

# 2cSNA<sup>®</sup> – Modern Network Management

## Technical Brief

### Overview

**2cSNA<sup>®</sup>** is an application designed to facilitate a specific downsizing project in those z/OS installations that are no longer in need of all the functionality provided by the classic network management products.

The changes in SNA network architectures due to the removal of Front End Processors (FEPs or NCPs), Terminal Cluster Controllers (such as 3172/4) and the ubiquitous presence of an IP infrastructure have made quite a few functions/tasks of SNA network management obsolete.

Yet users are still causing SNA sessions to be established – there is no doubt about that. In addition to this, Enterprise Extender itself is causing SNA activity that needs to be controlled. Therefore, some functionality of network management is still needed for SNA, although to a much lesser extent.

In addition to this, network management personnel are reluctant to drive their remaining SNA legacy network without any control instance at all and are traditionally leery of using MVS consoles or other vehicles for their commands.

Meanwhile, cost control mandates a thorough examination of all areas in which a redesign or re-orientation might save significant amounts of money.

## Functions of 2cSNA®

### NICF – Network Interrogation and Control Facility

Although SNA network control is no longer the main task of the personnel involved, the remaining interaction with VTAM and the system, as well as with the TCP/IP stack is still quite intensive and thus this functionality is a prime requirement.

**2cSNA®** NICF provides one or more “command console” panels for entering commands to the system, displaying information and even for seamlessly executing your own **2cSNA®** REXX programs.

These console panels can be created and removed at will and can be “associated” with a specific “home” system. Commands can be targeted against this home system or any other system known to **2cSNA®** – within the same sysplex.

An explanation of SNA/VTAM sense codes as well as many shortcut commands for often used functions is included.

This part of **2cSNA®** can be used in a focal point manner and obviates the need to physically log on to each system in parallel.

### SSHF – Session Status and History Facility

A central part of **2cSNA®** is its ability to gather Session Awareness Data (SAW) from VTAM and to display this to the user.

The highlight of this functionality is the ability to search through the historical data for sessions that have ended some time ago and to inspect the progress of these – especially in the case where such a session might have ended abnormally.

### SSTF – Session Setup/Termination Trace Facility

A unique feature of **2cSNA®**, SSTF allows the user to view the beginning and end of a session, even historical ones that ended some time ago, in terms of individual trace records.

Therefore, the complete sequence of events at session initiation time and prior to session termination is recorded in trace records and are available for inspection.

These trace records are shown in the same manner as the Award Winning AnSyNova Product **2cIP®** – in overview format or fully decoded down to the VTAM structures contained therein.

You can see the BIND, the BIND response, the session termination and control vectors/sense codes associated with this clearly and concisely.

## MARF – Message Assignment and Routing Facility

Another unique feature of **2cSNA**<sup>®</sup>, MARF allows the user to set up one or more panels which will receive messages from the system based on their Route-Codes, Message Ids and/or Jobnames. This makes it possible to read console messages from the system and thereby take part in operating the system.

The technical idea behind this is to keep an EMCS console open at all times in order catch messages from the system – instead of only activating the console for entering commands.

2cSNA performs the job of separating the captured messages stream and routing the messages to their individual panels. It also sets up the needed routes-codes for the EMCS console intelligently as the mathematical SET-UNION of all route-codes desired by the user.

Therefore it is possible to view CICS messages in one panel and (for example) IMS messages in a second panel. Each time the user presses ENTER, all panels are updated simultaneously with the messages destined for them.

## Some technical details

### The data gatherer

**2cSNA**<sup>®</sup> consists of an assembler coded started task, the **data gatherer** that utilises the VTAM CNM interface via the ISTDCLU to gather session awareness and session trace data. This interface is documented by IBM in

*z/OS Communications Server : SNA Programming : Chapter 12. Coding for the communication network management interface*

The data gatherer needs to be started separately on every system you wish to manage and needs to be in an APF authorized environment.

It stores accumulated SAW information in ECSA – typically you would reserve about 5 to 10 Megabytes for this purpose and most installations should have no problems donating this amount of space to **2cSNA**<sup>®</sup>.

When the ECSA area is used up, older SAW data is written out to a VSAM KSDS, whose size determines the period of time that can be kept on hand.

