Department of Veterans Affairs

VistA Scheduling Enhancements (VSE)  
Enhancement 1: An Aggregated View of VistA Clinic Profile Scheduling Grids for VistA Scheduling   
Enhancement 2: A Single Queue for Appointment Requests  
SD\*5.3\*627  
SD\*5.3\*642  
Enhancement 3: Resource Management Reporting  
SD\*5.3\*628

Technical Manual



May 2016

Revision History

Note: The revision history cycle begins once changes or enhancements are requested after the document has been baselined.

| Date | Version | Description | Author |
| --- | --- | --- | --- |
| 5/2/16 | 1.0 | Initial Baseline | D. Reed |

Table of Contents

[1. Overview 1](#_Toc436912101)

[1.1. Security 1](#_Toc436912102)

[1.2. Rules of Behavior 1](#_Toc436912103)

[1.3. Orientation 1](#_Toc436912104)

[2. Implementation and Maintenance 2](#_Toc436912105)

[2.1. System Requirements 2](#_Toc436912106)

[2.2. Package-Wide Variables 2](#_Toc436912107)

[2.3. Menu Diagram 2](#_Toc436912108)

[2.4. Routines 2](#_Toc436912109)

[2.5. Files and Tables 8](#_Toc436912110)

[2.5.1. File Access 9](#_Toc436912111)

[2.5.2. Client Application Dependencies and Files 9](#_Toc436912112)

[2.5.2.1. VistA Scheduling GUI Trace Log 13](#_Toc436912113)

[2.5.2.2. Request Management 15](#_Toc436912114)

[2.6. VistA Scheduling GUI Cross References 16](#_Toc436912115)

[2.7. Table File 29](#_Toc436912116)

[2.8. Internal Relations 89](#_Toc436912117)

[2.9. External Relations 89](#_Toc436912118)

[2.10. Published Entry Points 89](#_Toc436912119)

[2.11. Exported Option 89](#_Toc436912120)

[2.12. Parameter Definitions 90](#_Toc436912121)

[2.13. Security Keys 90](#_Toc436912122)

[2.14. Archiving and Purging 90](#_Toc436912123)

[3. Resource Management Reporting Implementation 91](#_Toc436912124)

[3.1. System Requirements 91](#_Toc436912125)

[3.2. Application Files 91](#_Toc436912126)

[3.3. Resource Management Reporting Java Application Build Process 92](#_Toc436912127)

[3.4. Data Flow Diagram 93](#_Toc436912128)

[3.5. Implementation of Report Filters 93](#_Toc436912129)

[3.6. Resource Management Reporting Application Functions 94](#_Toc436912130)

[3.6.1. Report Filter Functionality 94](#_Toc436912131)

[3.6.2. Java Classes Implementation 96](#_Toc436912132)

[3.6.3. Report Viewer Functionality 97](#_Toc436912133)

[3.6.4. Report Viewer Error Handling and Resolution 98](#_Toc436912134)

[3.6.4.1. Invalid Folder or Folder Does not exist: 98](#_Toc436912135)

[3.6.4.2. Missing Source Data File (SDEC\_VSE.xml): 99](#_Toc436912136)

[3.6.4.3. Java Application Exception Error Logs: 100](#_Toc436912137)

[4. Generating Online Documentation 101](#_Toc436912138)

[4.1. %INDEX 101](#_Toc436912139)

[4.2. List File Attributes 102](#_Toc436912140)

[4.3. Standards and Conventions Requirements and Exemptions 102](#_Toc436912141)

[4.4. Callable Routines 102](#_Toc436912142)

[5. Glossary 103](#_Toc436912143)

List of Figures

[Figure 1: Trace Log Capability Shortcut 13](#_Toc436912144)

[Figure 2: Show Trace Log Menu Option 14](#_Toc436912145)

[Figure 3: VistA Scheduling GUI Trace Log 14](#_Toc436912146)

[Figure 4: Request Query 15](#_Toc436912147)

[Figure 5: Query Results 15](#_Toc436912148)

[Figure 6: VistA Scheduling GUI Cross Reference 16](#_Toc436912149)

[Figure 7: VistA Scheduling GUI Cross Reference (cont.) 17](#_Toc436912150)

[Figure 8: VistA Scheduling GUI Cross Reference (cont.) 17](#_Toc436912151)

[Figure 9: VistA Scheduling GUI Cross Reference (cont.) 17](#_Toc436912152)

[Figure 10: VistA Scheduling GUI Cross Reference (cont.) 18](#_Toc436912153)

[Figure 11: VistA Scheduling GUI Cross Reference (cont.) 18](#_Toc436912154)

[Figure 12: VistA Scheduling GUI Cross Reference (cont.) 18](#_Toc436912155)

[Figure 13: VistA Scheduling GUI Cross Reference (cont.) 19](#_Toc436912156)

[Figure 14: VistA Scheduling GUI Cross Reference (cont.) 19](#_Toc436912157)

[Figure 15: VistA Scheduling GUI Cross Reference (cont.) 19](#_Toc436912158)

[Figure 16: VistA Scheduling GUI Cross Reference (cont.) 22](#_Toc436912159)

[Figure 17: VistA Scheduling GUI Cross Reference (cont.) 23](#_Toc436912160)

[Figure 18: VistA Scheduling GUI Cross Reference (cont.) 23](#_Toc436912161)

[Figure 19: VistA Scheduling GUI Cross Reference (cont.) 24](#_Toc436912162)

[Figure 20: VistA Scheduling GUI Cross Reference (cont.) 24](#_Toc436912163)

[Figure 21: VistA Scheduling GUI Cross Reference (cont.) 25](#_Toc436912164)

[Figure 22: VistA Scheduling GUI Cross Reference (cont.) 26](#_Toc436912165)

[Figure 23: VistA Scheduling GUI Cross Reference (cont.) 27](#_Toc436912166)

[Figure 24: VistA Scheduling GUI Cross Reference (cont.) 27](#_Toc436912167)

[Figure 25: VistA Scheduling GUI Cross Reference (cont.) 28](#_Toc436912168)

[Figure 26: Table File 29](#_Toc436912169)

[Figure 27: Table File (cont.) 30](#_Toc436912170)

[Figure 28: Table File (cont.) 31](#_Toc436912171)

[Figure 29: Table File (cont.) 32](#_Toc436912172)

[Figure 30: Table File (cont.) 32](#_Toc436912173)

[Figure 31: Table File (cont.) 33](#_Toc436912174)

[Figure 32: Table File (cont.) 34](#_Toc436912175)

[Figure 33: Table File (cont.) 35](#_Toc436912176)

[Figure 34: Table File (cont.) 36](#_Toc436912177)

[Figure 35: Table File (cont.) 37](#_Toc436912178)

[Figure 36: Table File (cont.) 38](#_Toc436912179)

[Figure 37: Table File (cont.) 39](#_Toc436912180)

[Figure 38: Table File (cont.) 40](#_Toc436912181)

[Figure 39: Table File (cont.) 41](#_Toc436912182)

[Figure 40: Table File (cont.) 42](#_Toc436912183)

[Figure 41: Table File (cont.) 43](#_Toc436912184)

[Figure 42: Table File (cont.) 44](#_Toc436912185)

[Figure 43: Table File (cont.) 45](#_Toc436912186)

[Figure 44: Table File (cont.) 46](#_Toc436912187)

[Figure 45: Table File (cont.) 47](#_Toc436912188)

[Figure 46: Table File (cont.) 48](#_Toc436912189)

[Figure 47: Table File (cont.) 49](#_Toc436912190)

[Figure 48: Table File (cont.) 50](#_Toc436912191)

[Figure 49: Table File (cont.) 51](#_Toc436912192)

[Figure 50: Table File (cont.) 52](#_Toc436912193)

[Figure 51: Table File (cont.) 53](#_Toc436912194)

[Figure 52: Table File (cont.) 54](#_Toc436912195)

[Figure 53: Table File (cont.) 55](#_Toc436912196)

[Figure 54: Table File (cont.) 55](#_Toc436912197)

[Figure 55: Table File (cont.) 56](#_Toc436912198)

[Figure 56: Table File (cont.) 57](#_Toc436912199)

[Figure 57: Table File (cont.) 58](#_Toc436912200)

[Figure 58: Table File (cont.) 59](#_Toc436912201)

[Figure 59: Table File (cont.) 60](#_Toc436912202)

[Figure 60: Table File (cont.) 61](#_Toc436912203)

[Figure 61: Table File (cont.) 62](#_Toc436912204)

[Figure 62: Table File (cont.) 63](#_Toc436912205)

[Figure 63: Table File (cont.) 64](#_Toc436912206)

[Figure 64: Table File (cont.) 65](#_Toc436912207)

[Figure 65: Table File (cont.) 66](#_Toc436912208)

[Figure 66: Table File (cont.) 67](#_Toc436912209)

[Figure 67: Table File (cont.) 68](#_Toc436912210)

[Figure 68: Table File (cont.) 69](#_Toc436912211)

[Figure 69: Table File (cont.) 70](#_Toc436912212)

[Figure 70: Table File (cont.) 71](#_Toc436912213)

[Figure 71: Table File (cont.) 72](#_Toc436912214)

[Figure 72: Table File (cont.) 73](#_Toc436912215)

[Figure 73: Table File (cont.) 74](#_Toc436912216)

[Figure 74: Table File (cont.) 76](#_Toc436912217)

[Figure 75: Table File (cont.) 77](#_Toc436912218)

[Figure 76: Table File (cont.) 78](#_Toc436912219)

[Figure 77: Table File (cont.) 79](#_Toc436912220)

[Figure 78: Table File (cont.) 80](#_Toc436912221)

[Figure 79: Table File (cont.) 81](#_Toc436912222)

[Figure 80: Table File (cont.) 82](#_Toc436912223)

[Figure 81: Table File (cont.) 83](#_Toc436912224)

[Figure 82: Table File (cont.) 84](#_Toc436912225)

[Figure 83: Table File (cont.) 85](#_Toc436912226)

[Figure 84: VistA Scheduling GUI Data Flow Diagram 93](#_Toc436912227)

[Figure 85: VistA Scheduling GUI Report Console 95](#_Toc436912228)

[Figure 86: Save/Export/Print Features 95](#_Toc436912229)

[Figure 87: Report Viewer 96](#_Toc436912230)

[Figure 88: Report Viewer Functionality 97](#_Toc436912231)

[Figure 89: Report Console 98](#_Toc436912232)

[Figure 90: Report Viewer 99](#_Toc436912233)

List of Tables

[Table 1: Routines and Description 2](#_Toc436912234)

[Table 2: Files Numbers and Names 8](#_Toc436912235)

[Table 3: File Access 9](#_Toc436912236)

[Table 4: Exported Option and Description 89](#_Toc436912237)

[Table 5: Parameter Definitions 90](#_Toc436912238)

[Table 6: Exported Security Keys 90](#_Toc436912239)

[Table 7: Java Class Name and Purpose 96](#_Toc436912240)

[Table 8: Terms and Definitions 103](#_Toc436912241)

# Overview

This manual provides Department of Veterans Affairs (VA) site managers with a technical description of the Veterans Health Information System and Technology Architecture (VistA) Scheduling Graphical User Interface (GUI) routines, files, menus, cross references, globals, and other necessary information required to effectively manage the system.

The VistA Scheduling GUI module has the following features:

* MS Windows user interface
* Graphical patient, clinic, provider, and resource scheduling
* Tightly linked to VistA patient and clinic data
* Graphical resource and clinic availability scheduling
* Printing and What You See Is What You Get (WYSIWYG) Print Preview of clinic schedules
* Graphical patient check-in linked to VistA/PCC Plus (PCC+) check-in
* Reschedule and manipulate appointments using standard Windows cut/paste metaphors utilities procedures
* Schedule multiple appointments during a time block
* Store and retrieve clinic availability patterns
* View schedules for multiple clinics simultaneously
* Resource Management Reporting for viewing metrics related clinic appointments and patient encounters in VistA

## Security

The VistA Scheduling GUI uses security keys to limit user’s ability to change system set-up parameters and patient information. In other words, not all VistA Scheduling GUI options are available to all users. Contact your site administrator to determine or change your security keys.

## Rules of Behavior

All VistA users are required to observe VA Rules of Behavior regarding patient privacy and the security of both patient information and VA computers and networks.

## Orientation

The VistA Scheduling GUI module has no VistA server menu options. The only VistA server preparation specifically required to run VistA Scheduling GUI is to install patches SD\*5.3\*627 and SD\*5.3\*628 using the Kernel Installation & Distribution System (KIDS) module and assign appropriate security keys to users. The rest of the module runs on the PC client and can be managed from there.

Interaction of VistA Scheduling GUI with the VistA system is accomplished entirely via the use of Remote Procedure Calls (RPCs).

# Implementation and Maintenance

VistA Scheduling GUI provides a Windows interface for the Patient Information Management System (PIMS) Scheduling software and is designed to interoperate with existing PIMS schedules.

## System Requirements

* Server
  + Cache version 5.0
  + Kernel version 8
  + PIMS version 5.3 patch 1012
  + VistA Scheduling Patch SD\*5.3\*627
  + VistA Scheduling Patch SD\*5.3\*628
* Client
  + .Net Version 4.0 or higher.
  + 4 GB RAM

## Package-Wide Variables

There are no package-wide variables associated with the PIMS package.

## Menu Diagram

VistA Scheduling GUI menus are discussed in detail in the *VistA Scheduling Enhancements Enhancement 1, Enhancement 2, and Enhancement 3 User Guide*.

## Routines

Table 1: Routines and Description

| Routine | Description |
| --- | --- |
| SDAM2 | ALB/MJK - Appt Mgt (cont) |
| SDAMWI1 | ALB/MJK - Walk-Ins (cont.) |
| SDAPIAP | ALB/MJK - Outpatient API/Appointments |
| SDB | FLA/RF,BSN/GRR - SET UP A CLINIC |
| SDB1 | ALB/GRR - SET UP A CLINIC |
| SDC | MAN/GRR,ALB/LDB - CANCEL A CLINIC'S AVAILABILITY |
| SDCNP0 | ALB/LDB - CANCEL APPT. FOR A PATIENT |
| SDCNSLT | ALB/HAG - LINK APPOINTMENTS TO CONSULTS |
| SDCODEL | ALB/RMO,ESW - Delete - Check Out |
| SDEC | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC01 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC01A | ALB/SAT - VISTA SCHEDULING RPCS |
| SDEC01B | ALB/SAT - VISTA SCHEDULING RPCS |
| SDEC02 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC03 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC04 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC05 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC06 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC07 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC07A | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC07B | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC08 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC09 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC12 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC13 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC14 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC15 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC16 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC17 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC18 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC19 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC20 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC21 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC22 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC23 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC24 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC25 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC25A | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC25B | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC26 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC27 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC28 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC29 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC30 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC31 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC32 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC33 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC34 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC35 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC36 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC37 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC38 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC40 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC44 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC45 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC46 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC47 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC48 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC49 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC50 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC51 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC52 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC52A | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC52B | ALB/SAT - VISTA SCHEDULING RPCS |
| SDEC53 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC54 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC54A | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC55 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC55A | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC56 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDEC57 | ALB/SAT - VISTA SCHEDULING RPCS |
| SDEC57A | ALB/SAT - VISTA SCHEDULING RPCS |
| SDECALV | ALB/SAT - VISTA SCHEDULING RPCs |
| SDECALV1 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDECALVR | ALB/SAT - VISTA SCHEDULING RPCs |
| SDECAPI | ALB/SAT - VISTA SCHEDULING RPCs |
| SDECAPI4 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDECDIQ1 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDECAR | ALB/SAT - VISTA SCHEDULING RPCS |
| SDECAR1 | ALB/SAT - VISTA SCHEDULING RPCS |
| SDECAR2 | ALB/SAT - VISTA SCHEDULING RPCS |
| SDECDEV | ALB/SAT - VISTA SCHEDULING RPCS |
| SDECDIQ1 | ALB/SAT - VISTA SCHEDULING RPCS |
| SDECDIS | ALB/SAT - VISTA SCHEDULING RPCS |
| SDECEKL | ALB/SAT - VISTA SCHEDULING RPCs |
| SDECERR | ALB/SAT - VISTA SCHEDULING RPCs |
| SDECF | ALB/SAT - VISTA SCHEDULING RPCs |
| SDECF2 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDECFUNC | ALB/SAT - VISTA SCHEDULING RPCs |
| SDECGMR | ALB/SAT - VISTA SCHEDULING RPCS |
| SDECI2 | ALB/SAT - VISTA SCHEDULING RELEASE 2 PRE/POST INSTALL |
| SDECIDX | ALB/SAT - VISTA SCHEDULING RPCS |
| SDECINI1 | ALB/SAT - VISTA SCHEDULING PRE/POST INSTALL |
| SDECINI2 | ALB/SAT - VISTA SCHEDULING PRE/POST INSTALL |
| SDECINIT | ALB/SAT - VISTA SCHEDULING PRE/POST INSTALL |
| SDECLK | ALB/SAT - VISTA SCHEDULING RPCS |
| SDECLOC | ALB/SAT – VISTA SCHEDULING RPCS |
| SDECNEW | ALB/SAT - VISTA SCHEDULING RPCs |
| SDECPAT | ALB/SAT - VISTA SCHEDULING RPCs |
| SDECPAT1 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDECPAT2 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDECPAT3 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDECPAT4 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDECPT | ALB/SAT - VISTA SCHEDULING RPCs |
| SDECPTCX | ALB/SAT - VISTA SCHEDULING RPCs |
| SDECPTPC | ALB/SAT - VISTA SCHEDULING RPCs |
| SDECPTPL | ALB/SAT - VISTA SCHEDULING RPCs |
| SDECRMG | ALB/SAT - VISTA SCHEDULING RPCS |
| SDECRMG1 | ALB/SAT - VISTA SCHEDULING RPCS |
| SDECRMG2 | ALB/SAT - VISTA SCHEDULING RPCS |
| SDECRPC | ALB/SAT - VISTA SCHEDULING RPCs |
| SDECRT | ALB/SAT - VISTA SCHEDULING RPCs |
| SDECRT0 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDECRT1 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDECRT2 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDECSFGR | ALB/SAT - VISTA SCHEDULING RPCs |
| SDECU | ALB/SAT - VISTA SCHEDULING RPCs |
| SDECU2 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDECUT1A | ALB/SAT - VISTA SCHEDULING RPCS |
| SDECUTL | ALB/SAT - VISTA SCHEDULING RPCs |
| SDECUTL1 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDECUTL2 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDECV | ALB/SAT - VISTA SCHEDULING RPCs |
| SDECWL | ALB/SAT - VISTA SCHEDULING RPCs |
| SDECWL1 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDECWL2 | ALB/SAT - VISTA SCHEDULING RPCs |
| SDECWL3 | ALB/SAT - VISTA SCHEDULING RPCS |
| SDM1A | SF/GFT,ALB/TMP - MAKE APPOINTMENT |
| SDN | SF/GFT,ALB/LDB - RECORD NO SHOWS |
| SDNACT | ALB/TMP - INACTIVATE A CLINIC |
| SDREACT | ALB/TMP - REACTIVATE A CLINIC |
| SDRRISRU | 10N20/MAH; Recall Reminder Utilities |
| SD628PST | SD\*5.3\*628 Post Install Routine |
| SDCED | VSE ENCOUNTER XREF |
| SDCED1 | VSE ENCOUNTER XREF |
| SDCED2 | VSE ENCOUNTER XREF |
| SDECRPT | VSE CLINIC APPOINTMENTS |
| SDECSTP | VSE RESOURCE MANAGEMENT REPORT STOP CODES |
| SDWLPL | IOFO BAY PINES/DMR,ESW - WAIT LIST PICK LIST |
| SDECXML | VSE RESOURCE MANAGEMENT REPORT XML DATA |
| SDECXML1 | VSE RESOURCE MANAGEMENT REPORT XML DATA |
| SDECXML2 | VSE RESOURCE MANAGEMENT REPORT XML DATA |
| SDECXML3 | VSE RESOURCE MANAGEMENT REPORT XML DATA |
| SDECXML4 | VSE RESOURCE MANAGEMENT REPORT XML DATA |
| SDECXML5 | VSE RESOURCE MANAGEMENT REPORT XML DATA |
| SDECXUTL | VSE RESOURCE MANAGEMENT REPORT XML UTILITY |

## Files and Tables

Table 2: Files Numbers and Names

| File Number | File Name |
| --- | --- |
| 44 | HOSPITAL LOCATION |
| 403.5 | RECALL REMINDERS |
| 403.56 | RECALL REMINDERS REMOVED |
| 409.3 | SD WAIT LIST |
| 409.822 | SDEC ACCESS GROUP |
| 409.824 | SDEC ACCESS GROUP TYPE |
| 409.823 | SDEC ACCESS TYPE |
| 409.834 | SDEC ADDITIONAL RESOURCE |
| 409.81 | SDEC APPLICATION |
| 409.84 | SDEC APPOINTMENT |
| 409.845 | SDEC PREFERENCES AND SPECIAL NEEDS |
| 409.85 | SDEC APPT REQUEST |
| 409.831 | SDEC RESOURCE |
| 409.832 | SDEC RESOURCE GROUP |
| 409.833 | SDEC RESOURCE USER |

