MCCF v1.0

Technical Manual



October 2017

Department of Veterans Affairs

Office of Information and Technology (OI&T)

Revision History

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

| Date | Revision | Description | Author |
| --- | --- | --- | --- |
| 10/19/2017 | 1.0 | Initial document | Patrick Whalen |
|  |  |  |  |

A Technical Manual is a required end-user document for all OI&T software releases. The intended audience for this document is local IT support, management, and development personnel for nationally released software. It provides sufficient technical information about the software for developers and technical personnel to operate and maintain the software with only minimal assistance from Product Support staff.

**Table of Contents**

[1. Introduction 1](#_Toc496199853)

[1.1. Purpose 1](#_Toc496199854)

[1.2. System Overview 1](#_Toc496199855)

[1.3. Document Orientation 2](#_Toc496199856)

[2. Implementation and Maintenance 3](#_Toc496199857)

[2.1. System Requirements 3](#_Toc496199858)

[2.2. System Setup and Configuration 3](#_Toc496199859)

[3. Files 4](#_Toc496199860)

[4. Routines 5](#_Toc496199861)

[5. Exported Options 5](#_Toc496199862)

[6. Mail Groups, Alerts, and Bulletins 5](#_Toc496199863)

[7. Public Interfaces 5](#_Toc496199864)

[7.1. Integration Control Registrations 5](#_Toc496199865)

[7.2. Application Programming Interfaces 6](#_Toc496199866)

[7.3. Remote Procedure Calls 6](#_Toc496199867)

[7.4. HL7 Messaging 6](#_Toc496199868)

[7.5. Web Services 6](#_Toc496199869)

[8. Standards and Conventions Exemptions 6](#_Toc496199870)

[8.1. Internal Relationships 6](#_Toc496199871)

[8.2. Software-wide Variables 6](#_Toc496199872)

[9. Security 6](#_Toc496199873)

[9.1. Security Menus and Options 6](#_Toc496199874)

[9.2. Security Keys and Roles 6](#_Toc496199875)

[9.3. File Security 6](#_Toc496199876)

[9.4. Electronic Signatures 6](#_Toc496199877)

[9.5. Secure Data Transmission 6](#_Toc496199878)

[10. Archiving 6](#_Toc496199879)

[11. Non-Standard Cross-References 6](#_Toc496199880)

[12. Troubleshooting 6](#_Toc496199881)

[12.1. Special Instructions for Error Correction 6](#_Toc496199882)

[12.2. National Service Desk and Organizational Contacts 6](#_Toc496199883)

[13. Acronyms and Abbreviations 6](#_Toc496199884)

# Introduction

The Medical Care Collection Fund System (MCCF) automates and modernizes the five VA work streams: eBilling, ePharmacy, eInsurance, ePayments, and eAdmin.

## Purpose

This Technical Manual is designed to provide the developers and technical personnel with information necessary to install, maintain, and troubleshoot MCCF.

## System Overview

**WORK STREAM MODULES**

eBilling

* Claims Tracking, Encounter Form Utilities, and Insurance Data Capture.

ePharmacy

* Provides a method for managing the medications given to Veterans who have visited a clinic or who have received prescriptions upon discharge from the hospital
* Automatically generates prescription labels, and prints refill request forms
* Medication histories are kept online to permit checks for potential interactions
* Profiles can be generated to assist the clinician in managing the patient's medication regimen
* Management reports aid the pharmacy in controlling inventory and costs

eInusrance

* Determination of eligibility for claimed insurance (Verification)