Trace data is not “pre-stored” in ECSA, but written out to a second VSAM KSDS immediately.

Both VSAM files’ sizes determine the period of time that can be kept before they wrap. Currently there is no provision for “dumping” the files to archives at wrap time and therefore no danger of complications or any administrative hassle.

It should be noted that there need to be a separate set of these VSAM files for each system that is to be monitored.

The data gatherer is written as a multi-subtask program with state of the art error handling (ESTAE, VTAM error exits etc.) and will fit in with your current system gracefully.

## The 2cSNA® ISPF application (TUI)

The TUI (textual user interface) of **2cSNA®** is to provide

1. An interface to the SAW / TRACE information provided by the data gatherer
2. A command interface to one or more systems in your installation
3. A platform for user-written REXX scripts to run against the command interface
4. Display of a list of currently active sessions as well as completed sessions
5. Display of general data for a specific session
6. Display of trace data for a specific session

The TUI runs under ISPF/TSO as an application that adheres closely to the standards imposed by IBM and SAA on such user interfaces. This means that there are in practice no restrictions on screen splitting, customisation and even writing REXX macros etc.

Full support for all 3270 Models (even Model 6, 62x160) and state of the art usage of ISPF attribute features give the user full flexibility in using the TUI.

Multiple Panels can be opened for different purposes and/or target systems and navigation between these stacked instances can be done via a rotary scrolling scheme using PF Keys or by point-and-shoot into the stack-list on the screen.

If only the purpose of using the TUI as a command interface to VTAM and MVS is considered, one can **log on to one TSO** and perform all tasks from that one focal point – of course always taking care to “target” commands to the correct destination system.

The command transport is done **transparently**, so that the user need not concern him with command syntax and command routing. **The most suitable transport method for the desired target system is selected automatically by 2cSNA®.**



**2cSNA**® provides for a seamless integration of your own set of REXX libraries, in which you can provide your own extension of the basic functionality using all the tools and power available to ISPF – you can use panels, pop-ups, bring back output into **2cSNA**® windows and so on. You can also issue commands, inspect and analyse the results or perform repetitive tasks and complex local automation.

*Please note that time based and event based automation (as opposed to user-initiated automation) should be moved to where it belongs: into the system automation package present at your site. That kind of automation cannot run in your TSO address space and the **2cSNA**® data gatherer makes no provision for automation functionality.*

As soon as you would like to view SAW data or to inspect session trace records, one must log on to the TUI on the right system, even if the VSAM KSDS files might be on shared DASD. This is because the **TUI** needs to actually **make contact** (within the same system) with the **data gatherer**.

Contact with the data gatherer is made via the ECSA area, which is not searched for (a process that could take up to 30 seconds), but rather “picked up” from MVS Name/Token Services which were set by the data gatherer. Therefore “making” contact (and not losing it) is a very quick process.

Once contact has been made, the TUI can display both the session table from ECSA as well as the “older” session data from the SAW VSAM KSDS. The amount of data to be seen as well as a selection by other criteria can be effected by using the **FILTER** command.

If a user wishes to see detail data on a session, he can request this. Subsequently, if desired and if such records have been recorded, he can invoke the trace display facility to display the trace records.

These can be viewed in sequence, expanded and dissected and displayed in many different modes down to the bits as described in the SNA Formats manual.

