



Lexicon Utility

Technical Manual and Developer's Guide

**Version 2.0
September 1996
Revised Sep 11, 2014**

Department of Veterans Affairs (VA)
Office of Information and Technology (OI&T)
Office of Enterprise Development (OED)

Revision History

Date	Description of Change	Author
09/24/2003	LEX*2.0*25 CSV changes: pp. 7, 15-16, 22, 24-25, 28-33, 35, 38, 46, 52, 64-69	N/A
06/12/2008	LEX*2.0*41 Changes related to the implementation of SNOMED CT codes in the Lexicon for the Lab Data Sharing Interoperability (LDSI) project.	Ferdinand Frankson, Kathleen Barnett
05/06/2009	LEX*2.0*62 Changes designed to implement advanced date testing.	Kimball Rowe, Michelle Dilley, Kathleen Barnett
09/22/2010	Existing Tech Manual into new template, edit, format. Added changes for LEX*2.0*73.	Kimball Rowe, Ruth Gong, Peri Koester
11/05/2012	LEX*2.0*58 Added new protocol and routine	Thomas Hackett
04/02/2014	Tech Writer Review-edited for grammar, TOC, etc.	Kimberlee Mann
04/21/2014	Added changes for LEX*2.0*80 (ICD-10)	Kimball Rowe
04/24/2014	Tech Writer Review	Kimberlee Mann
09/11/2014	Added changes for \$\$CODE^LEXTRAN and \$\$GETSYN^LEXTRAN1 for LEX*2.0*86	Kimball Rowe

Table of Contents

1. Preface	6
1.1 Scope of Manual.....	6
1.2 Audience	6
1.3 Related Manuals.....	6
2. Introduction.....	7
3. Clinical Lexicon Version 1.0 (GMPT).....	7
3.1 Problem List Expert Panel.....	7
3.2 End User Requirements	9
3.3 Application Requirements.....	9
4. Lexicon Version 2.0 (LEX).....	9
5. Lexicon Example Entry – Migraine Headache	11
5.1 Terms	11
5.2 Definition.....	11
5.3 Semantic Class/Type	11
5.4 Classification Systems/Codes	11
6. Package Components	12
6.1 Manager Options.....	12
6.1.1 Lexicon Management Menu	12
6.2 User Manager/Options.....	13
6.3 Code Set Versioning Options	14
6.4 Protocols.....	14
6.4.1 Lexicon Update.....	14
6.4.2 Lexicon Update Notification (example)	15
6.4.3 Mapping/Subset Update.....	15
7. Lexicon Files, Fields, and Indexes	17
8. Routines Supporting the Lexicon.....	19
8.1 Special Lookup.....	19
8.1.1 Description.....	19

8.2	Silent Lookup (New)	19
8.2.1	Routines	19
8.2.2	Description.....	19
8.2.3	Routines	19
8.2.4	Setup	20
8.2.5	Routines	20
9.	Setting/Displaying User Defaults	20
9.1	Description	20
9.2	Routines	21
10.	Edit	22
10.1	Description	22
10.2	Routines	23
11.	ICD-10 Support	23
11.1	Description	23
11.2	Routines	23
11.3	Indexing and Input Transformations	24
11.3.1	Description.....	24
11.3.2	Routines	24
12.	Code Set Versioning Queries	25
12.1	Description	25
12.2	Routines	25
12.2.1	ICD-10 Specific	26
12.3	Miscellaneous	26
12.3.1	Routines	26
13.	Package Characteristics and Usage	27
13.1	Supported Callable Routines	27
13.1.1	LEXSET	27
13.2	LEXCODE.....	45
13.3	LEX10CS (ICD-10 Specific).....	46
14.	Pruning the Output	49
14.1	LEX10CX (ICD-10 Specific).....	55

14.2	LEXSRC2.....	57
14.3	LEXTRAN	58
14.4	LEXTRAN1	61
14.5	LEXXM.....	65
14.6	LEXA.....	66
14.7	LEXAR	68
14.7.1	User Responses	69
14.7.2	Application Comment	71
15.	Other Supporting Package Components	72
15.1	LEXD* Namespaced Routines.....	72
15.1.1	EN1^LEXD*(<Application>)	72
15.2	Special Variables.....	73
15.2.1	Variables Affecting the Lookup.....	73
15.2.2	Global Arrays.....	74
15.2.3	Search Conditions.....	74
15.2.4	Local Arrays.....	75
15.2.5	Error Array LEX("ERR").....	76
15.2.6	Exact Match Concept LEX("EXC").....	76
15.2.7	Help Array LEX("HLP").....	77
15.2.8	List Array LEX("LIST").....	77
15.2.9	User Narrative LEX("NAR").....	77
15.2.10	Matches Found String LEX("MAT").....	77
15.2.11	Maximum Selection LEX("MAX").....	77
15.2.12	Minimum Selection LEX("MIN").....	77
15.2.13	Response from the User LEX("RES").....	78
15.2.14	Selection Array LEX("SEL").....	78
15.3	Controlling the View	80
15.3.1	View by Semantic Class and Types	80
15.3.2	View by Classification System	81
15.3.3	View by both Semantics and Classification Systems	81
15.3.4	View by Subset.....	81
15.3.5	Other Views.....	82

16. Searching the Lexicon:	82
Building and Re-ordering the List	Error! Bookmark not defined.
16.1 Matches Found ^TMP("LEXFND",\$J).....	82
16.2 Matches Reviewed ^TMP("LEXHIT",\$J).....	82
16.3 Matches Displayed LEX("LIST")	82
16.4 Example Search	82
17. Unresolved Narratives	83
17.1 User Unresolved Narratives	83
17.2 Application Unresolved Narratives	85
18. Re-indexing the Lexicon	85
19. Subsets	86
19.1 Logical Subset	86
19.2 Physical Subset	87
19.3 Application Subset	87
19.4 Creating an Application Subset	87
20. Integration Control Registrations (ICRs) Summary	90
20.1 ICRs with Lexicon as the Custodian	90
20.1.1 Retired/Withdrawn	90
20.1.2 Active/Pending	90
20.2 ICRs with Lexicon as the Subscriber	92
20.2.1 Retired/Expired/Withdrawn	92
20.2.2 Active/Pending	92
20.3 ICRs Supporting Lexicon External References	93
20.3.1 External Global References	93
20.3.2 External Routine References	95
21. Package Security	96
21.1 Use of data by Salt Lake City IRM Field Office Developers:	97
22. SACC Exemptions/Non-Standard Code	98
23. Appendix A: Classification Systems	98
24. Appendix B: Semantic Classes and Types	99

25.	Appendix C: Integration Control Registrations Detailed	113
26.	Lexicon as a Subscriber.....	113
27.	Lexicon as a Custodian.....	194

1. Preface

1.1 Scope of Manual

This manual provides technical information required to effectively set up and use the Veterans Health Information Systems and Technology Architecture (VistA) Lexicon Utility. It also contains material useful in linking to the Lexicon Utility.

1.2 Audience

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

1.3 Related Manuals

- [Lexicon Utility User Manual](#)
- [Lexicon Utility Installation Guide](#)

2. Introduction

VistA's Lexicon Utility is a dynamic dictionary of medical terms. The Lexicon maps coding schemes such as ICD-10, SNOMED CT, and DSM to major medical concepts. Currently, there are 36 different coding schemes represented in the Lexicon. The Lexicon Utility can support other coding schemes that are unique to the VA, such as the codes used by the Social Work Service or US Code Title 38 Chapter 4 for Service Connected Disabilities.

Working with outside sources, the Lexicon updates its terminology as the source files incorporate new terminologies and classification codes. The flexibility offered by this approach is tremendous. The Lexicon can draw from sources such as CMS, AMA and APA while maintaining compatibility with VA authoritative files. The Lexicon adjusts new terminology for use in the VA while retaining backward compatibility with older coding schemes (for example, transition from DSM-III-R to DSM-IV). Updates to the Lexicon are exported on a periodic basis.

The Lexicon supports usage by all clinical areas. With the mapping of application-specific term files, such as radiology, medicine, etc., to the Lexicon, Veterans Health Administration (VHA) achieves standardized clinical terminology. Terminology associated with Hybrid Open System Technology (HOST) applications can also be mapped to the Lexicon. This standardization on the part of VistA and HOST applications permits information mobility within VHA and with other industry-wide systems.

A specially designed interface permits the user to enter a medical term using familiar natural language. The computer codifies and stores this term, permitting retrieval and analysis by a wide variety of legitimate users of clinical data. This interface captures exactly what the user enters and maps it to a standardized term that is linked to a major concept. An application using the Lexicon can reflect what the user actually entered while maintaining the links to the needed coding system and the Lexicon.

The Lexicon's initial implementation was with the Problem List Application Version 1.0. It was completed on September 6, 1994 and released to the field on January 28, 1995.

The initial release of the utility was conceived and planned as a proof of concept. While not complete, the design of the Lexicon includes the ability to evolve as new demands are placed upon it.

3. Clinical Lexicon Version 1.0 (GMPT)

The following software requirements have been stated for the Lexicon Utility:

3.1 Problem List Expert Panel

- Single unique concepts
- Must support natural/native terminology
- All terms must map to ICD-9 (rescinded)

- Must specifically include problem list terminology from NANDA, Social Work, and Dental, as well as clinical findings and symptoms or other terminologies (to be identified)
- Must be flexible enough to map various coding schemes (examples follow)

Coding System	Name (Development Organization)
ACR	Index for Radiological Diagnosis (ACR)
AI/RHEUM	Disease/Findings Knowledge Base (NLM)
BIRADS	Breast Imaging Reporting and Data System (ACR)
COSTAR	Computer Stored Ambulatory Records, MA General Hospital
COSTART	Coding Symbols Thesaurus for Adverse Reaction Terms (FDA)
CPT-4	Current Procedural Terminology (AMA)
CRISP	Computer Retrieval of Info. on Scientific Projects (NIH)
DMIS ID	Defense Medical Information System Identifiers (DoD)
DSM	Diagnostic & Statistical Manual of Mental Disorders (APA)
DXPLAIN	Diagnostic Prompting System, MA General Hospital
HCPCS	HCFA Current Procedural Coding System (CMS)
HHCC	Home Health Care Component
ICD-10-CM	International Classification of Diseases Diagnosis (CMS)
ICD-10-PCS	International Classification of Diseases Procedures (CMS)
ICD-9-CM	International Classification of Diseases Diagnosis (CMS)
LOINC	Logical Observation Identifier Names and Codes (RII)
NANDA	Classification of Nursing Diagnosis (NANDA)
NIC	Nursing Intervention Classifications
NOC	Nursing Outcomes Classifications
OMAHA	Omaha Nursing Diagnosis and Interventions
SCC	Service Connected, US Code Title 38, Chapter 4
SNOMED	Systematized Nomenclature of Medicine (CAP)
SNOMED CT	Systemized Nomenclature of Medicine Clinical Terms (IHTSDO)
UMDNS	Universal Medical Device Nomenclature System (ECRI)

- Must be usable by a variety of applications and utilities within VistA
- Must support addition of terms at the site level (rescinded)
- Must be able to migrate to a nomenclature selected for use throughout VistA when that decision occurs
- Site modification to include edit display text (rescinded) and site specific shortcuts and synonyms (MTLU for v1.0, context sensitive shortcuts for v2.0)

3.2 End User Requirements

- Group terms by clinical categories (e.g., ICD-9 Major Clinical Categories)
- Place the most frequently used terms at the top of the selection list
- Accept the provider narrative if the search fails or the term was not found
- Build subsets of terms (based on specialty or clinic) restricting the lookup domain

3.3 Application Requirements

- Provide Silent Lookup using a multi-term search (CPRS)
- Build shortcuts for terms (based on specialty or clinic) gaining immediate access to terms without the benefit of a search (PL)
- Provide shortcut as a user default (PL)
- Add CPT terminology and codes to the Lexicon Utility (multiple applications)
- Provide entry point to retrieve an internal entry number based on a code from a classification system (PCE)

4. Lexicon Version 2.0 (LEX)

Terminology

Added

Terminology added since v1.0:

- Current Procedural Terminology (CPT-4)
- Diagnostic and Statistical Manual of Mental Disorders (DSM-IV)
- International Classification of Diseases (ICD-10-CM and ICD-10-PCS)
- Systemized Nomenclature of Medicine Clinical Terms (SNOMED CT)
- Defense Medical Information System Identifiers (DMIS ID)
- Service Connected, US Code, Title 38, Chapter 4 (updated)
- Breast Imaging Reporting and Data System (BIRADS)

Namespace LEX

Changed

We changed the namespace from GMPT to LEX. We renamed all routines and package variables from GMPT* to LEX* to conform to the new namespace.

Global Root ^LEX

Changed

We changed the global root from ^GMP and ^GMPT to ^LEX and ^LEXT respectively. This helps to prevent inadvertent deletion of Lexicon data. The difference between killing ^TMP and ^GMP is one character on a standard QWERTY keyboard, both controlled by the same

finger and located approximately a quarter of an inch from each other.

Shortcut Functionality **Added**

The Expression Type field (#757.01) has been changed from a set of codes to a pointer to the new file Expression Type, #757.011.

Concept Usage File #757.001 **Added**

The Concept Usage file records the usage of Lexicon by application performing lookups using the Special Lookup Routines. This file later determines the order of the selection list during lookup. The more frequently used terms float to the top of the list.

Expression Type File #757.011 **Added**

The Expression Type field (#757.01) has been changed from a set of codes to a pointer to the new file Expression Type, #757.011.

Mapping Definitions File #757.31 **Added**

This file is used to define a mapping from one coding system (source code) to another coding system (target code). The coding systems are found in the Coding Systems file #757.03.

Mappings File #757.32 **Added**

This file contains the mappings from one coding system to another coding system. Selection of a term or a code from one coding system can be translated to another coding system.

Codes File #757.02 **Changed**

The Codes file was modified to include a status multiple to record code activation dates and inactivation dates. The ACT cross-reference is generated from this multiple. This cross reference provides the Lexicon the ability to retrieve the appropriate code and text based on a date supplied by the calling routine. If a date is not supplied, then TODAY is used.

Coding Systems File #757.03 **Changed**

The Coding Systems file was modified by making the SOURCE TITLE field #2 an identifier for the purpose of lookup while editing the Change File #757.01. The IMPLEMENTATION DATE field #11 was added to document the implementation of each Coding System.

Character Positions File #757.033 **Added**

This file stores the name/title, description, explanation, and inclusions/examples of a character position in a code.

Subset Definition file #757.2 **Changed**

The DIC("S") value used by various applications and stored in the APPLICATION FILTER field (#10) has been modified to include the passing of a date.

Shortcut User Default **Added**

Context sensitive shortcuts are now a user default. For example, the user may have one set of shortcuts for searching using the Problem List application and another set defined for another

application.

