Enterprise Terminology Services (ETS)

Package Version 1.0

Technical Manual / Security Guide

Version 1.0

March 2017

Department of Veterans Affairs

Office of Information and Technology (OI&T)

Revision History

Date	Revision	Description	Author
Feb 2017	1.0	Initial Version	PST Team

Artifact Rationale

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

Table of Tables

Table of Figures

1. Introduction

This Enterprise Terminology Services (ETS) package exists to provide enterprise-wide access to common code sets that are needed in a health care system. It will contain FileMan files with the data and Application Program Interfaces (APIs) that can be used to provide access and process the data. The Integration Control Registrations will be supported and thus open to all VistA applications.

1.1. Purpose

This document is designed to include the FileMan data structures and APIs that are a part of this package. It will include information regarding routines and processes that are required to keep the data up-to-date with the most recent national standard data.

1.2. System Overview

The initial release of ETS will consists of the Logical Observation Identifiers Names and Codes (LOINC) files and APIs to access the data in those files. LOINC is a trademark of the Regenstrief Institute. The LOINC database provides sets of universal names and ID codes for identifying laboratory and clinical test results.

All additions, changes, and deprecation to entries in this file shall be done by the Standards & Terminology Services (STS) team. Creating and/or editing locally defined fields in the file are not permitted.

1.3. Document Orientation

This manual's intended audience is Information Resource Management (IRM) personnel, Applications Coordinators (ADPACs), Clinical Coordinators, and developers.

1.3.1. Disclaimers

This manual provides an overall explanation of this package functionality. This guide does not attempt to explain how the overall VistA programming system is integrated and maintained.

1.3.1.1. 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.

1.3.1.2. 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.

1.3.2. References

LOINC reference documents:

- Regenstrief Institute – http://www.regenstrief.org/resources/loinc/
- National Library of Medicine's Unified Medical Language System – https://www.nlm.nih.gov/research/umls/loinc_main.html

2. Implementation and Maintenance

2.1. System Requirements

2.1.1. Hardware Requirements

The ETS package should be installed on systems that are able to support running VistA applications.

2.1.2. Software Requirements

The following software packages must be installed prior to the ETS version 1.0 installation.

- VA FileMan version 22.2
- Kernel version 8.0
- MailMan version 8.0

2.1.3. Database Requirements

VA Fileman version 22.2 and Kernel version 8.0 are required to install and run the ETS application.

2.2. System Setup and Configuration

The initial release of ETS will be in patch ETS*1.0*1, which will be a standard Kernel Installation and Distribution System (KIDS) build. The installation is described in the patch description. After installation, the Enterprise Terminology Services package will be created on the system along with the LOINC databases and APIs.

3. Files

Table 1: Files List

File #	File Name	Global Location	Description
129.1	LOINC	^ETSLNC(129.1,	This LOINC file contains an extraction of the LOINC database. The LOINC database provides sets of universal names and ID codes for identifying laboratory and clinical test results.
129.11	LOINC COMPONENT	^ETSLNC(129.11,	This file contains the name of the component or analyte measured for the LOINC file (#129.1).
129.12	LOINC AXIS CODES	^ETSLNC(129.12,	The LOINC Axis Codes file contains a collection of LOINC codes used by the Axis fields in the LOINC file.
129.13	LOINC METHOD CODES	^ETSLNC(129.13,	This file contains a listing of method codes, which is one of the axis fields in a LOINC code and are defined by Regenstrief.
129.14	LOINC EXCLUDED WORDS FILE	^ETSLNC(129.14,	This file contains LOINC terms which are meant to be excluded from indexing and lookup.

4. Routines

The following is a list of routines that will exist after the installation of ETS (ETS*1.0*0).

- ETS10P0
- ETSLNC
- ETSLNC1
- ETSLNC2
- ETSLNC3
- ETSLNCIX
- ETSLNCTX

5. Exported Options

There are no exported options associated with this package.

6. Mail Groups, Alerts, and Bulletins

There are no mail groups, alerts, or bulletins associated with this package.

7. Public Interfaces

There are no public interfaces associated with this package.