## Sample Screens

```

AnSyNova 2cSNA V1.110157 05348
CMD_ONWB Command ==> /d net,majnodes
Stacked: LOG
Scroll==> HALF
From 56 to 78 of 78
----- Command issued successfully -----
56 13:12:21 ONWB IST089I LOC32 TYPE = LCL SNA MAJ NODE , ACTIV
57 13:12:21 ONWB IST089I LOC33 TYPE = LCL SNA MAJ NODE , ACTIV
58 13:12:21 ONWB IST089I LOC34 TYPE = LCL SNA MAJ NODE , ACTIV
59 13:12:21 ONWB IST089I LOC35 TYPE = LCL SNA MAJ NODE , ACTIV
60 13:12:21 ONWB IST089I LOC36 TYPE = LCL SNA MAJ NODE , ACTIV
61 13:12:21 ONWB IST089I LOC37 TYPE = LCL SNA MAJ NODE , ACTIV
62 13:12:21 ONWB IST089I LOC38 TYPE = LCL SNA MAJ NODE , ACTIV
63 13:12:21 ONWB IST089I SWEE3225 TYPE = SW SNA MAJ NODE , ACTIV
64 13:12:21 ONWB IST089I SWEE3226 TYPE = SW SNA MAJ NODE , ACTIV
65 13:12:21 ONWB IST089I SWEE3228 TYPE = SW SNA MAJ NODE , ACTIV
66 13:12:21 ONWB IST089I SWEE3229 TYPE = SW SNA MAJ NODE , ACTIV
67 13:12:21 ONWB IST089I SWEE3230 TYPE = SW SNA MAJ NODE , ACTIV
68 13:12:21 ONWB IST089I SWEE3280 TYPE = SW SNA MAJ NODE , ACTIV
69 13:12:21 ONWB IST089I SWEE3281 TYPE = SW SNA MAJ NODE , ACTIV
70 13:12:21 ONWB IST089I ATCPZHNB TYPE = APPL SEGMENT , ACTIV
71 13:12:21 ONWB IST089I ATCPZBNB TYPE = APPL SEGMENT , ACTIV
72 13:12:21 ONWB IST089I ATCPZFNB TYPE = APPL SEGMENT , ACTIV
73 13:12:21 ONWB IST089I END TYPE = CORSC SEGMENT , ACTIV
74 13:12:21 ONWB IST089I ISTLSXCF TYPE = LCL SNA MAJ NODE , ACTIV
75 13:12:21 ONWB IST089I ISTDSWMM TYPE = SW SNA MAJ NODE , ACTIV
76 13:12:21 ONWB IST089I ABOSVTAM TYPE = APPL SEGMENT , ACTIV
77 13:12:21 ONWB IST089I ACNMPNWB TYPE = APPL SEGMENT , ACTIV
78 13:12:21 ONWB IST1454I 78 RESOURCE(S) DISPLAYED
79 13:12:21 ONWB IST314I END

```

Fig. 1: Issuing a command to the local system



```
AnSyNova 2cSNA V1.110157 05348
SAW Command ==>>
Stacked: LOG CMD_ONWB
Scroll==> HALF
From 0 to 24 of 1754
```

