Memory Automatic

Colledia Control Documentation

By Paul Wilkins

Current Version : 1.58 (16th January 2008)

Index

Overview

This application was originally written as part of the new Colledia Control system installed for Television Studio 1 at Television Centre in September 2005. The automatic however has been written so as it can be used in any Colledia Control system. It stores snapshots settings of the current states of defined Colledia devices. These 'snapshots' can then be recalled to re-set these devices or reset to a defined "default" state. The devices can be Routers, GPIs and Infodriver values or a mix of any or all three.

Functionality

Designed to be as generic as possible, the Memory Automatic can store settings for a mix of core Colledia devices – routers, gpis and infodriver slots. Users can then save current settings into specified memories – either a single "Snapshot" memory or to a user specified name.

Memories can then be recalled either in their entirety, or as a subset. There is also the option of resetting the defined devices to a given default (eg bars to monitors, tone or silence for audio router destinations, infodriver slots all to 0……). The recall of the default settings can also be applied to either the whole or subsets of destinations, infodriver slots etc.

The Debug window of the application gives feedback on how many memories are stored, which was the most recent memory stored or recalled, together with debug or error messages from the application.

The application hooks into an infodriver – must be running for this application to run, and registers for revertives for all of the devices specified in the dev ini file for the application. Although the application is 4.5 compliant it obtains all its initialisation data from the device ini file of the infodriver it is hooking into – so that it can be used in older v2 systems if desired.

Memories are actually stored as comma delimited lists using the external infodriver slots 81 thru 4095 inclusively. Slot 80 holds the number of memory slots currently used. The contents of these slots are linked to named memories, the name of which is stored in the equivalent numbered entry in db*n* (whichever ini file specified in the automatic's settings by the entry - DATA_KEYNAMES_DATABASE) of the device ini file.

On start-up the Memory Automatic reads in the ini file definitions, registers for revertives, and checks to see that there is space for the default memory definition and for the "Snapshot" memory. Any other existing memories stored are also read in and kept internally. As memories are saved or deleted, so infodriver slots and db*n* entries, are used or cleared appropriately.

As mentioned, memories are stored using comma delimited lists in the infodriver slots. However should commas be present in the data of memories being stored this would cause problems when recalling. The entry "**DeLimiterCharacter** " in the [MemoryAuto] section of the dev_ini file can be set to another character if required :

```
[MemoryAuto]

DebugMode=1

LogMode=0

CommandInterval=50

DeLimiterCharacter=,    --- the comma is used by default for memory lists
```

Note – if choosing a different character please do not use | (vertical pipe) as this is used by the automatic as a special character. This is really only an issue when storing textual data in the memory system.

Configuration

The automatic requires careful configuration to function fully and correctly. For the purposes of discussion the following example will be used :

```
[DEFINED_DATA]
DEFINED_1=TESTCONFIG

[TESTCONFIG]
DATA_KEYNAMES_DATABASE=2
DATA_DEFINITION_1=RTR
DATA_DEFINITION_2=RTR
DATA_DEFINITION_3=GPI
DATA_DEFINITION_4=INF

RTR_1_DEVICE=101
RTR_1_DESTS=1,2,3,4,5,6,7,8,9
RTR_1_DEFAULT_SRC=1,1,1,1,2,2,2,2,2

RTR_2_DEVICE=102
RTR_2_DESTS=RANGE,1,99
RTR_2_DEFAULT_SRC=256,256

GPI_3_DEVICE=103
GPI_3_OUT=1,2,3,11,12,13
GPI_3_DEFAULT_STATE=0,0,0,1,1,1

INF_4_DEVICE=104
INF_4_SLOTS=RANGE,1001,1049
INF_4_DEFAULT_DATA=0,0
```

The current version of the application allows for only 1 DEFINED_DATA type per application instance. The defined name is then used as a section heading containing all the required data fields for that definition, so in the example TESTCONFIG is the defined name and appropriate section heading.

Within this named section heading "[TESTCONFIG]" is firstly, the entry to determine which device ini database file to be used to hold the memory entry names. This can be any of the 10 database files, though bear in mind database 0 and 1 can only hold entries up to 16 characters long.

There then needs to follow entries for DATA_DEFINITIONS. The application allows for up to 5 definitions (or devices) at present. These definitions can be any combination of RTR, GPI or INF. In the example 4 devices are defined.