7.1. Integration Control Registrations

Table 2: ICR List

Category	Definition
Supported Reference	ICR #6731 has been created to make all APIs included in this package available for use by all VistA packages.

7.2. Application Programming Interfaces

7.2.1. $$CHKCODE^ETSLNC: Verify code in LOINC Dictionary

This API checks to see if the LOINC code is in the LOINC Dictionary.

Format

$$CHKCODE^ETSLNC(CODE)

Input parameter

CODE: (Required) LOINC Code

Output

$$CHKCODE: 0 – Does Not Exist
1 – Exists
-1^<message> – Error

Figure 1: $$CHKCODE^ETSLNC Examples

```
>W $$CHKCODE^ETSLNC(1)
1

>W $$CHKCODE^ETSLNC(6)
0

>W $$CHKCODE^ETSLNC("XXX")
-1^LOINC Code data missing
```

7.2.2. $$COMLST^ETSLNC: Retrieve component list

This API retrieves a list of LOINC codes for a passed-in component. It returns both the LOINC code and Fully Specified Name.

Format

$$COMLST^ETSLNC(COM,SUB)

Input parameters

COM: (Required) Component IEN

SUB: (Optional) Subscript used to store the data in. If not passed in, the default is "ETSCOMP".

Output

$$COMLST: 1 – Success
0 – Component not used
-1^<message> – Error

^TMP(SUB,$J,"COMP",CODE) = Fully Specified Name

Note: Array is not initialized. The calling application is responsible for initializing the array.

Figure 2: $$COMLST^ETSLNC Example

```
>W $$COMLST^ETSLNC(2,"SAB")
1

^TMP("SAB",127896,"COMP",1)="ACYCLOVIR:SUSC:PT:ISOLATE:ORDQN:"
```

7.2.3. $$CSDATA^ETSLNC: Get Detailed Information about a Code

This API retrieves information for a specific code in a database. Note that this API was modeled after a Lexicon API with a similar name, and the input and output value match the format of that API as much as possible.

Format

$$CSDATA^ETSLNC(CODE,CSYS,CDT,.RESULT)

Input parameters

CODE: (Required) LOINC Code used for lookup.

SYS: (Optional) Coding System to perform the lookup on. For LOINC, "LNC" is the only valid value to send. If not passed in, defaults to "LNC".

CDT: (Optional): Date in Fileman Format to use to evaluate Status. If not passed in, defaults to TODAY.

RESULT: (Required – Passed by Reference) Array to return the information requested.

Output

$$CSYS: One of three values:
1 – Successful (Code exists in the LOINC dictionary)
0 – Unsuccessful
-1^<message> – Error

RESULT(“LEX”,1): Two pieces of data – IEN^ Fully Specified Name

RESULT(“LEX”,1,"N"): "IEN ^ Fully Specified Name"

RESULT(“LEX”,2): Two pieces of data – Activation Status ^Activation Effective Date

RESULT("LEX",2,"N"): "Status ^ Effective Date"

RESULT(“LEX”,8): Either Null (Active) or 1 (Inactive)

RESULT("LEX",8,"N"): "Deactivated Concept"

RESULT(“SYS”,1): IEN

RESULT("SYS",1,"N"): "IEN"

RESULT(“SYS”,2): Long Common Name

RESULT("SYS",2,"N"): "Long Common Name"

* Note that the "N" values are the description of the output.

Figure 3: $$CSDATA^ETSLNC Example

```
>W $$CSDATA^ETSLNC(6110,"LNC","",.RESULT)
1

ZW RESULT
RESULT("LEX",1)="6110^PLANTAGO LANCEOLATA AB.IGE:ACNC:PT:SER:QN:"
RESULT("LEX",1,"N")="IEN ^ Fully Specified Name"
RESULT("LEX",2)="1^3160101"
RESULT("LEX",2,"N")="Status ^ Effective Date"
RESULT("LEX",8)=""
RESULT("LEX",8,"N")="Deactivated Concept"
RESULT("SYS",1)=6110
RESULT("SYS",1,"N")="IEN"
RESULT("SYS",2)="ENGLISH PLANTAIN IGE AB [UNITS/VOLUME] IN SERUM"
RESULT("SYS",2,"N")="Long Common Name"
```

7.2.4. $$CSYS^ETSLNC: Retrieve the Coding System Information

This API returns coding system information for LOINC. Note that this API was modeled after a Lexicon API with a similar name, and the input and output value match the format of that API as much as possible.

