

Enterprise Terminology Service (ETS)

Package Version 1.0

Technical Manual / Security Guide



Version 1.1

June 2017

Department of Veterans Affairs

Office of Information and Technology (OI&T)

Revision History

Date	Revision	Description	Author
May 2017	1.1	The following sections were updated for ETS*1.0*1: <ul style="list-style-type: none">• Section 1.2 (System Overview)• Section 1.3 (Reference)• Section 2.2 (System Setup and Configuration)• Section 3 (Files)• Section 4 (Routines)• Section 7.1 (Integration Control Registrations)• Section 7.2 (Application Programming Interfaces)• Section 9.3 (File Security)• Section 13 (Acronyms and Abbreviations)	PST Team Tech Writer Review: T.K.
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

1	Introduction	1
1.1	Purpose.....	1
1.2	System Overview	1
1.3	Document Orientation.....	1
2	Implementation and Maintenance	2
2.1	System Requirements	2
2.2	System Setup and Configuration.....	3
3	Files	3
4	Routines	5
5	Exported Options	5
6	Mail Groups, Alerts, and Bulletins	5
7	Public Interfaces	5
7.1	Integration Control Registrations	5
7.2	Application Programming Interfaces	6
7.3	Remote Procedure Calls.....	30
7.4	HL7 Messaging.....	30
7.5	Web Services.....	30
7.6	Contingency Planning	30
8	Standards and Conventions Exemptions	30
8.1	Internal Relationships.....	30
8.2	Software-wide Variables	30
9	Security	30
9.1	Security Menus and Options.....	30
9.2	Security Keys and Roles	30
9.3	File Security.....	31
9.4	Electronic Signatures	32
9.5	Secure Data Transmission	32
10	Archiving	32
11	Non-Standard Cross-References	32
12	Troubleshooting	33
12.1	Special Instructions for Error Correction.....	33
12.2	National Service Desk and Organizational Contacts	33
13	Acronyms and Abbreviations	33

Table of Tables

Table 1: Files List	3
Table 2: ICR List	5
Table 3: File Security List	31
Table 4: National Service Desk Contacts	33
Table 5: Acronyms and Abbreviations	33

Table of Figures

Figure 1: \$\$CHKCODE^ETSLNC Example	6
Figure 2: \$\$COMLST^ETSLNC Example – Component IEN	7
Figure 3: \$\$COMLST^ETSLNC Example – Component Name	7
Figure 4: \$\$CSDATA^ETSLNC Example	8
Figure 5: \$\$CSYS^ETSLNC Example	9
Figure 6: \$\$DEPLST^ETSLNC Example	9
Figure 7: \$\$GETCODE^ETSLNC Example	10
Figure 8: \$\$GETNAME^ETSLNC Example – LOINC IEN	11
Figure 9: \$\$GETNAME^ETSLNC Example – LOINC Code	11
Figure 10: \$\$GETREC^ETSLNC Example	12
Figure 11: \$\$GETSTAT^ETSLNC Example – LOINC IEN	14
Figure 12: \$\$GETSTAT^ETSLNC Example – LOINC Code	14
Figure 13: \$\$HIST^ETSLNC Example	15
Figure 14: \$\$PERIOD^ETSLNC Example	16
Figure 15: \$\$TAX^ETSLNC Example – LOINC Code	17
Figure 16: \$\$TAX^ETSLNC Example – Text String	17
Figure 17: \$\$VERSION^ETSLNC Example	18
Figure 18: \$\$CSDATA^ETSRXN Example	18
Figure 19: \$\$CSYS^ETSRXN Example	19
Figure 20: \$\$GETDATA^ETSRXN Example	20
Figure 21: \$\$HIST^ETSRXN Example	22
Figure 22: \$\$NDC2RXN^ETSRXN Example – Eleven-digit NDC	22
Figure 23: \$\$NDC2RXN^ETSRXN Example – NDC in 5-4-2 format	22
Figure 24: \$\$NDC2RXN^ETSRXN Example – Twelve-digit NDC	23
Figure 25: \$\$PERIOD^ETSRXN Example	23
Figure 26: \$\$RXN2OUT^ETSRXN Example	24
Figure 27: \$\$TAX^ETSRXN Example	25
Figure 28: \$\$VUICLASS^ETSRXN Example	26
Figure 29: \$\$VUID2RXN^ETSRXN Example – Valid VUID	29
Figure 30: \$\$VUID2RXN^ETSRXN Example – Valid VUID and Term Type	29

1 Introduction

This Enterprise Terminology Service (ETS) package exists to provide enterprise-wide access to common code sets that are needed in a health care system. It will contain VA 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 ETS package will consist of common codesets and Application Programming Interfaces (APIs) to access the data in those files. The first codeset in ETS is the Logical Observation Identifiers Names and Codes (LOINC) files. The LOINC database provides sets of universal names and ID codes for identifying laboratory and clinical test results. LOINC is a trademark of the Regenstrief Institute.

The second codeset will be RxNorm databases, which are maintained by the National Library of Medicine (NLM).

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.

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.

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

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

RxNorm Reference documents:

- RxNorm files are available for download at <https://www.nlm.nih.gov/research/umls/rxnorm/docs/rxnormfiles.html>. A Unified Medical Language System® (UMLS®) login is required.
- RxNorm Technical Manual is available at https://www.nlm.nih.gov/research/umls/rxnorm/docs/2017/rxnorm_doco_full_2017-2.html
- UMLS Quick Start Guide is available at <https://www.nlm.nih.gov/research/umls/quickstart.html>

2 Implementation and Maintenance

2.1 System Requirements

Hardware Requirements

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

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

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 Enterprise Terminology Service package, routines, database definitions, and data are delivered as standard Kernel Installation and Distribution System (KIDS) builds. Prior to installation of the ETS patches, the VA Fileman and Kernel patches should be installed and up-to-date.

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.

File #	File Name	Global Location	Description
129.2	RXNORM CONCEPT NAMES AND SOURCES	^ETSRXN(129.2,	The RxNorm Concept Names and Sources file contains a subset of the data distributed by NLM in the RXNCONSO.RRF file. There is exactly one row in this file for each atom (each occurrence of each unique string or concept name within each source vocabulary) in RxNorm. The initial extract of data will be for the records where the Source Abbreviation (SAB) is one of the following: 'RXNORM', 'VANDF', 'NDDF', 'MMSL', or 'ATC'.
129.21	RXNORM SIMPLE CONCEPT AND ATOM ATTRIBUTES	^ETSRXN(129.21,	The RxNorm Simple Concept and Atom Attributes file contains a subset of the data distributed by NLM in the RXNSAT.RRF file. There is exactly one row in this table for each concept, atom, or relationship attribute that does not have a sub-element structure. Not all RxNorm concepts or RxNorm relations have entries in this file. This file includes all source vocabulary attributes that do not fit into other categories. The initial extract of data will be for the records where the Source Abbreviation (SAB) is either 'RXNORM', 'VANDF', or 'ATC', the Attribute Name (ATN) is not either 'UMLSAUI' or 'UMLSCUI', and the RXCUI is not null.
129.22	RXNORM RELATED CONCEPTS	^ETSRXN(129.22,	The RxNorm Related Concepts file contains a subset of the data distributed by NLM in the RXNREL.RRF file. There is one row in this table for each relationship between concepts or atoms known to RxNorm. The initial extract of data will be for the records where the Source Abbreviation (SAB) is 'RXNORM' and the RXCUI1 is not null.
129.23	RXNORM SEMANTIC TYPES	^ETSRXN(129.23,	The RxNorm Semantic Types file contains a subset of the data distributed by NLM in the RXNSTY.RRF file. There is exactly one row in this file for each Semantic Type assigned to each concept. All RxNorm concepts have at least one entry in this file. Many have more than one entry. The initial extract of data will be for the records where the Semantic Type (STY) is not null.
129.24	RXNORM SOURCE INFORMATION	^ETSRXN(129.24,	The RxNorm Source Information file contains a subset of the data distributed by NLM in the RXNSAB.RRF file. This file contains the sources for each of the RxNorm Files loaded into VistA. The initial extract of data will be for all records.

