2cSNA© – Modern Network Management

Technical Brief

Overview

2cSNA© © 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 Assignement and Routing Facility

Another unique feature of **2cSNA**©, 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© consists of an assembler coded started task, the ***data gatherer*** that utilises the VTAM CNM interface via the ISTPDCLU 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**©.

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 loosing 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

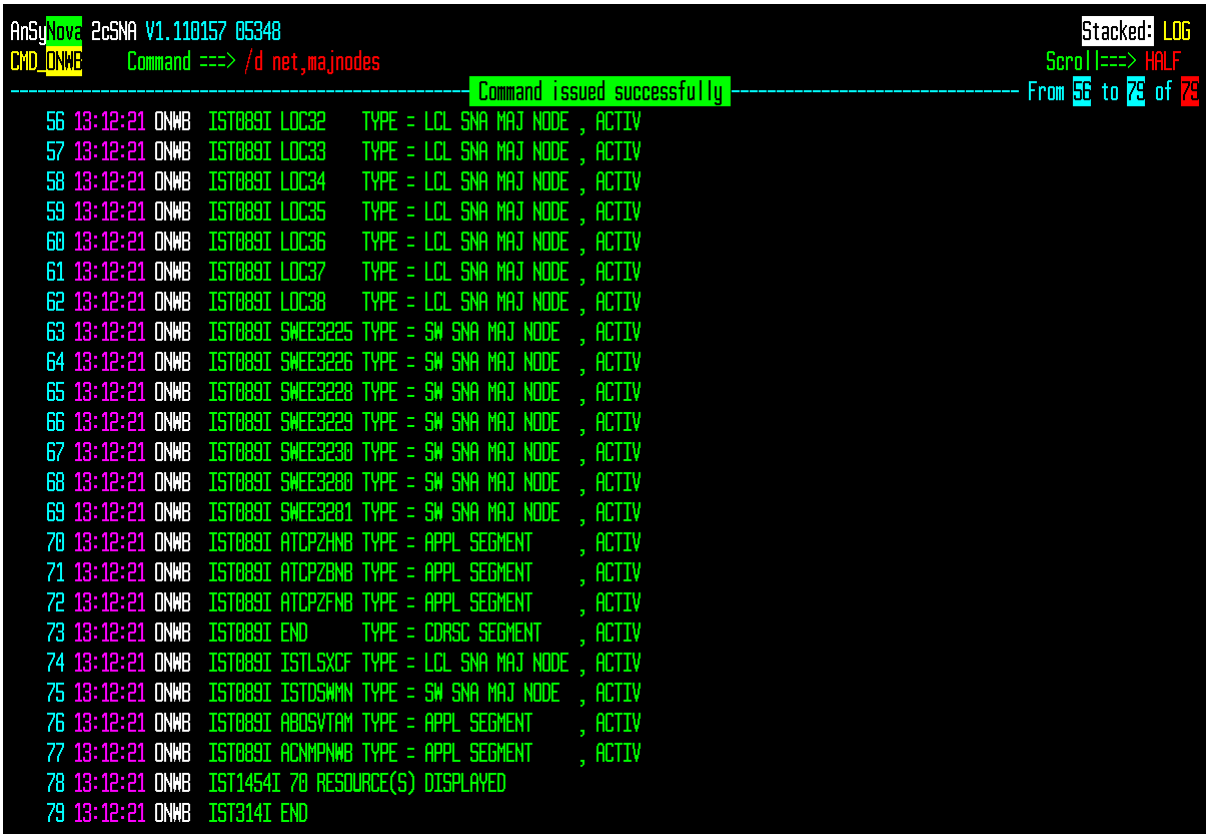

