

BBC Custom Control Library 12

Trace Control

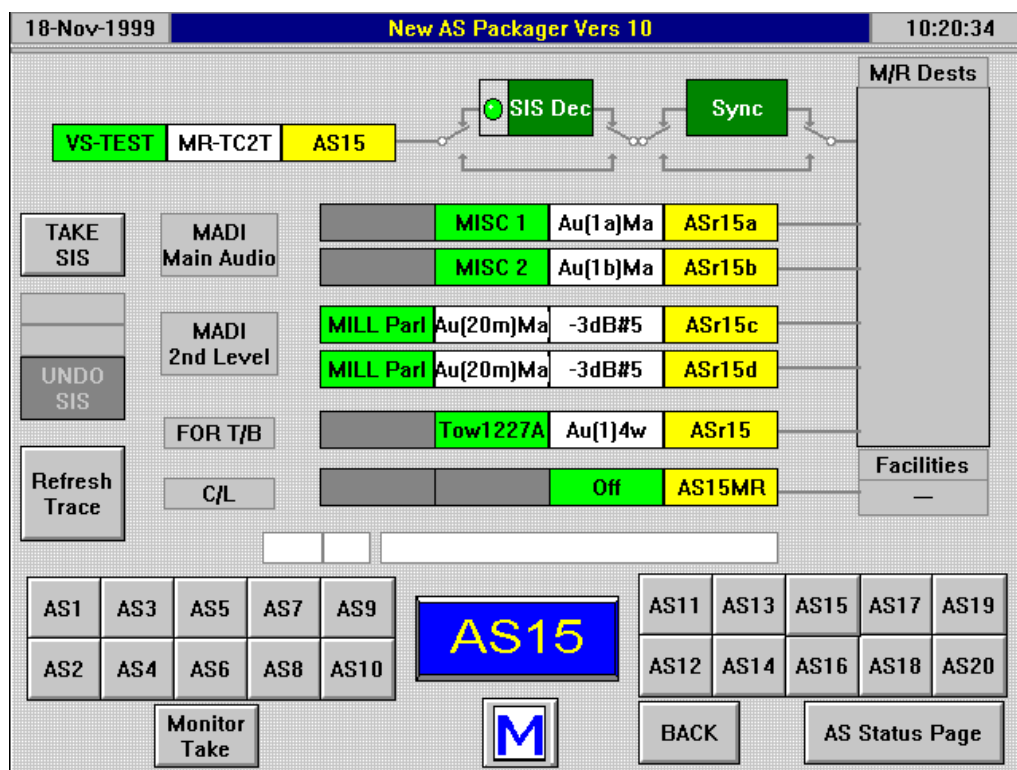
BNCS Software
From an original component written by David Yates
Enhancements by Steve Jensen
© Copyright Atos IT Solutions & Services 2014

Description

BBC_CC12.DLL is a Borland custom control DLL contains two controls - a router Tracebar and TraceButton.

The trace control is a control that displays the route a signal takes through a BNCS system. This enables you to trace back on a single control back to the earliest known point on a system. A destination router and index is given and the control works out the route back. Routes are determined using router names – this simply requires the names are consistent between devices on a system e.g. AS1 is a destination on router 22 and a source on router 62. No complex setup files are required to specify how devices are connected together.

This control appears as a single ID and is a "Smart button" in that it registers with CSI separately to the panel.



This example shows 7 BBC_CC12 Tracebar controls on one panel to trace the vision, 4 levels of audio, talkback and control line for one packaged source.

Setup

This control requires a settings file `c:\windows\bbc_cc12.ini`. This file contains a list of the routers that this Tracebar control needs and how they are associated. This file simply shows that there are *some* connections between routers, the actual connections are determined by router names. This is used for every instance of `BBC_CC12.DLL` and should contain all the routers that the Tracebar is to trace routes through.

```
[Tracebar]
Index_001=22,62,650,314,660,1
Index_002=62,22,650,314,660,325,
Index_003=650,22,1,62,314,660,
Index_004=314,62,22,650,660,325,
Index_005=1,22,650,
Index_006=322,21,
Index_007=21,322,
Index_008=325,62,2,
Index_009=660,650,22,
```

Note: For the V3 the path to `bbc_cc12.ini` file is specified by file: `bncs_config.ini` [Config] "ConfigPath" entry. For example: `ConfigPath=C:\BnCS\Windows`

The *Index_nnn* entries are simply an incrementing list of settings that the control reads until it finds a blank line.