4 Routines

The following is a list of routines that exist in the ETS namespace.

- ETS1PST
- ETSLNC
- ETSLNC1
- ETSLNC2
- ETSLNC3
- ETSLNCIX
- ETSLNCTX
- ETSRXN
- ETSRXN1
- ETSRXNTX

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 LOINC APIs included in this package available for use by all VistA packages.
Supported Reference	ICR #6758 has been created to make all RxNorm APIs included in this package available for use by all VistA packages.

7.2 Application Programming Interfaces

LOINC APIs

\$\$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 with check digit

Output

\$\$CHKCODE: IEN of the LOINC file or -1^<Error message>

Figure 1: \$\$CHKCODE^ETSLNC Example

```
>W $$CHKCODE^ETSLNC("42012-5")
42012
```

\$\$COMLST^ETSLNC: Retrieve component list

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

Format

```
$$COMLST^ETSLNC(COM,TYP,SUB)
```

Input parameters

COM: (Required) Component IEN or Component Name

TYP: (Optional) Input Type – Either "I" for IEN or "N" for Name. If not passed in, defaults to "N".

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

Output

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

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

Figure 2: \$\$COMLST^ETSLNC Example – Component IEN

```
>W $$COMLST^ETSLNC(18,"I","SAB")
1
^TMP("SAB",8826,"COMP",6984)="BETA LACTAMASE.EXTENDED SPECTRUM:SUSC:PT:ISOLATE:ORDQN:"
```

Figure 3: \$\$COMLST^ETSLNC Example – Component Name

```
>W $$COMLST^ETSLNC("BACLOFEN","N","SAB")
1
^TMP("SAB",32024,"COMP",9353)="BACLOFEN:MCNC:PT:SER/PLAS:QN:"
43897)="BACLOFEN:THRESHOLD:PT:SER/PLAS:ORD:"
74377)="BACLOFEN:MCNC:PT:UNK SUB:QN:"
74390)="BACLOFEN:MCNT:PT:XXX:QN:"
74391)="BACLOFEN:MCNC:PT:BLD:QN:"
75227)="BACLOFEN:MCNC:PT:URINE:QN:"
```

\$\$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 with check digit
SYS: (Optional) Coding System. 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 the current day.
RESULT: (Required – Passed by Reference) Array to return the information requested.

Output

- \$\$CSDATA:** 1 – Successful or 0 – Unsuccessful or -1^<Error message>
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 4: \$\$CSDATA^ETSLNC Example

```
>W $$CSDATA^ETSLNC("1-8","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"
```

\$\$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. For LOINC, the only valid value is "LNC". If not passed in, defaults to "LNC".

Output

\$\$CSYS: A delimited string of code system information or -1^<Error message>

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 5: \$\$CSYS^ETSLNC Example

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

\$\$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 6: \$\$DEPLST^ETSLNC Example

```
>W $$DEPLST^ETSLNC("SAB")
1
^TMP("SAB",127896,"DEPRECATED",1)="ACYCLOVIR:SUSC:PT:ISOLATE:ORDQN:"
```

\$\$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 or -1^<Error message>

Figure 7: \$\$GETCODE^ETSLNC Example

```
>W $$GETCODE^ETSLNC1(1)
1-8
```

\$\$GETNAME^ETSLNC: Retrieve LOINC Name Array

This API retrieves the LOINC Name Array, which includes the Fully Specified Name, Long Common Name, and Short 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 or 0 – no results found or -1^<Error message>

NAME: An array with the Name values. The array will be cleared upon entry. The output array will have the following subscripts:

NAME ("FULLNAME")=Fully Specified Name

NAME("LONGNAME")=Long Common

NAME("SHORTNAME")=Short Name

Figure 8: \$\$GETNAME^ETSLNC Example – LOINC IEN

```
>K NAME
>W $$GETNAME^ETSLNC(1,"I",.NAME)
1
>ZW NAME
NAME=""
NAME("FULLNAME")="ACYCLOVIR:SUSC:PT:ISOLATE:ORDQN:"
NAME("LONGNAME")="ACYCLOVIR [SUSCEPTIBILITY]"
NAME("SHORTNAME")="ACYCLOVIR SUSC ISLT"
```

Figure 9: \$\$GETNAME^ETSLNC Example – LOINC Code

```
>K NAME
>W $$GETNAME^ETSLNC("23123-3","C",.NAME)
>ZW NAME
NAME=""
NAME("FULLNAME")="FRANCISELLA TULARENSIS A RRNA:ACNC:PT:TISS:ORD:PROBE"
NAME("LONGNAME")="FRANCISELLA TULARENSIS A RRNA [PRESENCE] IN TISSUE BY DNA PROBE"
NAME("SHORTNAME")="F TULAR A RRNA TISS QL PRB"
```

\$\$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, defaults to "C".

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

Output

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

TMP Global: 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",#)
 "COMPONENT")
 "DATE LAST CHANGED")
 "EXAMPLE UCUM UNITS")
 "EXTERNAL COPYRIGHT NOTICE")
 "EXTERNAL COPYRIGHT NOTICE,#")
 "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 10: \$\$GETREC^ETSLNC Example

```

>W $$GETREC^ETSLNC("38213-5","C","SAB")
1
^TMP("SAB",8826,"RECORD","ACTIVATION HISTORY",1,"ACTIVATION EFFECTIVE DATE")="3150629^JUN 29, 2015"
  "ACTIVATION STATUS")="1^ACTIVE"
^TMP("SAB",8826,"RECORD","ADJUSTMENT")=""
  "CHALLENGE")=""
  "CHANGE REASON")=""
  "CHANGE TYPE")="MIN"
  "CHECK DIGIT")=5
  "CLASS")="MICRO"
  "CLASSTYPE")="2^CLINICAL"
  "CODE")="38213-5"
  "COMMENTS")=4
^TMP("SAB",8826,"RECORD","COMMENTS",1)="The Faces, Legs, Activity, Cry, and Consolability (FLACC) scale is a
behavioral "
  