1	13:15:48	_VTAM	NTS08001	DEBMW001.SSCP32	2005/12/22-12:11:32	<< STILL ACTIVE! >>	2	ISTVTCOS	0
2	13:15:48	_NTS08001	PIM16003	DEBMW001.SSCP32	2005/12/22-12:11:32	<< STILL ACTIVE! >>	128	MEDIUM T3278MSE	#CONNECT 0
3	13:15:48	_CIC1PVE1	IP322342	DEBMW001.SSCP30	2005/12/22-11:56:27	<< STILL ACTIVE! >>	61	MEDIUM M2SNAQ	#CONNECT 0
4	13:15:48	_TPXDSXX1	IP322342	DEBMW001.SSCP30	2005/12/22-11:56:27	2005/12/22-11:56:27	0	MEDIUM M2SNAQ	#CONNECT 0
5	13:15:48	_VTAM	IP322342	DEBMW001.SSCP32	2005/12/22-11:56:26	<< STILL ACTIVE! >>	2	ISTVTCOS	0
6	13:15:48	_D1BMWTPX	IP322342	DEBMW001.SSCP32	2005/12/22-11:56:26	2005/12/22-11:56:27	14	MEDIUM M2SNAQ	0
7	13:15:48	_CIC1PVE1	IP322341	DEBMW001.SSCP30	2005/12/22-11:55:31	2005/12/22-11:56:16	96	MEDIUM M2SNAQ	#CONNECT 0
8	13:15:48	_TPXDSXX1	IP322341	DEBMW001.SSCP30	2005/12/22-11:55:30	2005/12/22-11:55:31	0	MEDIUM M2SNAQ	#CONNECT 0
9	13:15:48	_VTAM	IP322341	DEBMW001.SSCP32	2005/12/22-11:55:27	2005/12/22-11:56:16	4	ISTVTCOS	0
10	13:15:48	_D1BMWTPX	IP322341	DEBMW001.SSCP32	2005/12/22-11:55:27	2005/12/22-11:55:30	20	MEDIUM M2SNAQ	0
11	13:15:48	_CIC1PVE1	IP322340	DEBMW001.SSCP30	2005/12/22-11:35:17	<< STILL ACTIVE! >>	128	MEDIUM M2SNAQ	#CONNECT 0
12	13:15:48	_TPXDSXX1	IP322340	DEBMW001.SSCP30	2005/12/22-11:35:16	2005/12/22-11:35:17	0	MEDIUM M2SNAQ	#CONNECT 0
13	13:15:48	_VTAM	IP322340	DEBMW001.SSCP32	2005/12/22-11:35:14	<< STILL ACTIVE! >>	2	ISTVTCOS	0
14	13:15:48	_D1BMWTPX	IP322340	DEBMW001.SSCP32	2005/12/22-11:35:14	2005/12/22-11:35:16	0	MEDIUM M2SNAQ	0
15	13:15:48	_TPX1PGL5	IP322339	DEBMW001.SSCP30	2005/12/22-11:23:40	2005/12/22-12:10:41	130	MEDIUM M2SNAQ	#CONNECT 0
16	13:15:48	_TPX1SGL1	IP322339	DEBMW001.SSCP30	2005/12/22-11:23:39	2005/12/22-11:23:40	17	MEDIUM M2SNAQ	#CONNECT 0
17	13:15:48	_VTAM	IP322339	DEBMW001.SSCP32	2005/12/22-11:23:25	2005/12/22-12:10:41	4	ISTVTCOS	0
18	13:15:48	_D1BMWTPX	IP322339	DEBMW001.SSCP32	2005/12/22-11:23:25	2005/12/22-11:23:39	21	MEDIUM M2SNAQ	0
19	13:15:48	_TPX1PGL7	IP322338	DEBMW001.SSCP30	2005/12/22-11:21:13	<< STILL ACTIVE! >>	128	MEDIUM M2SNAQ	#CONNECT 0
20	13:15:48	_TPX1SGL1	IP322338	DEBMW001.SSCP30	2005/12/22-11:21:12	2005/12/22-11:21:13	16	MEDIUM M2SNAQ	#CONNECT 0
21	13:15:48	_VTAM	IP322338	DEBMW001.SSCP32	2005/12/22-11:13:57	<< STILL ACTIVE! >>	2	ISTVTCOS	0
22	13:15:48	_D1BMWTPX	IP322338	DEBMW001.SSCP32	2005/12/22-11:13:57	2005/12/22-11:21:12	21	MEDIUM M2SNAQ	0
23	13:15:48	_CIC1PVE1	IP322337	DEBMW001.SSCP30	2005/12/22-11:08:03	<< STILL ACTIVE! >>	128	MEDIUM M2SNAQ	#CONNECT 0
24	13:15:48	_TPXDSXX1	IP322337	DEBMW001.SSCP30	2005/12/22-11:08:02	2005/12/22-11:08:03	17	MEDIUM M2SNAQ	#CONNECT 0

Fig. 2: A SAW data session list



```

AnSyNova 2cSNA V1.110157 05348
LUD Command ==>
Stacked: LOG CMD_ONWB SAW
Scroll ==> HALF
From 1 to 24 of 71

1 13:18:42 +----- D1BMWTPX -----+
2 13:18:42 ! Session Activation.... 2005/12/22-11:55:27 !
3 13:18:42 ! Termination.... 2005/12/22-11:55:30 !
4 13:18:42 +-----+-----+
5 13:18:42 +----- PRIMARY LU -----+ +----- SECONDARY LU -----+
6 13:18:42 ! Name.... D1BMWTPX ! ! Name.... IP322341 !
7 13:18:42 ! Subarea.. 00010020 ! ! Subarea.. 00010020 !
8 13:18:42 ! Element.. 03D7 ! ! Element.. 03B2 !
9 13:18:42 +-----+-----+
10 13:18:42 +----- ASSOCIATED NETNAMES/SSCPs -----+
11 13:18:42 ! CV2B SSCP32 DEBMW001 PNCP SSCP32 DEBMW001 !
12 13:18:42 ! CVE3 SSCP30 DEBMW001 CVE4 !
13 13:18:42 ! CVE5 SSCP30 DEBMW001 CVE6 SSCP32 DEBMW001 !
14 13:18:42 ! RSV0 ! RSV0 !
15 13:18:42 +-----+-----+
16 13:18:42 +----- SESSION START DATA -----+ +----- ACTIVATION PARMS -----+
17 13:18:42 ! ! ! !
18 13:18:42 ! Logmode..... M2SNAQ ! ! FM Profile... 3 !
19 13:18:42 ! Suba Cos..... MEDIUM ! ! TS Profile... 3 !
20 13:18:42 ! APPN Cos..... ! ! Pri Max RU... 512 !
21 13:18:42 ! Type..... ! ! Sec Max RU... 512 !
22 13:18:42 ! Class..... ! ! Dflt Size... 24x80 !
23 13:18:42 ! PCID.. CF5FD9FD3653E98D ! ! Alt Size... !
24 13:18:42 +-----+-----+ ! LU Type.... 2 !

