lccCheckDateTimeDrift Manual

Contents

Description 1
lccCheckDateTimeDrift 1
Installation 2
Using 2
Starting 3
Starting with an alternate Logic File 3
Logic File Description/Syntax 3
Logic File 3
Logic File – Examples 7
This example will pull records from two source files, and send email alerts if the Drift on any record is more than 120 seconds. 7
Definitions 8
Modifications 8

Description

This document describes how to use the lccCheckDateTimeDrift

lccCheckDateTimeDrift

is a Command Line program created to check the Date/Time values of multiple records and report how much difference (Drift) exists between them. The program can also be set with a 'Drift Threshold' to send an Alert if any of the records match/exceed that threshold (Drift).

Update 20201028: you can now use the key lcc:NTPServer to have servers checked against a (Network Time Protocol) NTP server.

The program is controlled by a Logic File.

Program Logic

- reads the Logic File
- loads all Source Paths
- loads all valid Records (lines)
- compares all records against the current date/time
- sorts records by the most difference
- show the record with the most 'Drift'
- reports all that Drifted more than the Drift Threshold

If you need a tool that can grab the Date/Time from multiple systems at the same time, we recommend:
lccAsyncCommands

Installation

- copy the lccCheckDateTimeDrift.exe to a folder
- create Logic File
- run lccCheckDateTimeDrift.exe

Using

Prerequisites: configure at least one Logic File.

The program will default to using a Logic File called:
lccCheckDateTimeDrift-logic.txt
in the same folder as the program. You can change to use by specifying a different one on the command line.

Starting

- run lccCheckDateTimeDrift.exe

Starting with an alternate Logic File

- Syntax: lccCheckDateTimeDrift.exe lcc:logicPath [Logic File]
- Example: lccCheckDateTimeDrift.exe lcc:logicPath ourLogic.txt

Logic File Description/Syntax

A Logic File is a Tab delimited text file. Any lines not recognized as a valid Key/Value pair, will be ignore and can be used as remarks/other.

The Logic File uses the syntax.
Syntax: **[Key]** *[tab]* **[Value]** … *[tab]* **[Value]**
Example: lcc:key value

Any extra tabs in a line after the expected ones are considered remarks and will be ignored. This is a nice way to document specific Key settings (see Log Levels in the Logic File example(s) for reference). Also, if you place a tab before a line, that will essentially make it a remark and will be ignored, which makes using/not using logic without removing quicker.

Any line not starting with an expected key is ignored, which makes placing remarks/formatting the logic easy.

Logic File

lcc:logLevel *(optional)(none to many per Logic File)*

Specifies what log information will be logged/shown. Provide a new line for each level desired.

Debug Levels

1 Checking Logic: displays checks against the Logic File.
2 Process Logic: display the logic being processed.
3 Stats: All Records: display each command process.

4 Stats: Largest Drift: display results from each command/thread.
5 Stats: Records Processed: display results from each command/thread.
6 Stats: Drift Threshold Met: display results from each command/thread.

Syntax: lcc:logLevel *[tab]* [#]

Example: lcc:logLevel 2
Example #2 (multiple): lcc:logLevel 2
lcc:logLevel 4

lcc:logPath *(optional, but highly recommended)(one per Logic File)*

The path/root name of the Log File. The program will place the '.log' extension automatically and will also append a Year/Month date.

lcc:NTPServer *(optional)(one per Logic File)*

If provided, the servers will be checked against the NTP server date/time. If not provided, the servers will be compared against each other.

Syntax: lcc:NTPServer *[tab]* [...server...]
Example: lcc:NTPServer someNTPServer.org

lcc:sourcePath *(mandatory)(one to many per Logic File)*

What file(s) contain the records (lines) to compare. You can provide more than one.

The file record(s) must have their columns separated by Tabs. Records must contain at least the Machine Name and Date value (in any column), but, can contain other columns. You will use other Keys to filter which columns to use.

Here is an example source record:

server1 20161006 15:00:20.941 Current time at \\server1 is 10/6/2016 3:00:21 PM

In this sample record, we would choose column 1 for the 'Name', and column '4' for the Date/Time. See keys 'lcc:sourceColName', 'lcc:sourceColDateTime'.

Any records (lines) not meeting minimum requirements will be ignored, i.e. you can provide files that have other lines in them.

In this case, we would define using columns 1 and 4 (using other keys).

Syntax: lcc:sourcePath *[tab]* [Path]
Example: lcc:sourcePath ourRecords.txt
Example #2: lcc:sourcePath \\server\share$\folder\ourRecords.txt

lcc:sourceColMin *(mandatory)(one per Logic File)*

What are the minimum number of columns a source record must have, after being split on Tabs, to be considered valid.

Syntax: lcc:sourceColMin *[tab]* [#]
Example: lcc:sourceColMind 4

lcc:sourceColName *(mandatory)(one per Logic File)*

What column # contains the Name of the device/other.