Format

$$CSYS^ETSLNC(SYS)

Input parameter

SYS: (Optional) Coding System to perform the Lookup on. For LOINC, the only valid value is "LNC". If not passed in, defaults to "LNC".

Output

$$CSYS: A delimited string of code system information

1. Not used
2. Three-character Source Abbreviation
3. Source Abbreviation
4. Nomenclature
5. Source Title
6. Source
7. Not used
8. Not used
9. Not used
10. Not used
11. Not used
12. Version ID
13. Implementation Date
14. Lookup Threshold

Figure 4: $$CSYS^ETSLNC Example

```
>W $$CSYS^ETSLNC("LNC")
^LNC^LNC^LOINC^Logical Observation Identifier Names and Codes^Duke University
Medical Center^^^^^^^JAN 01, 1994^20000
```

7.2.5. $$DEPLST^ETSLNC: Retrieve Deprecated List

This API retrieves the list of deprecated LOINC codes.

Format

$$DEPLST^ETSLNC(SUB)

Input parameter

SUB: (Optional) Subscript for the Temporary Deprecated List Array. If not passed in, defaults to "ETSDEP".

Output

$$DEPLST: 1 – Deprecated items found
0 – No Deprecated items found

^TMP(SUB,$J,"DEPRECATED",<CODE>) – Fully Specified Name

Figure 5: $$DEPLST^ETSLNC Example

```
>W $$DEPLST^ETSLNC("SAB")
1

^TMP("SAB",127896,"DEPRECATED",1)="ACYCLOVIR:SUSC:PT:ISOLATE:ORDQN:"
```

7.2.6. $$GETCODE^ETSLNC: Retrieve LOINC Code for given IEN

This API retrieves the LOINC code for the IEN passed in.

Format

$$GETCODE^ETSLNC(IEN)

Input parameter

IEN: (Required) LOINC Code IEN

Output
$$GETCODE: LOINC Code
1^<message> – Error

Figure 6: $$GETCODE^ETSLNC Examples

```
>W $$GETCODE^ETSLNC1(5)
5

>W $$GETCODE^ETSLNC1(0)
-1^LOINC Code not found
```

7.2.7. $$GETNAME^ETSLNC: Retrieve LOINC Name Array

This API retrieves the LOINC Name Array. Currently retrieves the Fully Specified Name and Long Common Name.

Format

$$GETNAME^ETSLNC(LOINC,TYPE,NAME)

Input parameter

LOINC: (Required) LOINC Code or IEN

TYPE: (Optional) Input Type - "C" for LOINC Code or "I" for LOINC IEN. If not passed in, defaults to "C"

NAME: (Required-Passed by Reference) – Array for output values

Output
$$GETNAME: 1 – success and results found
0 – no results found
-1^< message> – Error

NAME: An array with the Name values. The array will be cleared upon entry. The output array will have the following subscipts:
NAME ("FULLNAME")=Fully Specified Name
NAME("LONGNAME")=Long Common Name

Figure 7: $$GETNAME^ETSLNC Examples

```
>W $$GETNAME^ETSLNC(3,"C",.NAME)
1

>ZW NAME
NAME=""
NAME("FULLNAME")="ALMECILLIN:SUSC:PT:ISOLATE:ORDQN:MIC"
NAME("LONGNAME")="ALMECILLIN - BY TEST [SUSCEPTIBILITY] BY MINIMUM INHIBITORY
CONCENTRATION (MIC)"

>W $$GETNAME^ETSLNC(3,,.NAME)
1

>ZW NAME
NAME=""
NAME("FULLNAME")="ALMECILLIN:SUSC:PT:ISOLATE:ORDQN:MIC"
NAME("LONGNAME")="ALMECILLIN - BY TEST [SUSCEPTIBILITY] BY MINIMUM INHIBITORY
CONCENTRATION (MIC)"
```

7.2.8. $$GETREC^ETSLNC: Retrieve LOINC Information

This API retrieves all information about the LOINC CODE or IEN.