### File Access

Table 3: File Access

| File Number | GL | RD | WR | LYG | DD | DEL |
| --- | --- | --- | --- | --- | --- | --- |
| 44 | ^SC( | d | D | D | @ | @ |
| 403.5 | ^SD(403.5, |  |  | @ | @ | @ |
| 403.56 | ^SD(403.56, |  |  | @ | @ | @ |
| 409.3 | ^SDWL(409.3, |  | D | D | @ | @ |
| 409.822 | ^SDEC(409.822, |  |  |  |  |  |
| 409.824 | ^SDEC(409.824, |  |  |  |  |  |
| 409.823 | ^SDEC(409.823, |  |  |  |  |  |
| 409.834 | ^SDEC(409.834, |  |  |  |  |  |
| 409.81 | ^SDEC(409.81, |  |  |  |  |  |
| 409.84 | ^SDEC(409.84, |  |  |  |  |  |
| 409.85 | ^SDEC(409.85, |  |  |  |  |  |
| 409.845 | ^SDEC(409.845, |  |  |  |  |  |
| 409.831 | ^SDEC(409.831, |  |  |  |  |  |
| 409.832 | ^SDEC(409.832, |  |  |  |  |  |
| 409.833 | ^SDEC(409.833, |  |  |  |  |  |

### Client Application Dependencies and Files

Table 4: Clinical Scheduler Distributable Files

| Clinical Scheduler Files | Description |
| --- | --- |
| ClinSchd.exe | This is the main executable that launches the application. |
| ClinSchd.Infrastructure.dll | This library contains the application models, behaviors, events, interfaces, static information, threading capability and other common functionality used by the application and its supporting modules. |
| CancelAppt.dll | Required functionality for cancelling appointments and Appointment Requests. |
| ChangeDivision.dll | Functionality required for Changing Divisions and invoking Authentication functionality for Division. |
| CheckIn.dll | Functionality required to check in Patients for a selected appointment. |
| CheckOut.dll | Functionality required to check out patients for a selected appointment |
| DataAccess.dll | This is the primary Data Access Layer designed to interface with VistA remote procedure calls. |
| FindAppt.dll | Functionality required for searching and finding appointments based on criteria set. |
| Management.dll | Functionality required for managing Users, Clinics and Clinic Groups. |
| MarkAsNoShow.dll | Functionality required to Update Appointment status to “No Show”. |
| Navigation.dll | This library handles the layout and grouping of services and objects within the GUI display. |
| PatientAppt.dll | Functionality required to Create New Appointments. |
| PatientSelection.dll | Functionality required to select a Patient from the VistA Patient file. |
| Reports.dll | Functionality required to support the GUI reports implemented in E1/E2 and E3. |
| ResourceSelection.dll | Functionality required for users to select Resources, Clinics and Clinic Groups. |
| Ribbon.dll | This library contains the tabs and controls that display in the application that allow the user to view high level dashboard functionality as well as the tabs required to switch between Scheduling functions, User and System Management and Reports. |
| Task.dll | This library contains functionality for users to Manage Schedules and Appointments associated with Clinic, Provider and Clinic Group schedules. i.e., Creating, Modifying and Cancelling Appointments. |
| Microsoft Practices Libraries | The Microsoft Enterprise Library is a collection of reusable software components (application blocks) designed to assist software developers with common enterprise development cross-cutting concerns (such as logging, validation, data access, exception handling, and many others). Application blocks are a type of guidance; they are provided as source code, test cases, and documentation that can be used "as is," extended, or modified by developers to use on complex, enterprise-level line-of-business development projects. |
| 3rd Party Controls |  |
| DotNetBar for WPF (v 7.6.0.7) | LicenseKey - DAD80300DA22  <http://www.devcomponents.com/>  These components are used to host the Provider and ClinicGroup Schedule Controls. |
| Telerik Windows Controls  (v 2010.1.603.35) | Telerik Windows controls are used by the GUI for displaying dialog boxes, user input fields such as Text, Radio/Check Buttons, Drop Down and Combination Lists, and other UI elements. |
| **Client Configuration Files** |  |
| ClinSchd.exe.config | This file contains configuration for internal functionality as well as configuration information for the client to connect to the VistA server. The ***Application Settings*** section of this file allows the user to modify the default VistA server connection that the user will need to authenticate against during application start up.  <appSettings>  <add key="host" value="*ServerName*" />  <add key="port" value="*Port*" />  <add key="nspace" value="*Namespace*" />  </appSettings>  \*In addition to connecting to a valid VistA server, port and namespace, Users will be required to supply valid Access and Verify Codes with the proper keys and permissions in order to Authenticate. |
| ClinSchd.Infrastructure.xml | This file contains configuration information for the client infrastructure project. Data contained in this file does not need to be modified for environments. Contains dynamic application information for the ClinSchd.Infrastructure project. |
| Telerik Configuration Files |  |
| Microsoft.Practices Configuration Files |  |

#### VistA Scheduling GUI Trace Log

The VistA Scheduling GUI Trace Log is a custom GUI designed to inspect and debug server side VistA Remote Procedure Calls (RPCs). The UI for this application displays the input and results for RPC calls. To launch the GUI with access to the trace log, the executable must be launched with the following command “/trace”.

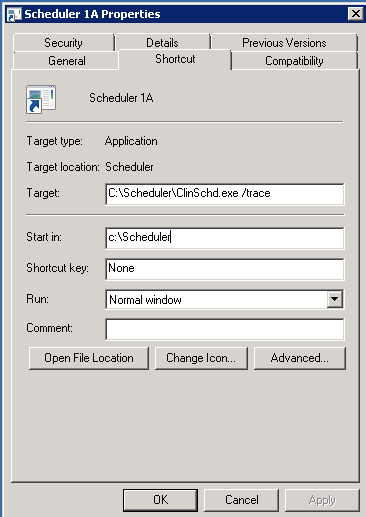


Figure 1: Trace Log Capability Shortcut

To launch the trace log, the user must right click in the Title Bar of the window and select the **Show Trace Log** option.

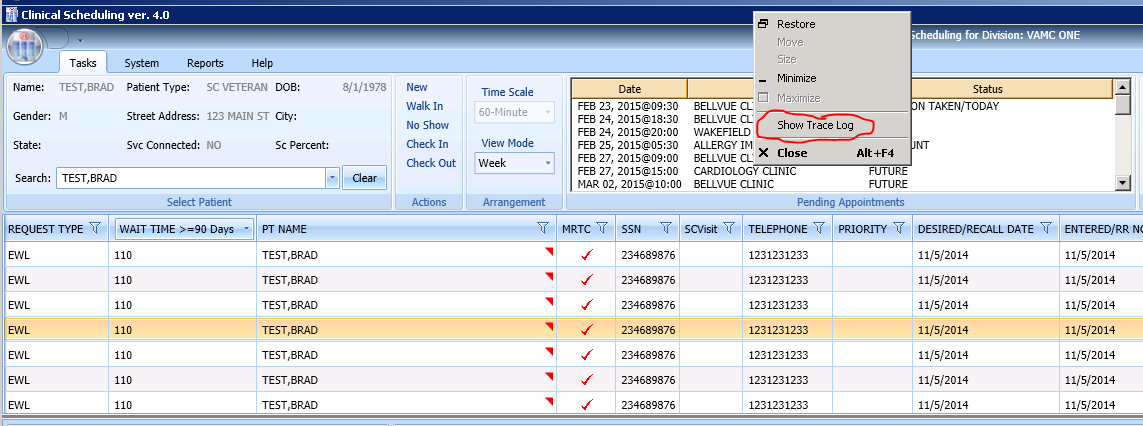


Figure 2: Show Trace Log Menu Option

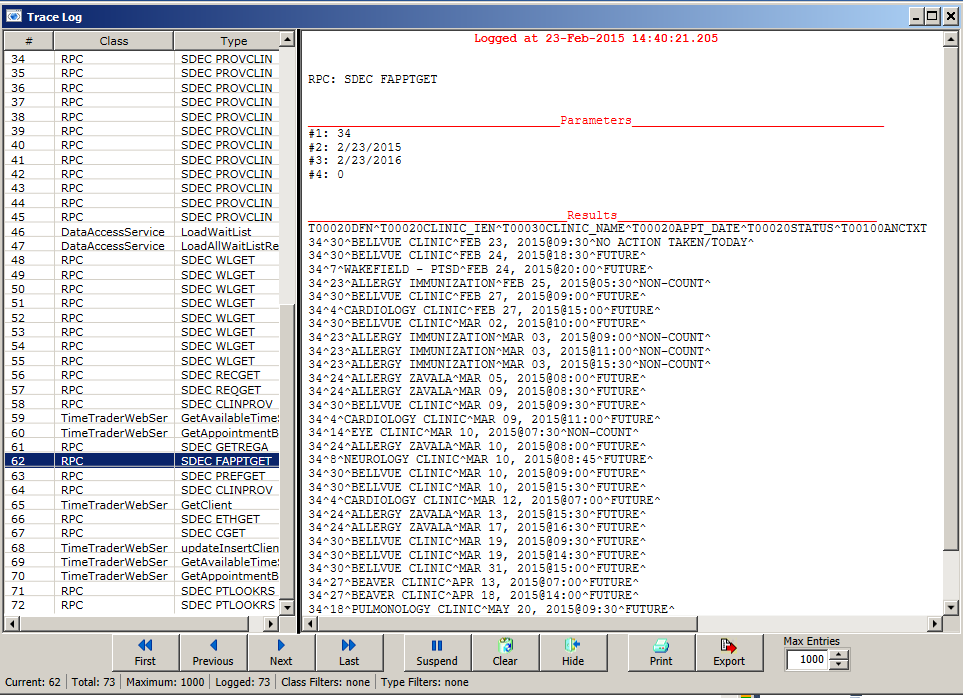


Figure 3: VistA Scheduling GUI Trace Log

Additional logging and debug information can be found in the Application folder (same folder as the ClinSchd.exe file) in a file named **trace.log**. This file can be opened in any text editor/viewer for inspection.

#### Request Management

The VSE scheduler creates and manages several different types of appointment requests: Appointments (APPT), Waitlist entries (EWL), Recalls, and Consults. These requests are retrieved in stored in different files in VistA. To view requests users will submit queries to the VistA server in order to retrieve individual requests. Users can search by patients name, request type, clinic or service/specialty clinic, priority group, wait time, service, connection, desired date, and origination date. Once the query is submitted to the server, the records are filtered based on the current query. Users can also specify a particular sort for the records based on: patient name, request type, clinic, wait time, priority group, origination date, desired date of appointment and service connection. The default sort is grouped by priority group, then by desired date, and then by origination date.

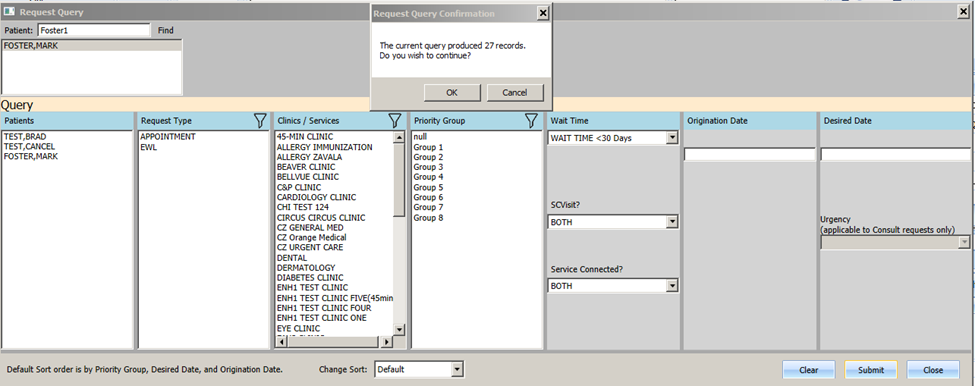


Figure 4: Request Query

All records meeting the query criteria are filtered and sorted on the server and returned to the GUI. Only 25 records are returned at a time. The Request Management grid contains paging functionality for the user to retrieve additional records beyond the initial 25.

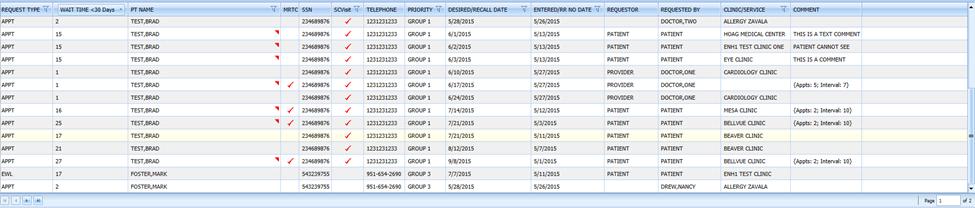


Figure 5: Query Results

## VistA Scheduling GUI Cross References



Figure 6: VistA Scheduling GUI Cross Reference

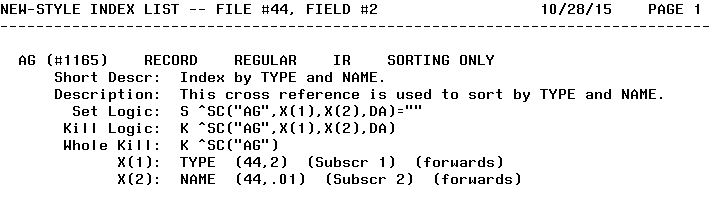


Figure 7: VistA Scheduling GUI Cross Reference (cont.)

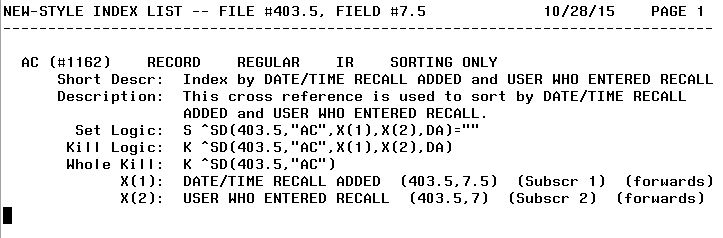
****

Figure 8: VistA Scheduling GUI Cross Reference (cont.)

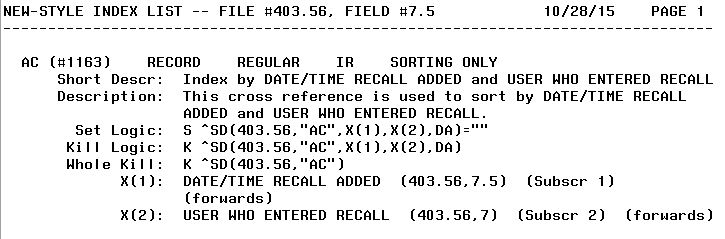


Figure 9: VistA Scheduling GUI Cross Reference (cont.)

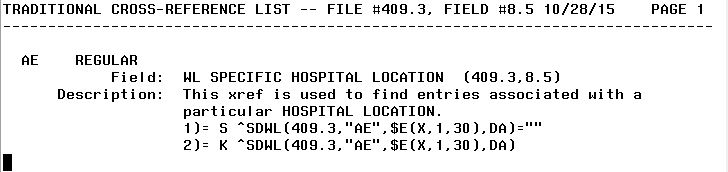


Figure 10: VistA Scheduling GUI Cross Reference (cont.)

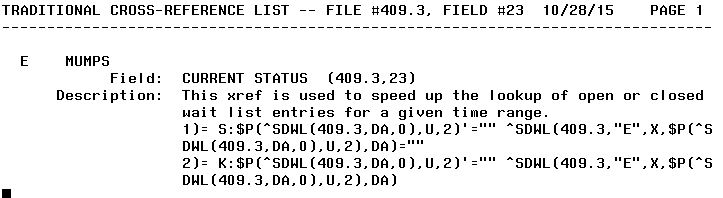


Figure : VistA Scheduling GUI Cross Reference (cont.)

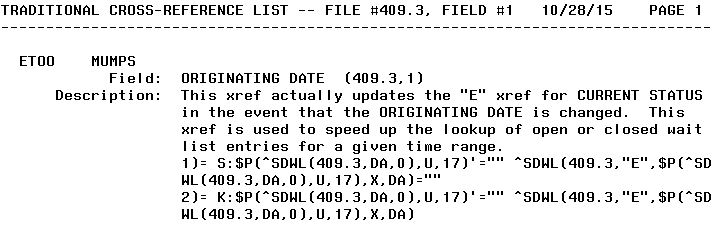


Figure : VistA Scheduling GUI Cross Reference (cont.)

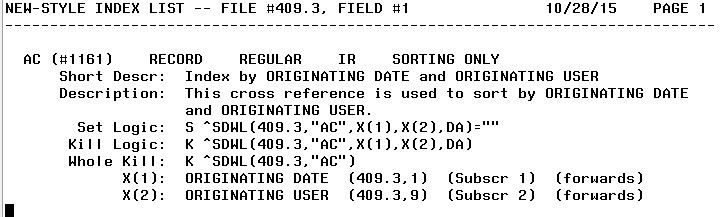


Figure : VistA Scheduling GUI Cross Reference (cont.)

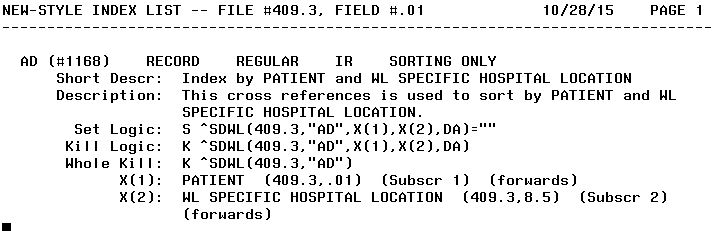


Figure : VistA Scheduling GUI Cross Reference (cont.)

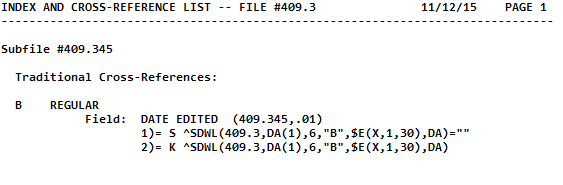


Figure 15: VistA Scheduling GUI Cross Reference (cont.)

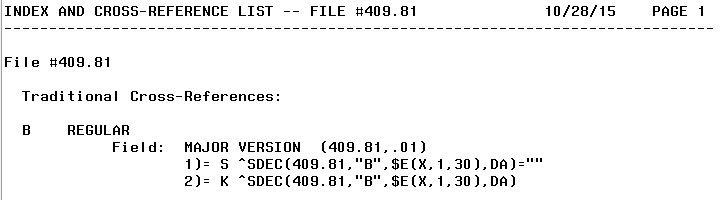
****

Figure 16: VistA Scheduling GUI Cross Reference (cont.)

****

Figure 17: VistA Scheduling GUI Cross Reference (cont.)

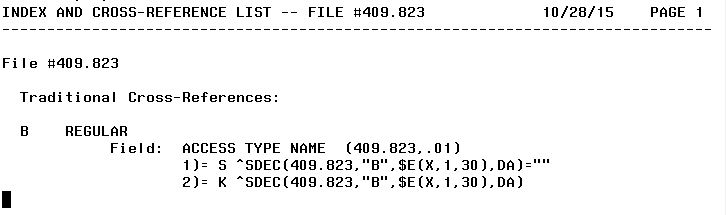
****

Figure 18: VistA Scheduling GUI Cross Reference (cont.)

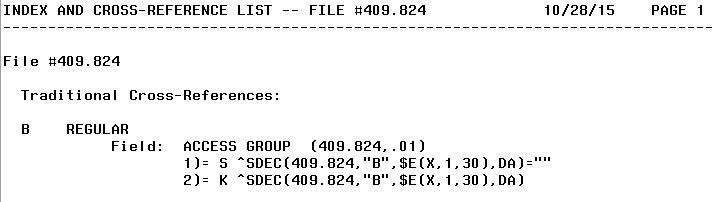
****

Figure 19: VistA Scheduling GUI Cross Reference (cont.)

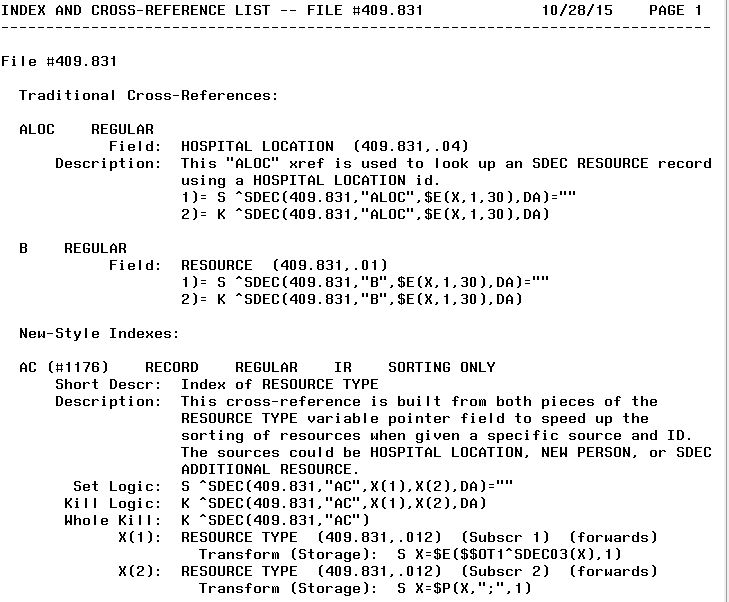


Figure 16: VistA Scheduling GUI Cross Reference (cont.)