It is not ideal for these parameters to be held within a separate ini file but this control works across a system. Were associations between routers to be changed all that needs changing would be this file once. If this information were to be held in each panel, each panel would need to be updated.

How it works – an example

Using the example `bbc_cc12.ini` file above.

In the above example, AS15 is destination 84 on router 22.

This destination is polled to get the current **source**, MR-TC2T.

Using the name, the control looks to see if MR-TC2T is a **destination** on device 22 (to see if it is a reentrant device). If it doesn't find this name in this database then it does through the list of devices for router 22 (the *Index_001* entry in the ini file above).

The destination database for device 62 is checked for the name MR-TC2T - this won't be found for this example. This process continues for the whole list of device numbers in the *Index_001* entry. MR-TC2T will be found in the destination database for device 1.

The **source** (VS-TEST) feeding destination MR-TC2T is added to the trace bar.

The process is then repeated looking for VS-TEST as a **destination** using the list of routers for device 1 (*Index_005*). This won't be found (this is the start of the route) and so the Tracebar stops.

When the list of routers is exhausted for a particular starting point the trace bar routine stops.

Single button and “Source Trace”

Version 1.00.02 and later of the trace bar has three additional styles of trace bar. As well as the original there is a single button version that just displays the leftmost cell. This shows you just the earliest source for whatever destination you give it. The Resource Workshop icons for these controls are:



- multiple horizontal destination-originated buttons



- single horizontal destination-originated button

The other two styles are the same except instead of giving it a destination to trace back from you give it a source. The Resource Workshop icons for these controls are:



- multiple horizontal source-originated buttons



- single horizontal source-originated button

You can't use source and destination trace from the same button – if you need this functionality use two buttons and fly the appropriate one onto the right place on the dialog.

Vertical Oriented Trace bars

Starting with Bbc_cc12.dll, version 1.3.0 and Bncs_cc12.dll, version 3.3.0 support for “vertical” orientation trace bar's is added. The vertical feature set is accessed by FOUR additional design time icons, similar to the above except a small “v” added to the middle right of the icon face.



- multiple vertical destination-originated buttons



- single vertical destination-originated button



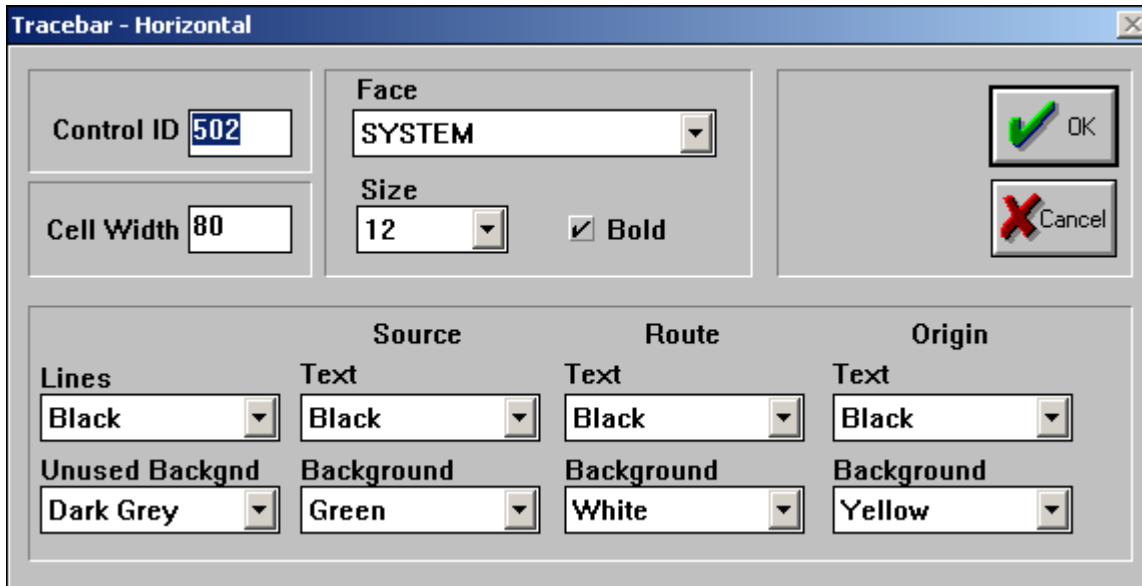
- multiple vertical source-originated buttons