```

Fig. 3: Session Detail

```

----- 2cSNA - TRACE ANALYSIS PANEL -----
Command ==>
Scroll ==> HALF

No RecNo H Length Description (short) LU Name LU Name Mode Mode From 1 to 22 of 26
--Trace TRACE
_ 1 1 - 48 SNA FID4 ReQ SC_ACTLU .....IP322341 < .....VTAM 00:00:00.000
_ 2 2 - 62 SNA FID4 Rs+ SC_ACTLU .....IP322341 < .....VTAM 00:00:00.000
_ 3 3 - 186 SNA FID4 ReQ SC_BIND .....IP322341 < .....D1BMWTPX 00:00:00.008
_ 4 4 - 48 SNA FID4 ReQ SC_SDT .....IP322341 < .....D1BMWTPX 00:00:00.013
_ 5 5 - 48 SNA FID4 Rs+ SC_SDT .....IP322341 < .....D1BMWTPX 00:00:00.013
_ 6 6 - 48 SNA FID4 ReQ RU(PFK5/EW)FMD .....IP322341 < .....D1BMWTPX 00:00:00.015
_ 7 7 - 48 SNA FID4 Rs+ IPR .....IP322341 < .....D1BMWTPX 00:00:00.015
_ 8 8 - 48 SNA FID4 Rs+ RU(00000000...)FMD .....IP322341 < .....D1BMWTPX 00:00:00.016
_ 9 9 - 48 SNA FID4 ReQ RU(PFK3/WSF)FMD .....IP322341 < .....D1BMWTPX 00:00:00.019
_ 10 10 - 48 SNA FID4 Rs+ IPR .....IP322341 < .....D1BMWTPX 00:00:00.019
_ 11 11 - 48 SNA FID4 ReQ RU(STRUF)FMD .....IP322341 < .....D1BMWTPX 00:00:00.306
_ 12 12 - 48 SNA FID4 Rs+ IPR .....IP322341 < .....D1BMWTPX 00:00:00.306
_ 13 13 - 48 SNA FID4 ReQ RU(PFK5)FMD .....IP322341 < .....D1BMWTPX 00:00:00.309
_ 14 14 - 48 SNA FID4 ReQ RU(PFK3)FMD .....IP322341 < .....D1BMWTPX 00:00:00.310
_ 15 15 - 48 SNA FID4 ReQ RU(STRUF)FMD .....IP322341 < .....D1BMWTPX 00:00:00.596
_ 16 16 - 48 SNA FID4 Rs+ IPR .....IP322341 < .....D1BMWTPX 00:00:00.596
_ 17 17 - 48 SNA FID4 ReQ Begin-Chain RU(PFK5)FMD .....IP322341 < .....D1BMWTPX 00:00:00.606
_ 18 18 - 48 SNA FID4 ReQ Mid-Chain .....IP322341 < .....D1BMWTPX 00:00:00.606
_ 19 19 - 48 SNA FID4 ReQ End-Chain .....IP322341 < .....D1BMWTPX 00:00:00.606
_ 20 20 - 48 SNA FID4 ReQ RU(ENTER)FMD .....IP322341 < .....D1BMWTPX 00:00:03.246
_ 21 21 - 69 SNA FID4 ReQ SC_UNBIND .....IP322341 < .....D1BMWTPX 00:00:03.251
_ 22 22 - 48 SNA FID4 Rs+ SC_UNBIND .....IP322341 < .....D1BMWTPX 00:00:03.252