Fig. 1: Issuing a command to the local system

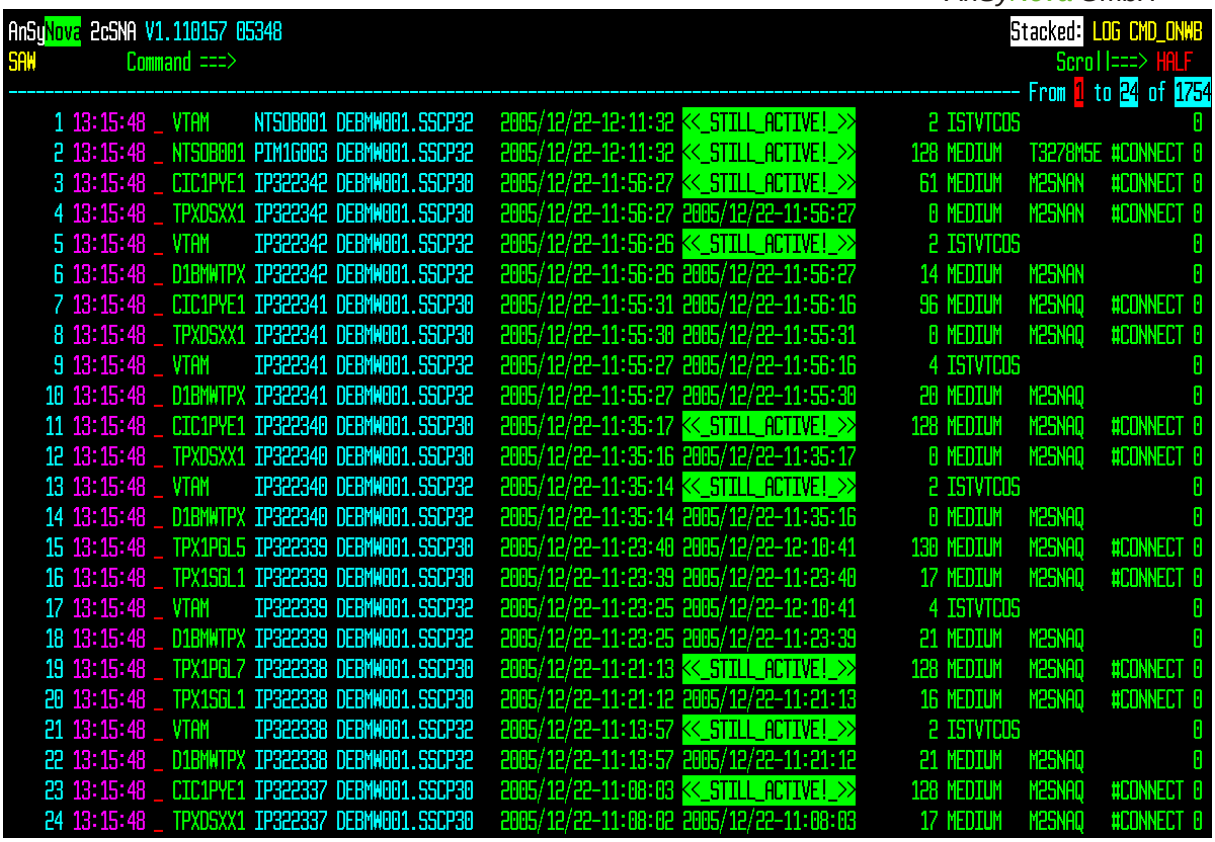

Fig. 2: A SAW data session list

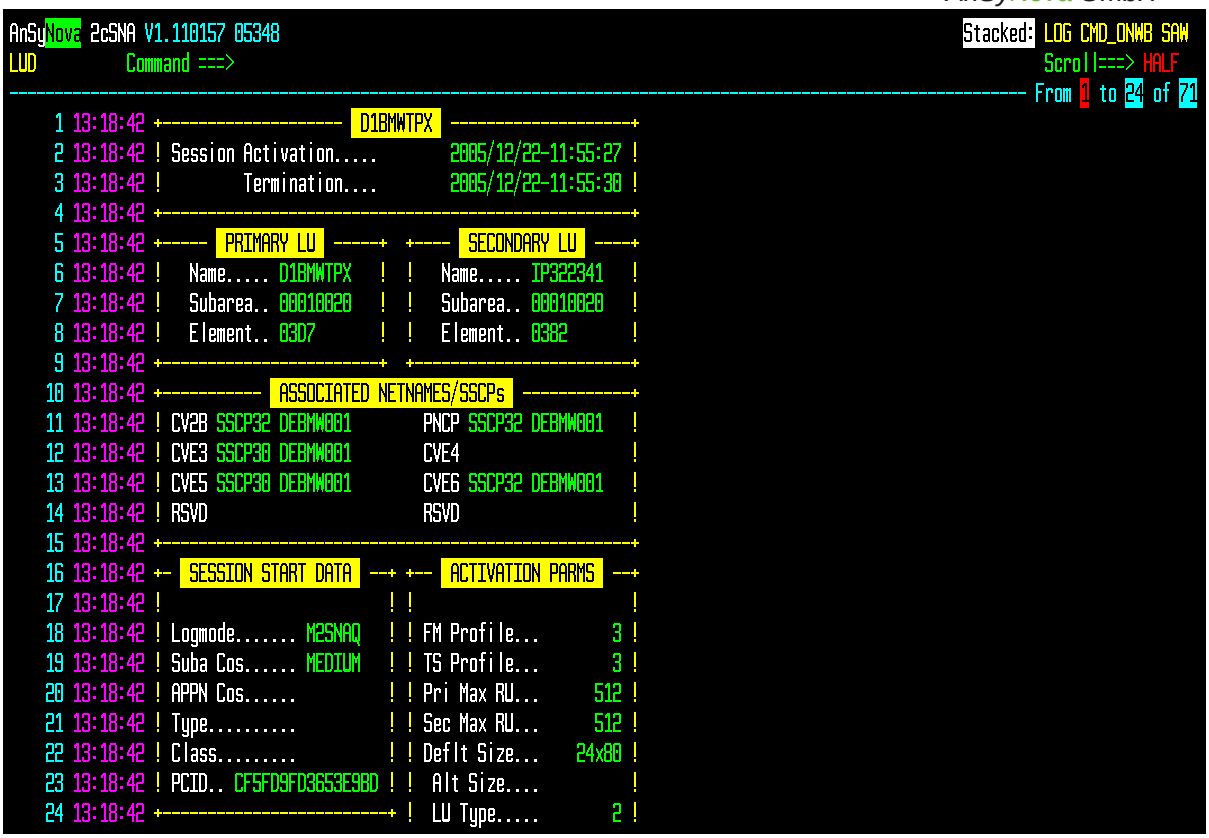

Fig. 3: Session Detail

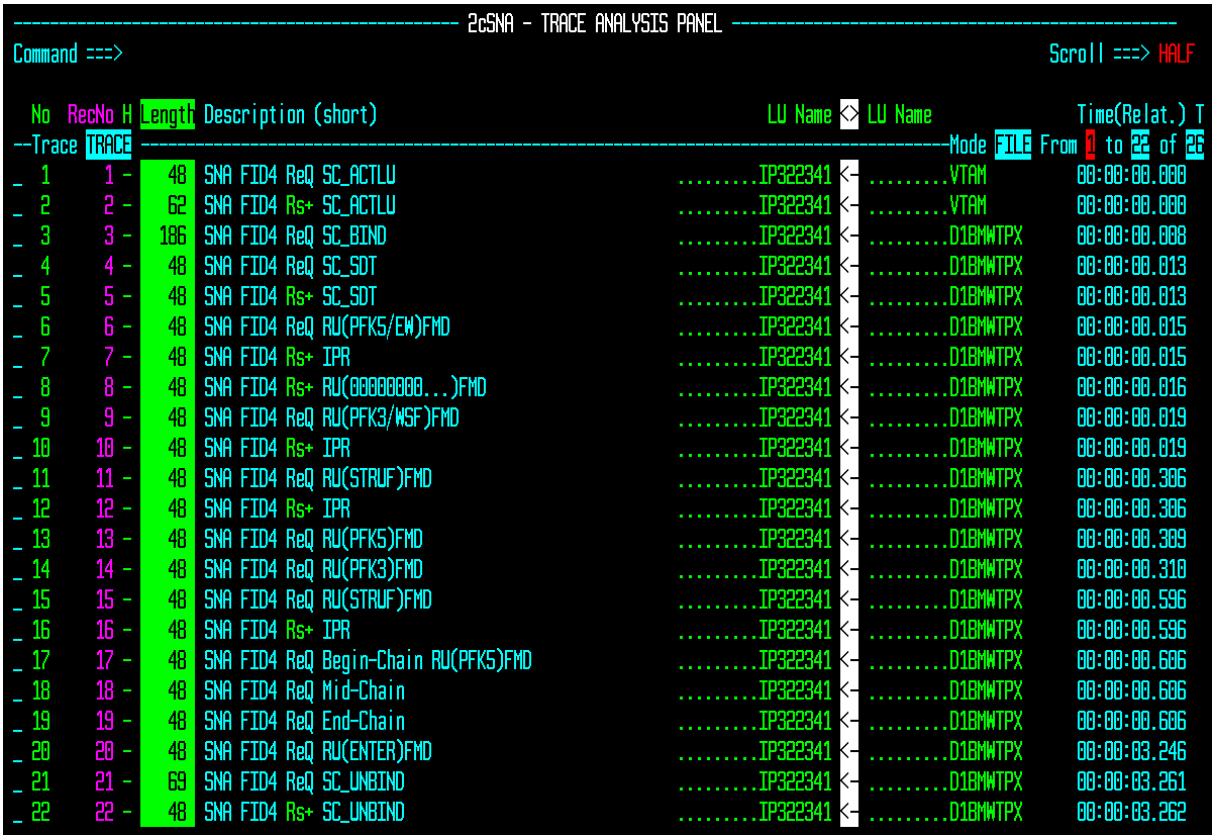

