

Department of Veterans Affairs
Decentralized Hospital Computer System

GENERIC CODE SHEET RELEASE NOTES

Version 2.0

March 1995

Information Systems Center
Washington, DC

Table of Contents

Release Notes..... 1

Revision History

Initiated on 12/22/04

Date	Description (Patch # if applic.)	Project Manager	Technical Writer
12/22/04	Updated to comply with SOP 192-352 Displaying Sensitive Data.		Mary Ellen Gray
12/22/04	Pdf file checked for accessibility to readers with disabilities.		Mary Ellen Gray

Release Notes

The Generic Code Sheet package is a Decentralized Hospital Computer Program (DHCP) software module which manages the input, editing, deletion, and transmission of code sheets from a local hospital computer system to a centralized computer system as defined by the code sheet.

Generic Code Sheet V 2.0 was developed to meet the new DHCP programming standards. The software in V. 2.0 is now using incremental and decremental locks, the TMP global versus the UTILITY global, and has pointers to the NEW PERSON file 200 instead of the USER file 3. The use of incremental and decremental locks eliminates the use of the GENERIC CODE SHEET LOCK file 2101.6 to control the locking process. In V. 1.5 this file was used to manage the locks for building, batching, and transmitting code sheets. If a process failed, the GENERIC CODE SHEET LOCK file would have to be cleared by the IRM Service. With the use of incremental and decremental locks, this file is no longer used to manage the locks, but is used to give users information on why a process is locked. The file no longer has to be cleared by the IRM Service.

Generic Code Sheet V 2.0 also includes new functionality to build, transmit, and manage the Financial Management System (FMS) code sheets. The transmission and management of the FMS code sheets is accomplished by the new GENERIC CODE SHEET STACK file 2100.1. This new file stores the FMS code sheets waiting to be transmitted. At a recurring time as set by the IRM Service, a TaskManager job will start up and transmit the FMS code sheets to Austin. The code sheets are grouped in MailMan messages using a "best-fit" strategy. One MailMan message may include 20 or 30 code sheets. This strategy eliminates the overhead associated with having one code sheet per MailMan message saving in disk space and network overhead.