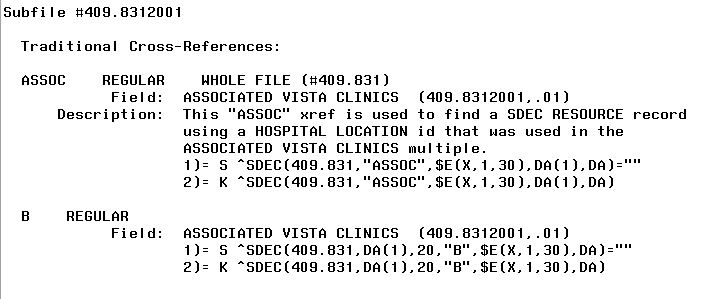


Figure 21: VistA Scheduling GUI Cross Reference (cont.)

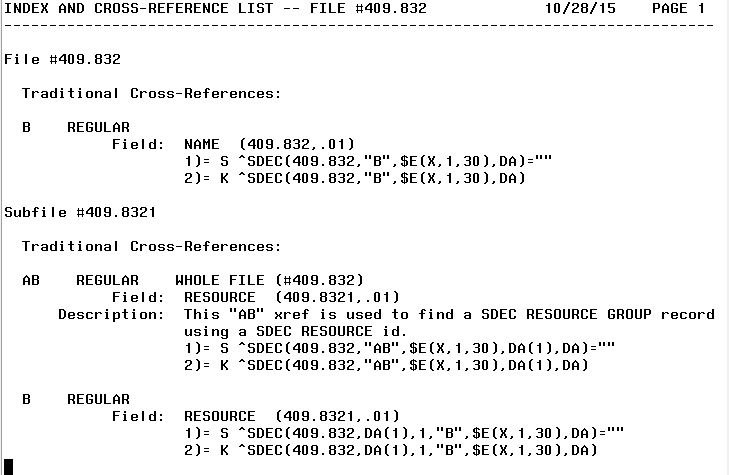


Figure 17: VistA Scheduling GUI Cross Reference (cont.)

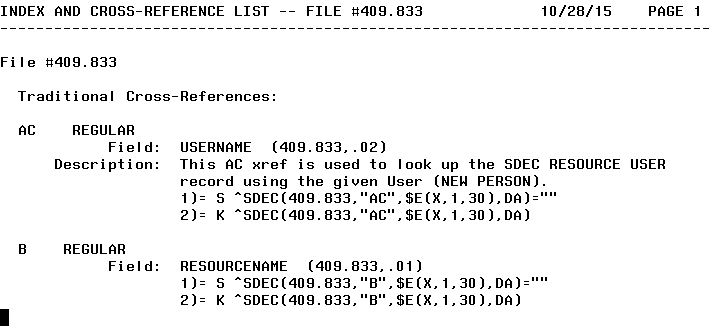


Figure : VistA Scheduling GUI Cross Reference (cont.)

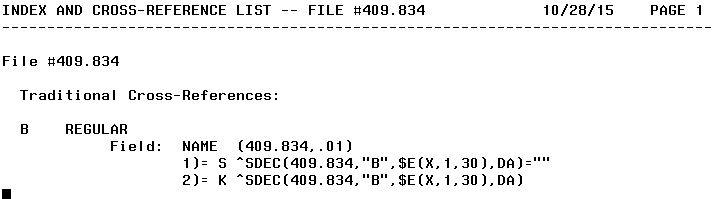


Figure : VistA Scheduling GUI Cross Reference (cont.)

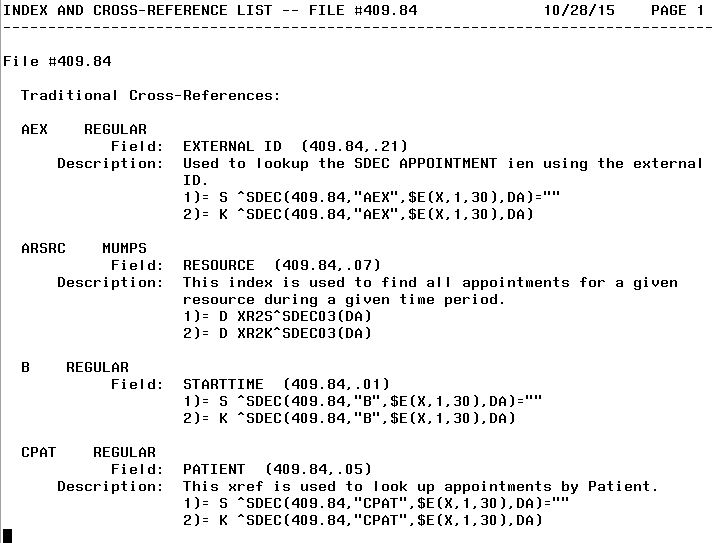


Figure : VistA Scheduling GUI Cross Reference (cont.)

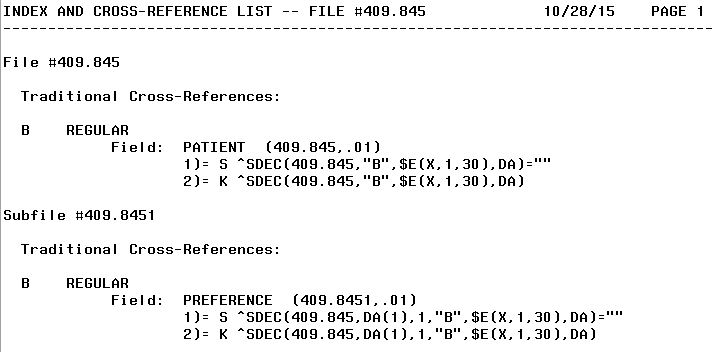


Figure : VistA Scheduling GUI Cross Reference (cont.)

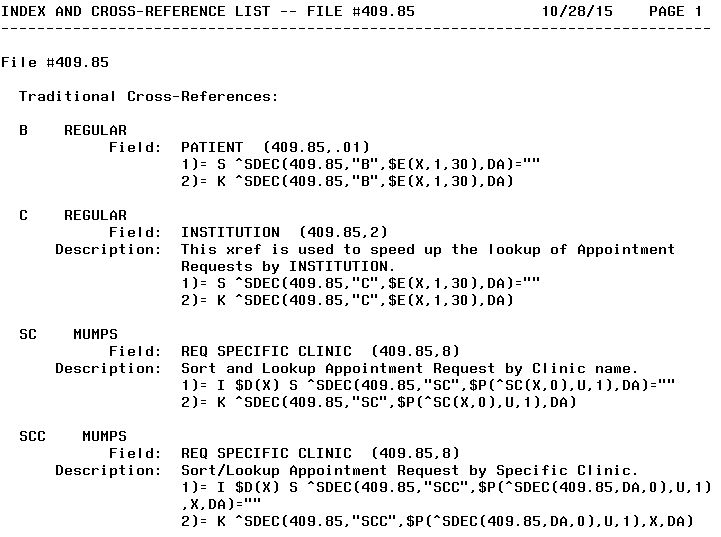


Figure : VistA Scheduling GUI Cross Reference (cont.)

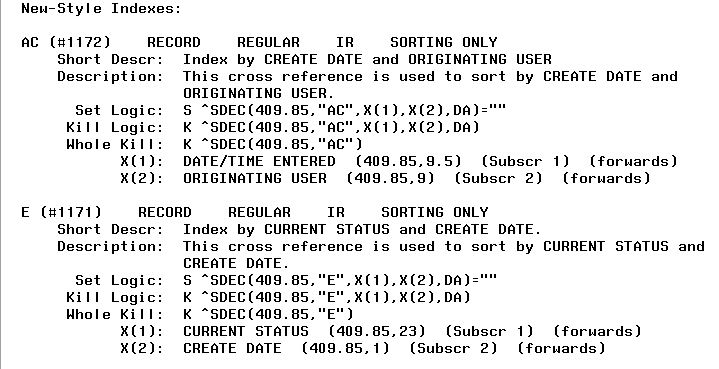


Figure : VistA Scheduling GUI Cross Reference (cont.)

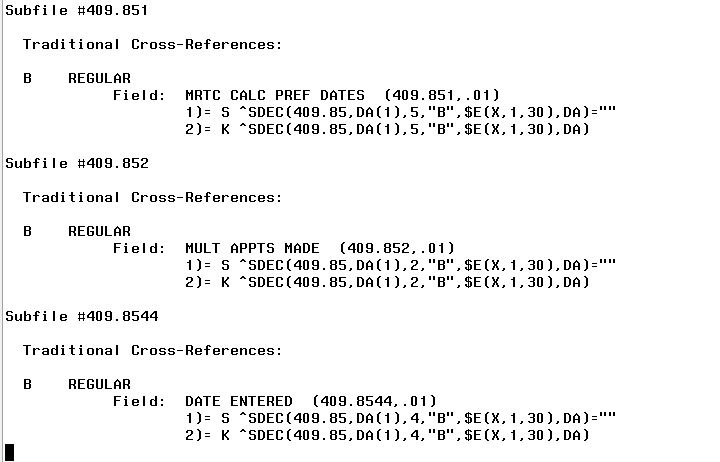


Figure : VistA Scheduling GUI Cross Reference (cont.)

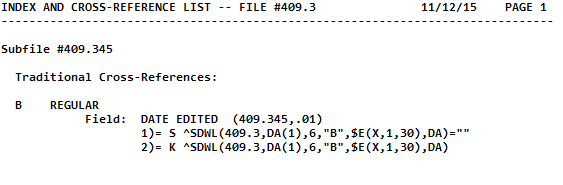


Figure : VistA Scheduling GUI Cross Reference (cont.)

## Table File

Figure 20 through Figure 80 depict the table file.

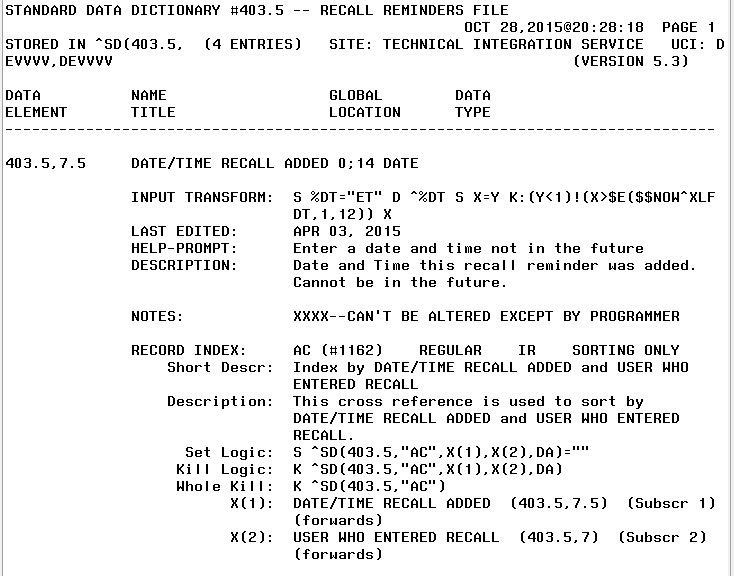


Figure : Table File

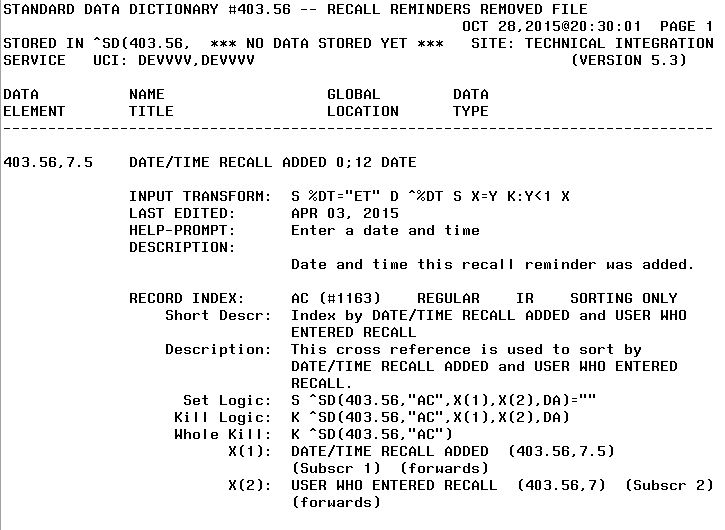


Figure : Table File (cont.)

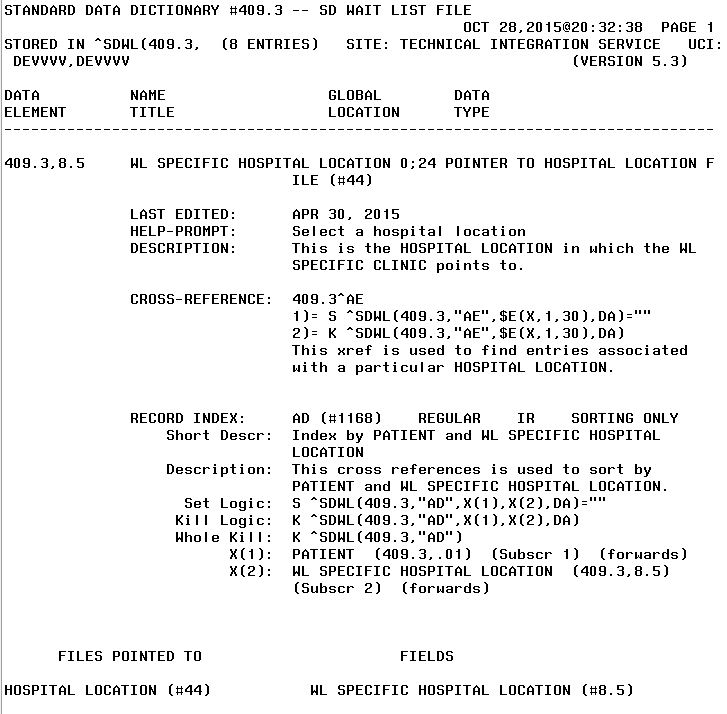


Figure : Table File (cont.)

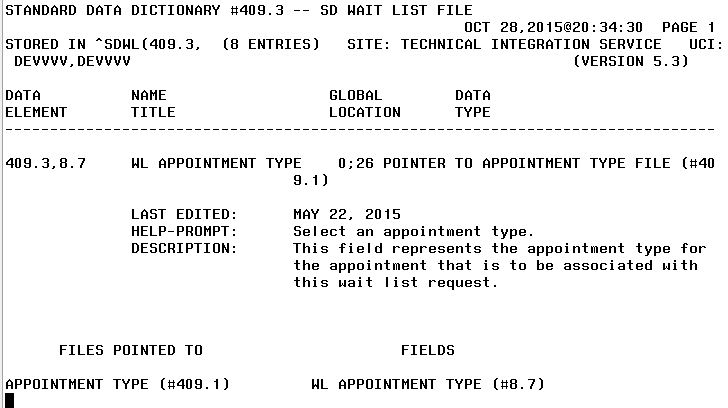


Figure : Table File (cont.)

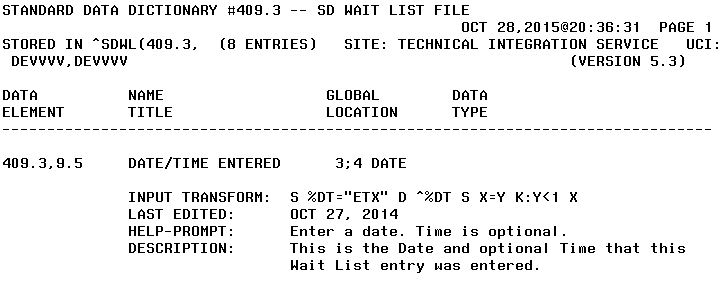


Figure : Table File (cont.)

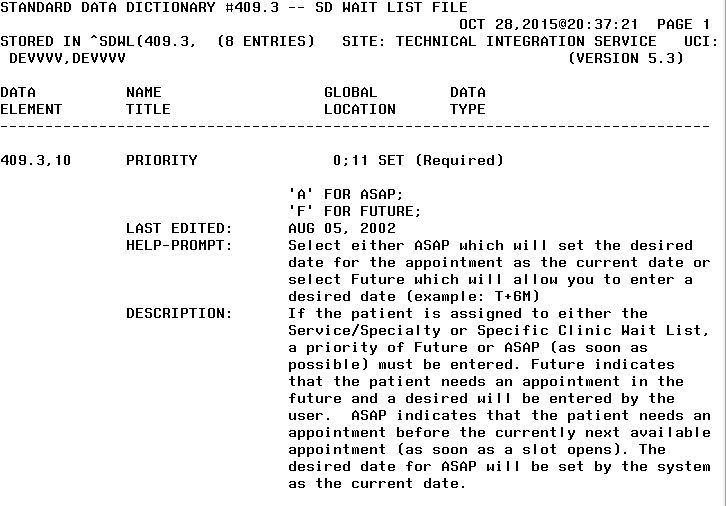


Figure : Table File (cont.)

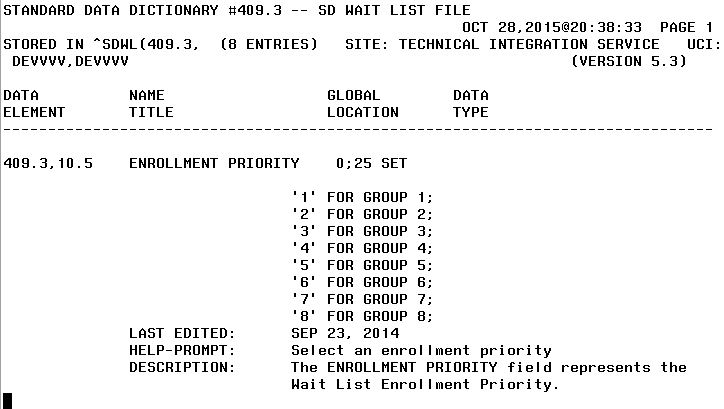


Figure : Table File (cont.)

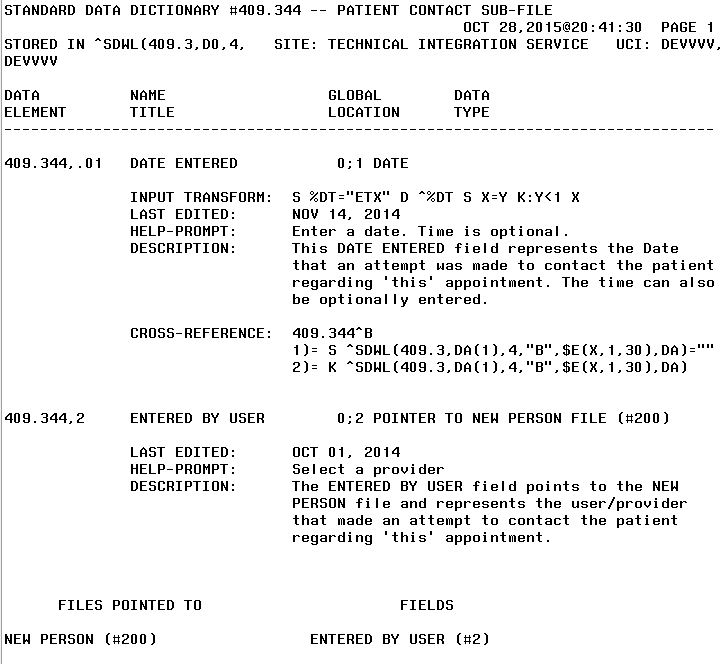


Figure : Table File (cont.)

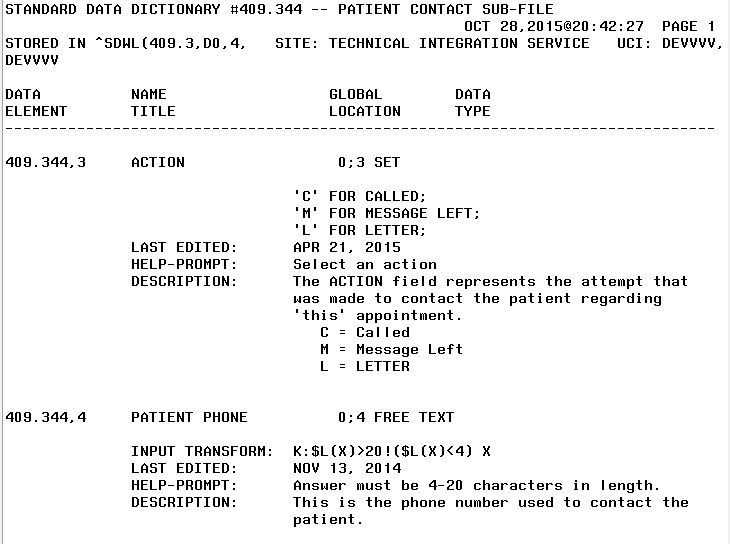


Figure : Table File (cont.)

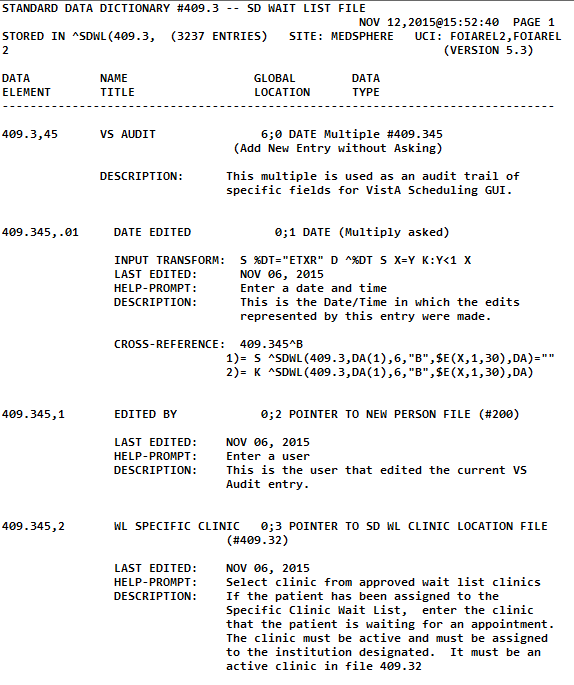


Figure : Table File (cont.)