Fig. 4: Trace Records

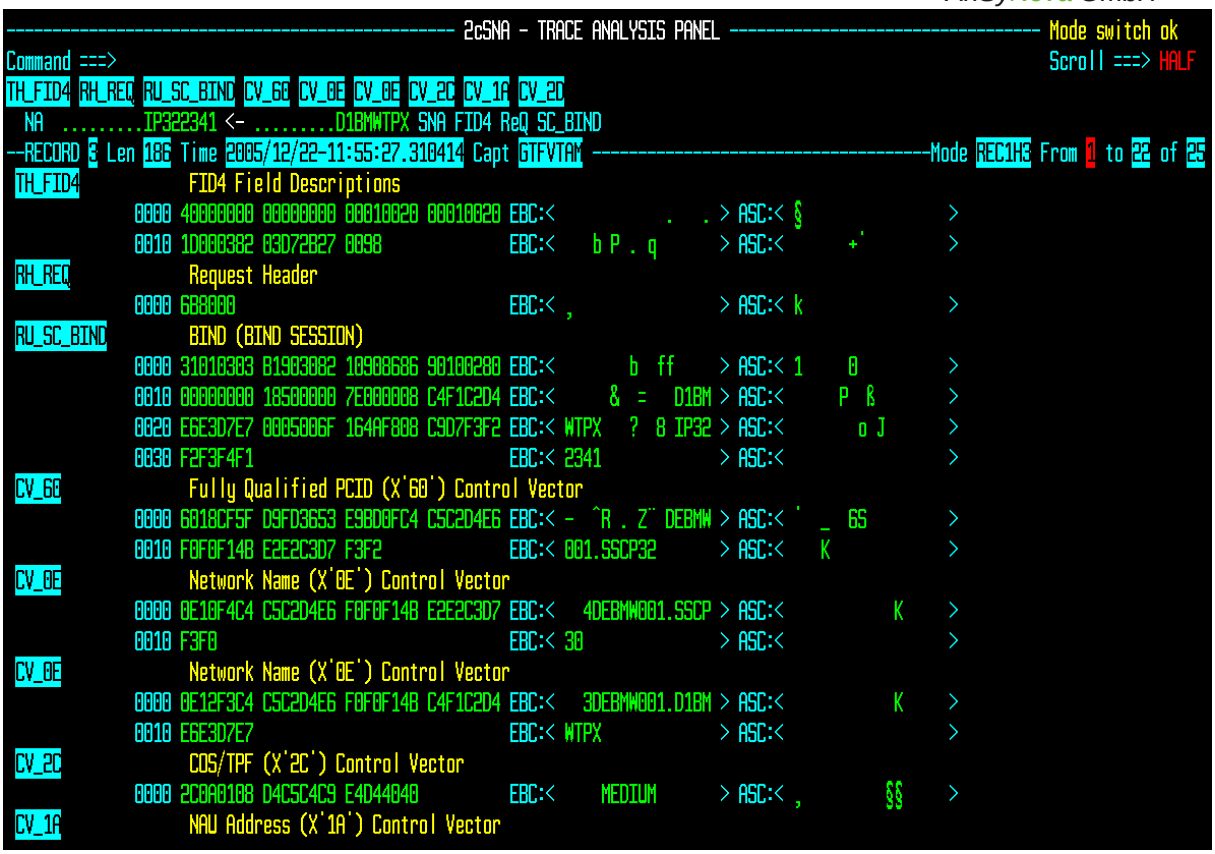

Fig.5: Trace Record (BIND) Medium Detail

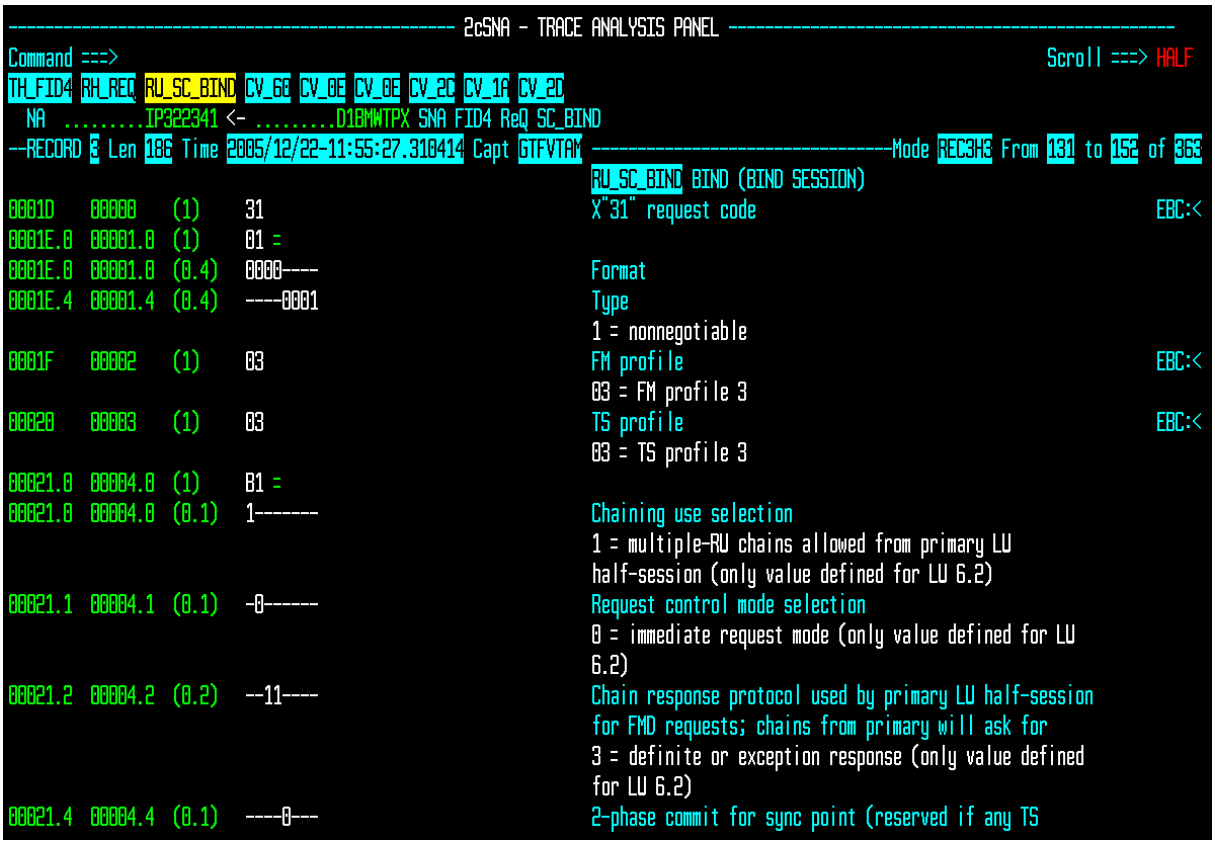

Fig. 6: Trace Record (BIND) Maximum Detail