Silent Lookup

Added

A Silent Lookup was added in support of GUI. The Lexicon Special Lookup routine has been modified to call the Silent Lookup so that the behavior of the loud lookup would be identical to the silent lookup. This lookup also includes:

- Reordering the selection list with the most frequently used at the top
- Placing the exact match at the top of the selection list

5. Lexicon Example Entry – Migraine Headache

5.1 Terms

Concept:	Migraine	
Symptoms:	Hemicrania	Directly Linked to Concept
	Migraine Headache	Directly Linked to Concept
Variants:	Hemicrania	Indirectly Linked (via Synonym)
	Hemicranias	Indirectly Linked (via Synonym)
	Migraines	Directly Linked to Concept

5.2 Definition

A periodic vascular headache, usually temporal, and unilateral in onset, commonly associated with irritability, nausea, vomiting, constipation or diarrhea, and often photophobia.

5.3 Semantic Class/Type

Diseases/Pathologic Processes	Signs and Symptoms
	Disease or Syndrome

5.4 Classification Systems/Codes

COSTAR	Computer Stored Ambulatory Records Term File	485/486
COSTART	Coding Symb Thesaurus - Adverse Reaction Terms	MIGRAINE
CRISP	CRISP Thesaurus, Nat Inst of Health	2056-6472
ICD-9-CM	Intl' Class of Diseases, 9th Rev, Clin Mod	346.9/346.
SNOMED D	Sys Nomen of Med, Diagnostic, 2nd Ed	D-8250

6. Package Components

6.1 Manager Options

6.1.1 Lexicon Management Menu

Defaults . . .

[LEX MGR DEFAULTS] Menu

This menu contains two options, one to modify user defaults and one to list user defaults.

Edit User/User Group Defaults LEXDMG

[LEX MGR USER DEFAULTS]

This option allows a manager to modify user defaults (filter, display, shortcuts, vocabulary) for either a single user or a group of users (based on service).

List User/User Group Defaults LEXDD1

[LEX MGR LIST DEFAULTS]

This option allows a manager to list user defaults to a device (filter, display, shortcuts, or vocabulary) for either a single user or a group of users (based on service). It also allows the manager to limit the listing to users with or without defaults.

Edit Lexicon . . .

[LEX MGR EDIT LEXICON] Menu

Very few fields in the Lexicon may be edited. This menu option contains two sub-options that allow managers to edit those [few] fields. One sub-option allows a manager to edit a term definition and the other to edit shortcuts (by context).

Edit Term Definition LEXEDF1

[LEX MGR EDIT DEFN]

This option lets you edit the definition of an expression. This definition is accessible during searches using the Lexicon help routine.

Edit Shortcuts by Context LEXSC

[LEX MGR EDIT SHORTCUTS]

This option lets managers add or delete shortcuts in a selected context.

Edit Search Threshold for a Coding System LEXDMGS

[LEX MGR EDIT SEARCH THRESHOLD]

This allows a manager to edit the search threshold for a coding system. That is the default number of record to examine before prompting the user to continue or refine the search.

6.2 User Manager/Options

Lexicon Utility Menu

[LEX UTILITY] Menu

This menu contains two sub-options, Look-up Term and User Defaults.

Look-up Term LEXLK

[LEX LOOK-UP]

This option lets you perform a simple lookup in the Lexicon and displays all the information known about the term selected.

User Defaults . . .

[LEX USER DEFAULTS] Menu

This menu contains five sub-options that let a single user modify or list user defaults, including the search filter, the display format, the preferred vocabulary, and shortcuts.

Filter EN^LEXDFL

[LEX USER FILTER]

This option lets the users either select or create their own filters to use while conducting searches in the Lexicon. The filter limits the response of the lookup based on the conditions found in the filter.

Display EN^LEXDCC

[LEX USER DISPLAY]

This option lets the user either select or create a display format which is used in presenting the selection list during searches in the Lexicon.

Vocabulary EN^LEXDVO

[LEX USER VOCABULARY]

This option lets the user select a default vocabulary (or subset of the Lexicon) to be used during a lookup (i.e., Nursing, Social Work, etc.).

Shortcuts New

[LEX USER SHORTCUTS] EN^LEXDCX

This option lets the user select a default set of shortcuts to use to rapidly access the Lexicon without the benefit of the special lookup.

List Defaults EN^LEXDDS

[LEX USER DEFAULT LIST]

This option lets the user list the current defaults (by application) to a device (terminal or printer).

6.3 Code Set Versioning Options

Code Sets

[LEX CSV] Menu

ICD-9 Diagnosis Code Set Query LEXQID
[LEX CSV ICD QUERY]

This option displays a single versioned entry from the ICD Diagnosis file #80 based on a date provided by the user. The date may be a future date.

ICD-9 Procedure Code Set Query LEXQIP
[LEX CSV ICP QUERY]

This option displays a single versioned entry from the ICD Operations/Procedure file #80.1 based on a date provided by the user. The date may be a future date.

CPT/HCPCS Procedure Code Set Query LEXQCP
[LEX CSV CPT QUERY]

This option displays a single versioned entry from the CPT/HCPCS file #81 based on a date provided by the user. The date may be a future date.

CPT Modifier Code Set Query LEXQCM
[LEX CSV MOD QUERY]

This option displays a single versioned entry from the CPT Modifier file #81.3 based on a date provided by the user. The date may be a future date.

ICD/CPT Code Set Change List LEXQC
[LEX CSV ICD/CPT CHANGE LIST]

This option produces a listing of ICD/CPT changes effective on the date provided by the user.

Code History

[LEX CSV HISTORY] LEXQH

This option produces a historical display of the versioned data for a selected code.

6.4 Protocols

6.4.1 Lexicon Update

[LEXICAL SERVICES UPDATE]

Extended Action

This Protocol is invoked by the Lexicon each time an update occurs with one of the coding systems that subscribe to this protocol. Currently there are two:

[ICD CODE UPDATE EVENT]

[ICPT CODE UPDATE EVENT]

Applications that require to be notified each time there is an update to either ICD-9 or ICD-10 coding systems may place an action on the [ICD CODE UPDATE EVENT] protocol. Applications that require to be notified each time there is an update to the CPT-4 coding system may place an action on the [ICPT CODE UPDATE EVENT] protocol.

6.4.2 Lexicon Update Notification (example)

[LEXICAL SERVICES UPDATE]

This protocol is invoked when a change to the ICD-9, ICD-10, CPT-4, or HCPCS coding system occurs.

[ICD CODE UPDATE EVENT]

This protocol is invoked when a change to the ICD-9 or ICD-10 coding systems occurs.

[GMPL SELECTION LIST CSV EVENT]

This Problem List protocol is invoked when a change to the ICD-9 coding system occurs. It reports any problems on the problem selection list that contains an inactive ICD-9 code.

[ORCM GMRC CSV EVENT]

This OERR/Consults protocol is invoked when a change to the ICD-9 coding system occurs. It reports any consult or procedure quick order with an inactive ICD-9 code.

[PXR CODE SET UPDATE ICD]

This Clinical Reminder protocol is invoked when a change to the ICD-9 coding system occurs. It reports all inactive ICD-9 codes in the Dialog file #801.41.

[ICPT CODE UPDATE EVENT]

This protocol is invoked when a change to the CPT-4 or HCPCS coding system occurs.

[PXR CODE SET UPDATE CPT]

This Clinical Reminder protocol is invoked when a change to the ICD-9 coding system occurs. It reports all inactive ICD-9 codes in the Clinical Reminders Dialog file #801.41.

6.4.3 Mapping/Subset Update

[LEX MAPPING CHANGE EVENT]

Action

Files, Pointers and Relationships

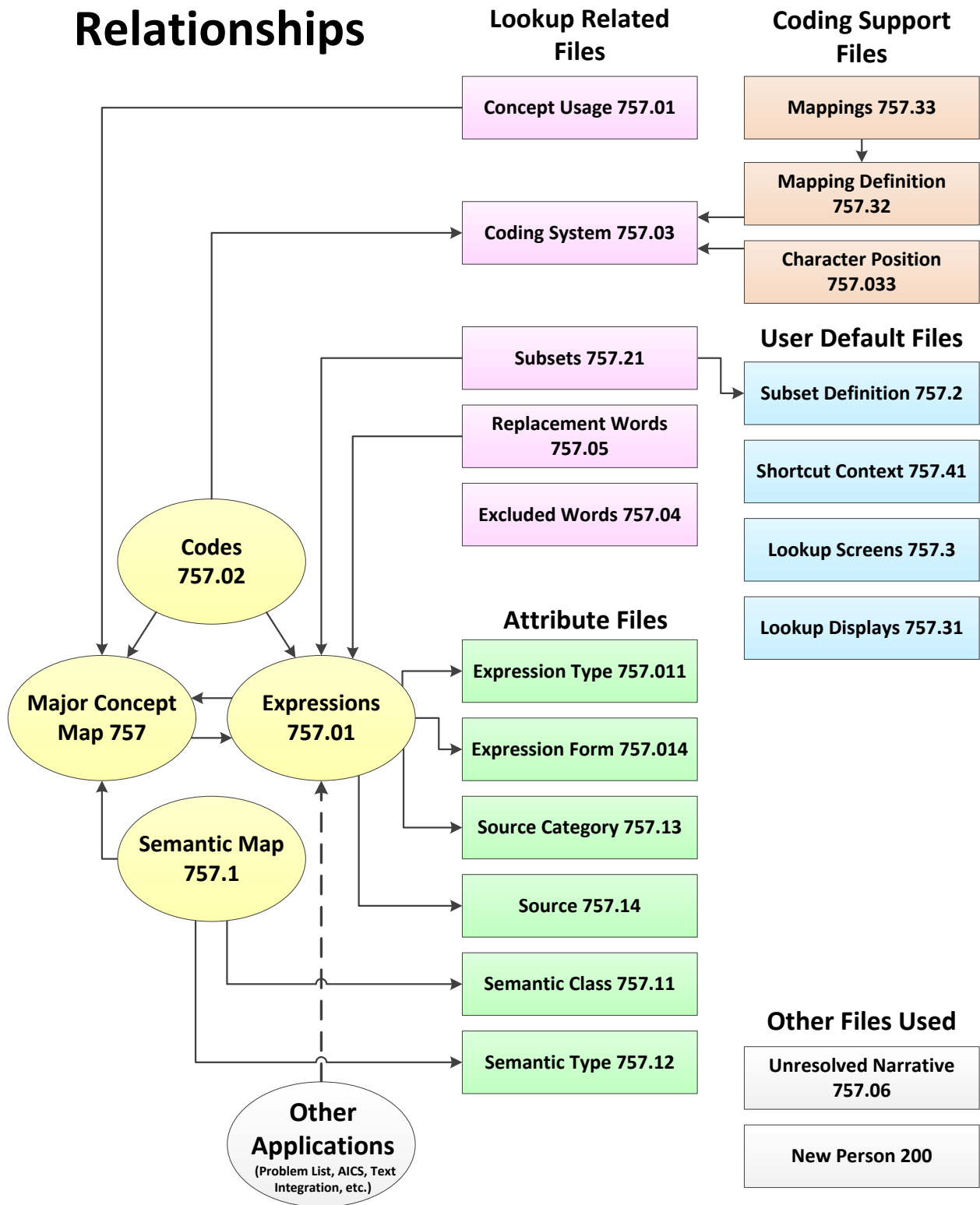


Chart 1: Chart of the Relationships between Lexicon File Components

7. Lexicon Files, Fields, and Indexes

Online documentation for the Lexicon Utility's files, fields, and cross-references may be obtained by using the FileMan's Data Dictionary Listing Utility for the file range 757-757.41 as follows:

>D ^DI

VA FileMan 21.0

Select OPTION: **DATA DICTIONARY UTILITIES**

Select DATA DICTIONARY UTILITY OPTION: **LIST FILE ATTRIBUTES**

START WITH WHAT FILE: SHORTCUT CONTEXT// **757** MAJOR CONCEPT MAP
(99861 entries)

GO TO WHAT FILE: MAJOR CONCEPT MAP// **757.41** SHORTCUT CONTEXT
(3 entries)

Select LISTING FORMAT: STANDARD// **<Enter>**

DEVICE: **<Enter>** VAX

STANDARD DATA DICTIONARY #757 -- MAJOR CONCEPT MAP FILE
STORED IN ^LEX(757, (99861 ENTRIES) SITE: SALT LAKE IRMFO
DATA NAME GLOBAL DATA
ELEMENT TITLE LOCATION TYPE

This file is a map of Major Concepts within the Lexicon Utility and contained in the expression file (#757.01). While the primary purpose of this file is for file maintenance, it also supports various other functions such as the display of classification codes by linking concepts to codes, and the ability to filter out unwanted search responses by linking concepts to semantic classes and types.

NOTE: Data Entries in this file should not be altered by the site.

DD ACCESS: @
RD ACCESS: @
WR ACCESS: @
DEL ACCESS: @
LAYGO ACCESS: @
AUDIT ACCESS: @

POINTED TO BY: MAJOR CONCEPT field (#.01) of the CONCEPT USAGE File (#757.001)

MAJOR CONCEPT field (#1) of the EXPRESSIONS File (#757.01)

MAJOR CONCEPT field (#3) of the CODES File (#757.02)

MAJOR CONCEPT field (#.01) of the SEMANTIC MAP File (#757.1)

CROSS

REFERENCED BY: EXPRESSION(B)

757,.01 EXPRESSION 0;1 POINTER TO EXPRESSIONS FILE
(#757.01) (Required)

OUTPUT TRANSFORM: S Y=\$P(^LEX(757.01,Y,1),U,1)

LAST EDITED: APR 19, 1996

DESCRIPTION: Pointer to the clinical expression in the Expression file (#757.01) which represents the preferred term for the Major Concept.

UNEDITABLE

CROSS-REFERENCE: 757^B
1)= S ^LEX(757,"B", \$E(X,1,30),DA)=""
2)= K ^LEX(757,"B", \$E(X,1,30),DA)

^LEX(757,"B",MCE,IEN) - where MCE is a pointer to the expression in the Expression file #757.01 which represents a clinical Major Concept Expression, and IEN is the Internal Entry Number in the Major Concept Map file #757
.....

.....

You may also retrieve the on-line documentation for any single file listed below by entering a single file number at START WITH WHAT FILE: prompt and not entering a file number at the GO TO WHAT FILE: prompt. The following is a listing of file numbers and file names contained in the Lexicon Utility package:

757	MAJOR CONCEPT MAP
757.001	CONCEPT USAGE
757.01	EXPRESSIONS
757.011	EXPRESSION TYPE
757.014	EXPRESSION FORM
757.02	CODES
757.03	CODING SYSTEMS
757.033	CHARACTER POSITIONS
757.04	EXCLUDED WORDS
757.05	REPLACEMENT WORDS
757.06	UNRESOLVED NARRATIVES
757.1	SEMANTIC MAP
757.11	SEMANTIC CLASSES
757.12	SEMANTIC TYPES
757.13	SOURCE CATEGORY
757.14	SOURCE
757.21	SUBSETS
757.3	LOOK-UP SCREENS
757.31	DISPLAYS
757.32	MAPPING DEFINITIONS
757.33	MAPPINGS
757.4	SHORTCUTS
757.41	SHORTCUT CONTEXT

8. Routines Supporting the Lexicon

8.1 Special Lookup

8.1.1 Description

Because of the requirement to reorder the list after the search and before user selection, this lookup now calls the Silent Lookup so that the search results from the Special Lookup and Silent Lookup remain consistent. The previous special lookup called the API to the Kernel Toolkit Multi-Term Lookup Utility (MTLU).