Format

$$GETREC^ETSLNC(LOINC,TYPE,SUB)

Input parameter

LOINC: (Required) LOINC Code or LOINC IEN

TYPE: (Optional) Input Type – "C: for Code (default) or "I" for IEN. If not passed in, default is "C".

SUB: (Optional) Subscript for ^TMP array storing the results. If not passed in, default is "ETSREC"

Output

$$GETREC: 1 – Record found
0 – No record found
-1^<message> – Error

Each node of the TMP global will be equal to the data in the field indicated by the node.
^TMP(SUB,$J,"RECORD","ACTIVATION HISTORY",#,"ACTIVATION EFFECTIVE DATE")
"ACTIVATION HISTORY",#,"ACTIVATION STATUS")
"ADJUSTMENT")
"CHALLENGE")
"CHANGE REASON")
"CHANGE TYPE")
"CHECK DIGIT")
"CLASS")
"CLASSTYPE")
"COMMENTS") or "COMMENTS",#)
"DATE LAST CHANGED")
"EXAMPLE UCUM UNITS")
"FULLY SPECIFIED NAME")
"IEN")
"LONG COMMON NAME")
"MASTER ENTRY FOR VUID")
"METHOD TYPE")
"NON-PATIENT SPECIMEN")
"PROPERTY")
"REPEAT OBSERVATION")
"SCALE TYPE")
"SHORTNAME")
"SNOMED CODE")
"SOURCE")
"SYSTEM")
"TIME ASPECT")
"TIME MODIFIER")
"UNITS")
"VA COMMON DISPLAY NAME")
"VERSION NUMBER")
"VUID")
"VUID EFFECTIVE DATE",#,"EFFECTIVE DATE/TIME")
"VUID EFFECTIVE DATE",#,"STATUS")

Figure 8: $$GETREC^ETSLNC Example

```
>W $$GETREC^ETSLNC(546,"C")
1
>D ^%G

Global ^TMP("ETSREC",$J
^TMP("ETSREC",139688,"RECORD","ACTIVATION HISTORY",1,"ACTIVATION EFFECTIVE
DATE")="3150629^JUN 29, 2015"
```

```
                        "ACTIVATION STATUS")="1^ACTIVE"
^TMP("ETSREC",139688,"RECORD","ADJUSTMENT")=""
                              "CHALLENGE")=""
                          "CHANGE REASON")=""
                            "CHANGE TYPE")="MIN"
                            "CHECK DIGIT")=2
                                  "CLASS")="MICRO"
                                "CLASSTYPE")="1^LABORATORY"
                                     "CODE")="546-2"
                                 "COMMENTS")=""
                       "DATE LAST CHANGED")="3150508^MAY 08, 2015"
                      "EXAMPLE UCUM UNITS")=""
                    "FULLY SPECIFIED NAME")="STREPTOCOCCUS.BETA-
HEMOLYTIC:ACNC:PT:THRT:ORD:ORGANISM SPECIFIC CULTURE"
                                      "IEN")=546
                         "LONG COMMON NAME")="STREPTOCOCCUS.BETA-HEMOLYTIC
[PRESENCE] IN THROAT BY ORGANISM SPECIFIC CULTURE"
                   "MASTER ENTRY FOR VUID")="YES"
                             "METHOD TYPE")="ORGANISM SPECIFIC CULTURE"
                    "NON-PATIENT SPECIMEN")=""
                                "PROPERTY")="ARBITRARY CONCENTRATION"
                      "REPEAT OBSERVATION")=""
                               "SCALE TYPE")="Ordinal"
                                "SHORTNAME")="B-HEM STREP THROAT QL CULT"
                            "SNOMED CODE")=""
                                 "SOURCE")="SH"
                                 "SYSTEM")="Throat"
                            "TIME ASPECT")="POINT"
                          "TIME MODIFIER")=""
                                  "UNITS")=""
                 "VA COMMON DISPLAY NAME")=""
                        "VERSION NUMBER")=""
                                  "VUID")=4683779
^TMP("ETSREC",139688,"RECORD","VUID EFFECTIVE DATE",1,"EFFECTIVE
                               DATE/TIME")="3050501^MAY 01, 2005"
                                 "STATUS")="1^ACTIVE"
```

7.2.9. $$GETSTAT^ETSLNC: Retrieve LOINC Status

This API retrieves the current status for a LOINC Code or IEN.