```



```

2)="pain assessment scale that can be used with non/pre-verbal patients (young child"
3)="ren). Zero, one or two points is assigned to each of the five categories; total "
4)="score ranges from zero to ten."
^TMP("SAB",8826,"RECORD","COMPONENT")="FLACC PAIN ASSESSMENT PANEL"
    "DATE LAST CHANGED")="3150508^MAY 08, 2015"
    "EXAMPLE UCUM UNITS")=""
    "EXTERNAL COPYRIGHT NOTICE")=3
^TMP("SAB",8826,"RECORD","EXTERNAL COPYRIGHT NOTICE",1)="Copyright (C) 2002 The Regents of the University of
Michigan Include the followi"
    2)="ng when printing the FLACC on documentation records, etc: Printed with permissio"
    3)="n (C) 2002, The Regents of the University of Michigan"
    "FULLY SPECIFIED NAME")="FLACC PAIN ASSESSMENT PANEL:-:PT:~PATIENT:-"
    "IEN")=38213
    "LONG COMMON NAME")="FLACC PAIN ASSESSMENT PANEL"
    "MASTER ENTRY FOR VUID")="YES"
    "METHOD TYPE")=""
    "NON-PATIENT SPECIMEN")=""
    "PROPERTY")="DASH"
    "REPEAT OBSERVATION")=""
    "SCALE TYPE")="DASH"
    "SHORTNAME")="FLACC PAIN ASSESSMENT PNL"
    "SNOMED CODE")=""
    "SOURCE")="SM"
    "SYSTEM")="~PATIENT"
    "TIME ASPECT")="POINT"
    "TIME MODIFIER")=""
    "UNITS")=""
    "VA COMMON DISPLAY NAME")=""
    "VERSION NUMBER")=""
    "VUID")=4681405
^TMP("SAB",8826,"RECORD","VUID EFFECTIVE DATE",1,"EFFECTIVE DATE/TIME")="3050501^MAY 01, 2005"
    "STATUS")="1^ACTIVE"

```

\$\$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 with Check Digit or LOINC IEN

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

Output

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

Figure 11: \$\$GETSTAT^ETSLNC Example – LOINC IEN

```
>W $$GETSTAT^ETSLNC(282,"I")
1^DEL
```

Figure 12: \$\$GETSTAT^ETSLNC Example – LOINC Code

```
>W $$GETSTAT^ETSLNC("2-6"),"C")
^Active
```

\$\$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 with Check Digit.
SYS: (Optional) Coding System for lookup. For LOINC, the valid value is "LNC". If not passed in, defaults to "LNC".
RESULT: (Required – Passed by Reference) Array returning the information requested.

Output

- \$\$HIST:** 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 13: \$\$HIST^ETSLNC Example

```
>W $$HIST^ETSLNC("3-4","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"
```

\$\$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 with Check Digit.
SYS: (Optional) Coding System to perform the Lookup. For LOINC, the only valid value is "LNC". If not passed in, 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 14: \$\$PERIOD^ETSLNC Example

```
W $$PERIOD^ETSLNC("3-4","LNC",.ARY)
1^^LNC^LOINC^Logical Observation Identifier Names and Codes

ARY(0)="1^^LNC^LOINC^Logical Observation Identifier Names and Codes"
ARY(3150629)="^^3;ETSLNC(129.1,^ALMECILLIN [SUSCEPTIBILITY] BY MINIMUM INHIBITORY
CONCENTRATION (MIC)"
ARY(3150629,0)="ALMECILLIN:SUSC:PT:ISOLATE:ORDQN:MIC"
```

\$\$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 the current day.
- SUB:** (Optional) Global subscript for ^TMP global. If not passed in, defaults to "ETSTAX".
- 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_ " "),#): Five-piece string of delimited values

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): Two-piece string of delimited values

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 15: \$\$TAX^ETSLNC Example – LOINC Code

```
>W $$TAX^ETSLNC("1-8","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:"
```

Figure 16: \$\$TAX^ETSLNC Example – Text String

```
>W $$TAX^ETSLNC("ALMECILLIN","LNC")
5

^TMP("LEXTAX",80148,0)=5
^TMP("LEXTAX",80148,1,"18857-3",5)="3150629^^^18857;ETSLNC(129.1,^ALMECILLIN [SUSCEPTIBILITY]"
^TMP("LEXTAX",80148,1,"18857-3",5,0)="18857-3^ALMECILLIN:SUSC:PT:ISOLATE:ORDQN:"
^TMP("LEXTAX",80148,1,"2-6",1)="3150629^3160301^^2;ETSLNC(129.1,^ALMECILLIN [SUSCEPTIBILITY] BY
MINIMUM LETHAL CONCENTRATION (MLC)"
^TMP("LEXTAX",80148,1,"2-6",1,0)="2-6^ALMECILLIN:SUSC:PT:ISOLATE:QN:MLC"
^TMP("LEXTAX",80148,1,"3-4",2)="3150629^^^3;ETSLNC(129.1,^ALMECILLIN [SUSCEPTIBILITY] BY MINIMUM
INHIBITORY CONCENTRATION (MIC)"
^TMP("LEXTAX",80148,1,"3-4",2,0)="3-4^ALMECILLIN:SUSC:PT:ISOLATE:ORDQN:MIC"
^TMP("LEXTAX",80148,1,"4-2",3)="3150629^^^4;ETSLNC(129.1,^ALMECILLIN [SUSCEPTIBILITY] BY DISK
DIFFUSION (KB)"
^TMP("LEXTAX",80148,1,"4-2",3,0)="4-2^ALMECILLIN:SUSC:PT:ISOLATE:ORDQN:AGAR DIFFUSION"
^TMP("LEXTAX",80148,1,"5-9",4)="3150629^^^5;ETSLNC(129.1,^ALMECILLIN [SUSCEPTIBILITY] BY SERUM
BACTERICIDAL TITER"
^TMP("LEXTAX",80148,1,"5-9",4,0)="5-9^ALMECILLIN:TITR:PT:ISOLATE+SER:QN:SBT"
```

\$\$VERSION^ETSLNC: Retrieve LOINC Version

This API retrieves the LOINC version.

Format

```
$$VERSION^ETSLNC()
```

Input parameter

None

Output

\$\$VERSION: LOINC Version Number or Null (No Version Data found) or -1^<Error message>

Figure 17: \$\$VERSION^ETSLNC Example

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

RxNorm APIs

\$\$CSDATA^ETSRXN: Get Detailed Information about a RXCUI

This API allows a user to retrieve information for a given RXCUI. 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

```
$$CSDAT^ETSRXN(CODE,SYS,CDT,.RESULT)
```

Input Parameters

CODE: (Required) RXCUI
SYS: (Optional) Coding System to perform the Lookup on. For RxNorm, this will be "RXN". If not passed in, defaults to "RXN".
CDT: (Optional) Date in FileMan format to use to evaluate Status. If not passed in, defaults to the current day.
RESULT: (Optional – Passed by Reference) Array returning the information requested.

Output Parameters

\$\$CSDATA: 1 if successful (value found in RXNCONSO file) or 0 if unsuccessful or -1^<Error message>
RESULT("LEX",1): Two pieces of data – IEN^ STR
RESULT("LEX",1,"N"): "IEN ^ Text (STR)"
RESULT("LEX",2): Two pieces of data – Activation Status (1:Active, 2:Inactive)^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("RXN",1): Three pieces of data - TTY^SUPPRESS^CVF
RESULT("RXN",1,"N"): "Term Type (TTY) ^ Suppression Flag (Suppress) ^ Content View Flag (CVF)"
* Note that the "N" values are the description of the output.