- single vertical source-originated button

Setup

The width of each cell is determined at design time, it will resize horizontally at run time to an integral number of cells



The dialog box 'Tracebar - Horizontal' contains the following settings:

- Control ID:** 502
- Cell Width:** 80
- Face:** SYSTEM
- Size:** 12
- Bold:** ☒
- OK:** 
- Cancel:** 

	Source	Route	Origin
Lines	Text: Black	Text: Black	Text: Black
Unused Backgnd	Background: Green	Background: White	Background: Yellow

The font and size, colour of the text, background and the lines are all defined at design time. Version 1.5.0 of bbc_cc12.dll (and later) and version 3.5.0 of bnCS_cc12.dll (and later) add the "Bold" font control (tick box).

It is possible to define the colours separately for:

- destination text, (the right hand side cell in the Tracebar)
- the source text (the leftmost cell in the Tracebar)
- the route text (the cells between the source and destination cells)
- the unused cells
- the lines that define the cells

The example panel shown above uses these different colours. You don't have to make them all different !

Writing to the control

The starting destination the control uses is set by writing a comma delimited string to the control of the format DEVICEID,INDEX

e.g. 22,1

this is destination index 1 on device 22.

Notification from the control

When one of the cells of the control is pressed the parent is notified i.e. in applcore the stringtable line corresponding to the ID of the control is executed.

When the text is read from the control the text returned is dependant upon the last cell pressed. i.e. in the above example if BD13 was pressed last then "BD13" will be the text returned from a Text Get.

TraceButton

This control has identical functionality to the TraceBar except that there is only a single "button" – effectively only the leftmost button of a TraceBar control. When the destination is set on this control the text will flash briefly as the route back is found. Return value from the control is the text as displayed.

Important Note

It is possible in certain circumstances to cause infinite loops using this control. (I know this cause I did it.....).

DO NOT get this control to update as a result of a revertive. This control itself works by polling devices individually as it works back through the route. It is therefore possible to set this control to poll which then causes a revertive which causes the control to poll etc etc etc. With CSI cache off this causes a constant stream of poll messages onto the network. With CSI cache on this causes the machine to hang.

Because of the number of times this control polls for information it is worthwhile (but not essential) that CSI is running with it's cache turned on.

Trace Timer and Register Monitoring Modes of Operation

Starting with version 3.04.00 the trace library offers trace timer and router reverts modes of operation.

Set Timer trace – "ST, <PollingTime>, <DeviceID>, <Index>

This command sets a timer running - within the TraceBar library - which performs a periodic update trace of the router indicated by the standard parameters **<DeviceID>** and **<Index>**. The timer will poll at the **<PollingTime>**. Minimum 1,000ms, maximum 10,000ms.

Delete Timer – "DT" Delete Timer

This command deletes / stops the timer update trace set by the previous ST command.

Example:

```
"TP 'ST,3000,5,15' 1 503   Start Timer polling dev 5, slot 15, panel 1, ID 503
"TP 'DT,5,15' 1 503       Delete & stop timer polling on panel 1, ID 503
```

Register for Revertives – "RR" Revertives

This command mode registers for revertives on the indicated by the standard parameters **<DeviceID>** and **<Index>**. When a revertive is received by the trace bar (indicating a route change) the trace bar performs a trace.