Format

$$GETSTAT^ETSLNC(LOINC,TYPE)

Input parameters

LOINC: (Required) LOINC Code or LOINC IEN

TYPE: (Optional) Input Type – "C" for Code or "I" for IEN. If not passed in, default is "C".

Output
$$GETSTAT: Current Status (Internal format^External Format)

Figure 9: $$GETSTAT^ETSLNC Examples

```
>W $$GETSTAT^ETSLNC(1,"C")
1^DEL

>W $$GETSTAT^ETSLNC(2,"I")
^Active

>W $$GETSTAT^ETSLNC(3)
3^DISCOURAGED
```

7.2.10. $$HIST^ETSLNC: Get Activation History for a LOINC Code

The API allows a user to extract the Activation History for a specified LOINC CODE. Note that this API was modeled after a Lexicon API with a similar name, and the input and output value match the format of that API as much as possible.

Format

$$HIST^ETSLNC(CODE,SYS,.RESULT)

Input Parameters

CODE: (Required) LOINC Code.

SYS: (Optional) Coding System for lookup. For LOINC, the valid values is "LNC". If not value passed, the value is defaulted to "LNC".

RESULT: (Required – Passed by Reference) Array returning the information requested.

Output

$$HIST: Either Number of Histories Found or "-1 ^ error message"

RESULT: Detailed History Information

RESULT(0) = Number of Activation Histories
RESULT(0,0) = LOINC Code^Source Abbreviation ("LNC")^Nomenclature ("LOINC")
RESULT(<Fileman Date>,<Status>) = <Comment>

Where <Date> is a date (in Fileman internal format) of the status change

<Status> is 1:Active or 0:Inactive

<Comment> is "Activated", "Inactivated", or "Re-activated", The comment is dependent on the current and previous statuses.

Figure 10: $$HIST^ETSLNC Example

```
>W $$HIST^ETSLNC(6110,"LNC",.RESULT)
3

>ZW RESULT
RESULT(0)=3
RESULT(0,0)="6110^LNC^LOINC"
RESULT(3140101,1)="Activated"
RESULT(3150101,0)="Inactivated"
RESULT(3160101,1)="Re-activated"
```

7.2.11. $$PERIOD^ETSLNC: Get Activation / Inactivation Periods for a Code

This API allows a user to view the activation periods of a given code. Note that this API was modeled after a Lexicon API with a similar name, and the input and output value match the format of that API as much as possible.

Format

$$PERIOD^ETSLNC(CODE,SYS,.RESULT)

Input Parameters

CODE: (Required) LOINC Code.

SYS: (Optional) Coding System to perform the Lookup. For LOINC, the only valid value is "LNC". If not passed, this defaults to "LNC".

RESULT: (Required – Passed by reference) Array to return the results.

Output

$$PERIOD: Five-piece string of delimited values or "-1^Error Message"

1. # of Activation Periods Found
2. Not used
3. Source Abbreviation ("LNC")
4. Nomenclature ("LOINC")
5. Source Title ("Logical Observation Identifier Names and Codes")

RESULT(0): Same output as $$PERIOD

RESULT(<Activation Date>): Four-piece string of delimited values

1. Inactivation Date (if inactivated)
2. Not Used

3. Variable Pointer to the LOINC File
4. LONG COMMON NAME

RESULT(<Activation Date>,0): FULLY SPECIFIED NAME

Figure 11: $$PERIOD^ETSLNC Example

```
>W $$PERIOD^ETSLNC(6110,"LNC",.RESULT)
1^^LNC^LOINC^Logical Observation Identifier Names and Codes

>ZW RESULT

RESULT(0)="1^^LNC^LOINC^Logical Observation Identifier Names and Codes"
RESULT(3061026)="^^6110;^ETS(129.1,^ENGLISH PLANTAIN IGE AB [UNITS/VOLUME] IN
SERUM"
RESULT(3061026,0)="PLANTAGO LANCEOLATA AB.IGE:ACNC:PT:SER:QN:"
```

7.2.12. $$TAX^ETSLNC: Get Taxonomy Information

This API allows a user to locate all of the valid codes in the LOINC database for a given string. Note that this API was modeled after a Lexicon API with a similar name, and the input and output value match the format of that API as much as possible.