Syntax: lcc:sourceColName *[tab]* [#]
Example: lcc:sourceColName 1

lcc:sourceColDateTime *(mandatory)(one per Logic File)*

What column # contains the Date/Time value.

This value should contain a Date/Time value somewhere in its value, like:

Current time at \\server1 is 10/6/2016 3:00:21 PM

This value will be split on all spaces ' ' to find Date/Time parts.

Syntax: lcc:sourceColName *[tab]* [#]
Example: lcc:sourceColName 4

lcc:dateTimeColMin *(mandatory)(one per Logic File)*

How many minimum columns (split on space) the Date/Time value, from key 'lcc:sourceColDateTime' should have, like:

Current time at \\server1 is 10/6/2016 3:00:21 PM

Has eight (8) columns.

Syntax: lcc:dateTimeColMin *[tab]* [#]
Example: lcc:dateTimeColMin 8

lcc:dateTimeColStart *(mandatory)(one per Logic File)*

Which column in the Date/Time value is where the actual Date/Time starts., from key 'lcc:sourceColDateTime', like:

Current time at \\server1 is 10/6/2016 3:00:21 PM
(1) (2) (3) (4) (5) (6) (7) (8)

Has the Date/Time start at column 6, when split on space. The program will start at that column and grab the next 3 columns (i.e. 4 columns). These four columns should have a value that has a syntax of:

MM/DD/YYYY HH:MM:SS AM/PM

Syntax: lcc:dateTimeColMin *[tab]* [#]
Example: lcc:dateTimeColMin 8

lcc:driftThreshold *(optional)(one per Logic File)*

What is the threshold at which a Date/Time value is considered incorrect/unsafe.

Syntax: lcc:dateTimeColMin *[tab]* [#]
Example: lcc:dateTimeColMin 8

lcc:SMTPServer *(optional)(one per Logic File)*

Email server to send email alerts to.

Syntax: lcc:SMTPServer *[tab]* [Name]
Example: lcc:SMTPServer ourEmailServer.edu

lcc:SMTPPort *(optional)(one per Logic File)*

Port to the SMTP server .

Syntax: lcc:SMTPPort *[tab]* [#]
Example: lcc:SMTPPort 25

lcc:SMTPFrom *(optional)(one per Logic File)*

Provides the from email address that messages are addressed from.

Syntax: lcc:SMTPFrom *[tab]* [Address]
Example: lcc:SMTPFrom ourmailbox@ourcollege.edu
Example #2: lcc:SMTPFrom 'our Mailbox' <ourmailbox@ourcollege.edu>

lcc:SMTPUser *(optional)(one per Logic File)*

If email server doesn't allow Anonymous authentication, this Key provides the User credentials id.

Syntax: lcc:SMTPUser *[tab]* [Id]
Example: lcc:SMTPUser ouruser

lcc:SMTPPassword *(optional)(one per Logic File)*

If email server doesn't allow Anonymous authentication, this Key provides the user credentials Password.

Syntax: lcc:SMTPPassword *[tab]* [Id]
Example: lcc:SMTPPassword ouruser

lcc:SMTPRecipient *(manatory)(one per Logic File)*

What recipient should receive an email alert. Provide this for each recipient.

Syntax: lcc:SMTPRecipient *[tab]* [Id]
Example: lcc:SMTPRecipient ourteam@college.edu
Example: lcc:SMTPRecipient specificperson@site.com

Logic File – Examples

This example will pull records from two source files, and send email alerts if the Drift on any record is more than 120 seconds.

```
lcc:logPath     lccCheckDateTime-log

        lcc:logLevel  1       Check Logic
        lcc:logLevel  2       Process Logic
        lcc:logLevel  3       Stats: All Records
```

```
        lcc:logLevel   4       Stats: Largest Drift
        lcc:logLevel   5       Stats: Records Processed
        lcc:logLevel   6       Stats: Drift Thresholds Met

lcc:sourcePath      ourDateTimeRecords.txt
lcc:sourcePath      \\server\share$\folder\moreRecords.txt

lcc:sourceColMin    4
lcc:sourceColName   1
lcc:sourceColDateTime       4

lcc:dateTimeColMin  8
lcc:dateTimeColStart 6

lcc:driftThreshold      120

lcc:SMTPServer      mail.server.edu
lcc:SMTPPort  25
lcc:SMTPUser ouruser@college.edu
lcc:SMTPPassword    ...
lcc:SMTPFrom        'lccCDTD Alerter' <ouruser@college.edu>
lcc:SMTPRecipient   ourteam@college.edu
lcc:SMTPRecipient   specificperson@site.com
```

Definitions

Modifications

NAME	DATE	MODIFICATION
David Mielcarek	10/7/2016	Created
David Mielcarek	10/28/2020	Added key lcc:NTPServer

End of document