These routines are responsible for:

- Obtaining the user input search string and passing it to Silent Lookup, then prompting for and getting the user's response.
- Storing and mailing Unresolved Narratives. An Unresolved Narrative occurs when the lookup either does not find a match or when it finds a match but the user, not satisfied with the results, does not select an expression from the list. These Unresolved Narratives are rolled-up into a mail message and submitted for inclusion in later releases of the Lexicon.
- Returning the standard FileMan variables and one additional variable Y(1) containing an active ICD code when one exists.

8.2 Silent Lookup (New)

8.2.1 Routines

LEXA1	Lookup (user input/special lookup routine)
LEXA2	Selection
LEXA3	Display
LEXA4	Unresolved Narratives

8.2.2 Description

This lookup searches the Lexicon and responds by building global and local arrays.

8.2.3 Routines

Lookup

LEXA	Lookup
LEXASC	Lookup by Shortcuts
LEXAB	Exact Match B Index
LEXABC	Lookup by Codes
LEXALK	Lookup by Keywords
LEXAFIL	Lookup with Filter

Interpret User Response

LEXAR	Interpret User Response
LEXAR2	Up-arrow, Jump, Null
LEXAR3	Help, Definition, MAX, Refresh
LEXAR4	Select Entry

LEXAR5	Select Entry
LEXAR6	Unresolved Narratives
LEXAR7	MAIL Narratives

Miscellaneous

LEXAL	List Builder (Global)
LEXAL2	List Builder (Array)
LEXAM	Setup/Parse User Input
LEXASO	Get Classification Sources
LEXAS	Spell Check User Input
LEXAS2	Spell Check User Input
LEXAS3	Spell Check User Input
LEXAS4	Spell Check User Input
LEXAS5	Spell Check User Input
LEXAS6	Spell Check User Input
LEXAS7	Spell Check User Input

8.2.4 Setup

8.2.5 Routines

LEXMTLU	Set up for XTLKKWL API
LEXSET	Set up App/User for Lookup
LEXSET2	Set up App/User for Lookup
LEXSET3	Set up App/User for Lookup
LEXSET4	Set up Functions
LEXSET5	Set up App/User for Lookup

9. Setting/Displaying User Defaults

9.1 Description

Currently, there are only four (4) types of user defaults

- Vocabulary: The word index to use during the search.
- Display: A string of classification coding systems to display during the search.
- Filter: A condition for selecting terms based on semantic or classification systems.
- Shortcut Context: A set of keywords resulting in immediate return of an expression.

These routines allow users and managers to either select from pre-existing defined default values or create their own.

Default	Select Default from a List	Create your own Default
Filter	Yes	Yes
Display	Yes	Yes
Shortcuts	Yes	No
Vocabulary	Yes	No

Additionally, managers may set defaults for multiple users based on service. Both users and managers may display user defaults. Users can only display their own defaults in a human readable format much like that shown in the section, “Controlling the View”. A manager may display the defaults for a single user or a user group, showing the actual data stored as the default values. Because of the complexity of the filtering by semantic classes and types (see Controlling the View), a large number of these routines (LEXDFL*) are devoted to the creation of these filtering strings.

9.2 Routines

Manager

LEXDMG	Manager Options
LEXDMGU	Select User/User Group
LEXDMGV	Verify Selections
LEXDMGS	Search Threshold
LEXDMGO	Overwrite Existing Defaults
LEXDMGT	Task to Modify Defaults

Default Filter

LEXDFL	Default Filter
LEXDFLS	Select a Filter
LEXDFLC	Create a Filter
LEXDFLT	Filter Type
LEXDFST	Filter by Semantics
LEXDFSB	Filter/Exclude Classes/Types
LEXDFSI	Include Semantic Classes/Types
LEXDFSE	Exclude Semantic Classes/Types
LEXDFSO	Filter by Sources
LEXDCCC	Create Filter by Source
LEXDFSS	Filter by Sources and Semantics

Default Display

LEXDCC	Default Display
LEXDCCS	Select a Display
LEXDCCC	Create a Display

Default Vocabulary

LEXDVO	Default Vocabulary
LEXDVOS	Select Default Vocabulary

Default Shortcut Context

LEXDCX	Default Shortcut Context
LEXDCXS	Select Default Shortcut Context

Display Defaults

LEXDD1	Display Defaults
LEXDD2	Build List
LEXDD3	Display List
LEXDD4	List Elements
LEXDDS	Single User Default
LEXDDSD	Single User Display
LEXDDSP	Single User Parse
LEXDDSS	Single User Save

Translate User Defaults

LEXDDT1	Translate String
LEXDDT2	Concatenate Translated String
LEXDDTC	Translate Shortcut String
LEXDDTD	Translate Display String
LEXDDTF	Translate Filter String
LEXDDTV	Translate Vocabulary String

Miscellaneous

LEXDFN	Default Names
LEXDFN2	Default Names
LEXDSV	Save Defaults
LEXDM	Select/Create/Delete Default
LEXDM2	Verify Default Delete
LEXDM3	Default Name
LEXDM4	Get Application/User/Service

10. Edit

10.1 Description

These routines provide managers at a site with the ability to edit the definition and the shortcuts associated with an expression. The definition is used as part of the Lexicon's help routines to assist in the selection of a term from a selection list. An edited definition is mailed to SLC IRMFO for review and

possible inclusion into a future release. The shortcuts are used to associate a keyword to a specific expression to rapidly access the Lexicon without engaging the look-up engine.

10.2 Routines

LEXEDF1	Select/Display/Mail Edited Definition
LEXEDF2	Edit Definition
LEXSC	Edit Shortcuts
LEXSC2	Edit Shortcuts
LEXSC3	Edit Shortcuts

11. ICD-10 Support

11.1 Description

The following routines support the International Classification of Diseases, Diagnosis (ICD) 10th revision coding system APIs and data sets:

11.2 Routines

LEX10CS	Supported ICD-10 APIs
LEX10CS2	Supported ICD-10 APIs (cont)
LEX10CX	ICD-10 Cross-Over
LEX10CX2	ICD-10 Cross-Over - Source
LEX10CX3	ICD-10 Cross-Over - Target
LEX10CX4	ICD-10 Cross-Over - Prompts
LEX10CX5	ICD-10 Cross-Over - Miscellaneous
LEX10DBC	Diagnosis Lookup by Code
LEX10DBR	Diagnosis Lookup by Root/Category
LEX10DBT	Diagnosis Lookup by Text
LEX10DL	Test ICD-10 Diagnosis Lookup
LEX10DLS	Test ICD-10 Diagnosis Lookup selection
LEX10DU	ICD-10 Diagnosis Utility
LEX10DX	ICD-10 Diagnosis
LEX10PL	Test ICD-10 Procedure Lookup
LEX10PLA	Test ICD-10 Procedure Lookup Abbreviations
LEX10PLS	Test ICD-10 Procedure Lookup Selection
LEX10PR	ICD-10 Procedure
LEX10TAX	Clinical Reminder ICD-10 Support

11.3 Indexing and Input Transformations

11.3.1 Description

Entry points for these routines are embedded into the Data Dictionary to maintain indexes and to control input transformations.

The indexes that are controlled in this manner include:

Index	File	Subscript	Routine
Main Word Index	757.01	AWRD	LEXNDX1
Subset Word Index	757.21	"A"_SUBSET	LEXNDX2
Application Index	757.2	APPS	LEXNDX2
Linked Word Index	757.01	AWRD	LEXNDX3
Linkages	757.05	ALINK	LEXNDX4/5
String Index	757.01	ASL	LEXNDX6
Shortcut Index	757.4	ARA	LEXNDX6
Code Set Indexes	757.02	ACT/APR/ADX	LEXNDX8
Mapping Indexes	757.33	G	LEXNDX9

Input transformations controlled in this manner include:

Field	File	Routine
Word (Excluded)	757.04	LEXERI
Word (Replace)	757.05	LEXERI
Replacement Term	757.05	LEXERI

11.3.2 Routines

LEXNDX1	Main Word Index
LEXNDX2	Sub-Set Word Index
LEXNDX3	Replacement Words
LEXNDX4	Linked Words Index
LEXNDX5	Linked Words
LEXNDX6	Index Strings/Shortcuts
LEXNDX8	Index Codes (757.02)
LEXNDX9	Index Mappings (757.33)
LEXERF	Functions for Exe/Rep Words
LEXERI	Exe/Rep Input Transformations
LEXRX	Re-Index Lexicon
LEXRXA	Re-Index 757 B
LEXRXB	Re-Index 757.001 B/AF
LEXRXC	Re-Index 757.01 B/ADC/ADTERM
LEXRXC2	Re-Index 757.01 AMC/AWRD

LEXRXC3	Re-Index 757.01 ASL/APAR
LEXRXD	Re-Index 757.02 B/ACODE/ACT
LEXRXD2	Re-Index 757.02 ADC/AMC/ASRC
LEXRXD3	Re-Index 757.02 ADCODE/APCODE
LEXRXD4	Re-Index 757.02 AVA/CODE/ADX/APR
LEXRXE	Re-Index 757.1 B/AMCC/AMCT/ASTT
LEXRXF	Re-Index 757.21 B/C/AA
LEXRXG	Re-Index 757.33 B/C/G
LEXRXG2	Re-Index 757.33 ACT/AMAP/AREV
LEXRXG3	Re-Index 757.33 ASRC/ATAR
LEXRXS	Re-Index - Small Files
LEXRXS2	Re-Index - Small Files
LEXRXXA	Re-Index Ask
LEXRXXM	Re-Index Miscellaneous
LEXRXXM2	Re-Index Miscellaneous (cont)
LEXRXXP	Re-Index Parse
LEXRXXS	Re-Index Save/Send
LEXRXXT	Repair/Re-Index - Task
LEXRXXT2	Repair/Re-Index - Task (cont)
LEXRXXT3	Repair/Re-Index - Task (cont)

12. Code Set Versioning Queries

12.1 Description

Entry points for these routines are embedded called by the [LEX CSV] name spaced options and display ICD or CPT codes and the versioned data associated with those codes.

12.2 Routines

LEXQC	Code Set (CSV) – Extract
LEXQC2	Code Set (CSV) – Save
LEXQC3	ICD/ICP/10D/10P
LEXQC4	CPT/MOD
LEXQCM	CPT Modifiers – Extract
LEXQCM2	CPT Modifiers – Save
LEXQCMA	CPT Modifiers – Ask
LEXQCP	CPT Procedures – Extract
LEXQCP2	CPT Procedures – Save
LEXQCPA	CPT Procedures – Ask
LEXQID	ICD Diagnosis – Extract
LEXQID2	ICD Diagnosis – Extract (cont)
LEXQID3	ICD Diagnosis – Extract (cont)
LEXQID4	ICD Diagnosis – Save
LEXQIDA	ICD Diagnosis – Ask
LEXQIP	ICD Procedure – Extract
LEXQIP2	ICD Procedure – Extract (cont)

LEXQIP3	ICD Procedure – Save
LEXQIPA	ICD Procedure – Ask
LEXQH	Code History – Main
LEXQHA	Code History – Ask
LEXQHL1	Code History – ICD Dx Extract
LEXQHL2	Code History – ICD Op Extract
LEXQHL3	Code History – CPT/HCPCS Extract
LEXQHL4	Code History – CPT Modifier Extract
LEXQHL5	Code History – Lexicon ICD/CPT Extract
LEXQHLM	Code History – Extract Misc
LEXQL	Code Lookup
LEXQL2	Code Lookup (List)
LEXQL3	Code Lookup (ICD)
LEXQL4	Code Lookup (CPT/Mod)
LEXQD	Defaults
LEXQO	Outputs
LEXQM	Miscellaneous

12.2.1 ICD-10 Specific

LEX10CS	ICD-10 Code Set Lookup
LEX10CS2	ICD-10 Code Set Lookup (continued)
LEX10CX	ICD-10 Cross-Over
LEX10CX2	ICD-10 Cross-Over - Source
LEX10CX3	ICD-10 Cross-Over - Target
LEX10CX4	ICD-10 Cross-Over - Prompts
LEX10CX5	ICD-10 Cross-Over - Miscellaneous
LEX10DBC	ICD-10 Diagnosis Lookup by Code
LEX10DBR	ICD-10 Diagnosis Lookup by Root
LEX10DBT	ICD-10 Diagnosis Lookup by Text
LEX10DL	ICD-10 Diagnosis Prototype Lookup
LEX10DLS	ICD-10 Diagnosis Prototype Selection
LEX10DU	ICD-10 Diagnosis Utility
LEX10DX	ICD-10 Diagnosis
LEX10PL	ICD-10 Procedure Prototype Lookup
LEX10PLA	ICD-10 Procedure Prototype Selection
LEX10PLS	ICD-10 Procedure Prototype Selection
LEX10PR	ICD-10 Procedures
LEX10TAX	Post-ICD-10 Taxonomies

12.3 Miscellaneous

12.3.1 Routines

LEXCODE	Convert Code to IEN
LEXSRC	Convert IEN to Code
LEXSRC2	Check Status of a Code
LEXTRAN	Retrieve Data for Specific Codes or Texts
LEXTRAN1	Retrieve Designations and Mappings for Specific Codes
LEXTRAN3	Mapping/Subset Updates

LEXTOKN	Parse Text into Words
LEXTOKN2	Special Case Words
LEXPRT	Print Utilities
LEXHLP	Help
LEXHLP2	Coding System Specific Help
LEXU	Miscellaneous Utilities
LEXU2	Miscellaneous Utilities (continued)
LEXU3	Miscellaneous Utilities (continued)
LEXU4	Miscellaneous Utilities (continued)
LEXUH	Miscellaneous Utilities (Help)
LEXLK	Demo Lookup
LEXLK2	Demo Lookup
LEXXM	Mixed Case
LEXXM1	Mixed Case 1 character
LEXXM2	Mixed Case 2 characters
LEXXM3	Mixed Case 3 characters
LEXXM4	Mixed Case 4 characters
LEXXM5	Mixed Case 5 characters
LEXXM6	Mixed Case 6-12 characters
LEXXMM	Mixed Case Miscellaneous

13. Package Characteristics and Usage

13.1 Supported Callable Routines

The following routines are supported:

13.1.1 LEXSET

CONFIG^LEXSET(<App>,<Subset>,<Date>)

ICR 1609

This entry point sets up the lookup variables for searching the Lexicon. It is not necessary to use this entry point for either Special Lookup or Silent Lookup since this entry point is embedded in Silent Lookup. You should use this entry point when:

- A search is to be conducted using the Kernel Toolkit's Multi-Term Lookup Utility (MTLU).
- It is desirable for an application to control the user defaults for a given situation (for example, the application may require the return of an ICD code).