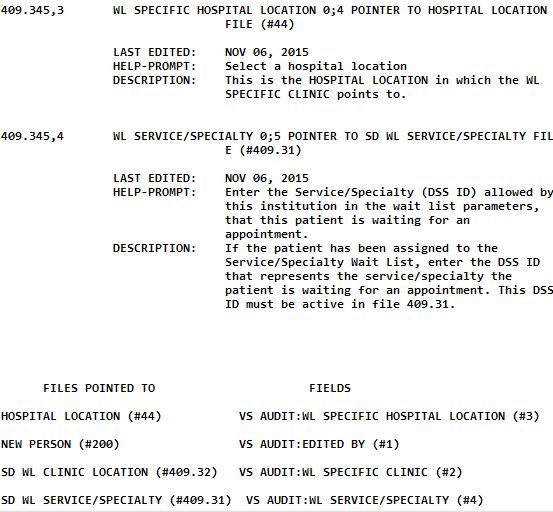


Figure : Table File (cont.)

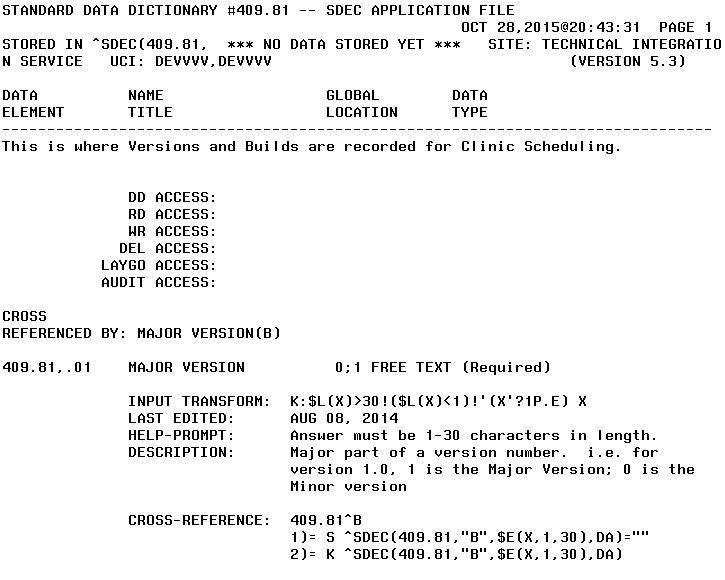


Figure : Table File (cont.)

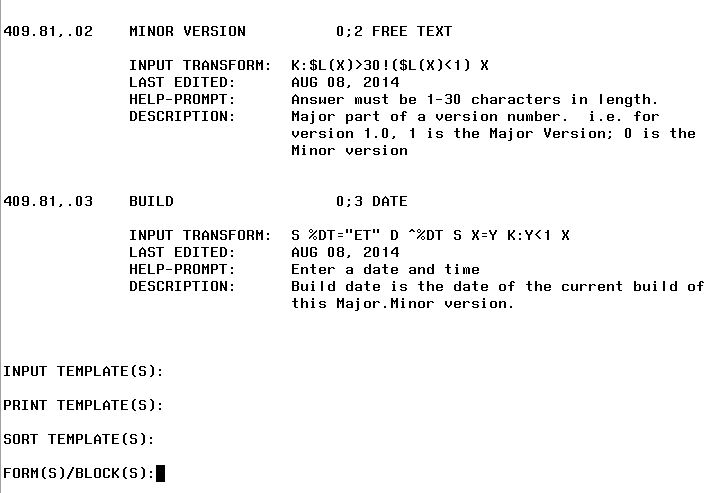


Figure : Table File (cont.)

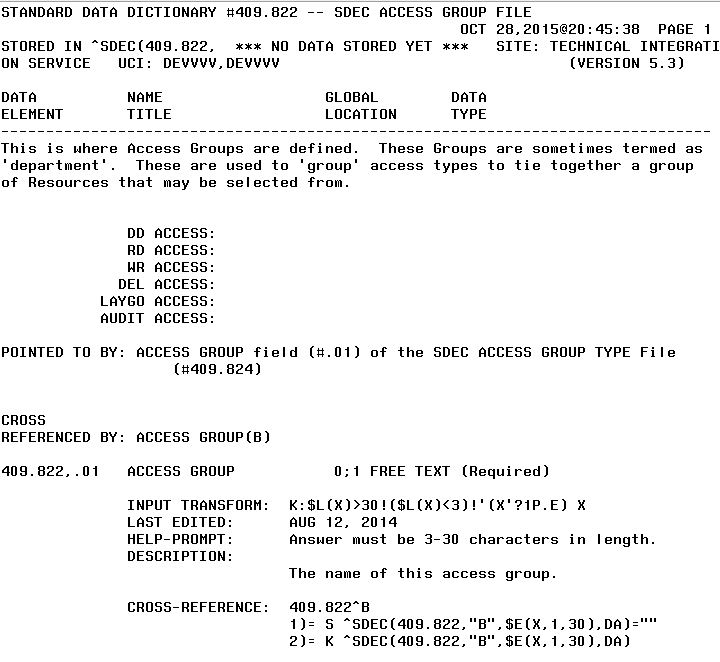


Figure : Table File (cont.)

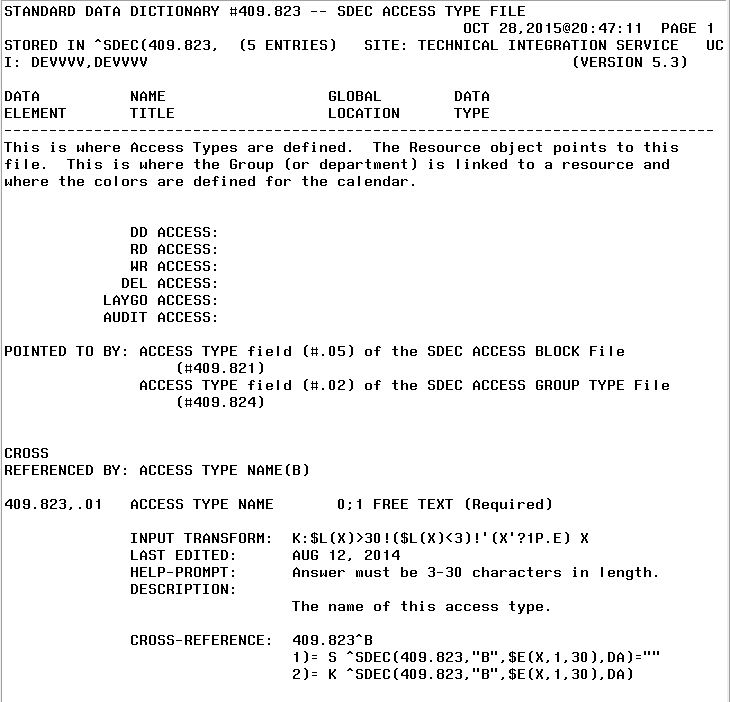


Figure : Table File (cont.)

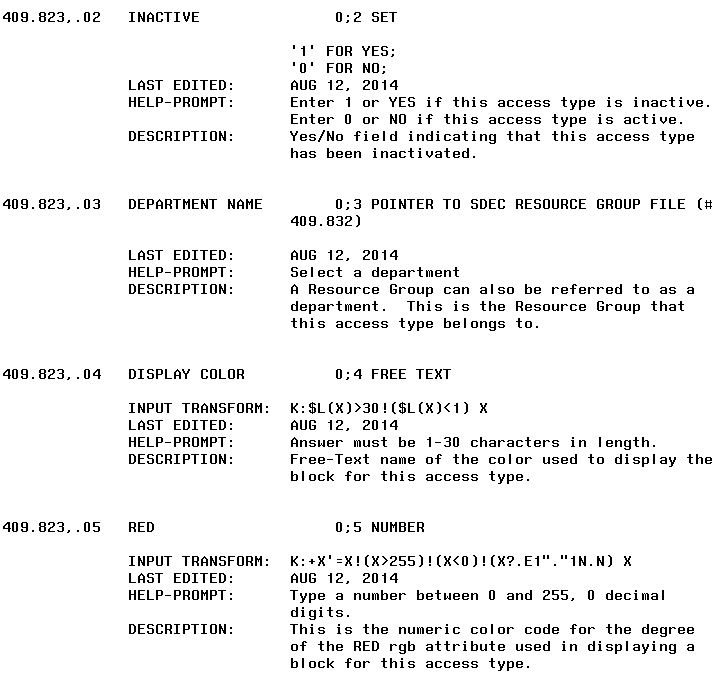


Figure : Table File (cont.)

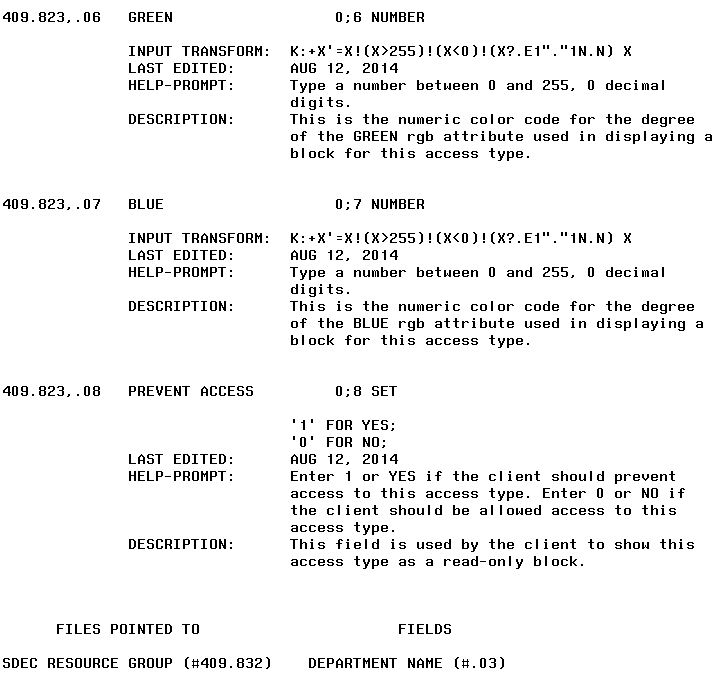


Figure : Table File (cont.)

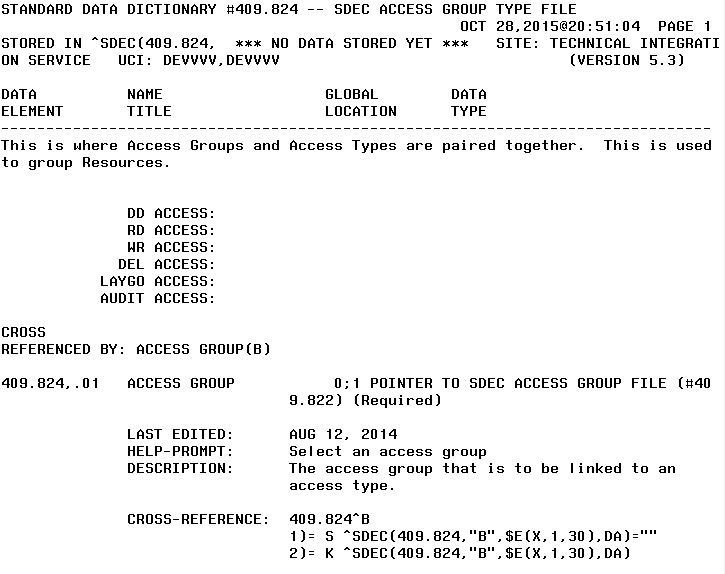


Figure : Table File (cont.)

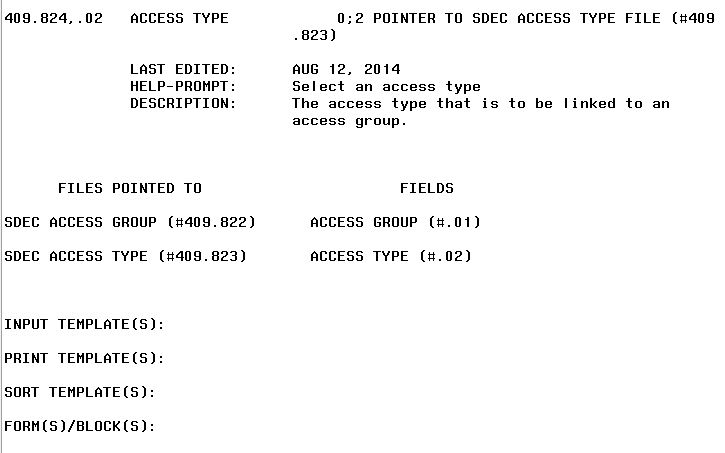


Figure : Table File (cont.)

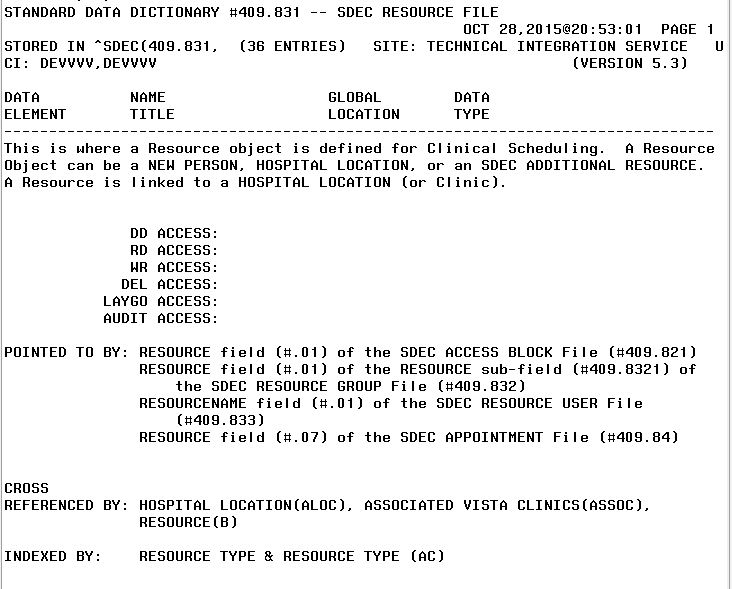


Figure : Table File (cont.)

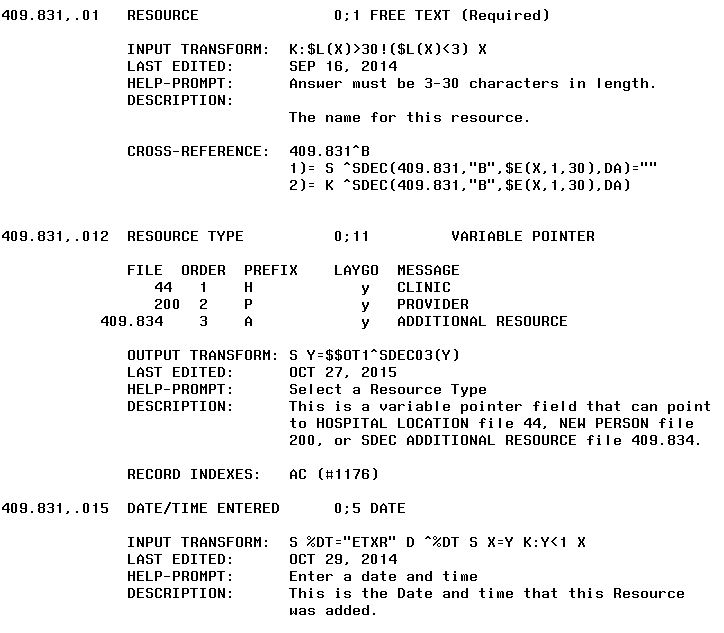


Figure : Table File (cont.)



Figure : Table File (cont.)

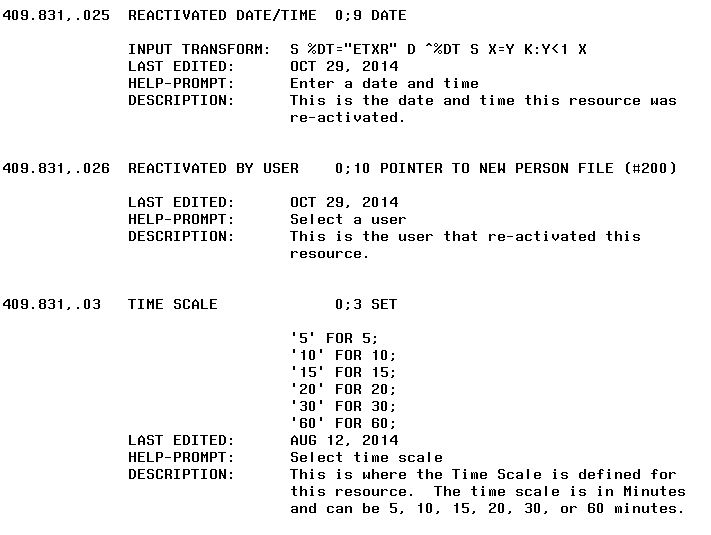


Figure : Table File (cont.)

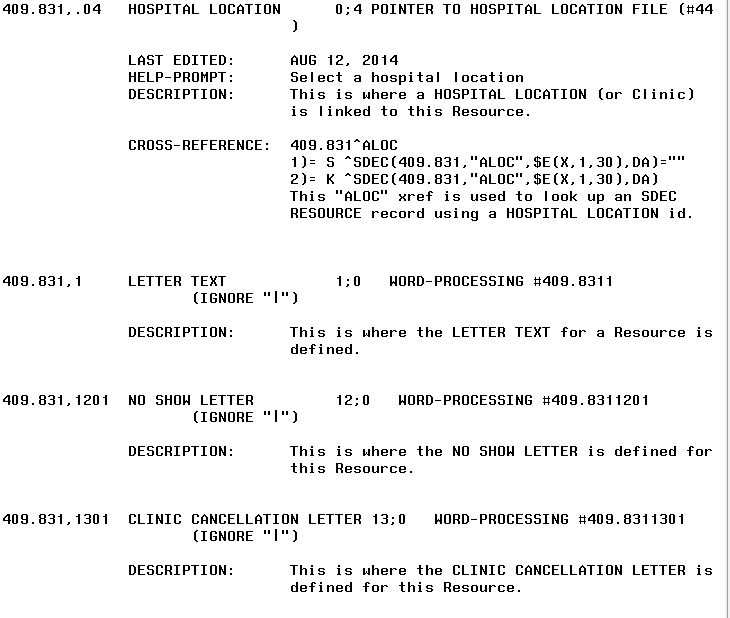


Figure : Table File (cont.)

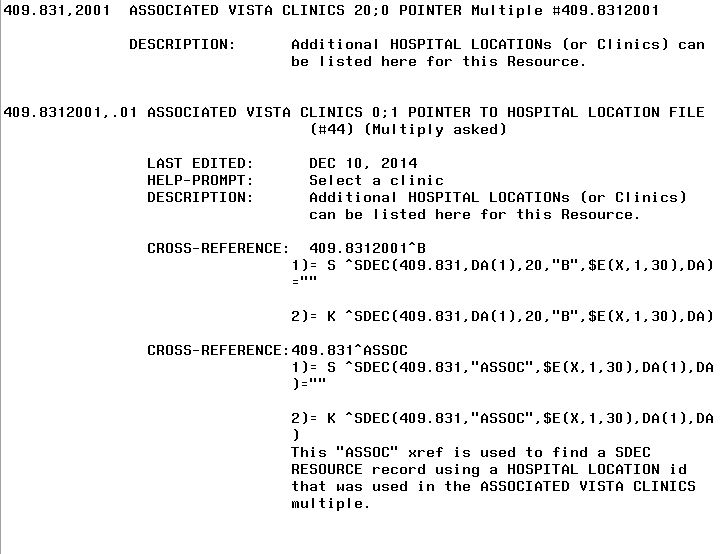


Figure : Table File (cont.)

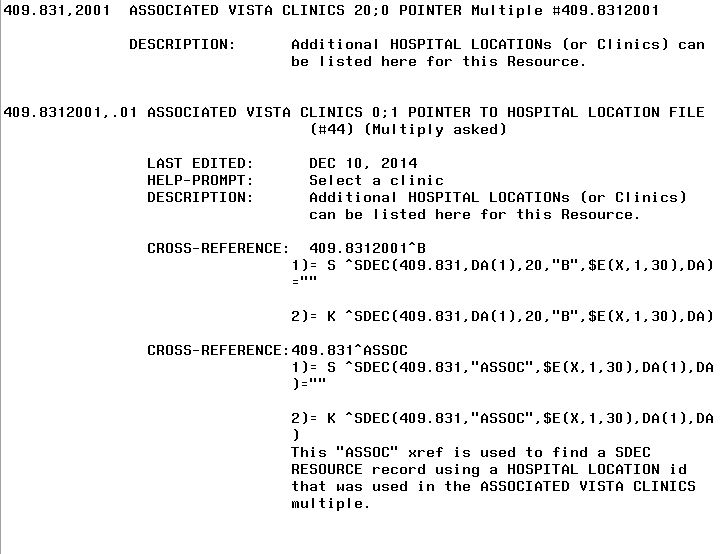


Figure : Table File (cont.)