There then must be the appropriate groups of 3 line entries for each of the type of device defined. So…

Routers require entries: **RTR_x_DEVICE, RTR_x_DESTS** and **RTR_x_DEFAULT_SRC.**

GPIs require entries: **GPI_x_DEVICE, GPI_x_OUT** and **GPI_x_DEFAULT_STATE.**

Infodrivers require entries: **INF_x_DEVICE, INF_x_SLOTS** and **INF_x_DEFAULT_DATA.**

Please be aware of the specific (and slightly differing) entry names for each different type of core device type.

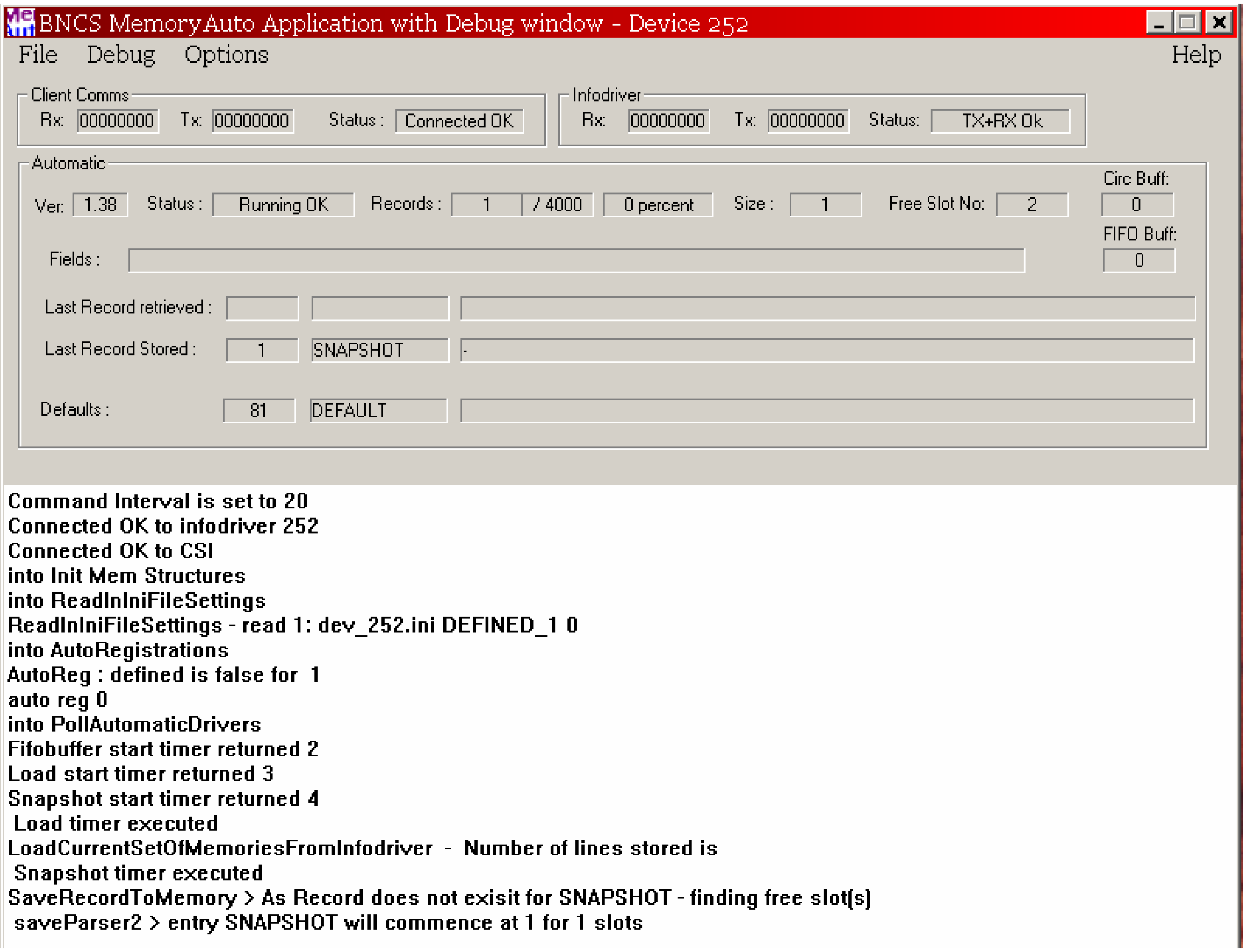
Device numbers are in the range 1 to 999.

Router Dests, GPI Outs and Infodriver Slots can be listed in two ways. Either as a discrete list e.g 1,2,3,4 or with the use of the Keyword "RANGE". If the latter is used then the term RANGE needs to be followed by the start and end of the group, e.g. RANGE, 1, 99 - means all dests, gpis or slots from 1 to 99. (simple really !).