This entry point searches the Subset Definition file (#757.2) and retrieves the application defaults, the subset defaults, and user defaults. Then it merges the three sets of defaults into a single list of defaults based on the information it has retrieved. For instance, if the application has defined the Overwrite flag as true, then the application defaults have precedence over any user defaults found and the user defaults are ignored. If the global root is anything other than ^LEX(757.01, then the user defaults for vocabulary and filter are ignored while the user defaults for display and shortcuts are used.

Input

This is the application identification and may be in the form of a name (e.g., PROBLEM LIST), a namespace (e.g., GMPL) or a pointer (e.g., Internal Entry Number—IEN) from an application definition in the Subset Definition file (#757.2). The default value for this parameter, if it is not supplied, is one (1), pointing to the Lexicon application definition. This is the same as the Application input parameter for LOOK^LEXA. A list of appropriate application identifiers is found in the Subset Definition file #757.2 in the AN index. At the time of this writing it included:

Application ID	Application or Purpose
CPT	CPT-4 Procedures
CHP	CPT-4 and HCPCS
DSM	Mental Health DSM-4
GMPL	Problem List
ICD	ICD-9-CM Diagnosis
10D	ICD-10 Diagnosis
10P	ICD-10 Procedures
LEX	Generic Lexicon Utility
PSN	Pharmacy (drug/form)
RA	Brest Imaging Rpt Data Sys BI-RADS
VAC	VA Frequently used Codes

This parameter represents the vocabulary subset to use during the search. This parameter is passed as a subset name (e.g., NURSING), or the subset mnemonic (e.g., NUR) or as a pointer to the Subset Definition file (#757.2). The default value for this parameter, if it is not supplied, is one (1) pointing to the main vocabulary (WRD subset) of the Lexicon located in file 757.01 and indexed by AWRD. This is the same as the Subset input parameter for LOOK^LEXA. A list of appropriate vocabulary subsets is found in the Subset Definition file #757.2 in the AA and AB indexes. At the time of this writing it included:

Subset ID	Purpose
DEN	Dental Terminology
IMM	Immunologic Terminology
NUR	Nursing Terminology
SOC	Social Work Terminology
WRD	General Use (default)
CPT	CPT Procedures

Subset ID	Purpose
DSM	Mental Health DSM-IV
ICD	ICD-9-CM Diagnosis
10D	ICD-10-CM Diagnosis
10P	ICD-10-PCS Procedures
NIC	Nursing Interventions
PL1	Problem List #1 General
PL2	Problem List #2 ICD Diagnosis and Procedures
VAC	VA Frequently used codes (ICD, ICP, DSM, CPT)

This is a date in FileMan format used to return the code that is active on the date supplied. If the date is not passed, then TODAY is used.

Output

If the variable LEXQ does not exist or is preset to 1, then the merged set of default values are placed in the global array ^TMP("LEXSCH",\$J) as in the case of a standard DIC lookup or Silent Lookup. The following is a brief summary of the global array:

Global Array Segment	Purpose
^TMP("LEXSCH",\$J,"ADF",0)	Application Default Flag
^TMP("LEXSCH",\$J,"APP",0)	Application
^TMP("LEXSCH",\$J,"DIS",0)	Display
^TMP("LEXSCH",\$J,"FIL",0)	Filter
^TMP("LEXSCH",\$J,"FLN",0)	File Number
^TMP("LEXSCH",\$J,"GBL",0)	Global (DIC)
^TMP("LEXSCH",\$J,"IDX",0)	Index
^TMP("LEXSCH",\$J,"LEN",0)	List Length
^TMP("LEXSCH",\$J,"OVR",0)	Overwrite User Default Flag
^TMP("LEXSCH",\$J,"SCT",0)	Shortcuts
^TMP("LEXSCH",\$J,"SVC",0)	Service
^TMP("LEXSCH",\$J,"UNR",0)	Unresolved Narrative Flag
^TMP("LEXSCH",\$J,"USR",0)	User (DUZ)
^TMP("LEXSCH",\$J,"VOC",0)	Vocabulary

If the variable LEXQ=0, then you should define the variable X as the user input and the merged set of default values are set into the appropriate local variables for making a direct call to the MTLU via the entry point ^XTLKKWL. The following variables are returned:

Variable	Content
DIC	Global Reference (root)
DIC ("S")	Search Filter (MUMPS code)
DIC (0)	Search Conditions (codes)
LEXAP	Application (pointer to file 757.2)
LEXQ	Silent lookup flag (codes)
LEXSHOW	Displayable Codes (free text)
LEXSUB	Subset (free text)
LEXUN	Unresolved Narratives (codes)
X	User input (free text)
XTLKGBL	Global Reference (root)
XTKLHLP	Help (MUMPS code)
XTLKKSCH("DSPLY")	Display (routine entry point)
XTLKKSCH("GBL")	Global Reference (root)
XTLKKSCH("INDEX")	Index to use ("A"_Subset)
XTLKSAY	MTLU display (codes)
XTLKX	User input (free text, same as X)

LEXU

\$\$ICDONE^LEXU(<IEN>,<Date>)	ICR 1573
\$\$CPTONE^LEXU(<IEN>,<Date>)	ICR 1573
\$\$DSMONE^LEXU(<IEN>,<Date>)	ICR 1573
\$\$D10ONE^LEXU(<IEN>,<Date>)	ICR 5679
\$\$P10ONE^LEXU(<IEN>,<Date>)	ICR 5679

These entry points (extrinsic functions) allow an application to retrieve a single code for a given classification system (ICD-9, CPT-4, and DSM-IV) and for a given internal entry number (IEN).

Input

<IEN>	This is an Internal Entry Number from the Lexicon Expression file #757.01.
<Date>	This is a date in FileMan format used to return the code that is active on the date supplied. If the date is not passed, then TODAY is used.

Output

Code	A single classification code, if one is found, or null, if no code is found.
\$\$ICDONE	One ICD-9 Diagnosis Code
\$\$CPTONE	One CPT Procedure Code
\$\$CPCONE	One HCPCS Code
\$\$D10ONE	One ICD-10 Diagnosis Code
\$\$P10ONE	One ICD-10 Procedure Code
\$\$DSMONE	One DSM-IV Diagnosis Code

\$\$DX^LEXU(<IEN>,<Date>) **ICR 5679**

This entry point is to be used as a screen Lexicon searches. It will screen out all terms not linked to either an ICD-9 or ICD-10 code. The code type (ICD-9 or ICD-10) is determined by date. If the date passed in is before the ICD-10 implementation date then it will screen on ICD-9 codes. If the date is on or after the ICD-10 implementation date then it will screen on ICD-10 codes.

Input

<IEN>	This is an internal entry number in the expression file #757.01. When performing FileMan lookups, set it to the variable +Y. (Required)
<Date>	This is the versioning date against which the codes found by the search will be compared in order to determine whether the code is active or inactive. Additionally, if the date passed is earlier than the ICD-10 implementation date then the screen will only consider ICD-9 codes. If the date is on or after the ICD-10 implementation date then the screen will only consider ICD-10 codes. If the date is not passed, then TODAY's date will be used. (Optional)

Assuming the variable <Date> is a valid FileMan format date:

Screen on ICD Diagnosis

S DIC("S")="I \$\$DX^LEXU(+Y,<Date>)"

<Date> is before the ICD-10 implementation date then screen on ICD-9 Diagnosis

<Date> is on or after the ICD-10 implementation date, then screen on ICD-10 Diagnosis

If the date is not passed, then TODAY is used.

Output

\$\$DX This is a Boolean value:

\$\$DX = TRUE If the Lexicon entry is linked to an active ICD
1 code of the type determined by date.

\$\$DX = FALSE If the Lexicon entry is not linked to an active ICD
0 code of the type determined by date.

\$\$ONE^LEXU(<IEN>,<Date>,<SAB>)

ICR 5679

Returns a single code for a given internal entry number (IEN) for a specified date and source (coding system).

Input

<IEN> Internal Entry Number in the Expression file ^LEX(757.01).

<Date> This is a date in FileMan format used to check if a code is active or inactive on a specified date. If not supplied, it will default to TODAY.

<SAB> Source, this is an internal entry number in file #757.03 or the 3 character source mnemonic (found on the ASAB cross-reference in file #757.03) or the SOURCE ABBREVIATION (.01 field in file #757.03).

Output

\$\$ONE A single code belonging to the specified coding system by the source abbreviation that is active on the date provided and assigned to the expression indicated by the internal entry number (IEN).

\$\$ALL^LEXU(<IEN>,<Date>,<SAB>) ICR 5679

Returns all codes for a given internal entry number (IEN) for a specified date and source (coding system).

Input

<IEN> Internal Entry Number in the Expression file #757.01.

<Date> This is a date in FileMan format used to check if a code is active or inactive on a specified date. If not supplied, it will default to TODAY.

<SAB> Source, this is an internal entry number in file #757.03 or the 3 character source mnemonic (found on the ASAB cross-reference in file #757.03) or the SOURCE ABBREVIATION (.01 field in file #757.03).

Output

\$\$ALL A string of codes for the source provided (one or more) delineated by a semi-colon or null if no codes are found for the source.

\$\$ICD^LEXU(<IEN>,<Date>) ICR 1573
\$\$D10^LEXU(<IEN>,<Date>) ICR 5679

This entry point (extrinsic function) allows an application to retrieve a series of ICD codes for a given internal entry number (IEN).

Input

<IEN> This is an Internal Entry Number from the Lexicon Expression file #757.01.

Output

Codes A string of ICD codes (one or more) delineated by a semi-colon or null if no codes are found.

Example

If **Y=33677^Diabetic Neuropathies** then the call **\$\$ICD^LEXU(+Y)** returns the string **250.60;357.2**, containing the ICD code for Diabetes with Neurological Manifestations and the ICD code for Polyneuropathy in Diabetes.

\$\$IMPDATE^LEXU(<SAB>) ICR 5679

This entry point (extrinsic function) returns the implementation date for a specified source.

Input

<SAB> Source, this is an internal entry number in file #757.03 or the 3 character source mnemonic (found on the ASAB cross-reference in file #757.03) or the SOURCE ABBREVIATION (.01 field in file #757.03).

Output

\$\$IMPDATE The date that a coding system was implemented in VistA in FileMan format.

\$\$CSYS^LEXU(<Sys>,<Date>)

ICR 5679

This entry point returns information about a coding system on file in the Coding System file #757.03.

Input

<Sys> Coding system identification system and can be in any of the following formats:

A nickname if one exist, i.e. HCPCS, DSM, NANDA, BIRADS

First three characters of source abbreviation (file #757.03, field .01)

Source Abbreviation (file #757.03, field .01), i.e., ICD, CPT, SCT

Nomenclature (file #757.03, field 1), i.e., ICD-9-CM, ICD-10-PCS

Type (only for ICD), i.e., "DIAG" or "PROC" (requires date)

<Date> Versioning date in FileMan format used to determine coding system if only the type is known (DIAG or PROC) and to determine the correct SDO version if one exists. If the date is not passed, then TODAY is used.

Output

\$\$CSYS A 13 piece caret (^) delimited string

- 1 IEN
- 2 SAB (3 character source abbreviation)
- 3 Source Abbreviation (3-7 char) (#.01)
- 4 Nomenclature (2-11 char) (#1)
- 5 Source Title (2-52 char) (#2)
- 6 Source (2-50 char) (#3)
- 7 Entries (numeric) (#4)
- 8 Unique Entries (numeric) (#5)
- 9 Inactive Version (1-20 char) (#6)
- 10 HL7 Coding System (2-40 char) (#7)
- 11 SDO Version Date (date) (757.08 #.01)
- 12 SDO Version Id (1-40 char) (757.08 #1)
- 13 Implementation Date (date) (#11)

\$\$HIST^LEXU(<Code>,<Sys>,.ARY)

ICR 5679

This entry point returns a code's activation history in an array passed by reference.

Input

<Code> This is a classification code found in the CODES file #757.02.
(Required)

<Sys> This is a coding system found in the CODING SYSTEMS file
#757.03. It can be in the form of a pointer, a source abbreviation, or
the name of a coding nomenclature. (Required)

Input/Output

.ARY This is an array of status effective dates and activation status passed
by reference. (Required)

ARY(0) = Number of Activation History

ARY(<date>,<status>) = Comment

Status

0 = Inactive

1 = Active

Comments include:

Activated

Inactivated

Re-activated

Revised

Reused

Output

\$\$HIST This is the number of activation history entries found
Or
-1 ^ Error Message

\$\$PERIOD^LEXU(<Code>,<Sys>,.ARY)

ICR 5679

This entry point returns the activation periods (active from and to) of a code in an array passed by reference.

Input

- <Code> This is a classification code found in the CODES file #757.02. (Required)
- <Sys> This is a coding system found in the CODING SYSTEMS file #757.03. It can be in the form of a pointer, a source abbreviation, or the name of a coding nomenclature. (Required)

Input/Output

.ARY This is an array of activation periods (including active on date and inactive on date when inactive) passed by reference. (Required)

ARY(0) 6 piece "^" delimited string

- 1 Number of Activation Periods found
- 2 Coding System (pointer to file 775.03)
- 3 Coding System Abbreviation
- 4 Coding System Nomenclature
- 5 Coding System Full Name
- 6 Coding System Source

or

#NAME?

ARY(<Activation Date>) = 4 piece "^" delimited string

- 1 Inactivation Date (conditional)
- 2 Pointer to Expression file #757.01 for the code in piece #2 above (required)
- 3 Variable Pointer IEN;Root of a national file (see below) Include when the code exist in an national file (conditional)

CPT Procedure code

IEN;ICPT(

ICD Diagnosis code	IEN;ICD9(
ICD Procedure code	IEN;ICD0(

4 Short Description from the SDO file (CPT or ICD)

ARY(<Activation Date>,0) = Lexicon Expression

Output

\$\$PERIOD This is the number of activation periods found:

Same as output variable ARY(0)

or

-1 ^ error message

\$\$NXSAB^LEXU(<SAB>,<Rev>)

ICR 5679

This API returns the next Source Abbreviation found in the CODING SYSTEMS file 757.03 using the ASAB cross-reference. It is the equivalent of \$O(^LEX(757.03,"ASAB",SAB)).

Input

<SAB> This is either a Source Abbreviation (SAB) from the .01 field of file 757.03 or null value to find the first SAB.

<Rev> This is a reverse flag (optional). If set to 1 the API will find the next Source Abbreviation in the reverse order (aka, previous SAB).

Output

\$\$NXSAB This is either the next Source Abbreviation (SAB) previous SAB (when reverse flag set to 1) or null if the input parameter SAB has no next SAB.