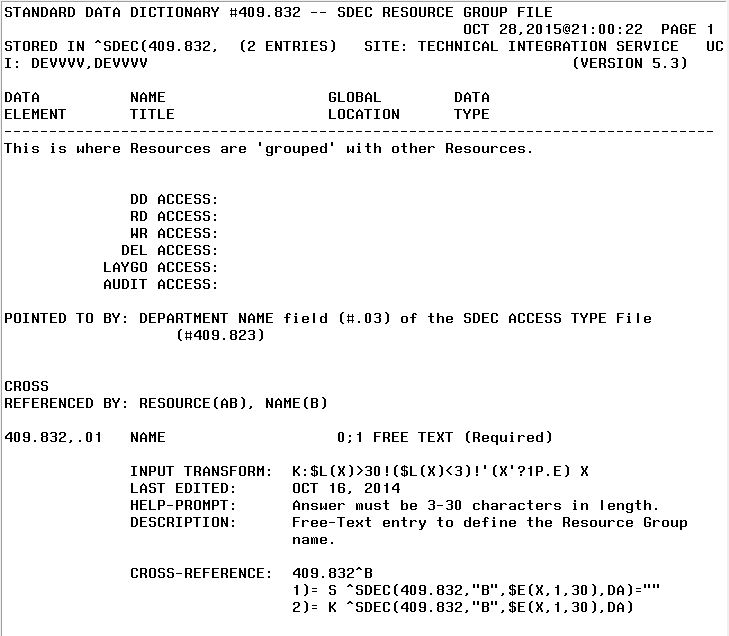


Figure : Table File (cont.)

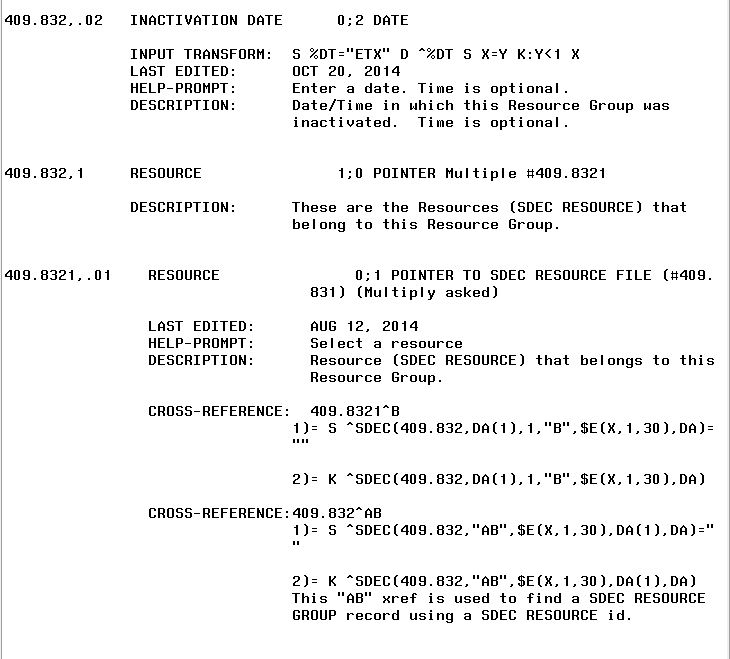


Figure : Table File (cont.)

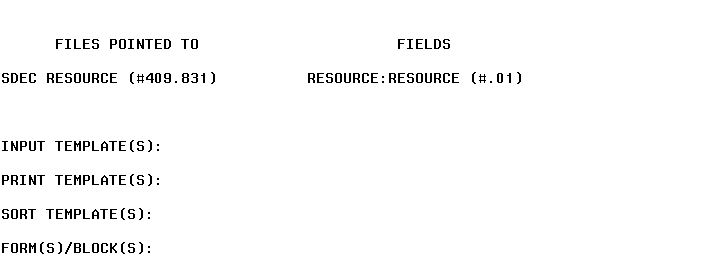


Figure : Table File (cont.)

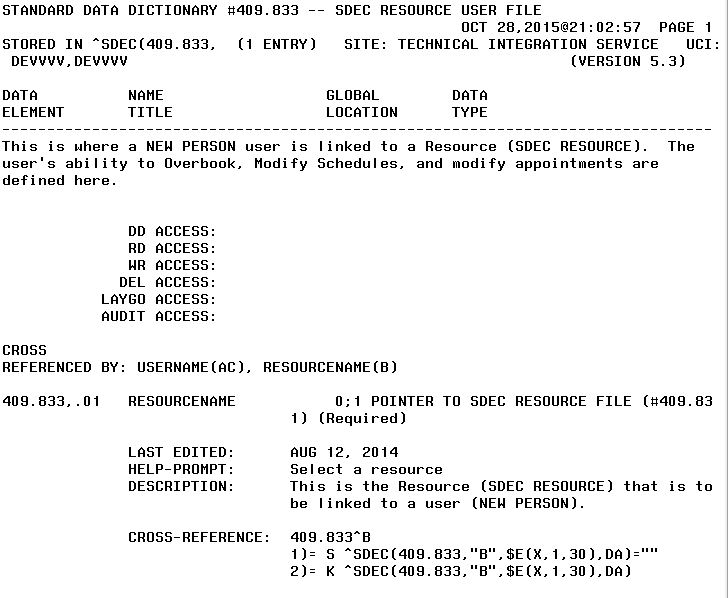


Figure : Table File (cont.)

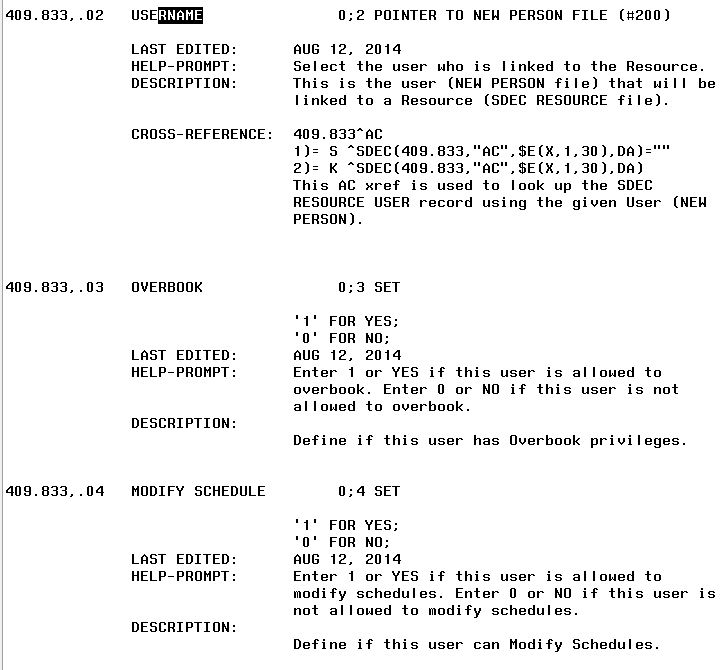


Figure : Table File (cont.)

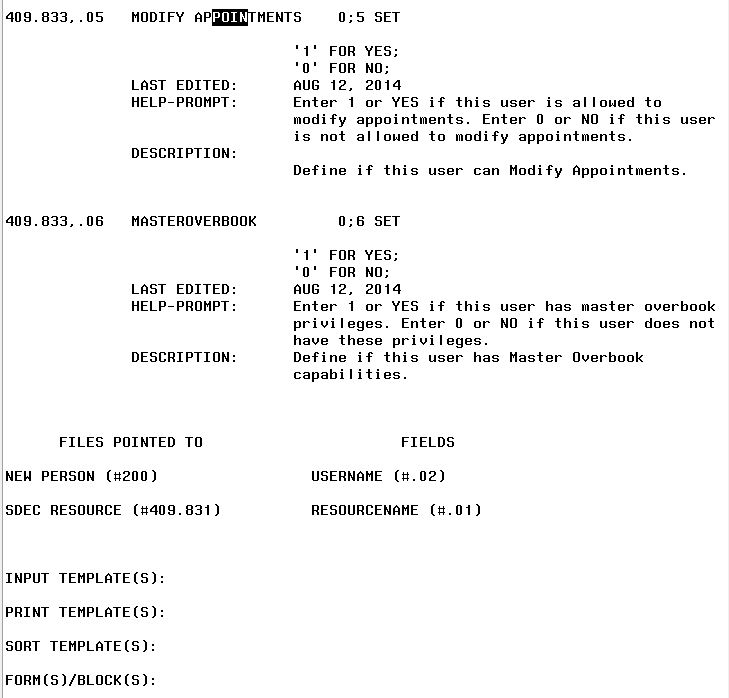


Figure : Table File (cont.)

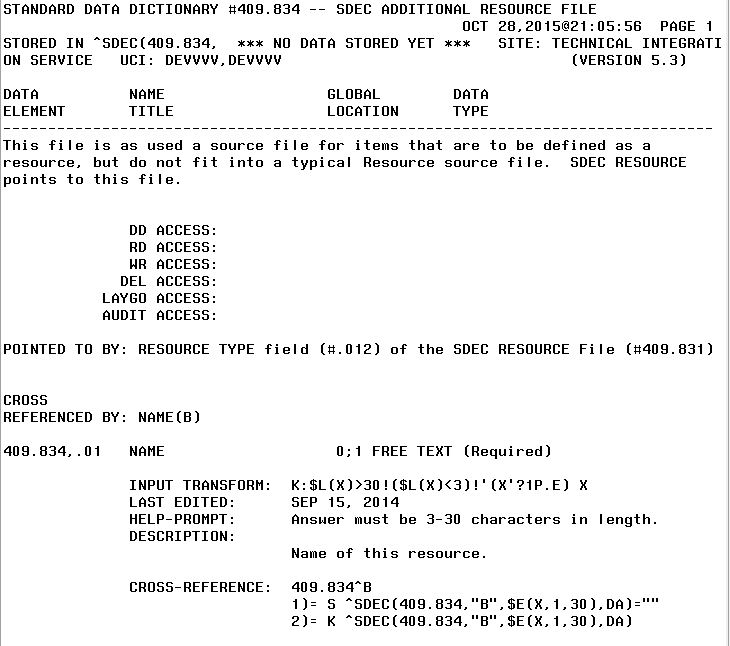


Figure : Table File (cont.)

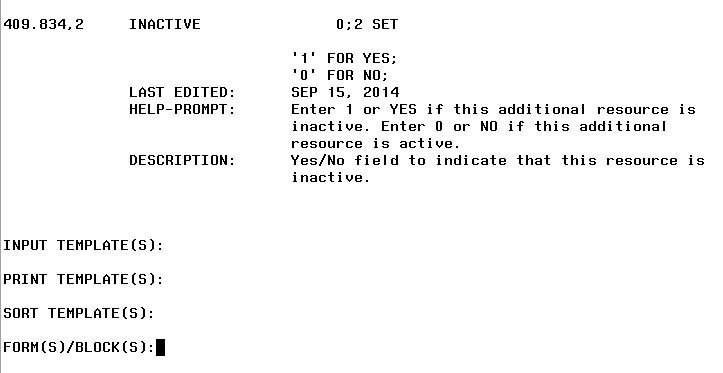


Figure 59: Table File (cont.)

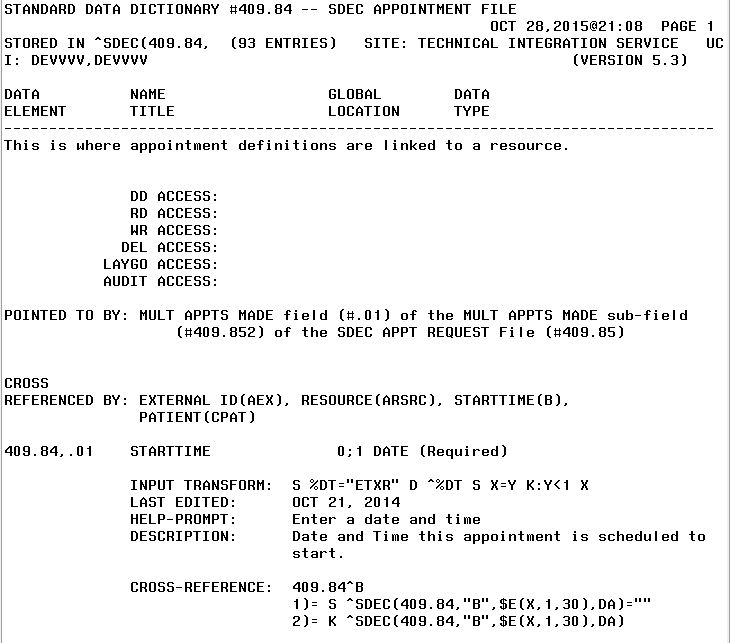


Figure : Table File (cont.)

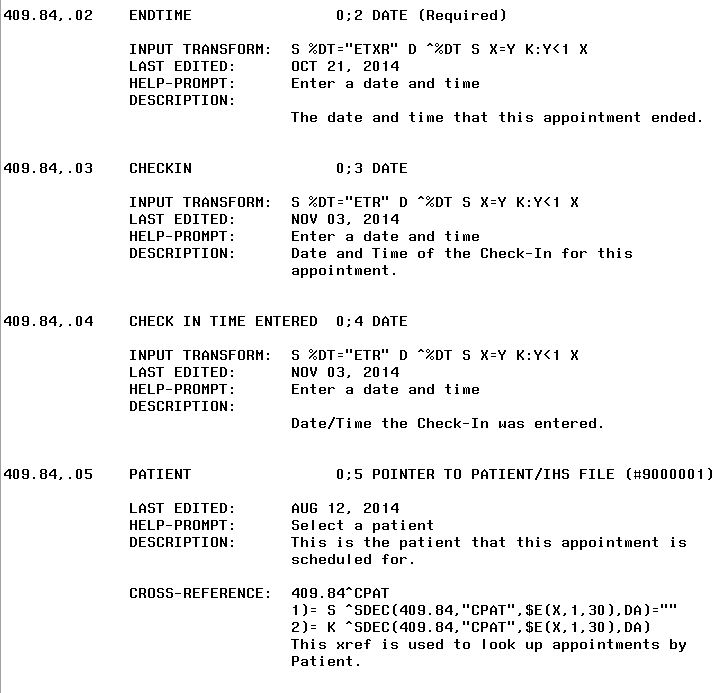


Figure : Table File (cont.)

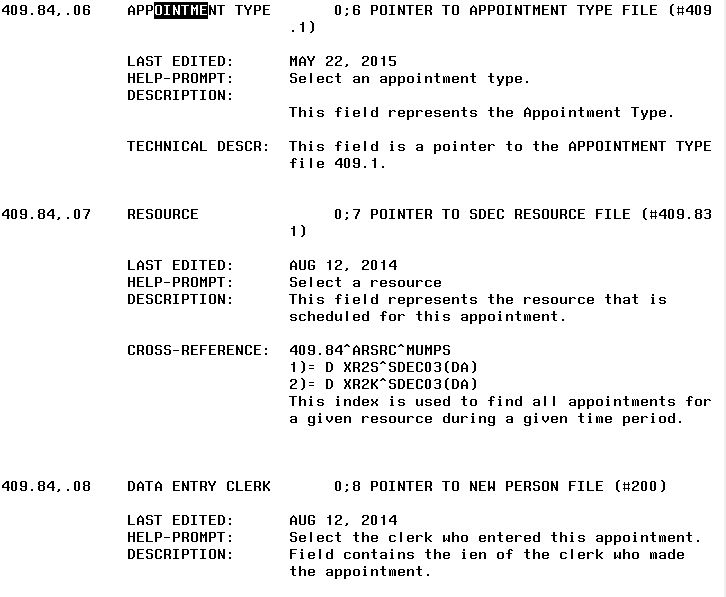


Figure : Table File (cont.)

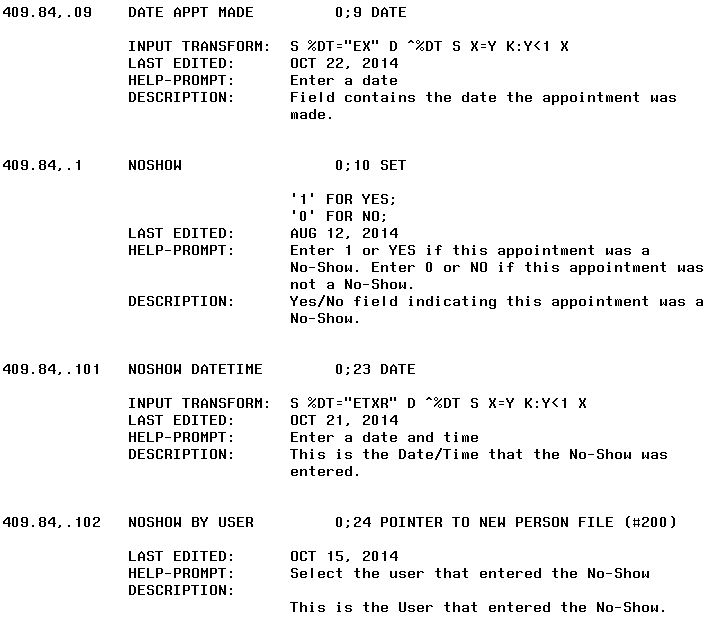


Figure : Table File (cont.)

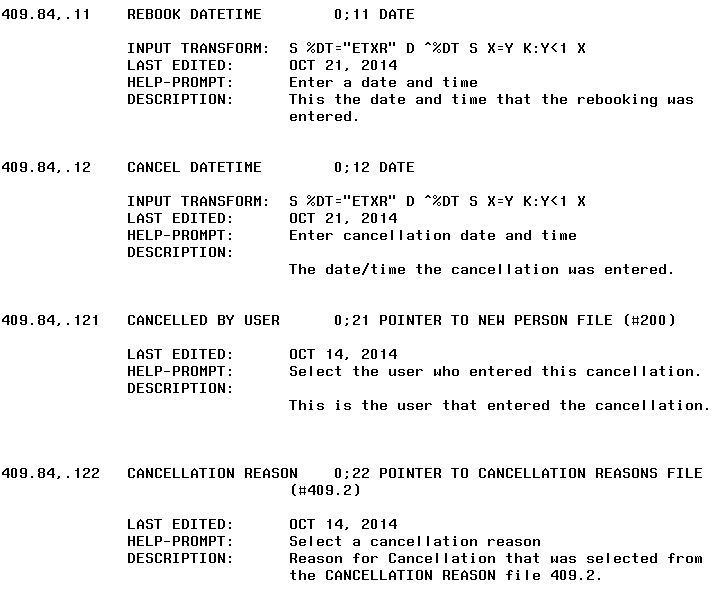


Figure : Table File (cont.)

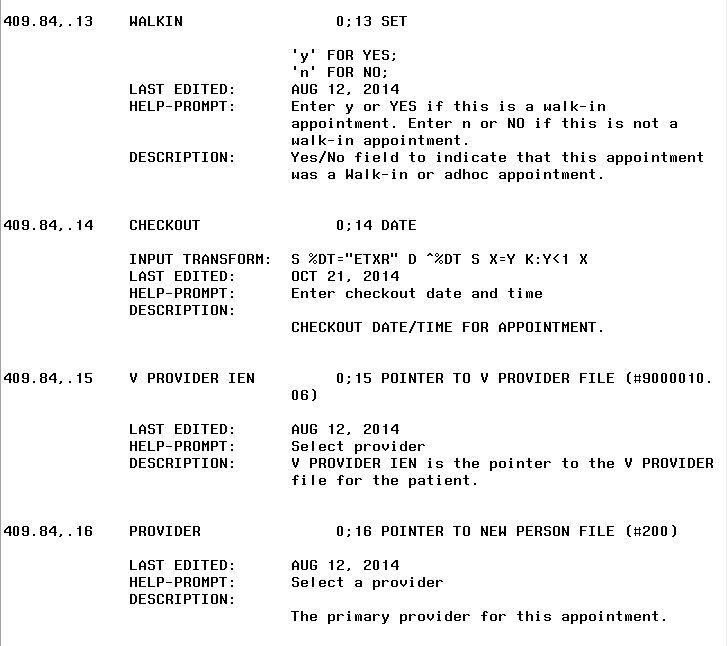


Figure : Table File (cont.)

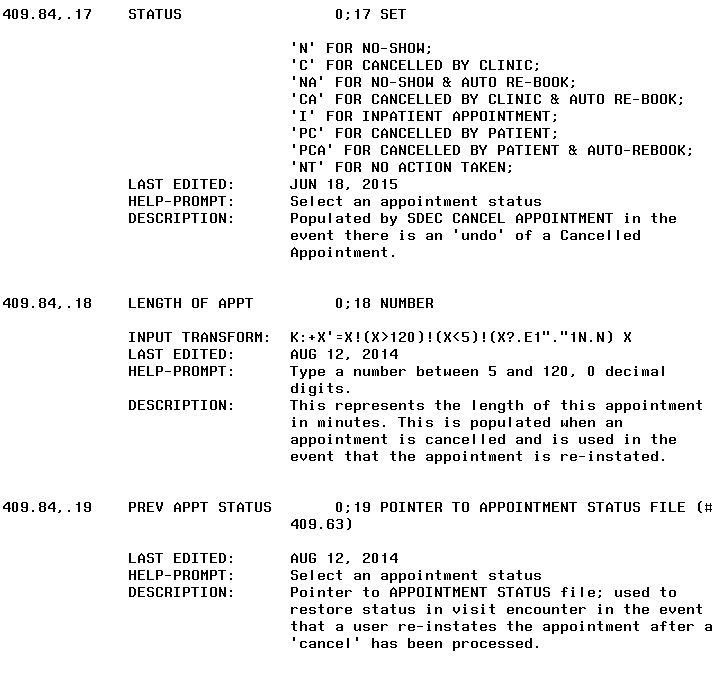


Figure : Table File (cont.)

