

NatCDCSP

Version 5.x Release Notes

Overview.....	2
Summary of Major Changes for Release 5.1.1	3
Enhancement: Support for Spanned Records	3
Enhancement: Output of Encountered Number of Occurrences.....	4
Enhancement: Time Differential Calculations Now Rounded	5
Installation.....	5

Overview

The purpose of this document is to alert the reader to new features and enhancements that comprise the version 5.x releases of NatCDCSP. This document is intended to be cumulative in scope, and allow the reader to see all changes that have occurred in the version 5.x release(s) of NatCDCSP that have occurred since its initial release in late December of 2012.

Summary of Major Changes for Release 5.1.1

NATCDCSP 5.1.1 was released on December 28th, 2012. In addition to containing all previous corrections and enhancements, the principle enhancement to version 5.1.1 of NATCDCSP was to handle the ADASEL output of “Spanned Records”, in addition to being able to handle the ADASEL output of “normal” records.

Enhancement: Support for Spanned Records

With the release of NatCDCSP 5.1.1, NatWorks provides a single module that can handle either “normal” PLOG records or it can handle “Spanned Records”. When using version 5.1.1 against files that are not internally handled as “Spanned Records”, nothing changes with this release and current processing remains the same as previous processing. This version of NatCDCSP can be substituted in place of a previous version of NatCDCSP, with there being no requirement for any other change.

When handling Spanned Records however, several rules must be adhered to:

- 1) **SRLOG=ALL**
The ADARUN Parameter SRLOG must be set to ALL (SRLOG=ALL). Per Software AG documentation this option forces ADABAS to write all segments of a Spann Record to the PLOG when a transaction occurs.
- 2) **No “Mixed Execution” Runs**
Within a single execution run of NatCDCSP, NatCDCSP will successfully handle either one or more “normal” files (just as it does today), or it can handle one or more “Spanned Records” files (see below), but it cannot process both “normal” records and “Spanned Records” in a single execution. NatCDCSP will detect what type of process it needs to adapt to by examining the contents of the first record it reads from the output of ADASEL (I.E. either “normal” record or a “Spanned Record”), and if it subsequently detects records in the other format execution will cease with an appropriate error message.
- 3) **Continued Limit of 32K**
For the immediate term, the NatQuery-generated post NatCDCSP processing programs will continue to have a limit of only being able to handle a 32K output record. This limitation will be removed in subsequent version of NatQuery through the use of Natural code that ius generated to READ and WRITE unformatted Work Files.
- 4) **Different JCL/Script**
When using NatCDCSP against normal files, the capability exists to process multiple files in a single pass. This is possible by first using ADASEL to write all “normal” transactions to say, EXPA1, with the output of EXPA1 then being sorted into ascending

sequence by FNR, ISN and relative Sequence Number. This initial sort immediately follows the execution of ADASEL and immediately precedes the execution of NATCDCSP.

When running NATCDCSP however against “Spanned Records”, this Sort is no longer possible due to the how ADABAS stores the Spanned Records into separate physical records.

In order to use NATCDCSP against Spanned Records then, a copy of the JCL/Script template (Production NATCDC SP Process template) should be created that removes this initial Sort. The ability to have NatCDCSP possibly reference more than one JCL template is now easily supported by NatQuery – which with the 5.3.6 release allows custom JCL/Script templates to be created and then be linked to the NatCDCSP process, with the JCL/Script template that is required being selectable at generation time.

5) Processing Multiple Spanned Record Files in one pass of NATCDCSP

When using NatCDCSP to process records from multiple Spanned Record files in a single pass, the following must be adhered to:

- a. Records from each Spanned Record file should be directed at their own respective EXPAX in the execution of ADASEL – not directed at the same EXPAX.
- b. When these records are then provided as input in the NATCDCSP module, they MUST be presented in ascending FNR sequence from lowest to highest FNR.

As an example, assume two files have spanned records, FNR=00010 and FNR=00011. ADASEL parameters are then constructed to support writing records from FNR=00010 to EXPA1 with records from FNR=00011 being written to EXPA2. These two files would then be concatenated together such that the file with records pertaining to FNR=00010 would be read prior to the file with records pertaining to FNR=00011.

Enhancement: Output of Encountered Number of Occurrences

In previous release of NatCDCSP, the utility would allow for the C* value of a recurring field to be selected for retrieval as part of the data. When NatCDCSP output the C* field however, it’s “real” contents would be overlaid with the number of occurrences that were requested for retrieval by the user – NOT the actual number of occurrences that was encountered.

This situation has been corrected with the release of NatCDCSP 5.1.1, which will now output the C* value that was encountered on each individual record and will no longer overlay this value with the number of occurrences that was requested.

Enhancement: Time Differential Calculations Now Rounded

There are two little known facts about PLOG records. The first is that Software AG does not store the date/timestamp of a transaction as part of the transactional record (the timestamp is derived from the date/timestamp of when the final PLOG Block is written), and the second is that these date/timestamps are not stored in Local Time – they are stored in GMT.

This means that in order to convert a PLOG timestamp, it is first necessary to calculate the difference between the local time zone and GMT to establish a Time Differential, and then apply that Time Differential to the date/timestamps stored in the PLOG so as to convert these date/timestamps back into local time.

Since it's initial release, NatCDCSP has had the ability to perform this calculation automatically (TD=AUTO), or this calculation could be manually over-ridden via a parameter. When NatCDCSP was told to do perform this calculation automatically however, the recalculation could be off by (in some extreme cases) several tenths of a second due to decimal precision and rounding issues.

This situation has been corrected with the release of NatCDCSP 5.1.1.

Installation

NatWorks supplies the NATCDCSP module as object code which has been converted into hexadecimal format so that it can be safely transported across networks without concern for corruption. Nothing changes with this release in regards to how a customer will handle and load the NatCDCSP module into Natural – the process remains exactly as it was before.

NatWorks handles the NATCDCSP module by compiling it using Software AG's Natural Optimizer Compiler to produce object code. We then unload that object code using SYSOBJH. We then run that object code through a simple Natural program, called CBIN2HEX, which converts this modules from a straight binary format (which is easily corruptible when moved across networks) into a hexadecimal format – typically giving the new module a name such as CDCSP511.HEX. We then make that modules available to our customers.

At the customer site, the customer typically uploads the “.HEX” file onto their server using a function of NatQuery. On the server, the module is run through a second NatWorks written program named CBIN2HEX, which converts the hexadecimal file back into a Binary object module. That Binary object module can then be loaded into Natural using SYSOBJH.

For complete information on Installing and Using NATCDCSP, please refer to the *NatCDC Installation and Operation Manual*.