\$\$CSDATA^LEXU(<Code>,<Sys>,<Date>,.ARY)

ICR 5679

This entry point returns information about a code from a specified coding system. It is intended to be similar to ICDDATA^ICDXCODE except it is not limited to ICD coding systems.

Input

<Code>	This is a code found in file #757.02 (CODES file).
<Sys>	This is a pointer to the CODING SYSTEMS file #757.03 that identifies the coding system that CODE belongs to. It is important to specify the coding system because some codes overlap various coding systems.
<Date>	This is the date that will be used to determine the status of the code in the CODE input parameter. The status will either be Inactive or Active.

Input/Output

.ARY	This is the name of a local array passed by reference that will contain the output.
------	---

ARY()

Lexicon Data

ARY("LEX",1)	IEN ^ Preferred Term
ARY("LEX",2)	Status ^ Effective Date
ARY("LEX",3)	IEN ^ Major Concept Term
ARY("LEX",4)	IEN ^ Fully Specified Name
ARY("LEX",5)	Hierarchy (if it exists)
ARY("LEX",6,0)	Synonyms/Other Forms
ARY("LEX",6,1)	Synonym #1
ARY("LEX",6,n)	#n
ARY("LEX",7,0)	Semantic Map
ARY("LEX",7,1,1)	Class ^ Type (internal)
ARY("LEX",7,1,2)	Class ^ Type (external)
ARY("LEX",7,1,n)	#n
ARY("LEX",7,1,n)	#n
ARY("LEX",8)	Deactivated Concept Flag

Coding System Data

ARY("SYS",1)	IEN
ARY("SYS",2)	Short Name
ARY("SYS",3)	Age High
ARY("SYS",4)	Age Low
ARY("SYS",5)	Sex
ARY("SYS",6,0)	MDC/DRG Pairing
ARY("SYS",6,1,1)	MDC
ARY("SYS",6,1,2)	DRGs
ARY("SYS",6,n,1)	#n
ARY("SYS",6,n,2)	#n
ARY("SYS",7)	Complication/Comorbidity
ARY("SYS",8)	MDC13
ARY("SYS",9)	MDC24
ARY("SYS",10)	MDC24
ARY("SYS",11)	Unacceptable as Principal Dx
ARY("SYS",12)	Major O.R. Procedure
ARY("SYS",13)	Procedure Category
ARY("SYS",14,0)	Description
ARY("SYS",14,1)	Text 1
ARY("SYS",14,n)	#n

Each data element will be in the following format:

ARY(ID,SUB) = DATA
ARY(ID,SUB,"N") = NAME

Where

ID Identifier, may be:

LEX for Lexicon data

SYS for Coding System data

SUB Numeric Subscript

DATA This may be:

A value if it applies and is found

Null if it applies but not found

N/A if it does not apply

NAME This is the common name given to the data element

Example:

```
S X=$$CSDATA^LEXU("C18.6",30,3141010,.ARY)
```

```
X=1
```

```
ARY("LEX",1)="267081^Malignant neoplasm of descending colon"
```

```
ARY("LEX",1,"N")="IEN ^ Preferred Term"
```

```
ARY("LEX",2)="1^3131001"
```

```
ARY("LEX",2,"N")="Status ^ Effective Date"
```

```
ARY("LEX",3)="267081^Malignant neoplasm of descending colon"
```

```
ARY("LEX",3,"N")="IEN ^ Major Concept Term"
```

```
ARY("LEX",4)=""
```

```
ARY("LEX",4,"N")="IEN ^ Fully Specified Name"
```

```
ARY("LEX",5)=""
```

```
ARY("LEX",5,"N")="Hierarchy (if exists)"
```

```
ARY("LEX",6,0)=0
```

```
ARY("LEX",6,0,"N")="Synonyms and Other Forms"
```

```
ARY("LEX",7,0)=1
```

```
ARY("LEX",7,0,"N")="Semantic Map"
```

```
ARY("LEX",7,1,1)="6^47"
```

```
ARY("LEX",7,1,1,"N")="Semantic Class ^ Semantic Type (internal)"
```

```
ARY("LEX",7,1,2)="Diseases/Pathologic Processes^Disease or Syndrome"
```

```
ARY("LEX",7,1,2,"N")="Semantic Class ^ Semantic Type (external)"
```

```
ARY("LEX",8)=""
```

```
ARY("LEX",8,"N")="Deactivated Concept Flag"
```

```

ARY("SYS",1)=501148
ARY("SYS",1,"N")="IEN"
ARY("SYS",2)="Malignant neoplasm of descending colon"
ARY("SYS",2,"N")="Short Name"
ARY("SYS",3)="
ARY("SYS",3,"N")="Age High"
ARY("SYS",4)="
ARY("SYS",4,"N")="Age Low"
ARY("SYS",5)="
ARY("SYS",5,"N")="Sex"
ARY("SYS",6,0)=0
ARY("SYS",6,0,"N")="MDC/DRG"
ARY("SYS",7)="
ARY("SYS",7,"N")="Complication/Comorbidity"
ARY("SYS",8)="
ARY("SYS",8,"N")="MDC13"
ARY("SYS",9)="
ARY("SYS",9,"N")="MDC24"
ARY("SYS",10)="
ARY("SYS",10,"N")="MDC24"
ARY("SYS",11)="
ARY("SYS",11,"N")="Unacceptable as Principal Dx"
ARY("SYS",12)="N/A"
ARY("SYS",13)="N/A"
ARY("SYS",14,0)=1
ARY("SYS",14,0,"N")="Description"
ARY("SYS",14,1)="MALIGNANT NEOPLASM OF DESCENDING
COLON"

```

Output

```

$$CSDATA      This is a boolean value:
               $$CSDATA = 1   TRUE   If the API is successful (fully or partial)
               $$CSDATA = 0   FALSE  If the API is unsuccessful
               or
               -1 ^ error message

```

\$\$MAX^LEXU(<Sys>)

ICR 5679

Input

<Sys> (Required) This is a pointer to the CODING SYSTEM file #757.03.

Output

\$\$MAX This is the value stored in the SEARCH THRESHOLD field #12 of the CODING SYSTEMS file 757.03. This value, along with the value of \$\$FREQ^LEXU, can be used to evaluate if a search should continue or be further refined.

\$\$FREQ The maximum number or records to inspect during a search based on the input text string.

\$\$MAX The maximum number of records to consider for a coding system before refining the search.

\$\$FREQ^LEXU(<Text>)

ICR 5679

Input

<Text> (Required) This is a text string intended as the input for a Lexicon search.

Output

\$\$FREQ This is the maximum number of records that must be inspected during a Lexicon search to find matching entries for the input search text.

If this number is too high, applications should consider re-prompting the user to either continue with the search or to further refine the search.

\$\$PAR^LEXU(<Text>,.ARY)

ICR 5679

Input

<Text> (Required) This is a text string intended as the input for a Lexicon search and will be parsed into words and placed in a local array.

.ARY Local array, passed by reference

Output

\$\$PAR This is the number of words parsed from the text.

ARY This is a local array containing the words parsed from the input text. The words are arranged in the order they are found in the text; in alphabetical order; and in the order they are used in the Lexicon search (frequency order)

Total words found

ARY(0)=#

Words listed in the order they appear in the input variable

```
ARY ( 1 ) =WORD1  
ARY ( n ) =WORDn
```

Words listed alphabetically with the frequency of occurrence

```
ARY ( "B" ,WORDA) =# (Frequency of Use)  
ARY ( "B" ,WORDB) =#
```

Words listed in the frequency order (the order used by the search)

```
ARY ( "L" , 1 ) =SEARCHWORD1  
ARY ( "L" , n ) =SEARCHWORDn
```

Special Variables used by the parsing logic:

- | | |
|----------------|--|
| LEXIDX | If this variable is set, the text will use the parsing logic used for setting cross-references. This is the default method. |
| LEXLOOK | If this variable is set, the text will use the parsing logic used for setup up for a Lexicon search (lookup). |

\$\$CAT^LEXU(<Code>)

ICR 5679

This entry point returns the category for a ICD-10 Diagnosis code.

Input

<Code> This is an ICD-10 Diagnosis code or sub-category.

Output

\$\$CAT This is the category to which the code or sub-category belongs.

\$\$RECENT(<SAB>)

ICR 5679

This API returns a boolean valued flag to indicate if the coding system identified by the source abbreviation has been recently updated with in a 90 day window (looking forward by 30 days and to the past by 60 days). This is to evaluate if a coding system was updated by a quarterly patch, and may be used to trigger an code set update protocol.

Input

<SAB> This is either a 3 character source abbreviation taken from the .01 field of the CODING SYSTEM file 757.03 or a pointer to the CODING SYSTEM file 757.03.

Output

\$\$RECENT This is a boolean valued flag

- 1 Indicates the coding system has been recently updated by a quarterly update by looking 30 days into the future and 60 days into the past for a changed made to the coding system
- 2 Indicates the coding system has NOT been recently updated by a quarterly patch

\$\$RUPD(SAB)

ICR 5679

This API returns a date the coding system identified by the source abbreviation has been updated based on a recent date (TODAY+3). This is to evaluate if a coding system was updated by a quarterly patch, and may be used to trigger an code set update protocol.

Input

<SAB> This is either a 3 character source abbreviation taken from the .01 field of the CODING SYSTEM file 757.03 or a pointer to the CODING SYSTEM file 757.03.

Output

\$\$RUPD This is date found for the last update to a coding system based on a recent date (TODAY+30)

\$\$LUPD(SAB,DATE)

ICR 5679

This API returns the last date the coding system identified by the source abbreviation has been updated based on the date supplied (optional). If no date is supplied, the last date will be returned.

Input

<SAB> This is either a 3 character source abbreviation taken from the .01 field of the CODING SYSTEM file 757.03 or a pointer to the CODING SYSTEM file 757.03.

<date> This is a date to use to retrieve the last update for a coding system (optional)

Output

\$\$LUPD This is date found for the last update to a coding system based on a recent passed or the last date updated if a date is not passed.

13.2 LEXCODE

EN^LEXCODE(<Code>,<Date>)

ICR 1614

This entry point allows an application to retrieve the internal entry numbers (IENs) and the text (as the FileMan Y variable) of the expressions associated with a classification code.

Input

<Code> (Required) Code taken from a classification system listed in Coding Systems file #757.03

<Date> (Optional) The date against which the codes found by the search will be compared in order to determine whether the code is active or inactive. If not passed, TODAY's date will be used.

Output

Local Array **LEXS**

LEXS(0)=Code
LEXS(SAB,0)=Number of Terms found for SAB
LEXS(SAB,0,"SAB")=Source Nomenclature ^ Name
LEXS(SAB,#)=IEN file 757.01^Display Text (term)
Example of returned array LEXS using code V62.4
LEXS(0)="V62.4"
LEXS("DS4",0)=1
LEXS("DS4",0,"SAB")="DSM-IV^Diagnostic & Statistical Manual of Mental Disorders"
LEXS("DS4",1)="303722^Acculturation Problem"
LEXS("ICD",0)=5
LEXS("ICD",0,"SAB")="ICD-9-CM^International Classification of Diseases, Diagnosis"
LEXS("ICD",1)="111638^Social maladjustment"
LEXS("ICD",2)="29696^Cultural Deprivation"
LEXS("ICD",3)="100676^Psychosocial Deprivation"
LEXS("ICD",4)="303722^Acculturation Problem"
LEXS("ICD",5)="111507^Social Behavior"

EXP^LEXCODE(<Code>,<Source>,<Date>)

ICR 5680

This entry point allows an application to retrieve an active preferred term for a coding system on the date provided.

Input

<Code> (Required) Code taken from the Codes file #757.02.

<Source> (Required) This is either the three character Source Abbreviation or a pointer to the Coding Systems file #757.03. Source abbreviations (SAB) may be found in Appendix A or the "ASAB" cross-reference if the Coding Systems file #757.03. It is used to distinguish between different coding systems with the same code (i.e., the code 300.01

occurs in both the ICD-9-CM and DSM-IV coding systems).

<Date>

(Optional) The date against which the codes found by the search will be compared in order to determine whether the code is active or inactive. If not passed, TODAY's date will be used.

Output

\$\$EXP

2 Piece "^" delimited string containing

Either:

- 1 Pointer to Expression file #757.01
- 2 Display Text (Expression)

or:

- 1 -1 Error Message

13.3 LEX10CS (ICD-10 Specific)

\$\$ICDSRCH^LEX10CS(<Text>,.ARY,<Date>,<Len>,<Fil>)

ICR 5681

This entry point searches for an ICD code and returns active ICD codes found up to the number defined by the input parameter Length. If the date is before the ICD-10 implementation date then the search will be conducted for ICD-9 codes. If the date passed is on or after the ICD-10 implementation date then the search will be conducted for ICD-10 codes.

Input

<Text>

Text or Code to search for. (Required)

.ARY

This is a local output array passed by reference. (Required)

<Date>

The date against which the codes found by the search will be compared in order to determine whether the code is active or inactive. If not passed, TODAY's date will be used. (Optional)

<Len>

This specifies the length of the list of codes. Default value is 30. (Optional)

<Fil>

This is a filter to apply to the search to screen out unwanted entries. It is MUMPS code in the form of a valid IF statement. It is the same as FileMan's DIC("S"). (Optional)

Output

\$\$ICDSRCH

2 Piece "^" delimited string the success/error conditions

A Positive number for successful search
not exceeding the Length of the list.

A Negative number for an unsuccessful search
or a search condition

-1^No codes found

No codes found and local array not returned

-2^Too many items found, please refine search

The list exceeds the number indicated by the list length.
However, those found up to the list length will be returned in
the array and the list will be marked as a pruned list.

ARY Output Array passed by reference containing the codes found

ARY(0)=# found ^ Pruning Indicator

ARY(1)=CODE ^ status effective date

ARY(1,"IDL")=ICD Dx long description (if code)

ARY(1,"IDL",1)=ICD Dx IEN ^ effective date

ARY(1,"IDS")=ICD Dx short description (if code)

ARY(1,"IDS",1)=ICD Dx IEN ^ effective date

ARY(1,"LEX")=Lexicon expression

ARY(1,"LEX",1)=Lexicon IEN ^ effective date

ARY(1,"SYN",1)=synonym #1

ARY(1,"SYN",m)=Synonym #m

Pruning Indicator: If the second piece of ARY(0) is greater than 0,
then the list has been pruned, limiting the list to the length specified
by the input parameter <Len>.

\$\$DIAGSRCH^LEX10CS(<Text>,.ARY,<Date>,<Len>,<Fil>) ICR 5681

This entry point searches for an ICD code and returns active ICD-10 codes found up to the number defined by the input parameter Length. This search is similar to \$\$ICDSRCH^LEX10CS except it searches only ICD-10 codes.