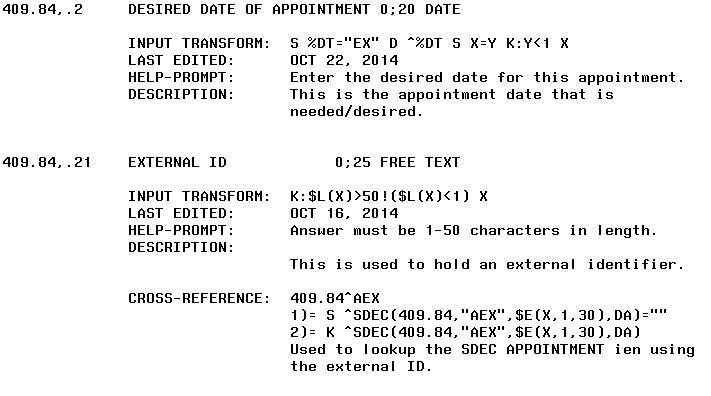


Figure : Table File (cont.)

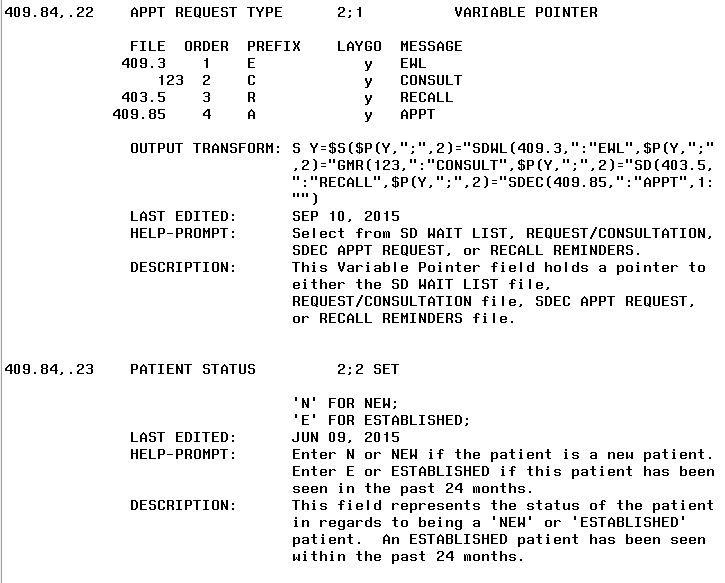


Figure : Table File (cont.)

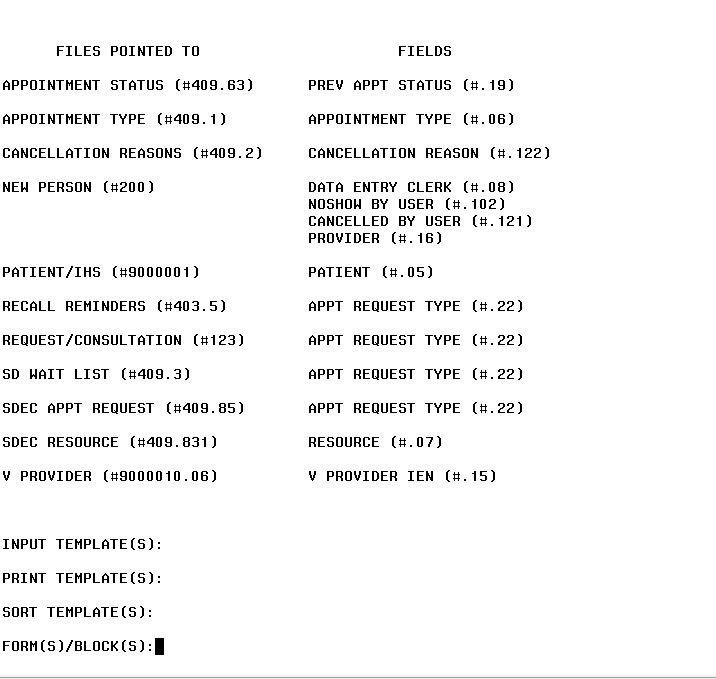


Figure : Table File (cont.)

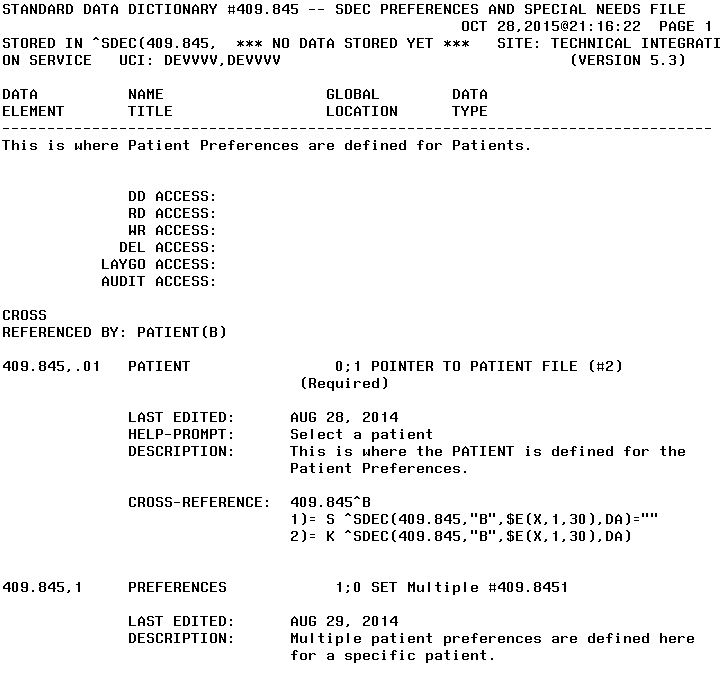


Figure : Table File (cont.)

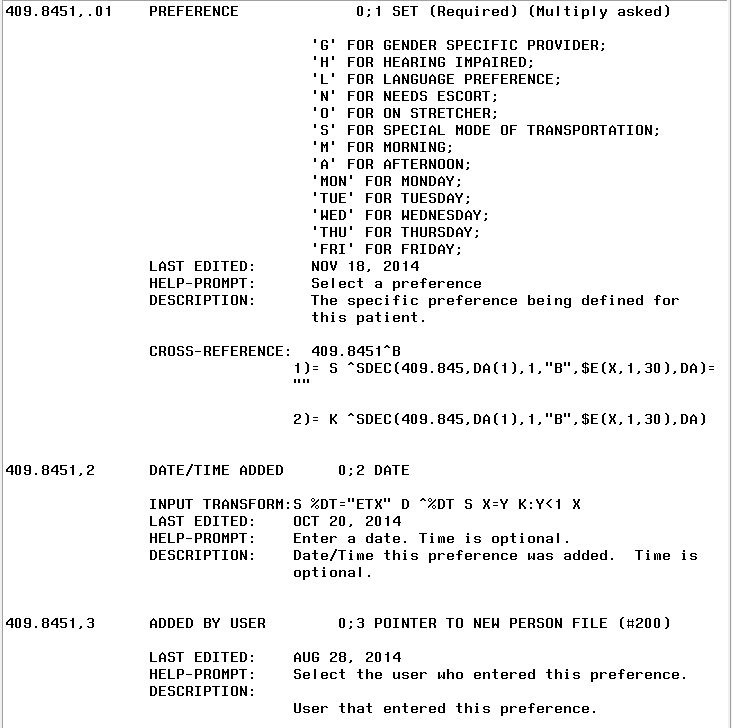


Figure : Table File (cont.)

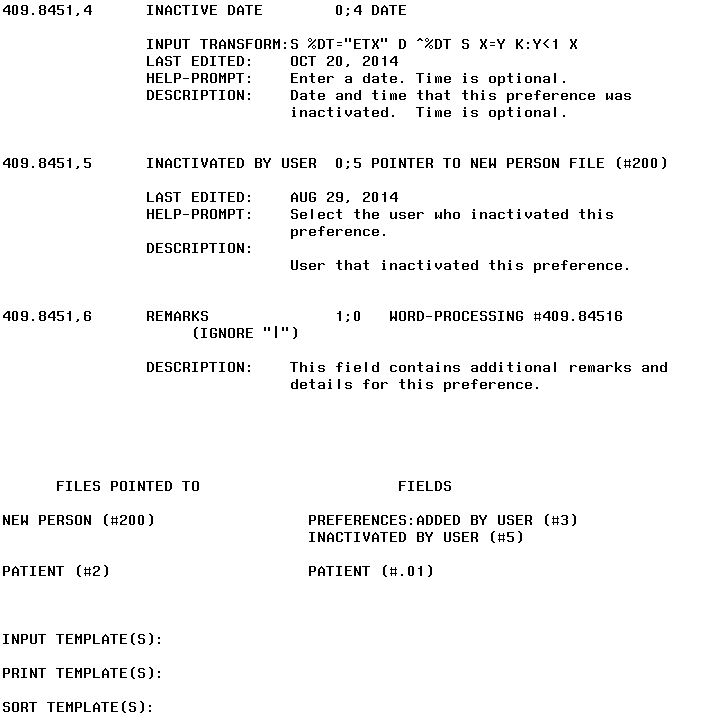


Figure : Table File (cont.)

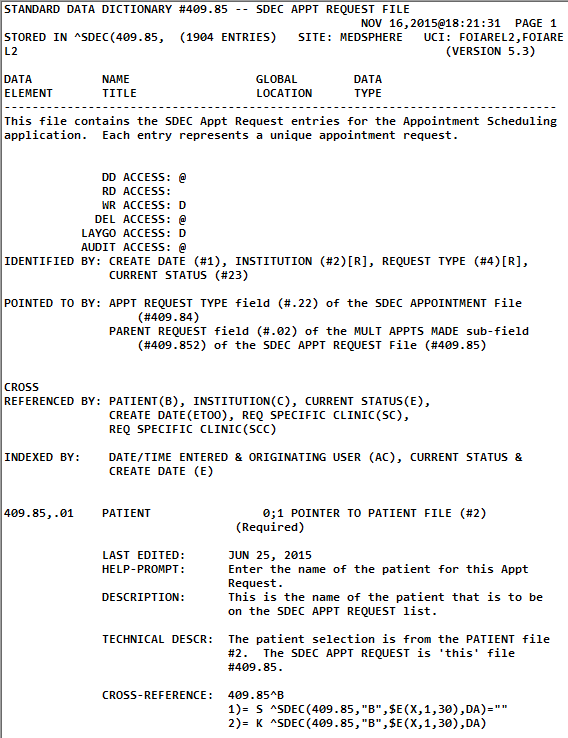
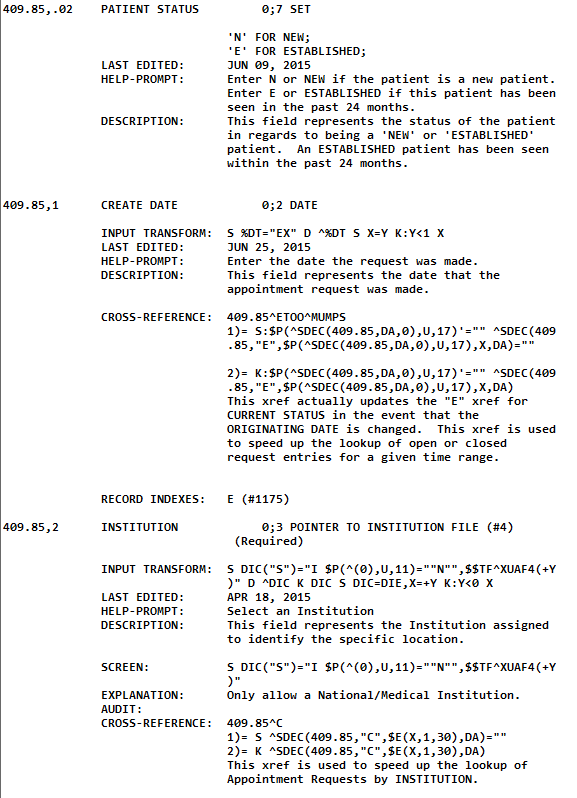


Figure : Table File (cont.)



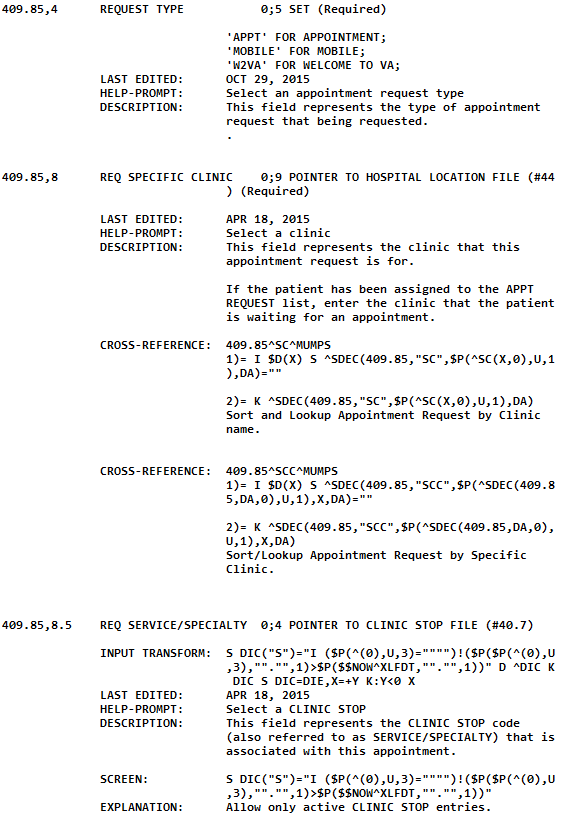
****

Figure 74: Table File (cont.)

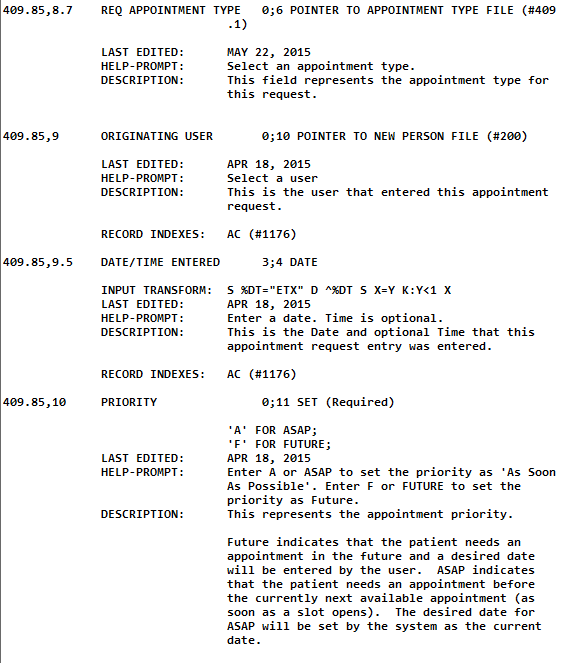


Figure 75: Table File (cont.)

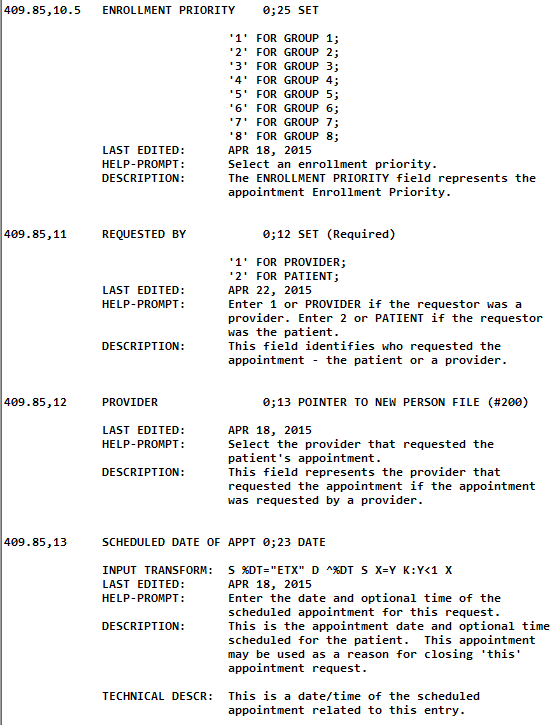


Figure 76: Table File (cont.)

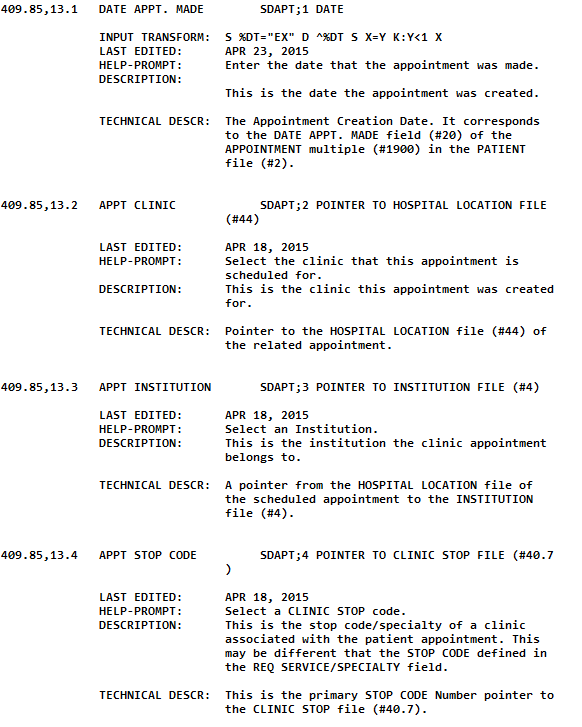


Figure 77: Table File (cont.)

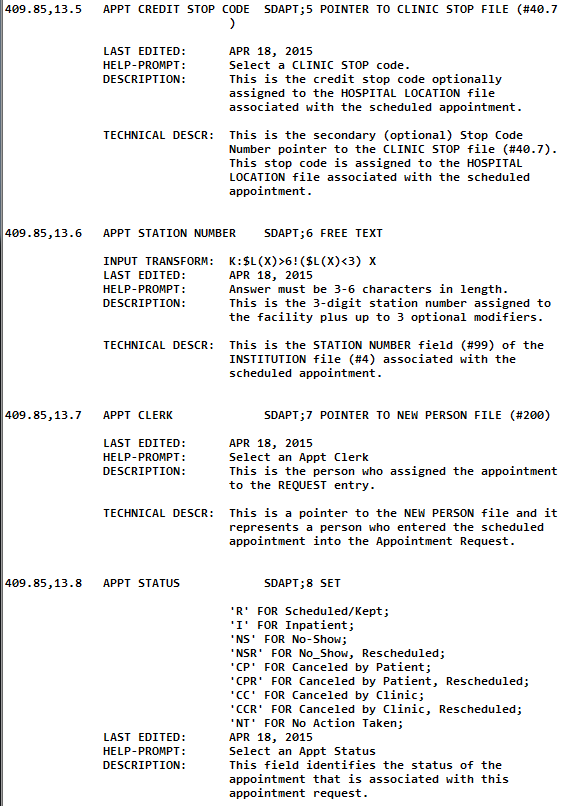


Figure 78: Table File (cont.)

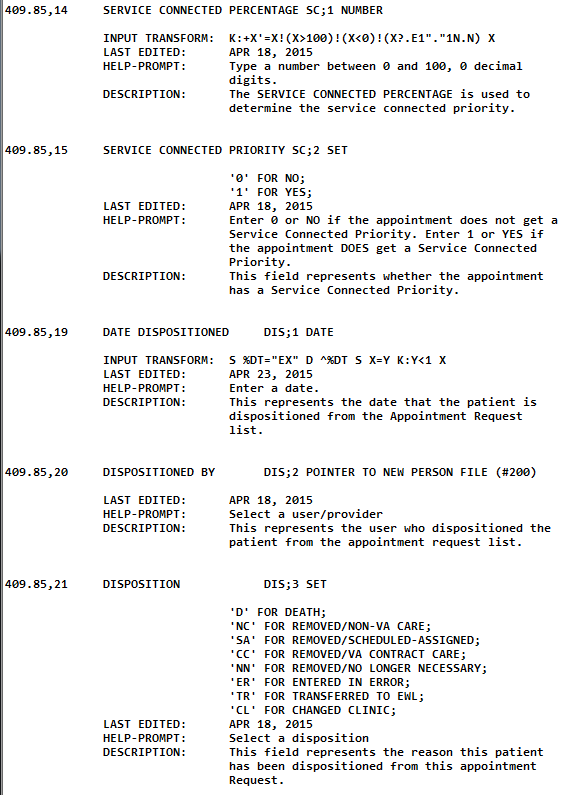


Figure 79: Table File (cont.)