ePayments

* Billing, Collections, Patient Account Profiling, EDI Lockbox

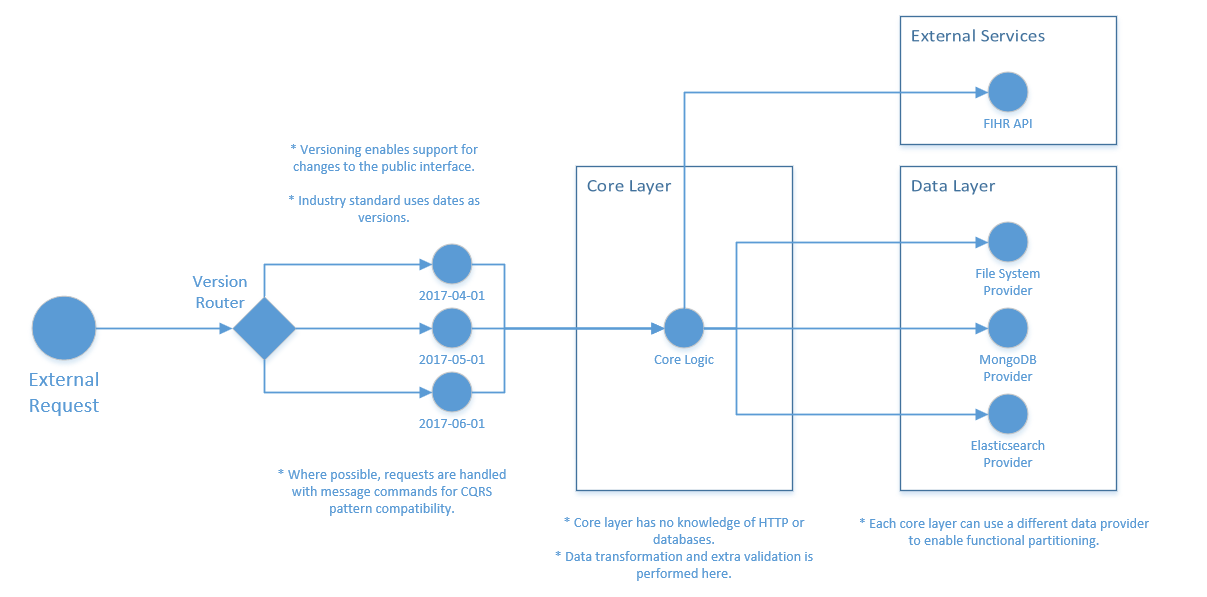
eAdmin

* Content Management

**ARCHITECTURE**

MCCF is a web-based system built on the Angular 2(Version 4) framework and Node,js. Angular is a TypeScript-based open-source front-end web application platform for building complex web applications. Node.js is a JavaScript run-time environment for executing JavaScript code server-side.

TAS backend services are driven by Node v6.

The following diagram lays out the general architecture of a service.

## Document Orientation

The MCCF Technical manual is intended for technical personnel, sofware developers and computer system managers. Technical personnel should be familiar with software delopment procedures, software development tools, and Unix operating system. Software developers should be familiar with Angular2, Node.js, JavaScript (ES6), TypeScript, CSS, and HTML.

The MCCF Technical Manual is divided into major sections for general clarity and simplification of the material being presented.The Implementation and Maintenance Section provides information on any aspect of the package that is site configurable.

### Disclaimers

#### Software Disclaimer

This software was developed at the Department of Veterans Affairs (VA) by employees of the Federal Government in the course of their official duties. Pursuant to title 17 Section 105 of the United States Code this software is not subject to copyright protection and is in the public domain. VA assumes no responsibility whatsoever for its use by other parties, and makes no guarantees, expressed or implied, about its quality, reliability, or any other characteristic. We would appreciate acknowledgement if the software is used. This software can be redistributed and/or modified freely provided that any derivative works bear some notice that they are derived from it, and any modified versions bear some notice that they have been modified.

#### Documentation Disclaimer

The appearance of external hyperlink references in this manual does not constitute endorsement by the Department of Veterans Affairs (VA) of this Web site or the information, products, or services contained therein. The VA does not exercise any editorial control over the information you may find at these locations. Such links are provided and are consistent with the stated purpose of the VA.

### References

# Implementation and Maintenance

For development MCCF requires the installation of Node.js and Angular2 client.