Then the list of default sources, states or data must follow in the same manner as the preceding Dests, Outs or Slots. Where a discrete list of destinations was stated then there must be an equally long discrete list of default sources.

Where the Range parameter is used you can only define the default source or state for the whole group. So from our example – the second router entry with RANGE,1,99 - the default source is given as 256,256 – meaning that when set to default, all these 99 destinations will have source 256 routed to them.

Debug Window and Error Messages



Client Comms
Rx: 00000000 Tx: 00000000 Status : Connected OK

Infodriver
Rx: 00000000 Tx: 00000000 Status: TX+RX Ok

Automatic
Ver: 1.38 Status : Running OK Records : 1 / 4000 0 percent Size : 1 Free Slot No: 2
Circ Buff: 0
FIFO Buff: 0
Fields :
Last Record retrieved :
Last Record Stored : 1 SNAPSHOT -
Defaults : 81 DEFAULT

```
Command Interval is set to 20
Connected OK to infodriver 252
Connected OK to CSI
into Init Mem Structures
into ReadInIniFileSettings
ReadInIniFileSettings - read 1: dev_252.ini DEFINED_1 0
into AutoRegistrations
AutoReg : defined is false for  1
auto reg 0
into PollAutomaticDrivers
Fifobuffer start timer returned 2
Load start timer returned 3
Snapshot start timer returned 4
 Load timer executed
LoadCurrentSetOfMemoriesFromInfodriver  -  Number of lines stored is
 Snapshot timer executed
SaveRecordToMemory > As Record does not exisit for SNAPSHOT - finding free slot(s)
 saveParser2 > entry SNAPSHOT will commence at 1 for 1 slots
```

Start up of the Automatic showing expected debug messages

The application window shows the current state of the automatic with network traffic counters, number of memory records, most recent record stored and retrieved, the default definition, number of entries in the internal buffers (circular and FIFO).

Error Debug messages will detail the function in which they originated from and some relevant data to aid debugging any problems.

Use of the Memory Automatic from Panels

Commands to the Memory Automatic may be written via slots 1,11,21,31 or 41 of the linked infodriver, and any answers to said commands will be returned from the command slot + 1, i.e. 2,12,22,32 or 42.

Slot 80 – holds the number of memories current stored.

Slot 81 – 4095 will hold any memory data, as a comma delimited list. When retrieving a memory this data is parsed in order and applied to the devices according to the order specified in the configuration.

So for the example configuration listed above, the CDL would consist of 114 numbers and 49 strings, where the first 9 numbers will relate to the 9 destinations for router device 101, the next 99 entries for router device 102, 6 gpi states etc….

Clearly where large ranges of defined devices exist it will mean that many slots may be used to store a single memory. The application keeps track of how many slots are used per memory stored, retrieved or deleted.

COMMANDS into the MEMORY AUTOMATIC : via slots 1,11,21,31,41

SAVE - save a memory - provide the name of the memory : SAVE|<memory name>.

Please use a vertical pipe | to delineate start of the saved memory name.

(Please do NOT enter the command SAVE|DEFAULT).

Memories are either added, if new, or updated if they already exist in the system.

SNAPSHOT – or SAVE|SNAPSHOT – will save the current known revertives for devices defined.

RECALL – Recall an existing memory :

There are two styles of recall command where either the entire memory is recalled or a subset of the memory :

RECALL|<memory name> - for a full recall

Or for a subset

RECALL|<memory name>|DEV=xxx,a,b,c,d,e,f,g,h or

RECALL|<memory name>|DEV=xxx,RANGE,a,b

Subsets of the defined data destinations, slots etc can be recalled. Multiple device subsets can also be recalled for example :

RECALL|<name>|DEV=101,1,2,3,DEV=102,50,51,52,53,DEV=102,RANGE,61,89

will recall subsets of destinations for routers 101 and 102.. The use of RANGE means very long lists are not required in the recall command.

DELETE – Deletes an existing memory : DELETE|<memory name>. Please note that neither the DEFAULT or SNAPSHOT memories can be deleted. A message will either confirm the deletion or say the memory was not found.

These commands can easily be linked to buttons on Colledia Panels. In the TC1 system the MEMORYFUNCTIONS panel is a generic panel to enable the save, recall, deletion, snapshot functionality of the memory automatic. This panel sends commands via slot 1 and get answers back from the automatic via slot 2.

In the TC1 Colledia system various subsets were defined in ObjectSettings.xml and used by this panel to enable subsets of vision or audio groups to be recalled as required.