Input

<Text>	Text or Code to search for. It is the same as FileMan's X. (Required)
.ARY	This is a local output array passed by reference. (Required)
<Date>	The date against which the codes found by the search will be compared in order to determine whether the code is active or inactive. If not passed, TODAY's date will be used. (Optional)
<Len>	This specifies the length of the list of codes. (Optional)
<Fil>	This is a filter to apply to the search to screen out unwanted entries. It is MUMPS code in the form of a valid IF statement. It is the same as FileMan's DIC("S"). (Optional)

Output

\$\$DIAGSRCH	2 Piece "^" delimited string the success/error conditions
	A Positive number for successful search not exceeding the Length of the list.
	A Negative number for an unsuccessful search or a search condition
	-1^No codes found
	No codes found and local array not returned
	-2^Too many items found, please refine search
	The list exceeds the number indicated by list length. However, those found up to the list length will be returned in the array and the list will be marked as a pruned list.
ARY	Output Array passed by reference containing the codes found

ARY(0)=# found ^ Pruning Indicator
ARY(1)=CODE or Category ^ code status effective date
ARY(1,"CAT")=Category Name
ARY(1,"IDL")=ICD Dx long description (if code)
ARY(1,"IDL",1)=ICD Dx IEN ^ effective date
ARY(1,"IDS")=ICD Dx short description (if code)
ARY(1,"IDS",1)=ICD Dx IEN ^ effective date
ARY(1,"LEX")=Lexicon expression
ARY(1,"LEX",1)=Lexicon IEN ^ effective date
ARY(1,"SYN",1)=synonym #1
ARY(1,"SYN",m)=Synonym #m

Pruning Indicator: If the second piece of ARY(0) is greater than 0, then the list has been pruned, limiting the list to the length specified by the input parameter <Len>.

14. Pruning the Output

The DIAGSRCH API builds an array of terms linked to ICD-10 codes. This API employs a two-staged search as follows:

Stage 1: Initial Search

It first checks to see if the input text string is a code or partial code. If it is, then a "lookup by code" begins and all codes that equal or begin with the input text are placed in the array. If the input text is not a code, then a "lookup by text" begins and all terms found that are linked to an ICD-10 code are placed in the array.

Stage 2: Search Pruning

If the search list does not exceed a predefined number of matches (default if unspecified to be 30), then the list will be passed back to the calling application. If the search list exceeds a predefined number of matches, then the list will be pruned using ICD-10-CM categories/sub-categories.

The rightmost character of each code in the original list will be removed. If the resulting text is either a diagnosis category or a sub-category then the category or sub-category will be added to the list and the code will be removed (category replaces code on the list). If there is no category or sub-category the code will remain on the list.

The list length will be checked again. If the new list length is less than the limit then the list will be passed back to the calling application. If the new list length is not less than the limit then the pruning continues, character by character, until the number of items on the list falls below the limit.

For example, suppose the limit is set to 6 and the search returns the following codes:

A12	A12.0	A12.45	A12.46	A13.49	E13
E13.31	E14.45	E14.567	S34.203	S34.204	S34.205

The search returned 12 codes and exceeds the limit of 6, so search results will be pruned, returning the following codes:

A12	A12.4	A13.4	E13	E13.3	E14.4
E14.5	E14.56	S34.20			

The pruned search returns 9 codes which still exceeds the limit of 6, so the pruning operation will be repeated until the limit does not exceed 6:

A12	A13	E13	E14	E14.5	S34.2
-----	-----	-----	-----	-------	-------

The additional pruning returns search results that is now equal to or less than the limit of 6. These categories/codes are placed in the array and returned to the calling application to present to the user. The calling application will issue all prompts and help text for user interaction.

\$\$PCSDIG^LEX10CS(<Frag>,<Date>)

ICR 5681

This entry point takes an ICD-10-PCS code, full or a partial (code fragment), and returns a list of all possibilities for the next character, with any definitions and examples available. If a full code is passed (7 characters), it will return the code's long description and status.

Input

- <Frag> This is an ICD-10-PCS Code (7 characters) or a fragment of an ICD-10-PCS Code (less than 7 characters). (Required)
- <Date> The date against which the codes found by the search will be compared in order to determine whether the code is active or inactive. If not passed, TODAY's date will be used. (Optional)

Output

`$$PCSDIG` This is a boolean value:

`$$PCSDIG = 1` When the input code fragment is valid or null

`$$PCSDIG = 0` When the input code fragment is invalid

`LEXPCDAT` Output local array containing characters in the next position and the character descriptions.

If the input parameter `<Frag>` is a valid code fragment or null, the return value of `LEXPCDAT` will be 1 and the following array will be returned.

`LEXPCDAT("NEXLEV ,char1, DESC)=char1 description`

`LEXPCDAT("NEXLEV ,char2, DESC)=char2 description`

`LEXPCDAT("NEXLEV ,charn, DESC)=charn description`

If the input parameter `<Frag>` is a valid code, the return value of `LEXPCDAT` will be 1 and the following array will be returned.

`LEXPCDAT("PCSDESC)=long description for code`

`LEXPCDAT("STATUS)=status_ ^ _effective date`

If the input parameter `<Frag>` is not a valid code fragment or null and it is not a valid code, the return value of `LEXPCDAT` will be 0 and no array will be returned.

`$$CODELIST^LEX10CS(<Sys>,<Spec>,.ARY,<Date>,<Len>,<Fmt>)` ICR 5681

This entry point creates a list of active codes based on an input code mask and date and places the list in an array specified by the calling application.

Input

`<Sys>` Coding system from the Coding Systems file #757.03. This can be a pointer, the .01 field or the abbreviated 3 character mnemonic (found on the ASAB cross-reference). (Required)

`<Spec>` This is a code from the coding system or a code mask. Any character position can be occupied by a question mark "?" to allow any value in that character position. The trailing character may be an asterisk indicating any characters that follow are allowable. The following are all valid; C71.0, C71.*, C7?.0, or 02V?0*. (Required)

.ARY	This is a local output array passed by reference. (Required)
<Date>	The date against which the codes found by the search will be compared in order to determine whether the code is active or inactive. If not passed, TODAY's date will be used. (Optional)
<Len>	This specifies the length of the list of codes. Default value is 30. (Optional)
<Fmt>	List Format. A value of 1 returns a detailed listing in the array, includes the code, code IEN in file #757.02, the code s effective date, the expression, and the expression IEN in file #757.01. A value of 0 (zero) returns a brief listing in the array (codes only). Default value is 0 (zero). (Optional)

Output

\$\$CODELIST 2 Piece "^" delimited string containing

Either:

Piece	Meaning
1	Positive value for success
2	Number of Codes Found

or:

Piece	Meaning
1	Negative number for error or condition
2	Error Message or Condition

Example errors/conditions

- 1 Coding system not specified (First parameter is missing)
- 2 Invalid coding system/source abbreviation (First parameter not valid)
- 3 No search specification (Second parameter missing)
- 4 Insufficient search specification (Second parameter too short)
- 5 Invalid search specification (Second parameter invalid)
- 6 Number of matches exceeds specified limit (More matches found, only the number specified will be returned)

^TMP(SUB,\$J, This is a global array subscripted as specified by the calling application, input parameter SUB. It contains a list of codes found in either a brief or detailed output..

Brief output array (<Fmt> = 0)

^TMP(SUB,\$J,0)=Total n
^TMP(SUB,\$J,1)=Code 1
^TMP(SUB,\$J,2)=Code 2
^TMP(SUB,\$J,n)=Code n

Detailed output array (<Fmt> = 1)

^TMP(SUB,\$J,0)=Total n
^TMP(SUB,\$J,1)=Code 1
^TMP(SUB,\$J,1,1)=Variable Pointer 1 ^ Code 1 ^ Date
^TMP(SUB,\$J,1,2)=Expression 1 IEN ^ Expression 1
^TMP(SUB,\$J,2)=Code 2
^TMP(SUB,\$J,2,1)=Variable Pointer 2 ^ Code 2 ^ Date
^TMP(SUB,\$J,2,2)=Expression 2 IEN ^ Expression 2
^TMP(SUB,\$J,n)=Code n
^TMP(SUB,\$J,n,1)=Variable Pointer n ^ Code n ^ Date
^TMP(SUB,\$J,n,2)=Expression n IEN ^ Expression n

Note: If the code is found in one of the VistA Code Set files controlled by a Standards Development Organization (SDO), then a variable pointer will be provided for that code in that file. Example of SDO controlled files include:

ICD DIAGNOSIS file #80
ICD OPERATION/PROCEDURE file #80.1
CPT file #81
DSM file #627.7

\$\$TAX^LEX10CS(<Text>,<Src>,<Date>,<Sub>,<Ver>) ICR 5681

This entry point searches the input text and build an array of codes that qualify for a taxonomy.

Input

<Text> This is a text string, a code or a code fragment to search for. (Required)

<Src> This is a string of sources delimited by an “^” up arrow. The sources may be either a pointer to the CODING SYSTEM file 757.03, or a source abbreviation (found in the ASAB cross reference of file 757.03) (Required)

Using source pointers to file 757.03
"1^2^30^31"

Using source abbreviations

"ICD^ICP^10D^10P"

<Date> This a date used processing versioned data. Also, when a versioned list is required (input parameter VER=1) it is used to suppress inactive codes from the list. (Optional, default is TODAY)

<Sub> This is the name of a subscript to use in the ^TMP global (Optional). This allows for applications to put the data in their own namespace. It also allows for multiple search results to exist.

^TMP(LEXSUB,\$J,
^TMP("LEXTAX",\$J, Default

<Ver> This is a boolean flag that indicates if the search is to be versioned. If set to 1, only active codes will be returned based on the date in the <Date> input parameter. If no date, then TODAY is used. Default value is 0 (zero). (Optional)

0 Return active and inactive codes
1 Version, return active codes only

Output

\$\$TAX This the number of codes found by the search or -1 ^ with an error message.

^TMP(SUB,\$J, This is the results of the search saved in the ^TMP global with the specified subscript arranged by source:

^TMP(SUB,\$J,SRC,(CODE_ " "),#)

5 piece "^" delimited string

- 1 Activation Date (can be a future date)
- 2 Inactivation Date (can be a future date)
- 3 Lexicon IEN to Expression File 757.01
- 4 Variable Pointer to a National file
- 5 Short Name from a National file

^TMP(SUB,\$J,SRC,(CODE_ " "),#,0)

2 piece "^" delimited string

- 1 Code (no spaces)
- 2 Lexicon Expression

14.1 LEX10CX (ICD-10 Specific)

EN^LEX10CX

ICR 5840

This entry point is an interactive lookup where the input coding system and code are not known. There is no input variables for this API, however, the variable LEXSAB can be preset to a coding system (.01 field in file 757.03), else wise the user will be prompted for a coding system. This API will display a selection list of suggested ICD-10 codes that have a similar textual content as the user selected code and coding system. If no suggestions are available or the user does not make a selection, then the user will be prompted for a ICD-10 code.

Input

None

Output

X This is a 4 piece "^" delimited string representing the source code.

- 1 Lexicon IEN for file 757.02
- 2 Expression
- 3 Code in selected Coding System
- 4 Coding System nomenclature

or null if search fails

Examples:

X="119899^Tobacco Use Disorder^305.1^ICD-9-CM"

X="7078519^Diabetes mellitus type 2^44054006^SNOMED CT"

Y This is a 4 piece "^" delimited string representing the target ICD-10 code.

- 1 Lexicon IEN for file 757.02
- 2 Expression
- 3 ICD-10 Diagnostic Code
- 4 ICD-10-CM

or -1 if search fails

Examples:

Y="5003360^Nicotine Dependence, unspecified, Uncomplicated^F17.200^ICD-10-CM"

Y="5002666^Type 2 Diabetes Mellitus without Complications^E11.9^ICD-10-CM"

EN2^LEX10CX(<Code>,<SAB>)**ICR 5840**

This entry point is an interactive lookup where the input coding system and code are known and supplied as input parameters CODE and SAB. This API will display a selection list of suggested ICD-10 codes that have a similar textual content as the specified code (CODE) and coding system (SAB). If no suggestions are available or the user does not make a selection, then the user will be prompted for a ICD-10 code. The output for EN2 is the same as entry point EN.

Input

<Code>	This is a code from the specified coding system identified by the input parameter SAB.
<SAB>	This is the coding system abbreviation (a three character representation of the coding system taken from the .01 field of the CODING SYSTEMS file 757.03)

Output

Same as EN^LEX10CX

EN3^LEX10CX(<Code>,<SAB>,.ARY,<Max>)**ICR 5840**

This entry point is a silent lookup for suggested ICD-10 codes for a code in another coding system. The code (CODE) and coding system abbreviation (SAB) are passed as input parameters. This API will create an array of suggested ICD-10 codes that have a similar textual content as the specified code (CODE) and coding system (SAB)..

Input

<Code>	This is a code from the specified coding system identified by the input parameter SAB.
<SAB>	This is the coding system abbreviation (a three character representation of the coding system taken from the .01 field of the CODING SYSTEMS file 757.03)
.ARY	This is a local array, passed by reference. This API kill the array before it starts to populate it. (see output variable ARY)
<Max>	This is the maximum number of suggestions to return in the array (optional, default 100)

Output

ARY This is a local array, passed by reference:

```

ARY("X")  Input
ARY("Y",0) Output  Number of Suggestions
ARY("Y",1) Output  First Suggestion
ARY("Y",n) Output  nth Suggestion

ARY("E")  Error Message

```

Both ARY("X") and ARY("Y",#) are 4 piece "^"

delimited strings:

- 1 Internal Entry Number (IEN) file 757.01
- 2 Expression (file 757.01, field .01)
- 3 Code (file 757.02, field 1)
- 4 Nomenclature (file 757.03, field 1)
i.e., SNOMED CT, ICD-9-CM or ICD-10-CM

Example:

```
ARY("X")="331786^Diabetes with Ketoacidosis, type I [Juvenile type], Uncontrolled^  
250.13^ICD-9-CM"
```

```
ARY("Y",0)=3
```

```
ARY("Y",1)="5002587^Type 1 Diabetes Mellitus with Ketoacidosis without Coma^  
E10.10^ICD-10-CM"
```

```
ARY("Y",2)="5002588^Type 1 Diabetes Mellitus with Ketoacidosis with Coma^  
E10.11^ICD-10-CM"
```

```
ARY("Y",3)="5002623^Type 1 Diabetes Mellitus with Hyperglycemia^E10.65^  
ICD-10-CM"
```

14.2 LEXSRC2

\$\$STATCHK^LEXSRC2(<Code>,<Date>,.ARY,<Src>)

ICR 4083

This entry point allows an application to retrieve the status of a code (active or inactive) and the effective date of the status. An optional array may be included to provide additional information about the code.

Input

<Code>	(Required) Code taken from the Codes file 757.02 (ICD/CPT/DSM etc).
<Date>	(Optional) Date to screen against (default TODAY).
.ARY	(Optional) Output Array, passed by reference.
<Src>	(Optional) Source Abbreviation. Source abbreviations (SAB) may be found in Appendix A or the ASAB cross-reference if the Coding Systems file 757.03. It is used to distinguish between different coding systems with the same code.