```

Fig. 4: Trace Records



```

----- 2cSNA - TRACE ANALYSIS PANEL -----
Command ==>
Mode switch ok
Scroll ==> HALF

THL_FID4 RH_REQ RU_SC_BIND CV_60 CV_0E CV_0E CV_20 CV_1A CV_20
NA .....IP322341 <- .....D18MWPX SNA FID4 ReQ SC_BIND
--RECORD 3 Len 186 Time 2005/12/22-11:55:27.310414 Capt GTFVTAY -----Mode RECD3H6 From 1 to 22 of 25
THL_FID4 FID4 Field Descriptions
0000 40000000 00000000 00010020 00010020 EBC:< . . > ASC:< $ >
0010 1D000382 03D72B27 0098 EBC:< b P . q > ASC:< + ' >
RH_REQ Request Header
0000 6B0000 EBC:< , > ASC:< k >
RU_SC_BIND BIND (BIND SESSION)
0000 31010303 81903082 10908686 90100200 EBC:< b ff > ASC:< 1 0 >
0010 00000000 18500000 7E000000 C4F1C2D4 EBC:< & = D18M > ASC:< P 8 >
0020 E6E3D7E7 0005006F 164AF008 C9D7F3F2 EBC:< WTPX ? 8 IP32 > ASC:< o J >
0030 F2F3F4F1 EBC:< 2341 > ASC:< >
CV_60 Fully Qualified PCID (X'60') Control Vector
0000 6018CF5F D9FD3653 E9BD0FC4 C5C2D4E6 EBC:< - ^R . Z' DEBMW > ASC:< ' _ 6S >
0010 F0F0F14B E2E2C3D7 F3F2 EBC:< 001.SSCP32 > ASC:< K >
CV_0E Network Name (X'0E') Control Vector
0000 0E10F4C4 C5C2D4E6 F0F0F14B E2E2C3D7 EBC:< 4DEBMW001.SSCP > ASC:< K >
0010 F3F0 EBC:< 30 > ASC:< >
CV_0E Network Name (X'0E') Control Vector
0000 0E12F3C4 C5C2D4E6 F0F0F14B C4F1C2D4 EBC:< 3DEBMW001.D18M > ASC:< K >
0010 E6E3D7E7 EBC:< WTPX > ASC:< >
CV_20 COS/TPF (X'2C') Control Vector
0000 2C0A0108 D4C5C4C9 E4D44040 EBC:< MEDIUM > ASC:< , $$ >
CV_1A NAU Address (X'1A') Control Vector

```

Fig.5: Trace Record (BIND) Medium Detail

```

----- 2cSNA - TRACE ANALYSIS PANEL -----
Command ==>
Scroll ==> HALF

THL_FID4 RH_REQ RU_SC_BIND CV_60 CV_0E CV_0E CV_20 CV_1A CV_20
NA .....IP322341 <- .....D18MWPX SNA FID4 ReQ SC_BIND
--RECORD 3 Len 186 Time 2005/12/22-11:55:27.310414 Capt GTFVTAY -----Mode RECD3H6 From 131 to 152 of 353
RU_SC_BIND BIND (BIND SESSION)
X'31" request code
EBC:<
0001D 0000 (1) 31
0001E.0 0001.0 (1) 01 =
0001E.0 0001.0 (0.4) 0000----
0001E.4 0001.4 (0.4) ----0001
Format
Type
1 = nonnegotiable
0001F 00002 (1) 03
FM profile
03 = FM profile 3
EBC:<
00020 00003 (1) 03
TS profile
03 = TS profile 3
EBC:<
00021.0 00004.0 (1) B1 =
00021.0 00004.0 (0.1) 1-----
Chaining use selection
1 = multiple-RU chains allowed from primary LU
half-session (only value defined for LU 6.2)
Request control mode selection
0 = immediate request mode (only value defined for LU
6.2)
00021.1 00004.1 (0.1) -0-----
Chain response protocol used by primary LU half-session
for FMD requests; chains from primary will ask for
3 = definite or exception response (only value defined
for LU 6.2)
00021.2 00004.2 (0.2) --11----
2-phase commit for sync point (reserved if any TS
00021.4 00004.4 (0.1) ---0---

```

Fig. 6: Trace Record (BIND) Maximum Detail