## System Requirements

CentOS Linux v7

### Hardware Requirements

### Software Requirements

Angular (version 4)

Node.js v6

MCCF requires a browser to run the application: Chrome v60, Internet Explorer v11

### Database Requirements

Elasticsearch 5.x

## System Setup and Configuration

The client-side code for MCCF is written in Angular TypeScript. A native Angular Client program, which is installed on a software developer’s computer, installs the code with the command npm install. The code is then compiled into JavaScript and HTML bundles with the command npm build. These bundles are “served” by a web server to the browser user interface. A software developer can run the client-side application locally with the command npm start and entering localhost:4200 into browser URL field.

The server-side code for MCCF is written in Node v6 ES6. The application is installed with a standard npm install command. Tests are performed with the mocha command. A software developer can run the application locally with the grunt command and accessing port 8193.

# Files

MCCF base code directory structure:

/mccf\_tas\_core  
/src  
 /app  
 /assets  
 /environments  
 index.html  
 polyfills.ts  
 test.ts  
 tsconfig.spec.json  
 favicon.ico  
 main.ts  
 styles.css  
 tsconfig.app.json  
 typings.d.ts  
/scripts  
.angular-cli.json  
.bowerrc  
.jazzignore  
inventory  
protractor.conf.js  
tsconfig.json  
.jshintrc  
Jenkinsfile  
README.md  
tslint.json  
.travis.yml  
/karma  
requirements.yml  
Vagrantfile  
ansible.cfg  
karma.conf.js  
runFortifyScan.sh\*  
editorconfig  
bower.json  
LICENSE  
/.git  
devmachine.yml  
package.json  
site.yml  
.gitignore  
/e2e  
package-lock.json

DEVELOPMENT AND PRODUCTION ENVIRONMENT DIFFERENCES

When Angular code is compiled on a developer’s computer, supporting module libraries are downloaded from the NPM (Node Package Manager) public repository. In production, MCCF does not allow this because the system is behind a VA firewall. When MCCF is in production, Angular uses the Sinopia service for installing modules.

# Routines

# Exported Options

# Mail Groups, Alerts, and Bulletins

# Public Interfaces

## Integration Control Registrations

| **Category** | **Definition** |
| --- | --- |
| Controlled Subscription | Describes attributes/functions that must be controlled in their use. The decision to restrict the ICR is based on the maturity of the custodian package. Typically, these ICRs are created by the requesting package based on their independent examination of the custodian package's features. For the ICR to be approved the custodian grants permission to other VistA packages to use the attributes/functions of the ICR; permission is granted on a one-by-one basis where each is based on a solicitation by the requesting package. An example is the extension of permission to allow a package (e.g., Spinal Cord Dysfunction) to define and update a component that is supported within the Health Summary package file structures. |
| Supported Reference | This applies where any VistA application may use the attributes/functions defined by the ICR (these are also called “Public”). An example is an ICR that describes a standard API such as DIE or VADPT. The package that creates/maintains the Supported Reference must ensure it is recorded as a Supported Reference in the ICR database. There is no need for other VistA packages to request an ICR to use these references; they are open to all by default.  **NOTE:** ICRs categorized as Supported References are listed on the DBA menu on FORUM and are open for use by everyone. |

## Application Programming Interfaces

## Remote Procedure Calls

## HL7 Messaging

## Web Services

# Standards and Conventions Exemptions

## Internal Relationships

## Software-wide Variables

# Security

## Security Menus and Options

## Security Keys and Roles

## File Security

## Electronic Signatures

## Secure Data Transmission

# Archiving

# Non-Standard Cross-References

# Troubleshooting

## Special Instructions for Error Correction

## National Service Desk and Organizational Contacts

# Acronyms and Abbreviations

MCCF – Medical care Collection fund

EDI – Electronic Data Interchange

TAS – Transaction Application Suite

NPM – Node Pacakge Manager

Template Revision History

| **D****ate** | **Version** | **Description** | **Author** |
| --- | --- | --- | --- |
| July 2016 | 1.1 | Updated instructional text to simplify content. | OI&T Documentation Standards Committee |
| June 2016 | 1.0 | Initial version | OI&T Documentation Standards Committee |