Format

$$TAX^ETSLNC(X,SRC,DT,SUB,VER)

Input parameters

X: (Required) Partial Code or Text String to look up.

SRC: (Optional) Targeted Source(s). For LOINC, valid value will be "LNC". If not passed in, defaults to "LNC"

DT: (Optional) Date in FileMan format to use to evaluate Status. If not passed in, defaults to TODAY.

SUB: (Optional) Global subscript for ^TMP global. If not passed in, defaults to "LEXTAX".

VER: (Optional) Search criteria - 0 for both Active and Inactive codes, 1 for Active codes only. If not passed in, defaults to 0.

Output

$$TAX: The number of LOINC codes found or -1 ^ error message

^TMP(SUB,$J,1,(CODE_" "),#) returns a five-piece " ^" string:

1. Activation Date
2. Inactivation Date (if inactivated)

3. Not used
4. Variable Pointer to the LOINC File – IEN concatenated with ";ETSLNC(129.1, "
5. Long Common Name

^TMP(SUB,$J,1,(CODE_" "),#,0) returns a two piece "^" delimited string

1. LOINC Code
2. Fully Specified Name

where SUB is the subscript passed via the SUB input parameter and CODE is the LOINC code.

Figure 12: $$TAX^ETSLNC Example

```
>W $$TAX^ETSLNC(6110,"LNC",,"TST",1)
1

^TMP("TST",57728,0)=1
^TMP("TST",57728,"6110 ",1)="3160101^^^61103;ETSLNC(129.1,^ENGLISH PLANTAIN IGE AB
[UNITS/VOLUME] IN SERUM"
^TMP("TST",57728,"6110 ",1,0)="6110^PLANTAGO LANCEOLATA AB.IGE:ACNC:PT:SER:QN:"
```

7.2.13. $$VERSION^ETSLNC: Retrieve LOINC Version

This API retrieves the LOINC version.

Format

$$VERSION^ETSLNC()

Input parameter

None

Output
$$VERSION: LOINC Version Number
Null – No Version Data found
-1^<File Definition Error>

Figure 13: $$VERSION^ETSLNC Example

```
>W $$VERSION^ETSLNC()
2.52
```

7.3. Remote Procedure Calls

There are no remote procedure calls associated with this package.

7.4. HL7 Messaging

There are no HL7 messages associated with this package.

7.5. Web Services

There are no Web services associated with this package.

7.6. Contingency Planning

This section does not apply to ETS.

8. Standards and Conventions Exemptions

8.1. Internal Relationships

There are no internal relationships defined with this package.

8.2. Software-wide Variables

There are no software-wide variables associated with this package.

9. Security

9.1. Security Menus and Options

There are no security menus or options associated with this package.

9.2. Security Keys and Roles

No security keys are part of this package.

9.3. File Security

Table 3: File Security List

File #	File Name	Global Location	File Security
129.1	LOINC	^ETSLNC(129.1,	DD ACCESS: @ RD ACCESS: WR ACCESS: @ DEL ACCESS: @ LAYGO ACCESS: @ AUDIT ACCESS: @
129.11	LOINC COMPONENT	^ETSLNC(129.11,	DD ACCESS: @ RD ACCESS: WR ACCESS: @ DEL ACCESS: @ LAYGO ACCESS: @ AUDIT ACCESS: @
129.12	LOINC AXIS CODES	^ETSLNC(129.12,	DD ACCESS: @ RD ACCESS: WR ACCESS: @ DEL ACCESS: @ LAYGO ACCESS: @ AUDIT ACCESS: @
129.13	LOINC METHOD CODES	^ETSLNC(129.13,	DD ACCESS: @ RD ACCESS: WR ACCESS: @ DEL ACCESS: @ LAYGO ACCESS: @ AUDIT ACCESS: @
129.14	LOINC EXCLUDED WORDS FILE	^ETSLNC(129.14,	DD ACCESS: @ RD ACCESS: WR ACCESS: @ DEL ACCESS: @ LAYGO ACCESS: @ AUDIT ACCESS: @

9.4. Electronic Signatures

There are no electronic signatures used or required by this package.