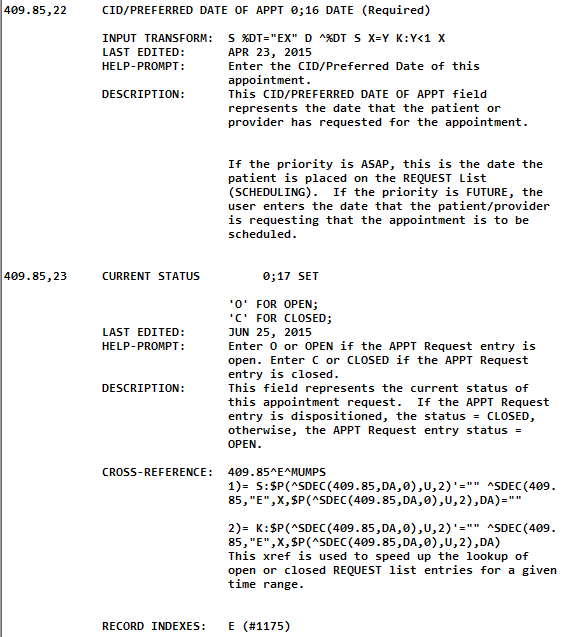


Figure 80: Table File (cont.)

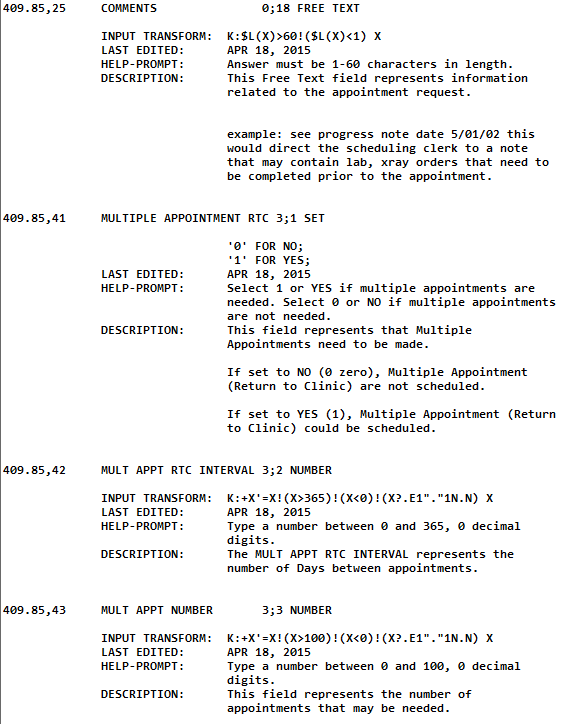


Figure 81: Table File (cont.)

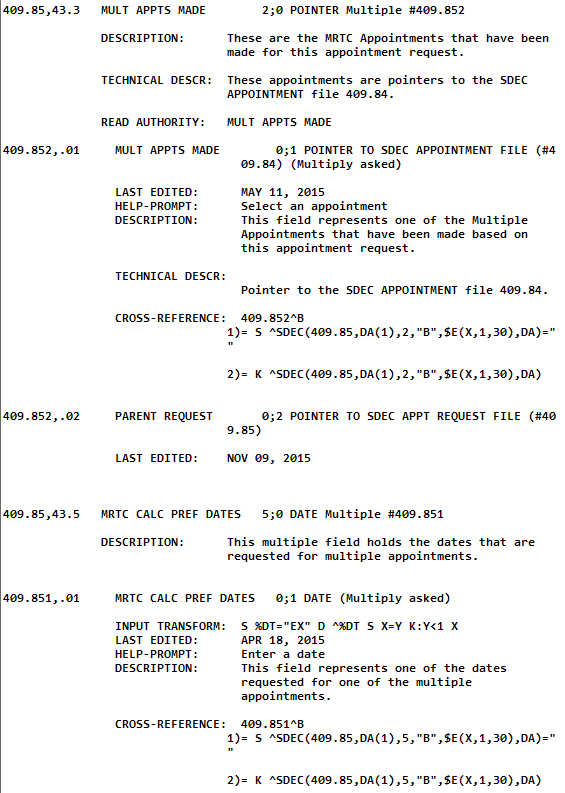


Figure 82: Table File (cont.)

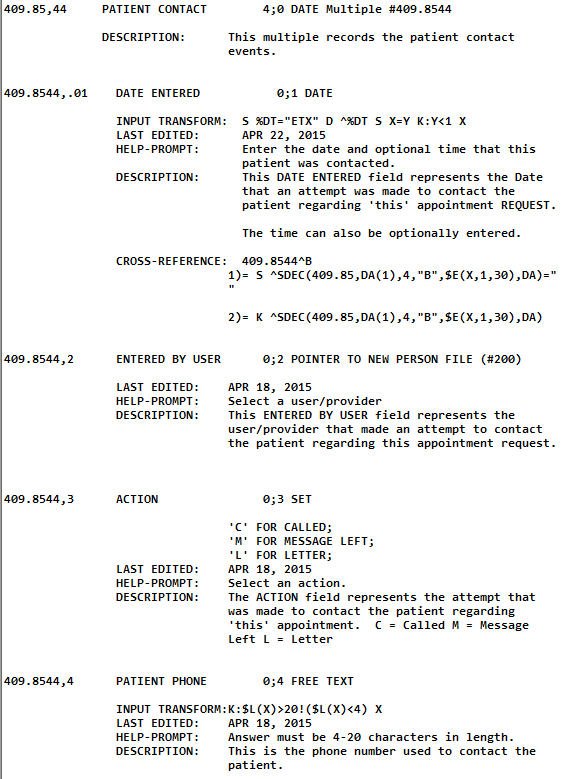


Figure 83: Table File (cont.)

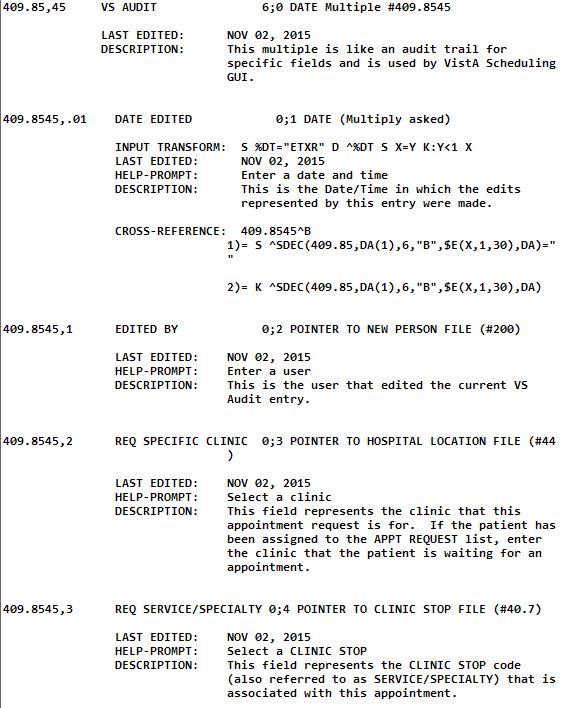


Figure 78: Table File

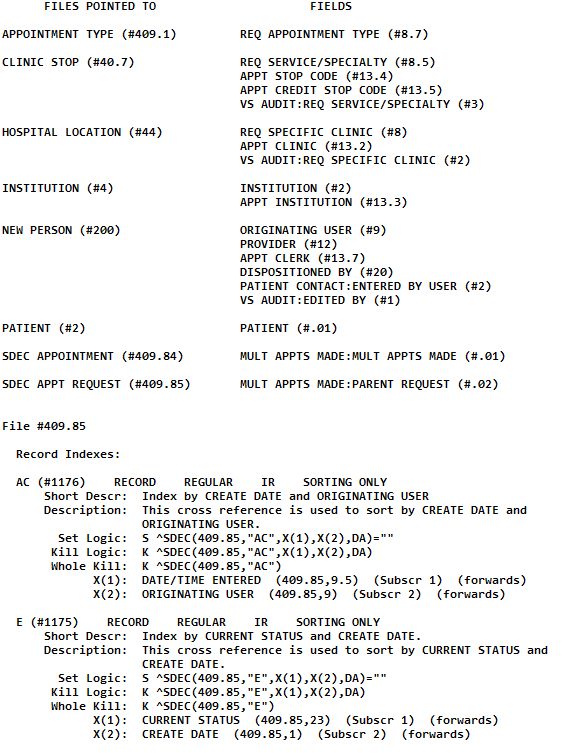


Figure 79: Table File



Figure 80: Table File

## Internal Relations

There are no documented internal relations in VistA Scheduling GUI.

## External Relations

No special integration agreements exist between VistA Scheduling GUI and any other package.

## Published Entry Points

No published entry points exist in VistA Scheduling GUI.

## Exported Option

Table 4: Exported Option and Description

| Option Name | Description |
| --- | --- |
| CS Dashboard Data Compile [SDEC REPORT DATA] | This option allows Scheduling Administrators to initiate the Clinical Scheduling Resource Management report data compilation.  This option is also scheduled to run every 24 hours at 1:00 am to ensure data is collected each day. |
| Add/Edit CS Resource Management Stop Codes [SDEC ADD CLINIC STOP CODES] | This option allows Clinical Scheduling Administrators to activate or inactivate Clinic Stop Codes related to the Report Types displayed in the Clinical Scheduling Resource Management report. |
| View CS Resource Management Clinics [SDEC VIEW CS CLINICS] | This option displays a list of clinic hospital locations that have a Primary Stop Code that matches a CS Resource Management Stop Code. |
| View CS Clinic Stop Codes [SDEC VIEW CS STOP CODES] | This option displays the Clinic Stop Codes configured for the Clinical Scheduling Resource Management Report. |
| CLINICAL SCHEDULING PROCEDURE CALLS [SDECRPC] | This option hosts RPCs in the SDEC namespace. CLINICAL SCHEDULING users must have access to this option in order to use CLINICAL SCHEDULING. |
| Refresh SDEC Index global | This option prepares the ^XTMP("SDEC","IDX" global and should be scheduled to run daily at 2am. |

## Parameter Definitions

Table 5: Parameter Definitions

| Parameter Name | Description |
| --- | --- |
| SDEC MENTAL HEALTH STOP CODES | This Parameter Definition identifies Stop Codes related to the Mental Health report type for the Resource Management Reporting. |
| SDEC PRIMARY CARE STOP CODES | This Parameter Definition identifies Stop Codes related to the Primary Care report type for the Resource Management Reporting. |
| SDEC SPECIALTY CARE STOP CODES | This Parameter Definition identifies Stop Codes related to the Specialty Care report type for the Resource Management Reporting. |

## Security Keys

Table 6: Exported Security Keys

| Key Name | Description |
| --- | --- |
| SDECZMENU | All Window Scheduling users must have this key to access the application. |
| SDECZMGR | This key should be assigned to users who manage the overall Scheduling application. This key gives access to the Scheduling Management menu option on the VSE applications System tab. |
| SDOB | Authorized to overbook in the clinic to maximum allowed, as defined in clinic set up. |
| SDMOB | Authorized to overbook in the clinic over the maximum allowed, as defined in clinic set up. |
| SDWL MENU | If user does not have this key assigned, they cannot enter a new wait list request (right click option “Transfer to EWL” from an open APPT request) or perform a disposition (right click option for EWL Disposition) of a EWL request. |

## Archiving and Purging

There is no archiving and purging in this module.

# Resource Management Reporting Implementation

Resource Management Reporting displays pertinent resource management metrics in a single view, the report is used by individual facilities and staff to measure and track supply, demand, and efficiency metrics related to clinic appointments and patient encounters in VistA.

## System Requirements

* System Requirements to run the executable JAR
* JRE (Java Run Time Environment) 1.6 or older.
* Jaspersoft – iReport viewer 5.6 or newer.
* System Requirements for application development
* JDK (Java Development Kit) 1.6
* Eclipse IDE
* Jaspersoft – iReport viewer 5.6 or newer.
* XML Editor (Textpad, EditPlus, Notepad)

## Application Files

**SDEC\_VSE\_ReportConsole.jar:** this is the executable JAR package file which launches the GUI for report console. The GUI will display all the available filter options to be selected before generating the report. The filters are explained below.

**SDEC\_VSE.xml**: This file acts as the source file from where the data will be collected, parsed and populated into the UI filters. This file is used again to generate the Jasper Reports.

**JRXML files:** are JasperReport template files, they are actually standard XML formatted files but have the .jrxml extension. All the JRXML files contain tag <jasperReport>, as root element, and also contain many sub-elements that constitute all of the report definition and properties. Resource Management Reporting uses an XML formatted file as the data source for rendering the report and charts. During the report rendering process the .jrxml file is compiled into a corresponding .jasper file.

**SDEC\_VSE\_ReportTemplate.jrxml:** This is the template file for the main report, it contains references to the compiled charts reports (.jasper) files. During the rendering process the jrxml file is compiled into a SDEC\_VSE\_ReportTemplate.jasper file.

**TotalDemandCharts.jrxml:** This is the template file for the Total Demand line chart subreport. The jrxml file is compiled into its own corresponding TotalDemandCharts.jasper file.

**TotalDemandSupplyCharts.jrxml:** This is the template file for the Total Demand and Total Supply combined column chart subreport. The jrxml file is compiled into its own corresponding TotalDemandSupplyCharts.jasper file.

**TotalSupplyCharts.jrxml:** This is the template file for the Total Supply line chart subreport. The jrxml file is compiled into its own corresponding TotalSupplyCharts.jasper file.

**VSE\_LOG.log:** This file is used to log any errors or warnings when the Resource Management Reporting java application is executed. This component was implemented using the Log4J java logging framework. Log4J is an open source API which lets the developer log any kind of statements as part of the application execution. This helps in tracking errors and other warnings to aid in troubleshooting any issues with the application.

The Resource Management application is utilizing the Log4J API to log any errors or warnings into a new file named “VSE\_LOG.log”. This file is created at runtime and stored with all of the application files in the designated application folder. The file contains the complete stack trace of the error/warning. The Java application creates one log file per each time it is executed by the user and this existing log file is always overwritten each time the application is launched or re-launched. However, if we have multiple errors/warnings as part of the same instance of the program execution, then the errors/warnings are appended to the existing log file.

## Resource Management Reporting Java Application Build Process

The application is mainly built using the Ant tool and a build script (build.xml). The build file defines the class path and the prerequisites which include all the external JAR dependencies. All of the dependencies will be copied into a single structure and the source files will be compiled. The compiled class files will be packaged along with the dependent JAR libraries, configurations, and XML files. We are using Ant version 1.9.4.

## Data Flow Diagram

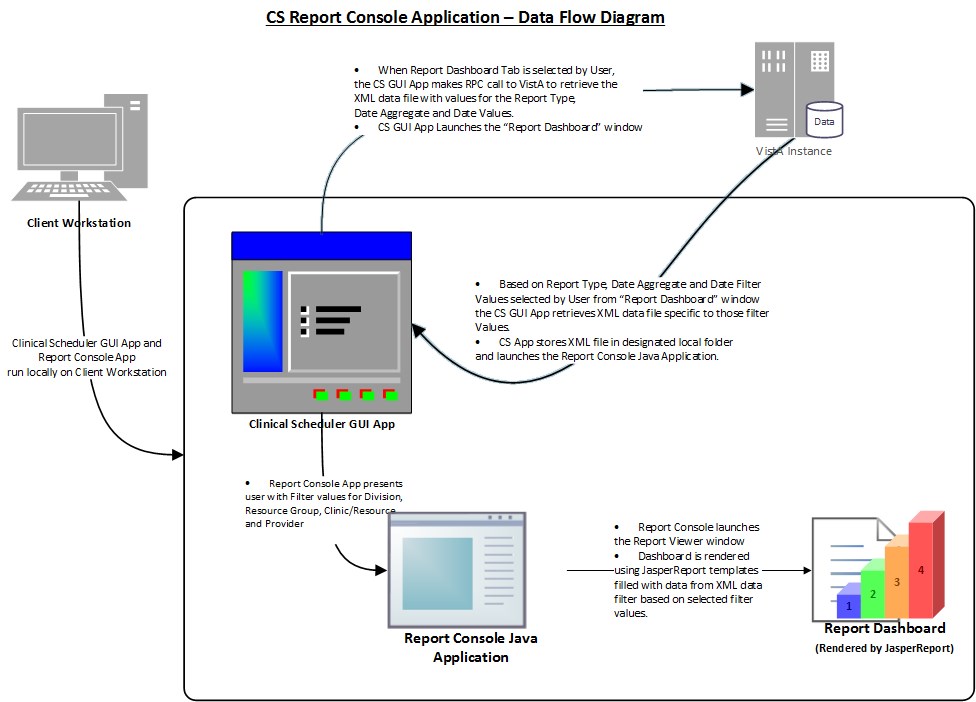


Figure 84: VistA Scheduling GUI Data Flow Diagram

## Implementation of Report Filters

Some of the filters (Report Type, Date and Date Aggregate) will be selected on the VistA Scheduling GUI by the user before the Report Console is started. These three filter values will be part of the SDEC\_VSE.xml document. When the UI for report console is initiated, the Java application will immediately look for these filters on the XML document and populate them as the default values on the Report Console. All other remaining filters would have generic values. Once the UI is up and running with these default filters, the User can select any value from the renaming filter lists to kick start the hierarchy.

Below are fragments of the XML source file which has the default filter values for Report Type, Date and Date Aggregate.

<?xml version="1.0" encoding="UTF-8"?>

<VAFacility Name="CHEYENNE VAMC" ID="442">

<ReportType ReportType="Primary Care">

<DateAggregate DateAgg="Year">

<Date ActivityDate="01/23/2015" BeginDate="01/23/2014" EndDate="01/23/2015" DateName="01/23/2014-01/23/2015">

….SOURCE XML BODY….

</Date>

</DateAggregate>

</ReportType>

</VAFacility>

## Resource Management Reporting Application Functions

### Report Filter Functionality

All the filters populated in the report UI will be part of the data parsed from the input SDEC\_VSE.xml. All the filters are based on a hierarchical structure, where each filter is dependent on the parent filter selection. The hierarchy and descriptions are described below:

* **VA Facility**: Name of the facility for which reports are generated.
* **Report Type**: defaults to the type from the VistA Scheduling GUI for which the report is being generated.
* **Date Aggregate**: duration of the appointment cycle also defaulted to the value from the VistA Scheduling GUI. The available values are Year, Quarter, Month, Week, and Day.
* **Date**: represents the activity date based on the date aggregate selected from the VistA Scheduling GUI. This filter normally has the range of dates available by week, month, quarter, and year.
* **Division**: This filter is populated with all the available Division names available for the selected date activity. Based on the data from the source XML, this filter will let the user select ‘ALL’ divisions.
* **Resource Group**: This filter contains a list of available resource groups for the selected division. When ‘All’ Divisions are selected, this filter will contain all the available Resource Groups for all the Divisions. Based on the data from the source XML, this filter will let the user select ‘ALL’ resource groups and is defaulted to ‘All’ groups.
* **Clinic/Resource Name**: This filter represents the Clinics & Resource names under the selected Resource Group. The filter will not have any default values and the user has to make a selection to continue with the report generation process. Based on the data from the source XML, this filter will let the user select ‘ALL’ clinics/resources.
* **Provider**: This filter represents the Provider names under the selected Resource Group & Clinics. The filter will not have any default values and the user has to make a selection to continue with the report generation process. Based on the data from the source XML, this filter will let the user select ‘ALL’ providers for the selected Clinics

Once all filters are selected, the ‘Generate Report’ button is activated and the User can generate the required report with charts (if chart data is available). See diagram below:

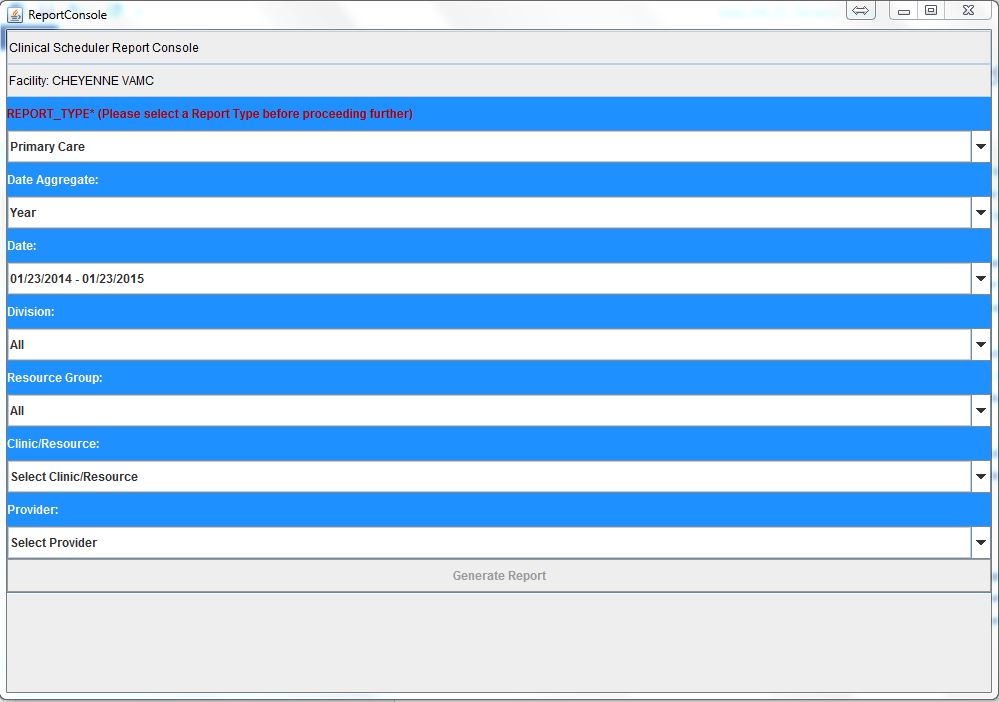


Figure 85: VistA Scheduling GUI Report Console

The report is opened in another applet window, the Report Viewer window, which lets the user print, save or export the report to 3 different formats, these are pdf, csv and rich text format (rtf) formats. The image below highlights the Save/Export and Print Features as implemented with the Report Viewer applet.

