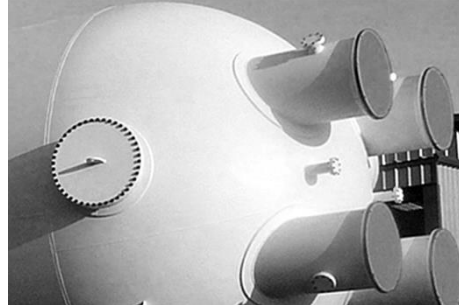


**CURTISS -  
WRIGHT**

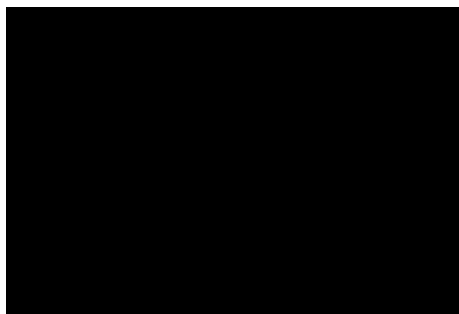
Nuclear Division



# R\*TIME Historian



Kevin Coble  
Jan 2020



# Historical Data in R\*TIME Version 15

- **R\*TIME Version 15 now has two methods for recording historical data.**
  - The archive system that has been present since version 9 of R\*TIME
    - Referred to as 'Legacy' archive throughout this discussion
  - A new system that uses 'Change Tracking' rather than periodic recording
    - Referred to as CTA or 'Change Tracking Archive' throughout this discussion

# 'Legacy' Archive Files

---

- **Defined size, both in length and number of points**
  - Disk space will be allocated for unused (spare) points at file creation
  - Fixed disk usage can be calculated and displayed at creation time
- **Fixed time frame (length)**
  - Can be circular, linear, or event (circular pre-event, linear post-event)
- **Fixed frequency**
  - Data stored every frequency cycle, whether it has changed or not
- **Only half the status word stored (extended status removed)**

# Change Tracking Archive Files

- **Directory instead of a single file**
  - One file for each point
- **Open-ended time frame**
- **No 'frequency' - all changes recorded with millisecond-level timestamps**
  - Timestamps set when value or status is stored
    - If from SOE, DAS time will be used
    - For most other data, SDZTIME will be read and used as the timestamp (SDZTIME is usually updated 20/second)
    - Data from external sources (such as wireless sensors) can provide the timestamp when the value or status changes are stored
- **Full status word stored**
- **Compression used to lower disk usage**

# Advantages of Each Method

Legacy	Change Tracking
Known disk usage	Often has lower disk usage for a fixed set of points/time
Tools more developed (backup, patch, etc.)	Data is timestamped to millisecond accuracy, not limited to file frequency
Lower disk head movement required	Retrieval for a small set of points (trends, reports) much faster, since each point has it's own file
	Can add/remove points while file is loaded
	Full status word available
	All times relative to a file creation time, allowing the time frame to be easily and quickly manipulated on a simulator.