Figure 18: \$\$CSDATA^ETSRXN Example

```
>K ARY
>W $$CSDATA^ETSRXN(749151,"RXN",,.ARY)
1
>ZW ARY
```

```

ARY("LEX",1)="271798^TRI-NORINYL 28 Day Pack"
ARY("LEX",1,"N")="IEN ^ Text (STR)"
ARY("LEX",2)="1^3170306"
ARY("LEX",2,"N")="Status ^ Effective Date"
ARY("LEX",8)=" "
ARY("LEX",8,"N")="Deactivated Concept"
ARY("RXN",1)="PSN^N^4096"
ARY("RXN",1,"N")="Term Type (TTY) ^ Suppression Flag (Suppress) ^ Content View Flag (CVF)"

```

\$\$CSYS^ETSRXN: Retrieve the Coding System Information

This API allows a user to retrieve information about the RxNorm Set. 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.

Input Parameters

SYS: Coding System to perform Lookup on (optional). For RxNorm, the valid value will be "RXN". If not passed in, it will default to "RXN".

Output Parameters

\$\$CSYS: A fourteen piece caret (^) delimited string or -1^<Error Message>

1. IEN – Not Used
2. SAB (3-character Source Abbreviation) – "RXN"
3. Source Abbreviation – "RxNorm"
4. Nomenclature – "RxNorm"
5. Source Title – "RxNorm"
6. Source – "National Library of Medicine"
7. Entries – Not Used
8. Unique Entries – Not Used
9. Inactive Version – Not Used
10. HL7 Coding System – Not Used
11. SDO Version Date – Not Used
12. SDO Version Id – Not Used
13. Implementation Date – Not Used
14. Lookup Threshold – Not Used

Figure 19: \$\$CSYS^ETSRXN Example

```

>W $$CSYS^ETSRXN("RXN")
^RXN^RXN^RxNorm^RxNorm^National Library of Medicine

```

\$\$GETDATA^ETSRXN: Return all information for a RXCUI

This API retrieves all information related to an RXCUI if a valid RXCUI is passed in.

Format

\$\$GETDATA^ETSRXN(RXCUI,SUB)

Input Parameters

RXCUI: (Required) RxNorm Concept ID

SUB: (Optional) Subscript for the TMP global. If not passed in, defaults to "ETSDATA".

Output Parameters

\$\$GETDATA: 1 if valid RXCUI was found or 0 for no RXCUI found or -1^<Error message>

TMP global: Return all data for the RXCUI data in a TMP global. The structure if the TMP global is:

```
^TMP(SUB,$J,RXCUI,"RXNCONSO")=Count
^TMP(SUB,$J,RXCUI,"RXNCONSO",RXNCONSO count,0)=IEN^RXCUI^SAB^TTY^CODE^SUPPRESS^CVF
^TMP(SUB,$J,RXCUI,"RXNCONSO",RXNCONSO count,1)=STR
^TMP(SUB,$J,RXCUI,"RXNREL")=Count
^TMP(SUB,$J,RXCUI,"RXNREL",RXNREL count,0)=IEN^RXCUI1^REL^RXCUI2^RELA^SAB^SUPPRESS^CVF
^TMP(SUB,$J,RXCUI,"RXNSAT")=Count
^TMP(SUB,$J,RXCUI,"RXNSAT",RXNSAT count,0)=IEN^RXCUI^CODE^SAB^SUPPRESS^CVF
^TMP(SUB,$J,RXCUI,"RXNSAT",RXNSAT count,1)=ATN
^TMP(SUB,$J,RXCUI,"RXNSAT",RXNSAT count,2)=ATV
^TMP(SUB,$J,RXCUI,"RXNSTY")=Count
^TMP(SUB,$J,RXCUI,"RXNSTY",RXNSTY count,0)=IEN^RXCUI^STY^CVF
```

Figure 20: \$\$GETDATA^ETSRXN Example

```
>W $$GETDATA^ETSRXN(29115,"OUT")
1
^TMP("OUT",27627,29115,"RXCONSO")=3
^TMP("OUT",27627,29115,"RXCONSO",1,0)="12004^29115^RXNORM^BN^29115^N^"
1)="Maalox"
^TMP("OUT",27627,29115,"RXCONSO",2,0)="12005^29115^MMSL^BN^47069^N^"
1)="Maalox"
^TMP("OUT",27627,29115,"RXCONSO",3,0)="12006^29115^MMSL^BN^5604^O^"
1)="Maalox (obsolete)"
^TMP("OUT",27627,29115,"RXNREL")=18
^TMP("OUT",27627,29115,"RXNREL",1,0)="86149^29115^RB^612^has_tradename^RXNORM^^"
^TMP("OUT",27627,29115,"RXNREL",2,0)="86150^29115^RB^6581^has_tradename^RXNORM^^"
^TMP("OUT",27627,29115,"RXNREL",3,0)="86151^29115^RO^210836^has_ingredient^RXNORM^^"
^TMP("OUT",27627,29115,"RXNREL",4,0)="86152^29115^RO^210847^has_ingredient^RXNORM^^"
^TMP("OUT",27627,29115,"RXNREL",5,0)="86153^29115^RO^210850^has_ingredient^RXNORM^^"
^TMP("OUT",27627,29115,"RXNREL",6,0)="86154^29115^RO^210852^has_ingredient^RXNORM^^"
^TMP("OUT",27627,29115,"RXNREL",7,0)="86155^29115^RO^366011^has_ingredient^RXNORM^^"
^TMP("OUT",27627,29115,"RXNREL",8,0)="86156^29115^RO^367863^has_ingredient^RXNORM^^"
^TMP("OUT",27627,29115,"RXNREL",9,0)="86157^29115^RO^571316^has_ingredient^RXNORM^^"
^TMP("OUT",27627,29115,"RXNREL",10,0)="86158^29115^RO^571327^has_ingredient^RXNORM^^"
^TMP("OUT",27627,29115,"RXNREL",11,0)="86159^29115^RO^571329^has_ingredient^RXNORM^^"
```



```

^TMP("OUT",27627,29115,"RXNREL",12,0)="86160^29115^RO^571331^has_ingredient^RXNORM^^"
^TMP("OUT",27627,29115,"RXNREL",13,0)="86161^29115^RO^1176865^has_ingredient^RXNORM^^"
^TMP("OUT",27627,29115,"RXNREL",14,0)="86162^29115^RO^1176866^has_ingredient^RXNORM^^"
^TMP("OUT",27627,29115,"RXNREL",15,0)="86163^29115^RO^1176867^has_ingredient^RXNORM^^"
^TMP("OUT",27627,29115,"RXNREL",16,0)="86164^29115^RO^1302338^has_ingredient^RXNORM^^"
^TMP("OUT",27627,29115,"RXNREL",17,0)="86165^29115^RO^1302339^has_ingredient^RXNORM^^"
^TMP("OUT",27627,29115,"RXNREL",18,0)="86166^29115^RO^1302340^has_ingredient^RXNORM^^"
^TMP("OUT",27627,29115,"RXNSAT")=3
^TMP("OUT",27627,29115,"RXNSAT",1,0)="4257^29115^29115^RXNORM^N^"
    1)="RXN_ACTIVATED"
    2)="01/28/2010"
^TMP("OUT",27627,29115,"RXNSAT",2,0)="4258^29115^29115^RXNORM^N^"
    1)="RXN_BN_CARDINALITY"
    2)="multi"
^TMP("OUT",27627,29115,"RXNSAT",3,0)="4259^29115^29115^RXNORM^N^"
    1)="RXN_OBSOLETED"
    2)="12/04/2009"
^TMP("OUT",27627,29115,"RXNSTY")=2
^TMP("OUT",27627,29115,"RXNSTY",1,0)="15479^29115^Inorganic Chemical^"
^TMP("OUT",27627,29115,"RXNSTY",2,0)="15480^29115^Pharmacologic Substance^"

```

\$\$HIST^ETSRXN: Get Activation History for a RXCUI

The API allows a user to extract the Activation History for a given RXCUI. 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^ETSRXN(CODE,SYS,.RESULT)

Input Parameters

CODE: (required) RXCUI
SYS: (optional) Coding System to perform the Lookup on. For RxNorm, this will be "RXN". If not passed in, defaults to "RXN".
RESULT: (optional) Array returning the information requested (Passed by Reference).