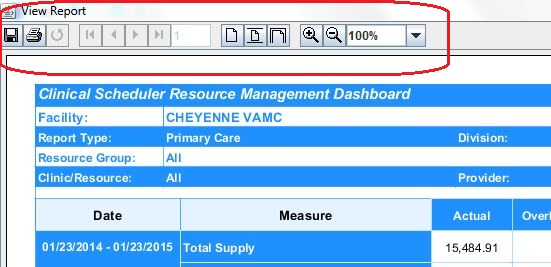


Figure 86: Save/Export/Print Features

Once the report is generated and saved, the user can switch back to the report console window at any point and generate different reports with a different set of filters. See image below for an example of the Report Viewer UI.

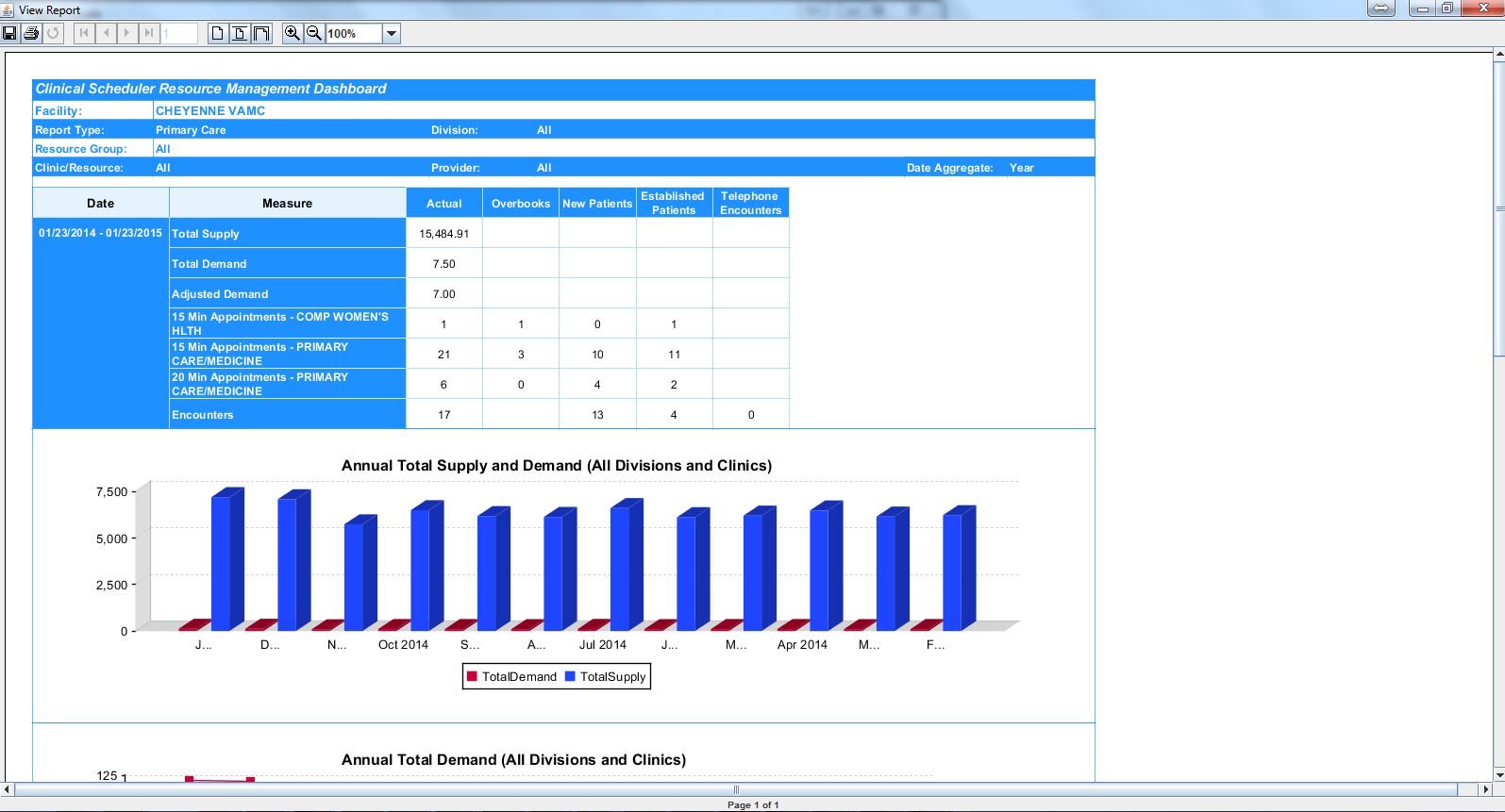


Figure 87: Report Viewer

### Java Classes Implementation

The Resource Management Reporting application is a Swing based Java application developed using Java and JasperReport components. The Java classes utilized in the project are outlined below:

Table 7: Java Class Name and Purpose

| Class Name | Purpose |
| --- | --- |
| ReportConsole.class | Main implementation class which invokes the GUI. |
| XMLParser.class | This is a DOM/XPath based parser used to parse the GUI filter data from the incoming source XML from Resource Management Reporting. |
| ReportsAndCharts.class | This is the reports implementation class which takes the data from the GUI and looks through the source XML for data needed to generate the reports and charts |
| XPathQueryReplacer.class | This is a utility class used to append the incoming facility name into the Jasper Reports query. |
| Utilities.class | This is another utility method which stores all required utility methods. |
| VSEConstants.class | This is a method which holds all constant values for the application’s use. |
| Build.xml | This is the ant script which is used for building the jar file with all dependencies. |

### Report Viewer Functionality

The Reports tab of the VISTA SCHEDULING GUI application launches the Resource Management Reporting Filter window where a user can input filters values such as the Report Type, Date Aggregate and Date values for the reports that they wish to display. This UI is shown below:

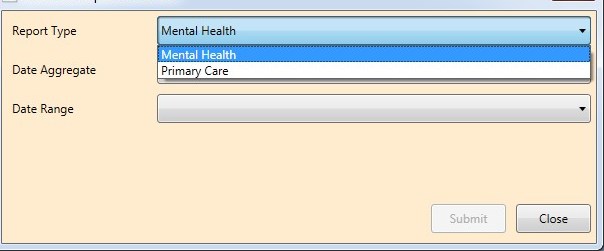


Figure 88: Report Viewer Functionality

After a User selects the first 3 filters for a Report and they click the submit button, they are presented with the Report Console Application with another set of filter inputs that they can use to further filter report data.

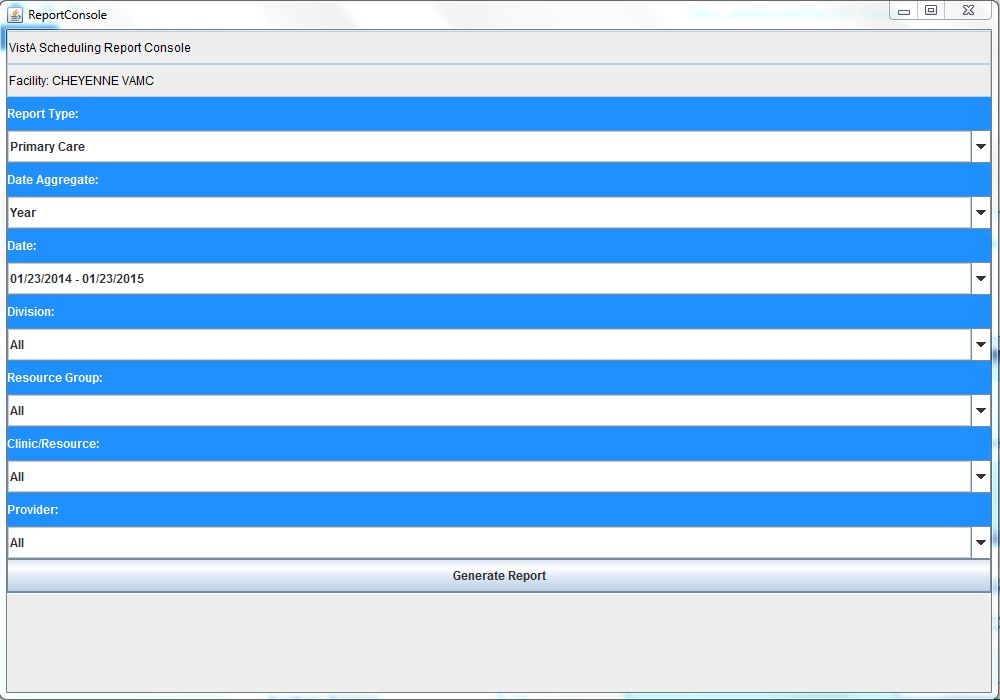


Figure 89: Report Console

### Report Viewer Error Handling and Resolution

The VISTA SCHEDULING GUI application is designed to launch the Resource Management Reporting Application within a Java Virtual Machines (JVM) window by executing the compiled java application (.jar) file with an argument to provide the location of the folder where the data source file (SDEC\_VSE.xml) and the JasperReport template files (.jrxml). An example of the command to launch the Resource Management Reporting Application with the required argument is shown below:

***Java -jar SDEC\_VSE\_ReportConsole.jar C:\Users\VA\_User\Documents***

If the path to the folder provided in the argument does not exist or is otherwise invalid, then the user will see an error window.

Some of the causes and potential resolution of this errors are listed in the sections below.

#### Invalid Folder or Folder Does not exist:

When the folder location provided to the Resource Management Reporting java application does not exist, an error window is displayed. The VS GUI application uses the HOMEPATH windows environment variable to determine the documents folder for storing the application working files, i.e. the .xml and .jrxml files, check the value set for the HOMEPATH variable to confirm that the folder is valid and that the user is has access to this folder.

#### Missing Source Data File (SDEC\_VSE.xml):

The VISTA SCHEDULING GUI application uses RPCs to download the source data XML file from the central server to the folder path set by the HOMEPATH environment variable on the user’s local workstation. When a user launches the Resource Management Reporting Application, this XML files is accessed and parsed by the application in order to render the report content, if there was an issue with downloading the XML file either because of access permission issue or an invalid folder path and the file is missing from the designated HOMEPATH folder, then the user would see an empty Report Viewer window as shown below:

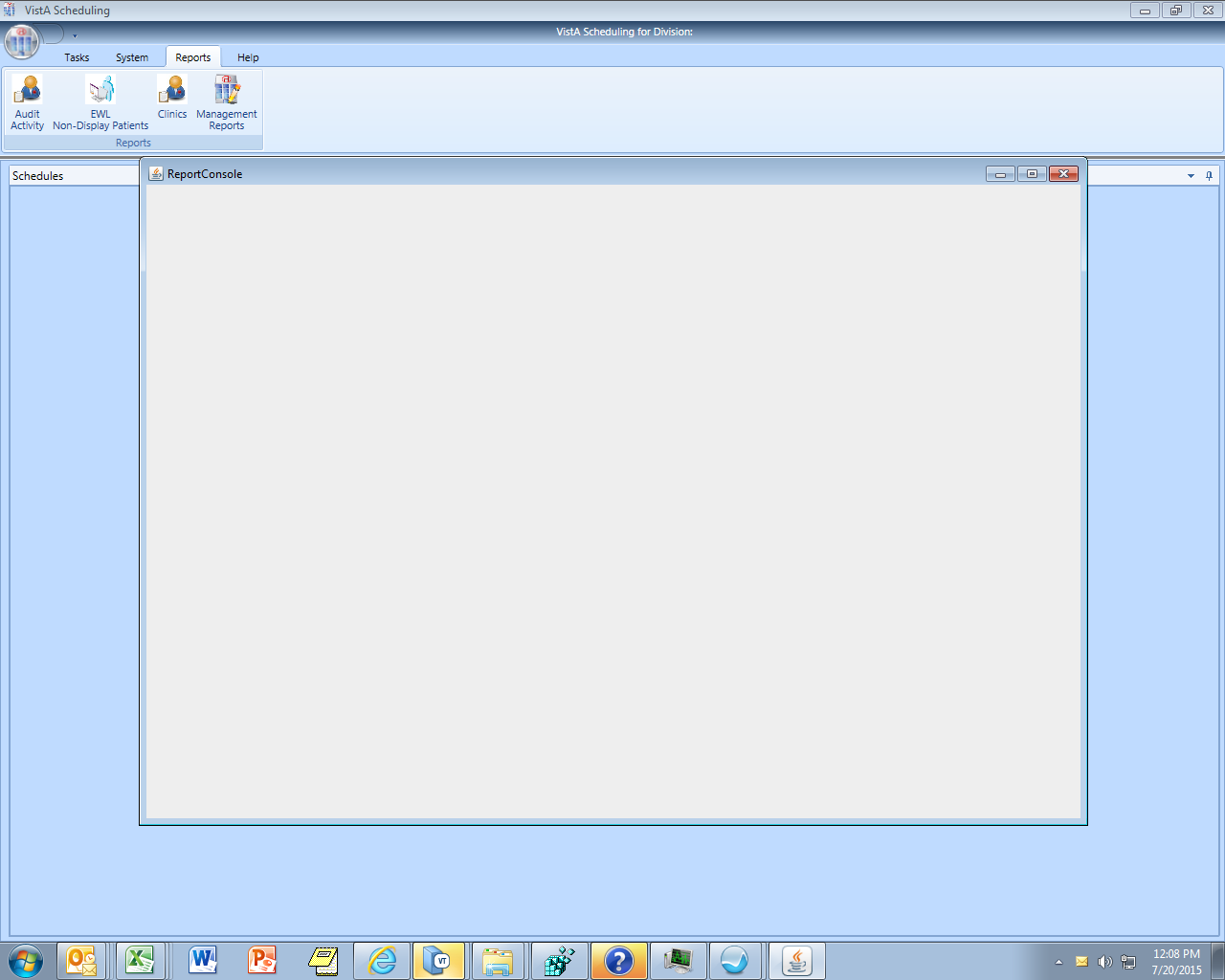


Figure 90: Report Viewer

In order to resolve this issue, check that the user has the right permissions to the HOMEPATH folder and also check that the folder name that is set in this environment variable does not have invalid characters such as "$" or "&" that may can cause the process of parsing the folder name to fail. If there are invalid characters in the folder name then this issue may have to be escalated to the development team to review and resolve.

#### Java Application Exception Error Logs:

The Resource Management Reporting Application is designed to log java exception errors to a log file “SDEC\_VSE\_LOG.log”. This log file is usually created in the same working folder as the .XML data source file and .JRXML JasperReport template files. The java exceptions logged into the file provide critical information for diagnosing issues with launching and using the Resource Management Reporting Application. An excerpt from the contents of the log file below shows a scenario where the data source XML file was not found by the application at runtime, in this case the issue will have to be resolved by ensuring that the XML file was properly created and loaded into the correct folder as expected by the application.

*Jun 01, 2015 7:53:59 PM gov.va.med.scheduling.utilities.Utilities logger*

*INFO: Exception in XML Parser*

*Jun 01, 2015 7:53:59 PM gov.va.med.scheduling.utilities.Utilities logger*

*INFO: java.io.FileNotFoundException: C:\temp\Test My Documents\SDEC\_VSE.XML (The system cannot find the file specified)*

*at java.io.FileInputStream.open(Native Method)*

*at java.io.FileInputStream.<init>(Unknown Source)*

*at java.io.FileReader.<init>(Unknown Source)*

*at gov.va.med.scheduling.infra.XMLParser.replaceAposInXML(XMLParser.java:304)*

*at gov.va.med.scheduling.infra.XMLParser.loadFilters(XMLParser.java:49)*

*at gov.va.med.scheduling.ui.ReportConsole.getFacilityName(ReportConsole.java:433)*

*at gov.va.med.scheduling.ui.ReportConsole.<init>(ReportConsole.java:53)*

*at gov.va.med.scheduling.ui.ReportConsole.main(ReportConsole.java:420)*

# Generating Online Documentation

This section describes a few methods to generate VistA Scheduling GUI system technical documentation. VistA Scheduling GUI software technical documentation, in addition to that which is located in the Help prompts throughout the VistA Scheduling GUI module, can be generated through the use of several Kernel options. These include, but are not limited to, the following:

* %INDEX
* VA FileMan
* Data Dictionary Utilities
* List File Attributes

For further information about other utilities that supply online technical information, consult the VistA Kernel Reference manual.

## %INDEX

This option analyzes the structure of a routine to determine in part if the routine adheres to VistA programming standards. The %INDEX output can include the following components:

* Compiled list of errors and warnings
* Routine listing
* Local variables
* Global variables
* Naked globals
* Label references
* External references

Running %INDEX for a specified set of routines allows users to discover any deviations from VistA programming standards that exist in the selected routines and to see how routines interact with one another. For example, which routines call, or are called by, other routines.

## List File Attributes

VA FileMan option allows users to generate documentation pertaining to files and file structure. Using the standard format of this option yields the following data dictionary information for a specified file:

* File name and description
* Identifiers
* Cross-references
* Files pointed to by the file specified
* Files that point to the file specified
* Input, print, and sort templates

In addition, the following applicable data is supplied for each field in the file:

* Field name, number, title, and description
* Global location
* Help prompt
* Cross-references
* Input transform
* Date last edited
* Notes

Using the Global Map format of this option generates an output that lists the following information:

* All cross-references for the file selected
* Global location of each field in the file
* Input, print, and sort templates

## Standards and Conventions Requirements and Exemptions

There are no exemptions to the Standards and Conventions (SAC) standards for this version.

## Callable Routines

Not Applicable.

# Glossary

Table 8: Terms and Definitions

| **Term** | **Definition** |
| --- | --- |
| Archiving | The storing of historical or little used data off-line (often on tape) |
| Banner | A line of test with a user’s name and domain |
| Browser | An interactive application that displays ASCII text on a terminal that supports a scroll region. The text can be in the form of a word-processing field or sequential local or global array. The user is allowed to navigate freely within the document |
| Callable Entry Points | Places in a routine that can be called from an application program |
| Cross-reference | An indexing method whereby files can include pre-sorted lists of entries as part of the stored database. Cross-references (x-refs) facilitate look-up and reporting. |
| Default Facility | A user selects a facility identification to work with patients registered to that facility |
| DHCP | Dynamic Host Configuration Protocol. A standardized network protocol used on Internet Protocol (IP) networks for dynamically distributing network configuration parameters, such as IP addresses for interfaces and services. |
| Entry Point | Entry point within a routine that is referenced by a “DO” or “GOTO” command from a routine internal to a package |
| File | A set of related records or entries treated as a single unit |
| FileMan | The database management system for VistA |
| Global | In Massachusetts General Hospital Utility Multi-Programming System (MUMPS), global refers to a variable stored on disk (global variable) or the array to which the global variable may belong (global array) |
| HHS | Health and Human Services |
| INDEX (%INDEX) | A Kernel utility used to verify routines and other MUMPS code associated with a package. Checking is done according to current ANSI MUMPS standards and VistA programming standards. This tool can be invoked through an option or from direct mode (>D^%INDEX). |
| Init | Initialization of an application package. The initialization step in the installation process builds files from a set of routines (the init routines). Init is a shortened form of initialization. |
| Internal Entry Number (IEN) | The number used to identify an entry within a file. Every record has a unique Internal Entry Number. |
| Information Resource Management (IRM) | VA personnel responsible for Information Systems Management and security |
| Kernel | The set of MUMPS software utilities that function as an intermediary between the host operating system and application packages, such as Laboratory and Pharmacy. The Kernel provides a standard and consistent user and programmer interface between application packages and the underlying MUMPS implementation. These utilities provide the foundation for VistA. |
| KIDS | Kernel Installation & Distribution System |
| Menu | A list of choices for computing activity. A menu is a type of option designed to identify a series of items (other options) for presentation to the user for selection. When displayed, menu-type options are preceded by the word “Select” and followed by the word “option” as in Select Menu Management option: (the menu’s select prompt). |
| Namespace | A unique set of two to four alpha characters that are assigned by the database administrator to a software application |
| Option | An entry in the Option file. As an item on a menu, an option provides an opportunity for users to select it, thereby invoking the associated computing activity. Options may also be scheduled to run in the background, non-interactively, by TaskMan. |
| Queuing | Requesting that a job be processed at a later time rather than within the current session |
| PIMS | Patient Information Management System |
| Remote Procedure Call (RPC) | An RPC is an entry in the REMOTE PROCEDURE file that points to specific M code to execute when called by an external Windows application |
| Routine | A program or sequence of instructions called by a program that may have some general or frequent use. MUMPS routines are groups of program lines that are saved, loaded, and called as a single unit via a specific name. |
| RPC | Remote Procedure Calls |
| User Class Identification (UCI) | A computing area |
| SAC | Standards and Conventions |
| Up-Hat (^) | A circumflex, also known as a “hat,” or “caret,” that is used as a piece delimiter in a global. The up-hat is denoted as “^” and is types by pressing **Shift + 6** on the keyboard. |
| Utility | A callable routine line tag or function. A universal routine usable by anyone |
| VA | Department of Veterans Affairs |
| Variable | A character or group of characters that refers to a value. MUMPS recognizes 3 types of variables: local variables, global variables and special variables. Local variables exist in a partition of the main memory and disappear at sign-off. A global variable is stored on disk, potentially available to any user. Global variables usually exist as parts of global arrays. |
| VistA | Veterans Health Information System and Technology Architecture |
| VSE | VistA Scheduling Enhancements |
| WYSIWYG | What You See Is What You Get |