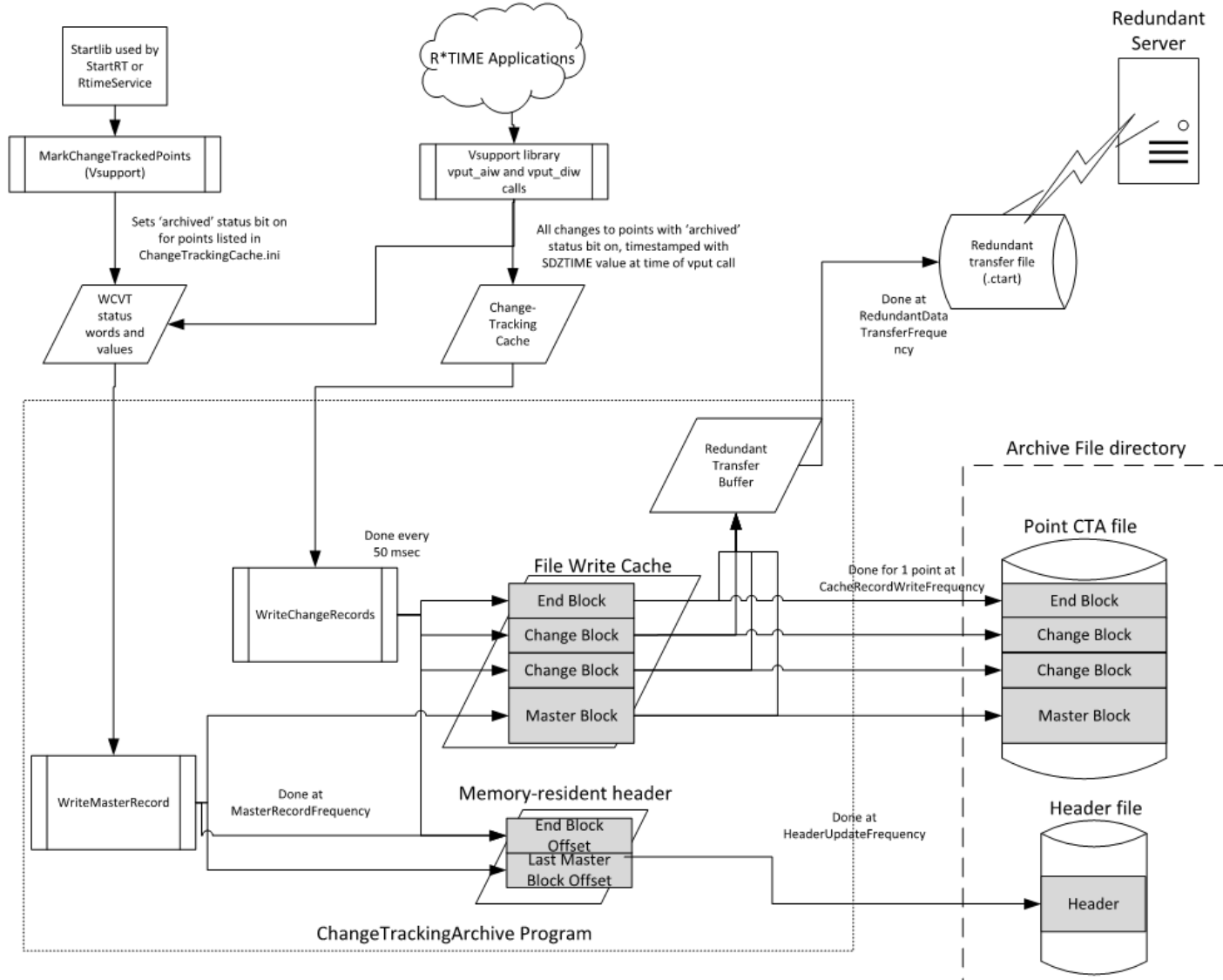
# Disadvantages of Each Method

Legacy	Change Tracking
Extra disk read on retrieval (all points must be read for the time frame, not just the ones requested)	Unable to be sure of maximum disk requirements (estimation calculations can be performed, discussed below). Base R*TIME can be configured to monitor disk utilization and activate alarms if desired.
	Due to required caching, a catastrophic error (i.e. power failure of system) can result in more data loss than with Legacy system. The maximum amount depends on configuration parameters and disk speed.

# Using Change Tracking Archive

- To effectively use the new CTA archives, it helps to understand what is happening ‘under the hood’
- The remainder of this talk aims to help you understand the internals enough to help set tuning parameters when CTA archives are created.
- The CTA\_Create display (used to create a new CTA archive) defaults all these tuning parameters to usable values, but you will often want to change them based on the size of the file and the system usage

# Change Tracking Archive Data Flow





# Change Tracking Cache

- **The first cache in that picture was the Change Tracking cache.**
  - Almost all 'vput\_w' calls will now check to see if the data being stored represents a change, and if the point being referenced is marked with the new status bit indicating it is being change-tracked. If these conditions are met, the change is put into the Change Tracking cache with a timestamp.
    - Timestamp passed in (SOE's), or from SDZTIME if not passed
  - There is a cache for each database and point type (the analog and digital cache are separate)
  - The cache is created at system startup, sized based on a configuration entry for each database in the ChangeTrackingCache.ini file
  - If this cache is full, new changes will be lost (the Vsupport library is not associated with CTA directly, and so cannot do anything about a full cache)
    - Therefore it must be sized to meet the worst-case number of changes in a CTA record cycle (which is currently set to 250 milliseconds)

# Change Tracking Archive

- **The ChangeTrackingArchive program periodically retrieves the changes from the Change Tracking cache and processes them into all active CTA archive files.**
  - There will be one copy of the ChangeTrackingArchive running for each database that requires archiving
  - The changes are put into a cache created by the program for each archived point
    - The change record is compressed by making the time and changed data (value or status) relative to a 'Master Block' that is periodically added to the cache – discussed in more detail later.
  - If this makes the cache full, it is written immediately to the point's associated disk file
    - Writing out-of-order like this will increase disk head movement, and will therefore degrade performance on non solid-state drives
  - The point caches are checked in a round-robin fashion, and the first N caches that have data are written to their associated disk files. N is a configurable value when the CTA archive file is created.
    - By going in order, we are attempting to minimize disk head movement
  - Value changes for analog points may be skipped if the value has not changed by more than a specified amount. This delta is a new R\*TIME version 15 database field.
    - If positive, the delta is treated as an absolute difference that must occur for the change to be recorded. If negative, the absolute value represents a percentage change

# Master Blocks

- **At file creation, activation, and at a definable periodic rate, the ChangeTrackingArchive program adds a 'Master Block' to each point cache.**
  - Contains a full time (still relative to the file time), not a short offset used for changes
  - Contains the full value (for analog points) and status words for the point at the time of the Master Block
  - All subsequent Change Blocks are relative to the previous Master Block in value, status, and time. This allows those blocks to be smaller, as they only have to contain the difference from the last master block
  - Due to the smaller amount of bits allowed for time in a Change Block, there has to be a Master Block no more than a day before the Change Block occurred. This means that Master Blocks must be written at least daily.