Output Parameters

\$\$HIST: Number of Histories Found or '-1^<Error message>
RESULT(0): Number of Histories Found
RESULT(0,0): A four-piece "^" delimited string
 1. IEN from RXNCONSO
 2. Source Abbreviation – "RXN"
 3. Nomenclature – "RXNORM"
 4. TTY from RXNCONSO
RESULT(0,1): STR
RESULT(<date>,<status>): Comment

Where: <date> is the date in FileMan internal date format of the last Activation Status

<status> is 1 (Active) or 0 (Inactive)
 <comment> is further information ("Activated", "Inactivated", "Re-activated", "Revised"). The comment is dependent on the current and previous statuses.

Figure 21: \$\$HIST^ETSRXN Example

```
>K ARY
>W $$HIST^ETSRXN(749151,"RXN",.ARY)
1
>ZW ARY
ARY(0)=1
ARY(0,0)="271798^RXN^RXNORM^PSN"
ARY(0,1)="TRI-NORINYL 28 Day Pack"
ARY(3170306,1)="Activated"
```

\$\$NDC2RXN^ETSRXN: Return RXCUI for a NDC

This API allows a user to extract the RxNorm Concept number if a valid NDC code is provided.

Format

\$\$NDC2RXN^ETSRXN(NDC,SUB)

Input Parameters

NDC: (Required) NDC Code.

SUB: (Optional) Subscript for TMP global. If not passed in, defaults to "ETSNDC".

Output Parameters

\$\$NDC2RXN: RxNorm Concept ID or -1^<Error Message>

^TMP global: Temporary Global with list of RXCUI and other pertinent data. The structure of the TMP global is:

^TMP(SUB,\$J,RXCUI Count,0)=IEN^RXCUI^SAB^SUPPRESS

Figure 22: \$\$NDC2RXN^ETSRXN Example – Eleven-digit NDC

```
>W $$NDC2RXN^ETSRXN("13668004529","OUT")
1
^TMP("OUT",4471,1,0)="554600^751139^RXNORM^N"
```

Figure 23: \$\$NDC2RXN^ETSRXN Example – NDC in 5-4-2 format

```
>W $$NDC2RXN^ETSRXN("13668-0045-29","OUT")
1
```

```
^TMP("OUT",4471,1,0)="554600^751139^RXNORM^N"
```

Figure 24: \$\$NDC2RXN^ETSRXN Example – Twelve-digit NDC

```
>W $$NDC2RXN^ETSRXN("013668004529","OUT")
1
^TMP("OUT",4471,1,0)="554589^751139^VANDF^N"
```

\$\$PERIOD^ETSRXN: Get Activation/Inactivation Periods for a RXCUI

This API allows a user to view the activation periods for a given RXCUI. 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^ETSRXN(CODE,SYS,RESULT)

Input Parameters

CODE: (Required) RXCUI

SYS: (Optional) Coding System to perform the Lookup on. For RxNorm, this will be "RXN". If not passed in, defaults to "RXN".

RESULT: (Optional) Array returning the information requested (Passed by Reference)

Output Parameters

\$\$PERIOD: 5 piece "^" string (below) or -1^<Error message>

1. Number of Activation Periods Found
2. Term Type
3. Source Abbreviation – "RXN"
4. Nomenclature – "RXNORM"
5. Source Title – "RXNORM"

RESULT(0): Same as \$\$PERIOD

RESULT(<Activation Date>): Four-piece "^" string:

1. Inactivation Date
2. Not used
3. Variable Pointer to the RXNCONSO file - "<IEN>;ETSRXN(129.2,"
4. Not Used

RESULT(<Activation Date>,0): STR

Where Activation Date is the date the RXCUI and SAB was activated

Figure 25: \$\$PERIOD^ETSRXN Example

```
>K ARY
>W $$PERIOD^ETSRXN(749151,"RXN",.ARY)
```

```
1^PSN^RXN^RXNORM^RXNORM
```

```
>ZW ARY
```

```
ARY(0)="1^PSN^RXN^RXNORM^RXNORM"
```

```
ARY(3170306)="^^271798;ETSRXN(129.2^"
```

```
ARY(3170306,0)="TRI-NORINYL 28 Day Pack"
```

\$\$RXN2OUT^ETSRXN: Retrieve a VUID and NDC for a RXCUI

This API allows a user to extract a VA Unique ID and NDC if a valid RXCUI is provided.

Format

```
$$RXN2OUT^ETSRXN(RXCUI,SUB)
```

Input Parameters

RXCUI: (Required) RxNorm Concept ID

SUB: (Optional) Subscript for the TMP global. If not passed in, defaults to "ETSOUT".

Output Parameters

\$\$RXN2OUT: Two pieces (see below) or -1^<Error Message>

1. VUID count
2. NDC count

^TMP global: Return all data for the RXCUI data in a TMP global. The structure if the TMP global is:

```
^TMP(SUB,$J,RXCUI,"VUID")=Count
```

```
^TMP(SUB,$J,RXCUI,"VUID",VUID Count,0)=IEN^RXCUI^SAB^TTY^VUID^SUPPRESS
```

```
^TMP(SUB,$J,RXCUI,"VUID",VUID Count,1)=STR
```

```
^TMP(SUB,$J,RXCUI,"NDC")=Count
```

```
^TMP(SUB,$J,RXCUI,"NDC",NDC Count,0)=IEN^RXCUI^CODE^SAB^SUPPRESS
```

```
^TMP(SUB,$J,RXCUI,"NDC",NDC Count,1)=ATN
```

```
^TMP(SUB,$J,RXCUI,"NDC",NDC Count,2)=ATV
```

Figure 26: \$\$RXN2OUT^ETSRXN Example

```
>W $$RXN2OUT^ETSRXN(751139,"OUT")
```

```
2^6
```

```
^TMP("OUT",4011,751139,"NDC",1,0)="554587^751139^4026525^VANDF^O"
```

```
1)="NDC"
```

```
2)="000173063306"
```

```
^TMP("OUT",4011,751139,"NDC",2,0)="554588^751139^4026525^VANDF^Y"
```

```
1)="NDC"
```

```
2)="000173063310"
```

```
^TMP("OUT",4011,751139,"NDC",3,0)="554589^751139^4026525^VANDF^N"
```

```
1)="NDC"
```

```
2)="013668004529"
```

```

^TMP("OUT",4011,751139,"NDC",4,0)="554590^751139^4026525^VANDF^N"
    1)="NDC"
    2)="013668004570"
^TMP("OUT",4011,751139,"NDC",5,0)="554600^751139^751139^RXNORM^N"
    1)="NDC"
    2)=13668004529
^TMP("OUT",4011,751139,"NDC",6,0)="554601^751139^751139^RXNORM^N"
    1)="NDC"
    2)=13668004570
^TMP("OUT",4011,751139,"VUID")=2
^TMP("OUT",4011,751139,"VUID",1,0)="273225^751139^VANDF^CD^4026525^N"
    1)="LAMOTRIGINE 25MG TAB,35,KIT"
^TMP("OUT",4011,751139,"VUID",2,0)="273227^751139^VANDF^AB^4026525^N"
    1)="LAMICTAL 25MG TAB,35 STARTER KIT"

```

\$\$TAX^ETSRXN: Return list of VUIDs in same value set

This API is for the creation of Taxonomy lists. This API will return all VUIDs for the products in the same value set as the valid VUID passed in.