Output

\$\$STATCHK 2 or 3 Piece String containing the code's status, the IEN, and if the status exist, the effective date, else -1 in lieu of the IEN.

The following are possible outputs:

1 ^ IEN ^ Effective Date	Active Code
0 ^ IEN ^ Effective Date	Inactive Code
0 ^ IEN	Not Active
0 ^ -1	Code not Found

.ARY (Optional) A local array (when passed by reference) containing the ASTM Triplet, the Major Concept Map and the Semantic Map.

ARY(0)	Code, a 2 Piece String containing: 1 IEN in the CODES file #757.02 2 A Code (external)
ARY(1)	Expression, a 2 Piece String containing: 1 IEN in the EXPRESSION file #757.01 2 The Code's Expression (external)Code is Inactive, and not found in the Lexicon
ARY(2)	Coding System, a 4 Piece String containing: 1 IEN in the CODING SYSTEMS file #757.03 2 Source Abbreviation (i.e., ICD or CPT) 3 Source Nomenclature (i.e., ICD-9-CM or CPT-4) 4 Source Full Name
ARY(3)	Major Concept, a 3 Piece String containing: 1 IEN in the MAJOR CONCEPT MAP file #757 2 IEN in the EXPRESSIONS file #757.01 3 The Major Concept expression, which may be different from the code's expression in ARY(1)
ARY(4,#)	Semantics (multiple), a 5 Piece String: 1 IEN in the SEMANTIC MAP file #757.1 2 IEN in the SEMANTIC CLASS file #757.11 3 IEN in the SEMANTIC TYPE file #757.12 4 External Semantic Class 5 External Semantic Type

14.3 LEXTRAN

\$\$CODE^LEXTRAN(<Code>, <Src>,<Date>,.ARY,<IENS>,<ID>,<INC>) ICR 4912

This API retrieves concept data for a given code and coding system.

Input

<Code>	This is a source code taken from one of the classification systems listed in Coding Systems file #757.03. e.g., ICD, CPT, DSM, NANDA, etc.
<Src>	This is a coding system identifier that identifies one of the coding systems listed in Coding Systems file #757.03. e.g., ICD, CPT, DSM, NANDA, etc.
<Date>	This is a date in FileMan format used to ensure that the expressions returned are for active codes on the date supplied. If the date is not passed, then TODAY is used.
<ARY>	This is the output array (defaults to LEX if none specified).

<IENS>	Include expression IENS in output array (optional) 1 return IENS (2nd piece) 0 do not return IENS (default)
<ID>	Designation Identifiers (optional) 1 return Designation IDs (3rd piece) 0 do not return Designation IDs (default)
<INC>	Include Deactivated Terms (optional) 1 return Deactivated Terms 0 do not return Deactivated Terms (default)

Output

\$\$CODE	<p>If API finds an active code for the source</p> <p>"1^CODE"</p> <p>ARY - an array containing information about the code</p> <p>ARY(0) - a five piece string:</p> <ol style="list-style-type: none"> 1. code 2. hierarchy 3. version 4. legacy code 5. code status <p>ARY("F") fully specified name</p> <p>ARY("P") preferred term</p> <p>ARY("S",n) synonyms (n is the nth synonym)</p> <p>if call cannot find specified code on file</p> <p>"-2^"_NAME_" code "_CODE_" not on file"</p> <p>where NAME is the source name and</p> <p>CODE is the code</p> <p>if call finds an inactive code for the source</p> <p>"-4^"_NAME_" code "_CODE_" not active for "_DATE"</p> <p>ARY - an array containing information about the code</p> <p>ARY(0) - a five piece string:</p> <ol style="list-style-type: none"> 1. code 2. hierarchy 3. version 4. legacy code 5. code status <p>otherwise</p> <p>"-1^error text"</p> <p>example of ARY array:</p>
----------	--

```

ARY(0)="67922002^Substance^20050701^T-C2500^1"
ARY("F")="Serum (Substance)"
ARY("P")="Serum"

```

TEXT^LEXTRAN(<Text>,<Date>,<Sub>, <Src>,<ARY>) ICR 4913

This API retrieves concept data for a given designation and coding system.

Input

<Text>	This is a designation.
<Date>	This is a date in FileMan format used to ensure that the expressions returned are for active codes on the date supplied. If the date is not passed, then TODAY is used.
<Sub>	This is a subset identifier. The subset specified must be one of the subsets defined in the subset definitions file (757.2).
<Src>	This is a coding system identifier that identifies one of the coding systems listed in Coding Systems file #757.03. E.g., ICD, CPT, DSM, NANDA, etc.
<ARY>	This is the output array (defaults to LEX if none specified).

Output

LEX	<p>or passed array name - an array containing information about the code</p> <p>LEX(0) - a five piece string:</p> <ol style="list-style-type: none"> 1. code 2. hierarchy 3. version 4. legacy code 5. code status <p>otherwise</p> <p>"-1^error text"</p> <p>example of LEX array:</p> <pre> LEX(0)="67922002^Substance^20050701^T-C2500^1" LEX("F")="Serum (Substance)" LEX("P")="Serum" </pre>
-----	--

VERSION^LEXTRAN(<Src>,<Code>,<Date>) ICR 5011

This API retrieves version information for a given coding system and code.

Input

<Src>	This is a coding system identifier that identifies one of the coding systems listed in Coding Systems file #757.03. e.g., ICD, CPT, DSM, NANDA, etc.
-------	--

<Code> This is a source code taken from one of the classification systems listed in Coding Systems file #757.03. e.g., ICD, CPT, DSM, NANDA, etc.

<Date> This is a date in FileMan format used to ensure that the expressions returned are for active codes on the date supplied. If the date is not passed, then TODAY is used.

Output

\$\$VERSION 1^Version
or
-1^error message

TXT4CS^LEXTRAN(<Text>, <Src>,<ARY>, <Sub>) ICR 4914

This API determines whether a given designation is valid for a specified coding system.

Input

<Text> This is a designation.

<Src> This is a coding system identifier that identifies one of the coding systems listed in Coding Systems file #757.03. E.g., ICD, CPT, DSM, NANDA, etc.

<ARY> This is the output array (defaults to LEX if none specified).

<Sub> This is a subset identifier. The subset specified must be one of the subsets defined in the subset definitions file (#757.2).

Output

\$\$TXT4CS 1^number of finds
plus
LEX or passed array name - an array containing discovered concept IDs and expression type for finds, e.g. LEX(113912006)="MAJOR CONCEPT"
or
-1^error message

14.4 LEXTRAN1

\$\$GETSYN^LEXTRAN1(<Src>,<Code>,<Date>,ARY,<IEN>,<ID>,<INC>) ICR 5006

This API will return an array for a given code and coding system. The array will contain all synonyms for the concept including the preferred term and the fully specified name. If any of the parameters are incorrect or unrecognizable, the API will return an error message indicating the nature of the error.

Input

<Src> This is the mnemonic for a code system (mandatory). The allowable code system mnemonics are those that exist in the "B" index of the coding systems file (#757.03) This is code system source abbreviation Lexicon.

<Code> This is a code of a classification system that is stored in the Lexicon. Classification systems include SNOMED CT, ICD, CPT, HCPCS, etc.

<Date>	This is the effective date; the default if no date is specified is the current system date (optional).
<IEN>	If this parameter is set to 1 the expression IEN will be included in the return array. Default is 0 - exclude IENS from return array.
<ID>	Designation Identifiers (optional) 1 return Designation IDs (3rd piece) 0 do not return Designation IDs (default)
<INC>	Include Deactivated Terms (optional) 1 return Deactivated Terms 0 do not return Deactivated Terms (default)

Input/Output

ARY	This is the name of the output array. The default, if no array name is specified, is 'LEX' (optional)
-----	---

The format of the output is as follows:

If valid code and source are passed

1^no of synonyms

LEX("P") = preferred term or major concept name^IEN

LEX("F") = fully specified name^IEN (if one exists)

LEX("S",n) = the nth synonym found^IEN (if they exist)

The presence of IEN in the return array is determined by the <IEN> parameter.

If the call does not find the code for the specified source it will return "-2^_NAME_" code "_CODE_" not on file" where NAME is the source name CODE is the code.

If an invalid source is passed the call will return "-1^source not recognized"

\$\$GETFSN^LEXTRAN1(<Src>,<Code>,<Date>)

ICR 5007

This API returns the fully specified name for a given coding system and code. If any of the passed parameters are incorrect or unrecognizable, the API will return an error message indicating the nature of the error.

Input

<Src>	This is the mnemonic for a coding system (mandatory). The allowable code system mnemonics are those that exist in the "B" index of the coding systems file (#757.03) This is code system source abbreviation Lexicon.
<Code>	This is a code that belongs to a coding system that is stored in the Lexicon. Coding systems include SNOMED CT, ICD, CPT, HCPCS, etc.
<Date>	This is the effective date; the default if no date is specified is the

current system date (optional).

Output

\$\$GETFSN If the API finds an active code for the source
1^FSN
where FSN is the fully specified name
or
If the API cannot find specified code on file
-8^_NAME_" code "_CODE_" has no FSN"
where NAME is the source name and CODE is
the code
or
-1 ^ error message

\$\$GETPREF^LEXTRAN1(<Src>,<Code>,<Date>) ICR 5008

This API returns the preferred term for a given coding system and code. If any of the parameters are incorrect or unrecognizable, the API will return an error message indicating the nature of the error.

Input

<Src> This is the mnemonic for a code system (mandatory). The allowable code system mnemonics are those that exist in the "B" index of the coding systems file (#757.03). This is the Lexicon code system source abbreviation.

<Code> This is a code belonging to a coding system that is stored in the Lexicon. Coding systems include SNOMED CT, ICD-9-CM, CPT, HCPCS, etc.

<Date> This is the effective date; the default if no date is specified is the current system date (optional).

Output

\$\$GETPREF If call finds an active code for the source
1^PREF
where PREF is the preferred name
or
If call cannot find specified code on file
-2^_NAME_" code "_CODE_" not on file"
where NAME is the source name and CODE is
the code
or
-1 ^ error message

\$\$GETDES^LEXTRAN1(<Src>,<Text>,<Date>)

ICR 5009

Input

<Src>	This is the mnemonic for a code system (mandatory). The allowable code system mnemonics are those that exist in the "B" index of the coding systems file (#757.03). This is the Lexicon code system source abbreviation.
<Text>	This is the displayable text of the expression for which the designation code is being sought (mandatory).
<Date>	This is the effective date; the default if no date is specified is the current system date (optional).

Output

\$\$GETDES	If call finds an active code for the source 1^DESIG where DESIG is the designation code or If call cannot find specified code on file -2^_NAME_" code "_CODE_" not on file" where NAME is the source name and CODE is the code. or -1 ^ error message
------------	---

\$\$GETASSN^LEXTRAN1(<Code>,<Map>,<Date>,ARY)

ICR 5010

This API returns an array containing the mappings for a specified code for a specified mapping identifier. If any of the parameters are incorrect or unrecognizable, the API will return an error message indicating the nature of the error.

Input

	This is a code belonging to a coding system that is stored in the Lexicon. Coding systems include SNOMED CT, ICD, CPT, HCPCS, etc.
<Map>	This is the mapping identifier (mandatory). This allows the system to determine which map is to be used for translation. The map must be defined in the mapping definition file (#757.32).
<Date>	This is a code belonging to a coding system that is stored in the Lexicon. Coding systems include SNOMED CT, ICD, CPT, HCPCS, etc.

Input/Output

ARY

This is the name of the output array. The default, if no array name is specified, is 'LEX' (optional) the output array will have the following format:

```
LEX(n, CODE)=""  
where n is the nth mapped code  
CODE is the code which is mapped to
```

e.g.

```
>S V=$$GETASSN(15250008,"SCT2ICD")
```

```
LEX=2  
LEX(1,"371.30")=""  
LEX(2,"371.40")=""
```

which shows that SNOEMD CT code 15250008 is mapped to two ICD-9-CM codes.

If the API finds no active mappings for passed arguments

```
0^0
```

If a bad argument is passed for a parameter, then the API returns

```
#NAME?
```

If the API cannot find specified code on file

```
-2^_NAME_" code "_LEXCODE_" not on file"
```

where NAME is the source name and CODE is the code

14.5 LEXXM

\$\$MIX^LEXXM(<Text>)

ICR 5781

This entry point converts any text to a modified mix case.

Input

<Text>

This is a string of text in any case possible.

Output

\$\$MIX

This is the same text, in a modified mix case.

Example:

arthropathy in behcet's syndrome involving
other specified sites

Traditional Mixed Case (FileMan):

Arthropathy In Behcet's Syndrome Involving
Other Specified Sites

Lexicon Modified Mixed Case (\$\$MIX^LEXXM):

Arthropathy in Behcet's Syndrome involving
other specified sites

14.6 LEXA

INFO^LEXA(<IEN>,<Date>)

ICR 1597

This entry point allows an application to build the LEX("SEL") selection array for any term in the Lexicon based on the internal entry number from the Expression file (#757.01). This entry point is meant to be used outside of the Lexicon lookup. It allows the application to retrieve information pertaining to the term to include synonyms, lexical variants, definitions, classification codes, semantic class and type, and linkages to the major VA classification files (e.g., ICD, CPT, DSM).

Input

<IEN> This is an Internal Entry Number from the Lexicon Expression file #757.01.
<Date> This is a date in FileMan format used to control which classification codes are returned based on the date supplied. If the date is not passed, then TODAY is used. Only active codes are returned.

Output

LEX("SEL") Local array LEX("SEL") containing the following segments:

Array Segment	Content
LEX("SEL", "EXP")	Expressions
LEX("SEL", "SIG")	Definitions when one exists
LEX("SEL", "SRC")	Sources (classification codes)
LEX("SEL", "STY")	Semantic Class and Type
LEX("SEL", "VAS")	VA Sources

The LEX("SEL") array and all of its segments are described in the section titled Special Variables under the subheading Local Arrays.

LOOK^LEXA(<X >,<App>,< Len>, <Sub>,<Date>) ICR 2950

This entry point is silent and intended to support Graphical User Interface (GUI) development.

Input

<X >
<App>

Equivalent to FileMan's variable X and contains the text to search for. This is the application identification and may be in the form of a name, namespace, or a pointer (Internal Entry Number - IEN) from an application definition in the Subset Definition file (#757.2).

The default value for this parameter, if it is not supplied, is one (1), pointing to the Lexicon application definition.

Included in this application definition are a number of application defaults which assist in searching the Lexicon. Application defaults include the global root, index, filter, display format, vocabulary, shortcuts, user default flag, overwrite user default flag, and the unresolved narrative flag. These are described in the Special Variable section of this manual.