9.5. Secure Data Transmission

There are currently no data transmissions that are part of this package.

10. Archiving

Currently, none of the ETS data is archived.

11. Non-Standard Cross-References

There are no non-standard cross-references.

12. Troubleshooting

12.1. Special Instructions for Error Correction

There are no special instructions.

12.2. National Service Desk and Organizational Contacts

The table below has the National Service Desk contacts.

Table 4: National Service Desk Contacts

Name	Role	Gov or Non-Gov	FTE	Org	Contact Info
Tier 1: National Service Desk	NSD Tier 1 Support	Gov	No change to existing VistA support FTE	NSD	1-855-NSD-HELP
Tier 2: National Service Desk	NSD Tier 1 Support	Gov	No change to existing VistA support FTE	NSD	Service Desk Manager (SDM) trouble tickets
Tier 3: Regional Application Service Line	Install Patch – Tier 3	Gov	No change to existing VistA support FTE	OI&T Field Operations	Region 1 – Kaylene Zimmer 360-816-6158 Region 2 – Stefan Test 850-293-0858 Region 3 – Jenny Stafford 901-601-0641 Region 4 – Mike Fournier 603-998-4714 Region 6 – Derik Frioud 303-370-7715

13. Acronyms and Abbreviations

Table 5: Acronyms and Abbreviations

Term	Definition
ADPAC	Applications Coordinator
API	Application Program Interface
CBO	Chief Business Office
DBIA	Database Integration Agreement
DFN	Data File Number
ETS	Enterprise Terminology Services
FTE	Full-Time Employee
HL7	Health Level Seven
ICR	Integration Control Registrations
ID	Identifier
IEN	Internal Entry Number
IRM	Information Resource Management
IT	Information Technology
KIDS	Kernel Installation and Distribution System
LOINC	Logical Observation Identifiers Names and Codes
M / MUMPS	Massachusetts General Hospital Utility Multi-Programming System
NSD	National Service Desk
NSR	New Service Request
OED	Office of Enterprise Development
OI&T	Office of Information and Technology
PIMS	Patient Information Management System
PMO	Program Management Office
RPC	Remote Procedure Call
RSD	Requirements Specification Document
SACC	Standards and Conventions Committee
SDM	Service Desk Manager
SSOi	Single Sign-On and Patient Context Management
STS	Standards and Terminology Services
VA	Department of Veterans Affairs
VistA	Veterans Health Information Systems and Technology Architecture

Term	Definition
Access Code	The unique sequence of characters assigned to the user by the site system manager. The access code in conjunction with the verify code is used to identify authorized users.
Application	A collection of computer programs and files developed specifically to meet the requirements of a user or group of users.
Archive	The process of moving data that is no longer actively used to a separate storage for long-term retention.
Field	A data element in a file.
FileMan	The VistA database manager.
Global	A collection of variables (fields) stored on disk that persist beyond routine or process completion. M VistA Server Globals are records stored in structured data files by M.
Kernel	A set of utilities that support data processing on VistA M Servers.
Option	Commands presented to a computer user by an applications. Typically, options are presented on a menu and have specific entry and exit actions.
Procedure	A reusable part of a computer program that performs a single function.
Purge	The action/process of deleting a file or data from a file.
Remote Procedure Call	An inter-process communication protocol that allows invocation of a program subroutine or procedure to execute in shared network space.
Required Field	A field which must have a data value entered by the user or passed as a parameter to computer program or subroutine.
Routine	A set of commands and arguments related, stored, and executed as a single M program.