Herrmann, JG 2007, "On-line water quality parameters as indicators of distribution system contamination", *Journal of the American Water Works Association*, vol 99, no. 1, pp. 66-77.

Hart, DB, McKenna, SA, Klise, KA, Cruz, VA & Wilson, MP 2007, "CANARY: A water quality event detection algorithm development and testing tool", *Proceedings of ASCE World Environmental and Water Resources Congress 2007*, ASCE, Tampa FL.

Klise, KA & McKenna, SA 2006a, "Multivariate applications for detecting anomalous water quality", *Proceedings of the 8th Annual Water Distribution Systems Analysis (WDSA) Symposium*, ASCE, Cincinnatti OH.

Klise, KA & McKenna, SA 2006b, "Water quality change detection: multivariate algorithms", *Proceedings of SPIE Defense and Security Symposium 2006*, Internation Society for Optical Engineering (SPIE), Orlando FL.

McKenna, SA, Hart, DB, Klise, KA, Cruz, VA & Wilson, MP 2007, "Event detection from water quality time series", *Proceedings of ASCE World Environmental and Water Resources Congress (EWRI) 2007*, ASCE, Tampa FL.

McKenna, SA, Klise, KA & Wilson, MP 2006, "Testing water quality change detection algorithms", *Proceedings of the 8th Annual Water Distribution Systems Analysis Symposium (WDSA)*, ASCE, Cincinnatti OH.

The MathWorks 2002, *Signal Processing Toolbox User's Guide*, 6th edn, The MathWorks.

US EPA 2005, "WaterSentinel Online Water Quality Monitoring as an Indicator of Drinking Water Contamination", EPA, U.S. Environmental Protection Agency, 817-D-05-002.

EPA 600/R-08/040A | March 2009 | www.epa.gov/ord

United States
Environmental Protection
Agency

Office of Research and Development
National Homeland Security Research Center
Cincinnati, OH 45268

PRESORTED STANDARD
POSTAGE & FEES PAID
EPA
PERMIT NO. G-35

Official Business
Penalty for Private Use
$300