# Change Blocks

- **For each change processed, a Change Block is added**
  - The change block is relative to the previous master block, both in time, status, and (for analog points) value.
  - Time is a 20-bit number of milliseconds since the Master block – this allows for a little over 1 million milliseconds, or a little over a day for the maximum time between Master Blocks
  - The data is a list of ‘bit-chunks’ that are different between the changed value/status and the Master Block, in one of three flavors
    - 4 4-bit chunks
    - 3 8-bit chunks
    - 1 32-bit chunk
  - If the change cannot be represented by one of these bit-chunk lists, then two 32-bit chunk changes are chained together as a two-word

# Memory Sizing

- **File cache sizing is in units of ‘words’ – a 64-bit (8 byte) word**
- **A Master Block is a fixed size**
  - Two words for digital points (time and status words – digital values are part of the status word)
  - Three words for analog points (time, value, and status words)
- **A Change Block is one or two words**
  - If the relative time and compressed change bits can fit in 64 bits, one word is used, else the Change Block uses two words to store the change
  - A value of 1.1 words-per-change is often used when estimating cache/disk sizes
- **An End Block is one word**
  - Always one at the end of the used cache or disk space, unused cache or disk follows
- **The following formula can then be used to estimate the cache/disk size needed:**
  - Cache –  $(\text{MasterBlockRate} * \text{MasterBlockSize} + 1.1 * \text{Changes}) * \text{WriteTime}$ 
    - If write time is less than Master Block Rate, round up to 1 Master Block
  - Disk –  $(\text{MasterBlockRate} * \text{MasterBlockSize} + 1.1 * \text{Changes}) * \text{TotalTime}$

# Header File

- **Each CTA archive has a header file that records the current content locations of the point data**
  - Periodically updated with the location of the End block for each point in the CTA archive
  - At archive load time, the ChangeTrackingArchive must determine the location in each point file where the next data should be written. The Header file gives a good starting point for the search, as it has the last recorded end location for each point
    - Header file is not written on every change or Master block addition, so will not usually have the accurate end point for many of the points
    - But, it is much closer than starting at the beginning of each point file for the search, so will greatly speed up loading of a CTA archive
    - A higher Header write frequency will lessen the load time of an archive set, but will add to the disk write usage for the archive

# Redundant Data Transfer

- **If the CTA file is created as a redundant file, then all recorded data must be passed from the master server to the standby server**
  - This is done by accumulating all writes to point caches into a 'Redundant Transfer set'. This set is periodically (on a configurable time basis) written to a file to be transferred to the other server
  - On startup, the ChangeTrackingArchive program tries to transfer all pending Redundant Transfer sets to the other server
  - On startup, and periodically, the ChangeTrackingArchive program on each server searches for any pending Redundant Transfer sets that have been transferred and processes them into point caches, and ultimately the point archive files

# Managing the CTA archives

- **The CTA\_Monitor display is the main display for monitoring and configuring the Change Tracking Archive system**
  - Shows status of the Change Tracking cache for the associated database
  - Shows the status of the first 5 loaded CTA archive sets
    - Start and End times
    - Average and maximum point cache usage
  - Has buttons to interact with the system for that database
    - Create a new CTA archive
    - Load an existing CTA archive
    - Modify a loaded CTA archive
    - Unload and/or Deactivated a CTA archive



# Creating a CTA archive

---

- **The CTA\_Monitor has a 'New' button for creating a new CTA archive**
  - Activates the CTA\_Create display
- **The CTA\_Create display has all the options for a CTA Archive**
  - Point list
  - Cache sizes
  - Master Block rate
  - Header update rate
  - Redundant transfer rate
- **Demo**

# Archive File Converter

- **Converts off-line Legacy files into CTA files**
  - Starts with one or more Legacy files
    - Can merge multiple Legacy files into a single CTA file
    - Converter coded so that other source files can be added as needed
  - Uses the 'delta' database field (if available) for analog difference determination – the same as the CTA recorder
  - Creates a CTA archive directory with all points and a specifiable spare count, caching sizes, etc. (in case you want to activate the file)
    - Can be loaded or activated using the CTA\_Monitor display if put into an appropriate (configured) directory on an R\*TIME system with the database the Legacy file was recording from.
- **Takes about 10 sec/gigabyte of Legacy file to convert (subject to system disk access speed)**
- **Often results in a much smaller file. A 10:1 ratio is common, especially if there are many non-changing digitals in the Legacy file.**



***CURTISS -  
WRIGHT***