UnRegister for Revertives – "UR" UnRegister

This command disables the previous RR command, i.e. unregisters for revertives on the indicated router.

Example:

```
"TP 'RR,5,15' 1 503   Router Register for revertives, panel 1, ID 503
"TP 'UR,5,15' 1 503   Router UnRegister for revertives, panel 1, ID 503
```

Info Driver Support

Starting with version 3.05.02 the trace library offers registration for Info Driver revertives mode of operation (with the LAWO router for example). The access to the registration for an Info Driver is of the same form as the for a standard Router – with a small change of command syntax.

For a Router use:

```
"TP 'RR,5,15' 1 503   Router Register for revertives, dev 5, slot 15, panel 1, ID 503
```

For an Info Driver use:

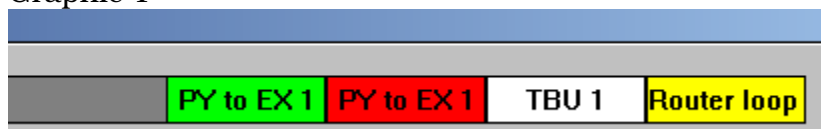
```
"TP 'IR,5,15' 1 503   Info Driver Register for revertives, dev 5, slot 15, panel 1, ID 503
```

Trace Loop Detection

Starting with version 3.06.00, the V3 (32bit) version of CC12 (bncs_cc12.dll) supports detection of router loops (the 16bit library bbc_cc12.dll does not include this feature). Prior to version 3.06.00 router loops (where the source is connected to the destination) would cause the CC12 to continue to trace indefinitely.

The following two graphics demonstrate the proposed graphical indicator triggered when a loop is detected. In the example the destination "PY to EX 1" has found that it is connected to source "PY to EX 1". This causes the tracing to stop and the library highlights the destination pane and toggles between BLACK on RED and RED on BLACK (foreground / background colours).

Graphic 1



Graphic 2



Acknowledgment

Much as I'd like to take credit (and large pay rises) for this control, the original concept for tracing routes using router names comes from Mr Peter Nash.

Revision History

- 04-Jan-1999 – Original issue V2 BBC_CC12.dll “trace” library. Dave Yates
- July 2005 – Version 3.00.00 – V3 port to 32bit (file: Bncs_cc12.dll). Steve Jensen
- March-2008 – Version 3.00.01 – Changed maximum number of routers from 32 to 64. Steve Jensen
- February-March-2010 – BBC_CC12.DLL, version 1.3.0 & Bncs_cc12.dll, version 3.3.0 added "vertical" trace bar support. Steve Jensen
- June-2011 – Starting with BNCS_CC12.dll version: 3.04.00 (V2 library Bbc_cc12 not updated) – Added timer-based and register for router reverts Tracebar monitoring functionality. Added documentation for the vertical oriented trace bars. Steve Jensen
- August-2011 – **Test** Release Version 3.04.03, dated: 15-August-2011 various bug fixes associated with the “register for router reverts” Tracebar functionality.
- Oct-2011 - Enhanced – added font “Bold” feature. bbc_cc12.dll version: 1.5.0, dated: 28-Sept-2011, and bncs_cc12.dll version: 3.5.0, dated: 18-Oct-2011
- August-2012 – Enhanced – Added support for Info Driver registration to the V3 bncs_cc12 library to support LAWO router scheme in W1/North project, Release Version: 3.05.02, dated: 14-Aug-2012
- November-2012 – updated bbc_cc12.dll to release version 1.5.1, updated version date and version number and rebuilt bncs_cc12.dll version 3.05.03, dated: 18Nov2012
- August-2014 – Support for loop detection from bncs_cc12.dll version 3.06.00 and forward. Updated Timer limits.
- September-2014 – Correction for several bugs in the trace facility regarding registration & re-registration – version: 3.06.01
- October-2014 – Correction for issues in Get Text support – version: 3.06.02