Format:

```
$$TAX^ETSRXN(VUID,SUB)
```

Input Parameters

VUID: (Required) VA Unique ID

SUB: (Optional) Subscript for TMP global. If not passed in, defaults to "ETSTAX"

Output Parameters

\$\$TAX: Number of return values or -1^<Error message>

^TMP global: Return list of VUID with the same value set. The structure if the TMP global is:

```
^TMP(SUB,$J, User-Entered VUID ,"VUID")=Count
```

```
^TMP(SUB,$J, User-Entered VUID ,"VUID",VUID Count,0)=IEN^RXCUI^SAB^TTY^VUID^SUPPRESS
```

```
^TMP(SUB,$J, User-Entered VUID ,"VUID",VUID Count,1)=STR
```

```
^TMP(SUB,$J, User-Entered VUID ,"VUID",VUID Count,2)=ACTIVATION DATE
```

Figure 27: \$\$TAX^ETSRXN Example

```

>W $$TAX^ETSRXN(4001147,"OUT")
1

^TMP("OUT",25599,4001147,"VUID")=1
^TMP("OUT",25599,4001147,"VUID",1,0)="137076^313725^VANDF^IN^4020804^N"
    1)="VITAMIN E OIL"
    2)="MAR 06, 2017"

```

\$\$VUICLASS^ETSRXN: Return list of VUIDs with the same drug class

This API will return the VUIDs for all products sharing the same ATC drug class as the valid VUID passed in.

Format

\$\$VUICLASS^ETSRXN(VUID,SUB)

Input Parameters

VUID: (Required) VA Unique ID

SUB: (Optional) Subscript for TMP global. If not passed in, defaults to "ETSCLA"

Output Parameters

\$\$VUICLASS: Number of return values or -1^<Error Message>

^TMP global: Return list of VUID with the same drug class. The structure if the TMP global is:

^TMP(SUB,\$J, User-Entered VUID ,"VUID")=Count

^TMP(SUB,\$J, User-Entered VUID ,"VUID",VUID Count,0)=IEN^RXCUI^SAB^TTY^VUID^SUPPRESS

^TMP(SUB,\$J, User-Entered VUID ,"VUID",VUID Count,1)=STR

Figure 28: \$\$VUICLASS^ETSRXN Example

```
>W $$VUICLASS^ETSRXN(4025394,"OUT")
78

^TMP("OUT",25599,4025394,"VUID")=78
^TMP("OUT",25599,4025394,"VUID",1,0)="34736^198051^VANDF^CD^4025402^N"
1)="OMEPRAZOLE 20MG CAP,EC"
^TMP("OUT",25599,4025394,"VUID",2,0)="34737^198051^VANDF^CD^4025403^N"
1)="OMEPRAZOLE 20MG CAP,EC,UD"
^TMP("OUT",25599,4025394,"VUID",3,0)="34738^198051^VANDF^AB^4025402^N"
1)="OMEPRAZOLE 20MG EC CAP"
^TMP("OUT",25599,4025394,"VUID",4,0)="34739^198051^VANDF^AB^4025403^N"
1)="OMEPRAZOLE 20MG EC CAP UD"
^TMP("OUT",25599,4025394,"VUID",5,0)="39642^199119^VANDF^CD^4025400^N"
1)="OMEPRAZOLE 10MG CAP,EC"
^TMP("OUT",25599,4025394,"VUID",6,0)="39643^199119^VANDF^CD^4025401^N"
1)="OMEPRAZOLE 10MG CAP,EC,UD"
^TMP("OUT",25599,4025394,"VUID",7,0)="39644^199119^VANDF^AB^4025400^N"
1)="OMEPRAZOLE 10MG EC CAP"
^TMP("OUT",25599,4025394,"VUID",8,0)="39645^199119^VANDF^AB^4025401^N"
1)="OMEPRAZOLE 10MG EC CAP UD"
^TMP("OUT",25599,4025394,"VUID",9,0)="42826^200329^VANDF^CD^4025404^N"
1)="OMEPRAZOLE 40MG CAP,EC"
^TMP("OUT",25599,4025394,"VUID",10,0)="42828^200329^VANDF^AB^4025404^N"
1)="OMEPRAZOLE 40MG EC CAP"
^TMP("OUT",25599,4025394,"VUID",11,0)="103956^251872^VANDF^CD^4017059^N"
1)="PANTOPRAZOLE NA 20MG TAB,EC"
^TMP("OUT",25599,4025394,"VUID",12,0)="103960^251872^VANDF^AB^4017059^N"
1)="PANTOPRAZOLE NA 20MG EC TAB"
^TMP("OUT",25599,4025394,"VUID",13,0)="112690^283669^VANDF^CD^4015647^N"
1)="PANTOPRAZOLE NA 40MG/VIL INJ"
```

```

^TMP("OUT",25599,4025394,"VUID",14,0)="112691^283669^VANDF^AB^4015647^N"
1)="PANTOPRAZOLE NA 40MG/VIL INJ"
^TMP("OUT",25599,4025394,"VUID",15,0)="127713^311277^VANDF^CD^4025395^N"
1)="LANSOPRAZOLE 30MG CAP,EC"
^TMP("OUT",25599,4025394,"VUID",16,0)="127714^311277^VANDF^CD^4025396^N"
1)="LANSOPRAZOLE 30MG CAP,EC,UD"
^TMP("OUT",25599,4025394,"VUID",17,0)="127715^311277^VANDF^AB^4025395^N"
1)="LANSOPRAZOLE 30MG EC CAP"
^TMP("OUT",25599,4025394,"VUID",18,0)="127716^311277^VANDF^AB^4025396^N"
1)="LANSOPRAZOLE 30MG EC CAP UD"
^TMP("OUT",25599,4025394,"VUID",19,0)="139058^314200^VANDF^CD^4014681^N"
1)="PANTOPRAZOLE NA 40MG TAB,EC"
^TMP("OUT",25599,4025394,"VUID",20,0)="139059^314200^VANDF^CD^4017265^N"
1)="PANTOPRAZOLE NA 40MG TAB,EC,UD"
^TMP("OUT",25599,4025394,"VUID",21,0)="139063^314200^VANDF^AB^4014681^N"
1)="PANTOPRAZOLE NA 40MG EC TAB"
^TMP("OUT",25599,4025394,"VUID",22,0)="139064^314200^VANDF^AB^4017265^N"
1)="PANTOPRAZOLE NA 40MG EC TAB UD"
^TMP("OUT",25599,4025394,"VUID",23,0)="155754^351260^VANDF^CD^4017058^N"
1)="LANSOPRAZOLE 30MG TAB,ORAL DISINTEGRATING,SA"
^TMP("OUT",25599,4025394,"VUID",24,0)="155755^351260^VANDF^AB^4017058^N"
1)="LANSOPRAZOLE 30MG SA DISINTEGRATING TAB"
^TMP("OUT",25599,4025394,"VUID",25,0)="155764^351261^VANDF^CD^4017057^N"
1)="LANSOPRAZOLE 15MG TAB,ORAL DISINTEGRATING,SA"
^TMP("OUT",25599,4025394,"VUID",26,0)="155765^351261^VANDF^AB^4017057^N"
1)="LANSOPRAZOLE 15MG SA DISINTEGRATING TAB"
^TMP("OUT",25599,4025394,"VUID",27,0)="179067^402014^VANDF^CD^4016813^N"
1)="OMEPRAZOLE 20MG TAB,SA"
^TMP("OUT",25599,4025394,"VUID",28,0)="179074^402014^VANDF^AB^4016813^N"
1)="OMEPRAZOLE 20MG SA TAB"
^TMP("OUT",25599,4025394,"VUID",29,0)="210285^486499^VANDF^CD^4025128^N"
1)="ESOMEPRAZOLE NA 20MG/VIL INJ"
^TMP("OUT",25599,4025394,"VUID",30,0)="210286^486499^VANDF^AB^4025128^N"
1)="ESOMEPRAZOLE NA 20MG/VIL INJ"
^TMP("OUT",25599,4025394,"VUID",31,0)="210295^486501^VANDF^CD^4025127^N"
1)="ESOMEPRAZOLE NA 40MG/VIL INJ"
^TMP("OUT",25599,4025394,"VUID",32,0)="210296^486501^VANDF^AB^4025127^N"
1)="ESOMEPRAZOLE NA 40MG/VIL INJ"
^TMP("OUT",25599,4025394,"VUID",33,0)="233993^596843^VANDF^CD^4025394^N"
1)="LANSOPRAZOLE 15MG CAP,EC"
^TMP("OUT",25599,4025394,"VUID",34,0)="233994^596843^VANDF^CD^4025393^N"
1)="LANSOPRAZOLE 15MG CAP,EC,UD"
^TMP("OUT",25599,4025394,"VUID",35,0)="233995^596843^VANDF^AB^4025394^N"
1)="LANSOPRAZOLE 15MG EC CAP"
^TMP("OUT",25599,4025394,"VUID",36,0)="233996^596843^VANDF^AB^4025393^N"
1)="LANSOPRAZOLE 15MG EC CAP UD"
^TMP("OUT",25599,4025394,"VUID",37,0)="239906^606726^VANDF^CD^4034873^N"
1)="ESOMEPRAZOLE 20MG (22.3MG W/MAG) CAP,EC"
^TMP("OUT",25599,4025394,"VUID",38,0)="239907^606726^VANDF^AB^4034873^N"
1)="ESOMEPRAZOLE 20MG (22.3MG W/MAG) EC CAP"
^TMP("OUT",25599,4025394,"VUID",39,0)="239925^606730^VANDF^CD^4025392^N"
1)="ESOMEPRAZOLE MAGNESIUM 40MG CAP,EC"
^TMP("OUT",25599,4025394,"VUID",40,0)="239927^606730^VANDF^AB^4025392^N"
1)="ESOMEPRAZOLE MAGNESIUM 40MG EC CAP"
^TMP("OUT",25599,4025394,"VUID",41,0)="242089^616539^VANDF^CD^4025610^N"
1)="OMEPRAZOLE 20MG/SODIUM BICARBONATE 1100MG
CAP, ORAL"
^TMP("OUT",25599,4025394,"VUID",42,0)="242090^616539^VANDF^AB^4025610^N"
1)="OMEPRAZOLE 20/NA BICARB 1100MG ORAL CAP"
^TMP("OUT",25599,4025394,"VUID",43,0)="242100^616541^VANDF^CD^4025611^N"
1)="OMEPRAZOLE 40MG/SODIUM BICARBONATE 1100MG
CAP, ORAL"
^TMP("OUT",25599,4025394,"VUID",44,0)="242101^616541^VANDF^AB^4025611^N"

```

```

1)="OMEPRAZOLE 40/NA BICARB 1100MG ORAL CAP"
^TMP("OUT",25599,4025394,"VOID",45,0)="258797^692576^VANDF^CD^4026004^N"
1)="ESOMEPRAZOLE MAGNESIUM 20MG/PKT PWDR,ORAL"
^TMP("OUT",25599,4025394,"VOID",46,0)="258798^692576^VANDF^AB^4026004^N"
1)="ESOMEPRAZOLE MAG 20MG/PKT ORAL PWDR"
^TMP("OUT",25599,4025394,"VOID",47,0)="258807^692578^VANDF^CD^4026005^N"
1)="ESOMEPRAZOLE MAGNESIUM 40MG/PKT PWDR,ORAL"
^TMP("OUT",25599,4025394,"VOID",48,0)="258808^692578^VANDF^AB^4026005^N"
1)="ESOMEPRAZOLE MAG 40MG/PKT ORAL PWDR"
^TMP("OUT",25599,4025394,"VOID",49,0)="274553^753557^VANDF^CD^4026576^N"
1)="OMEPRAZOLE 40MG/SODIUM BICARBONATE 1680MG/PKT
PWDR,ORAL"
^TMP("OUT",25599,4025394,"VOID",50,0)="274554^753557^VANDF^AB^4026576^N"
1)="OMEPRAZOLE 40MG/NA BICARB/PKT ORAL PWDR"
^TMP("OUT",25599,4025394,"VOID",51,0)="274564^753562^VANDF^CD^4026578^N"
1)="OMEPRAZOLE 20MG/SODIUM BICARBONATE 1680MG/PKT
PWDR,ORAL"
^TMP("OUT",25599,4025394,"VOID",52,0)="274565^753562^VANDF^AB^4026578^N"
1)="OMEPRAZOLE 20MG/NA BICARB/PKT ORAL PWDR"
^TMP("OUT",25599,4025394,"VOID",53,0)="279750^763306^VANDF^AB^4027563^N"
1)="PANTOPRAZOLE NA 40MG/PKT ORAL PWDR"
^TMP("OUT",25599,4025394,"VOID",54,0)="279751^763306^VANDF^CD^4027563^N"
1)="PANTOPRAZOLE NA 40MG/PKT PWDR,ORAL"
^TMP("OUT",25599,4025394,"VOID",55,0)="282969^797058^VANDF^AB^4029288^N"
1)="OMEPRAZOLE MAGNESIUM 10MG/PKT ORAL PWDR"
^TMP("OUT",25599,4025394,"VOID",56,0)="282970^797058^VANDF^CD^4029288^N"
1)="OMEPRAZOLE MAGNESIUM 10MG/PKT PWDR,ORAL"
^TMP("OUT",25599,4025394,"VOID",57,0)="282985^797063^VANDF^AB^4033640^N"
1)="OMEPRAZOLE MAGNESIUM 2.5MG/PKT ORAL PWDR"
^TMP("OUT",25599,4025394,"VOID",58,0)="282986^797063^VANDF^CD^4033640^N"
1)="OMEPRAZOLE MAGNESIUM 2.5MG/PKT PWDR,ORAL"
^TMP("OUT",25599,4025394,"VOID",59,0)="297599^833204^VANDF^CD^4028554^N"
1)="DEXLANSOPRAZOLE 30MG CAP,EC"
^TMP("OUT",25599,4025394,"VOID",60,0)="297600^833204^VANDF^AB^4028554^N"
1)="DEXLANSOPRAZOLE 30MG EC CAP"
^TMP("OUT",25599,4025394,"VOID",61,0)="297608^833213^VANDF^CD^4028555^N"
1)="DEXLANSOPRAZOLE 60MG CAP,EC"
^TMP("OUT",25599,4025394,"VOID",62,0)="297609^833213^VANDF^AB^4028555^N"
1)="DEXLANSOPRAZOLE 60MG EC CAP"
^TMP("OUT",25599,4025394,"VOID",63,0)="308364^854868^VANDF^CD^4014188^N"
1)="RABEPRAZOLE NA 20MG TAB,EC"
^TMP("OUT",25599,4025394,"VOID",64,0)="308365^854868^VANDF^CD^4014189^N"
1)="RABEPRAZOLE NA 20MG TAB,EC,UD"
^TMP("OUT",25599,4025394,"VOID",65,0)="308369^854868^VANDF^AB^4014188^N"
1)="RABEPRAZOLE NA 20MG EC TAB"
^TMP("OUT",25599,4025394,"VOID",66,0)="308370^854868^VANDF^AB^4014189^N"
1)="RABEPRAZOLE NA 20MG EC TAB UD"
^TMP("OUT",25599,4025394,"VOID",67,0)="348953^994005^VANDF^AB^4030075^N"
1)="ESOMEPRAZOLE 20MG/NAPROXEN 375MG EC TAB"
^TMP("OUT",25599,4025394,"VOID",68,0)="348954^994005^VANDF^CD^4030075^N"
1)="ESOMEPRAZOLE MAGNESIUM 20MG/NAPROXEN 375MG
TAB,EC"
^TMP("OUT",25599,4025394,"VOID",69,0)="348968^994008^VANDF^AB^4030076^N"
1)="ESOMEPRAZOLE 20MG/NAPROXEN 500MG EC TAB"
^TMP("OUT",25599,4025394,"VOID",70,0)="348969^994008^VANDF^CD^4030076^N"
1)="ESOMEPRAZOLE MAGNESIUM 20MG/NAPROXEN 500MG
TAB,EC"
^TMP("OUT",25599,4025394,"VOID",71,0)="494820^1437489^VANDF^CD^4032848^N"
1)="RABEPRAZOLE NA 5MG CAP,SPRINKLE"
^TMP("OUT",25599,4025394,"VOID",72,0)="494821^1437489^VANDF^AB^4032848^N"
1)="RABEPRAZOLE NA 5MG SPRINKLE CAP"
^TMP("OUT",25599,4025394,"VOID",73,0)="498386^1483318^VANDF^CD^4032850^N"
1)="RABEPRAZOLE NA 10MG CAP,SPRINKLE"

```



```

^TMP("OUT",25599,4025394,"VUID",74,0)="498387^1483318^VANDF^AB^4032850^N"
1)="RABEPRAZOLE NA 10MG SPRINKLE CAP"
^TMP("OUT",25599,4025394,"VUID",75,0)="559304^1811631^VANDF^AB^4036078^N"
1)="ASPIRIN 81MG/OMEPRAZOLE 40MG EC TAB"
^TMP("OUT",25599,4025394,"VUID",76,0)="559305^1811631^VANDF^CD^4036078^N"
1)="ASPIRIN 81MG/OMEPRAZOLE 40MG TAB,EC"
^TMP("OUT",25599,4025394,"VUID",77,0)="559313^1811632^VANDF^AB^4036079^N"
1)="ASPIRIN 325MG/OMEPRAZOLE 40MG EC TAB"
^TMP("OUT",25599,4025394,"VUID",78,0)="559314^1811632^VANDF^CD^4036079^N"
1)="ASPIRIN 325MG/OMEPRAZOLE 40MG TAB,EC"

```

\$\$VUI2RXN^ETSRXN: Retrieve RXCUI(s) for a valid VUID

This API allows a user to extract the RxNorm Concept number(s) if a valid VA Unique ID is provided.

Format

\$\$VUI2RXN^ETSRXN(VUID,TTY,SUB)

Input parameters

VUID: (Required) VA Unique ID

TTY: (Optional) Term Type in Source

SUB: (Optional) Subscript for ^TMP global. If not passed in, defaults to "ETSRXN"

Output parameters

\$\$VUI2RXN: Number of records found or -1^<Error message>

^TMP global: Return list of RXCUIs. The structure if the TMP global is:

^TMP(ETSSUB,\$J,RXCUI Count,0)=IEN^RXCUI^SAB^TTY^CODE^SUPPRESS

^TMP(ETSSUB,\$J,RXCUI Count,1)=STR

Figure 29: \$\$VUID2RXN^ETSRXN Example – Valid VUID

```

>W $$VUI2RXN^ETSRXN(4026525,"","OUT")
2
^TMP("OUT",4471,1,0)="273227^751139^VANDF^AB^4026525^N"
1)="LAMICTAL 25MG TAB,35 STARTER KIT"
^TMP("OUT",4471,2,0)="273225^751139^VANDF^CD^4026525^N"
1)="LAMOTRIGINE 25MG TAB,35,KIT"

```

Figure 30: \$\$VUID2RXN^ETSRXN Example – Valid VUID and Term Type

```

>W $$VUI2RXN^ETSRXN(4026525,"CD","OUT")
1
^TMP("OUT",4471,1,0)="273225^751139^VANDF^CD^4026525^N"
1)="LAMOTRIGINE 25MG TAB,35,KIT"

```

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: @
129.2	RXNORM CONCEPT NAMES AND SOURCES	^ETSRXN(129.2,	DD ACCESS: @ RD ACCESS: WR ACCESS: @ DEL ACCESS: @ LAYGO ACCESS: @ AUDIT ACCESS: @

File #	File Name	Global Location	File Security
129.21	RXNORM SIMPLE CONCEPT AND ATOM ATTRIBUTES	^ETSRXN(129.21,	DD ACCESS: @ RD ACCESS: WR ACCESS: @ DEL ACCESS: @ LAYGO ACCESS: @ AUDIT ACCESS: @
129.22	RXNORM RELATED CONCEPTS	^ETSRXN(129.22,	DD ACCESS: @ RD ACCESS: WR ACCESS: @ DEL ACCESS: @ LAYGO ACCESS: @ AUDIT ACCESS: @
129.23	RXNORM SEMANTIC TYPES	^ETSRXN(129.23,	DD ACCESS: @ RD ACCESS: WR ACCESS: @ DEL ACCESS: @ LAYGO ACCESS: @ AUDIT ACCESS: @
129.24	RXNORM SOURCE INFORMATION	^ETSRXN(129.24,	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
ATN	Attribute Name
CBO	Chief Business Office
DBIA	Database Integration Agreement
DFN	Data File Number
ETS	Enterprise Terminology Service

Term	Definition
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
NDC	National Drug Code
NLM	National Library of Medicine
NSD	National Service Desk
NSR	New Service Request
OED	Office of Enterprise Development
OI&T	Office of Information and Technology
SAB	Source Abbreviation
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
STY	Semantic Type
UMLS	Unified Medical Language System
VA	Department of Veterans Affairs
VistA	Veterans Health Information Systems and Technology Architecture
VUID	VA Unique ID
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.

Term	Definition
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.