At the time of this writing, there are six (6) application definitions:

Name	Namespace	IEN
Lexicon	LEX	1
Problem List	GMPL	4
ICD Diagnosis	ICD	12
CPT Procedures	CPT	13
Mental Health	DSM	14
ICD, CPT, and DSM Terminology	VAC	15

To conduct a search of the Lexicon using the application defaults for the Problem List, you may pass this parameter as:

Name "PROBLEM LIST" - This form is not case sensitive, and can be found in either the "B" or "C" index of file #757.2.

Namespace "GMPL" - Namespace - This form is not case sensitive, and can be found in the "AN" index of file #757.2.

Pointer 4 - This form is numeric, and is an Internal Entry Number (IEN) of file #757.2.

<Len>

This is a numeric value which controls the returning list length in the local array LEX("LIST"). See the section Building/Re-ordering the List for how this variable is used. The default value for this parameter when not supplied is five (5).

<Sub>

This parameter represents the vocabulary subset to use during the search. These subsets are defined in the Subset Definition file (#757.2). This parameter may be in one of three forms. To use the Nursing subset, you may pass the parameter as:

Name NURSING - This form is not case sensitive and may

be found in either the B or C index of file #757.2.

Mnemonic NUR - This form is not case sensitive. The mnemonic may be found in either the AA" or AB index of file #757.2.

Pointer 2 - This form is numeric, and it is an Internal Entry Number (IEN) of file #757.2.

<Date> This date is used to return the appropriate coding and classification information with the expression found during the search. The default value for this parameter is TODAY.
Lookup by Code - When the user searches the Lexicon for a specific classification code (i.e., ICD, CPT etc.), the date will be used to return the expression of the code that was active on the date supplied.
Lookup by Expression (text) - When the list of possible expressions is presented to the user for selection, codes displayed with the expressions include only those codes that were active on the date supplied.

Output

Full descriptions of the global and local arrays may be found in the section on Special Variables.

`^TMP("LEXFND",$J,<freq>,<IEN>)`

This global array contains all of the entries found during the search. The <freq> is a negative number based on the frequency of use for a given term. IEN is the internal entry number in the Lexicon Expression File (#757.01).

`^TMP("LEXHIT",$J,<seq>)`

This global array contains the entries reviewed by the user. The Lexicon Utility reorders the list based on frequency of use and assigns a sequence number representing where on the list this entry is located.

`LEX("LIST")`

This local array contains only those entries on the list which are currently being reviewed by the user. The third parameter to the look-up defines the length of this list.

14.7 LEXAR

EN^LEXAR(<Response >,<Date>)

<Response> This entry point is designed to interpret the user's response to the selection list. It takes two types of input:

1. A user's response
2. A comment from an application

<Date> This date is used to return the appropriate coding and classification information with the expression found during the search. This parameter only comes into play when the <Response> from the user

is a number where the user is selecting an expression from the list.

14.7.1 User Responses

User Response	Action	Results
Null	PGDN	<p>A null response has the effect of advancing the list by the number of entries defined by the third input parameter of the lookup.</p> <ol style="list-style-type: none"> 1. If a null response is received from the user and the user is not at the end of the list, then the next # of entries is placed on the list in the local array.
		<pre>LEX=<total matches found> LEX("EXC")=<exact match concept - optional> LEX("EXM")=<exact match - optional> LEX("LIST",#)=<entry> LEX("LIST",#)=<entry> LEX("LIST",#)=<entry> LEX("LIST",#)=<entry> LEX("LIST",#)=<entry> LEX("LIST",#)=<entry> LEX("MAX")=<last entry reviewed> LEX("MIN")=<first entry review> LEX("NAR")=<user input - optional></pre> <ol style="list-style-type: none"> 2. If a null response is received from the user and the user is at the end of the list, then the list is killed and the dialog with the user is considered over. If the application uses the Unresolved Narratives the user narrative may be returned as the user's response to save the narrative in the Unresolved Narrative file (#757.06). <pre>LEX=0 LEX("NAR")=<user input></pre>
^	QUIT	<p>Ends the dialog with the user by quitting the selection process, killing the selection list and setting LEX=0</p>
		<pre>LEX=0 LEX("NAR")=<user input></pre>
^^	EXIT	<p>Ends the dialog with the application and kills all LEX namespaced variables.</p>
^#	JUMP	<p>An up-arrow followed by a numeric value where the number is a specified entry on the list allows the user to jump from one location on the list to another.</p>
		<pre>LEX=<total matches found> LEX("EXC")=<exact match concept - optional> LEX("EXM")=<exact match - optional> LEX("LIST",#)=<entry> LEX("LIST",#)=<entry> LEX("LIST",#)=<entry> LEX("LIST",#)=<entry> LEX("LIST",#)=<entry> LEX("LIST",#)=<entry> LEX("MAX")=<last entry reviewed></pre>

User Response	Action	Results
		LEX("MIN")=<first entry review> LEX("NAR")=<user input - optional>
?	HELP	Places standard help in the array LEX("HLP").
		LEX=<total matches found> LEX("EXC")=<exact match concept - optional> LEX("EXM")=<exact match - optional> LEX("HLP",#)=<help text> LEX("LIST",#)=<entry> LEX("LIST",#)=<entry> LEX("LIST",#)=<entry> LEX("LIST",#)=<entry> LEX("LIST",#)=<entry> LEX("MAX")=<last entry reviewed> LEX("MIN")=<first entry review> LEX("NAR")=<user input - optional> LEX("LIST",#)=<entry> LEX("MAX")=<last entry reviewed> LEX("MIN")=<first entry review> LEX("NAR")=<user input - optional>
?#	HELP	A question mark followed by a numeric value where the number is a specified entry on the list. If the entry specified has a definition, that definition is placed in the array LEX("HLP").
		LEX=<total matches found> LEX("EXC")=<exact match concept - optional> LEX("EXM")=<exact match - optional> LEX("HLP",#)=<definition text> LEX("HLP",#)=<definition text> LEX("HLP",#)=<definition text> LEX("HLP",#)=<definition text> LEX("LIST",#)=<entry> LEX("LIST",#)=<entry> LEX("LIST",#)=<entry> LEX("LIST",#)=<entry> LEX("LIST",#)=<entry> LEX("MAX")=<last entry reviewed> LEX("MIN")=<first entry review> LEX("NAR")=<user input - optional>
-	PGUP	Backs up the list by the number of entries defined by the third parameter of the lookup.
		LEX=<total matches found> LEX("EXC")=<exact match concept - optional> LEX("EXM")=<exact match - optional> LEX("LIST",#)=<entry> LEX("LIST",#)=<entry> LEX("LIST",#)=<entry> LEX("LIST",#)=<entry> LEX("LIST",#)=<entry> LEX("LIST",#)=<entry> LEX("MAX")=<last entry reviewed> LEX("MIN")=<first entry review> LEX("NAR")=<user input - optional>
#	SELECT	Selects an entry from the list and kills the list. The selected

User Response	Action	Results
		entry, and information pertaining to the entry, is placed in the array LEX("SEL").
		<pre> LEX=<total matches found> LEX("SEL")=<exact match - optional> LEX("SEL","EXP",0)=<expressions> LEX("SEL","EXP",#)=<IEN^expression text> LEX("SEL","SIG",0)=<definition> LEX("SEL","SIG",#)=<definition text> LEX("SEL","SRC",0)=<sources> LEX("SEL","SRC",#)=<source^code> LEX("SEL","VAS",0)=<VA sources> LEX("SEL","VAS",#)=<file^VP^code^IEN> </pre> <p>If a date is passed, then the sources listed in the LEX("SEL","SRC") array will contain active codes based on the date provided. If no date is passed, only active codes for TODAY will be listed.</p>
User Input	Unresolved Narrative	If the list does not exist (in the case of receiving a null response at the end of the list), and the application uses unresolved narratives, and the user's original input string to the lookup is returned to the Lexicon, then the user's input and pertinent information about the search are saved in the Unresolved Narrative file (#757.06).

14.7.2 Application Comment

Application Comment	Action
IEN^"Comment"	<p>This is a special case of this entry point (similar to the use of Unresolved Narratives), and can only be used outside of the Lexicon Lookup (i.e. LEX does not exist). This response to this entry point allows an application to comment on an actual term contained in the Lexicon, save this comment in the Unresolved Narratives file (#757.06), and have that comment returned to the developers along with the user's unresolved narratives. This special case is left up to the discretion of the calling application developers.</p> <p>Examples of application comments might be:</p> <p>IEN^Diagnostic term maps to 799.9</p> <p>This type of comment might be used by an application that requires a valid ICD with a diagnostic term (as is the case with Problem List). If the Lexicon returns the term without an ICD or with an ICD code not found in the ICD-9 file (#80), then the application could take advantage of this entry point to instruct the developers of the Lexicon to have Medical Records Technicians take a look at the term and evaluate the term to an appropriate ICD code for future iterations of the Lexicon.</p> <p>IEN^RBBB suggested shortcut - Right bundle branch block</p> <p>This type of comment might be used when the user input RBBB fails to</p>

Application Comment	Action
	return a selection list, and on a subsequent search the user entered bundle branch block and selected Right bundle branch block, implying that RBBB was intended to have found "Right bundle branch block."

15. Other Supporting Package Components

15.1 LEXD* Namespaced Routines

15.1.1 EN1^LEXD*(<Application>)

This is a series of callable routines established for the expressed purpose of setting user defaults for a given application and intended for applications to create options to change the user lookup defaults for that application. All of these routines assume DUZ is set to the current user. Two conditions must be met for these routines to be used:

1. The application indicated by the input parameter has an application definition in the Subset Definition file (#757.2).
2. The application definition permits user defaults (Application User Defaults flag in file #757.2 is set to 1).

Input

<Application>

This is the application identification and may be in the form of a name (i.e., PROBLEM LIST", a namespace (i.e., GMPL) or a pointer (Internal Entry Number - IEN) from an application definition in the Subset Definition file (#757.2). The default value for this parameter, if not supplied, is one (1), pointing to the Lexicon application definition. This is the same as the Application input parameter for LOOK^LEXA.

EN1^LEXDFL(<Application>) ICR 1599

This entry point allows a user to select or create a default filter for the application identified by the input parameter application.

EN1^LEXDCC(<Application>) ICR 1601

This entry point allows a user to select or create a default display for the application identified by the input parameter application.

EN1^LEXDVO(<Application>) ICR 1603

This entry point allows a user to select a default vocabulary for the application identified by the input parameter application.

EN1^LEXDCX(<Application>) ICR 1605

This entry point allows a user to select a set of shortcuts (based on context) for the application identified by the input parameter application.

15.2 Special Variables

15.2.1 Variables Affecting the Lookup

LEXLL

This variable is taken from the third parameter to the entry point LOOK^LEXA and is a numeric value and controls the returning list length in the local array LEX("LIST").

LEXSUB

This variable is taken from the fourth parameter to the entry point LOOK^LEXA and the second input parameter to the entry point CONFIG^LEXSET. It represents the vocabulary subset to use during the search. This subset is indexed at either the AA or AB index of the Subset Definition file (#757.2). This parameter may be in one of three forms:

For example, to use the Nursing subset you may pass the parameter as:

Name	NURSING	The name is found in either the B or C index of file #757.2.
Mnemonic	NUR	The mnemonic is found in either the AA or AB index of file #757.2.
Pointer	2	This is an Internal Entry Number (IEN) of file #757.2.

LEXQ

This variable is used to tell the setup routine CONFIG^LEXSET which type of search variable to return:

LEXQ=1

Returns search variables for the silent lookup LOOK^LEXA (version 2+) and the loud lookup ^LEXA1 (version 2+) which is called by ^DIC and uses silent calls. These search variables are placed in the global array ^TMP("LEXSCH", \$J). **The default for LEXQ when it does not exist is 1.**

LEXQ=0

Returns the search variables for the loud lookup using the Kernel Toolkit's Multi-Term Lookup Utility (MTLU) entry point ^XTLKKWL (Version 1.0).

NOTE: X must be preset to the user input prior to calling CONFIG^LEXSET with LEXQ=0.

LEX

This variable indicates the current status of the dialog between the Lexicon and either the user or the calling application.

LEX > 0

The lookup is still engaged, the selection list stored at ^TMP("LEXHIT") still exists, and the lookup is waiting for a user response.

LEX = 0

The lookup has disengaged, the selection lists stored at ^TMP("LEXHIT") and ^TMP("LEXFND") have been deleted, and the lookup is waiting for a

response from the application. There are only two conditions that can set LEX to 0. They are:

- a. The user has reviewed the entire selection list and not made a selection. If this is the case, the global array ^TMP("LEXSCH",\$J) still exists. At this point, the Lexicon is waiting to receive either the user narrative (to be saved in the Unresolved Narratives file #757.06) or any other response to proceed with cleaning up the environment before disengaging the dialog with the application.
- b. The user has made a selection from the list. If this is the case, the local array LEX("SEL") is present. The Lexicon considers the dialog with the application over, and leaves the cleanup of the environment to the calling application (by killing LEX).

LEX does not exist.

This condition occurs when:

- a. The user's input to LOOK^LEXA is null or contains an up-arrow (^) ending the dialog between the application and the Lexicon.
- b. The user's response while reviewing the list contains double up-arrows (^) ending the dialog between the user and the Lexicon.

LEXVDT

This is a package wide variable and is taken from the fifth parameter to the entry point LOOK^LEXA and is the date (FileMan format) to use to find active codes and terms based on the given date. If not passed, TODAY is used. This date is used also by several other APIs related to data lookup and extraction. As a general guideline, this date should be either the date that service was provided to patient or the date that the term or code is used. This variable is also used in the FileMan (loud) lookup and can be set prior to calling ^DIC. After calling LOOK^LEXA, this variable will remain in the environment. The calling application are responsible for NEWing or KILLing this variable after the lookup.

15.2.2 Global Arrays

Found Array ^TMP("LEXFND")

This global array contains the list of expressions found during the search. This global array continually grows smaller as ^TMP("LEXHIT") grows larger.

Hit Array ^TMP("LEXHIT")

This global array contains the list of expressions found during the search. It is built by reordering the list in ^TMP("LEXFND") as the user reviews the list. The exact match (if any) at the top of the list, is immediately followed by other expressions found in the order of frequency of use. This array grows larger as the user reviews the list by adding entries to the list from ^TMP("LEXFND"). It is deleted when the Lexicon disengages the dialog with the user (the user either entered an up-arrow ^ or has reviewed the entire list and did not make a selection).

15.2.3 Search Conditions

ID	Search Conditions	Version 1.0 Equivalent
APP	Calling Applications	LEXAP
DIS	Display	LEXSHOW
EXC	Exact Match Concept	
EXM	Exact Match	
FIL	Filter	DIC("S")
ID	Search Conditions	Version 1.0 Equivalent
FLN	File Number	
GBL	Global	DIC, XTLKGBL, XTLKKSCH("GBL")
IDX	Index to Search on	XTLKKSCH("INDEX")
LEN	List Length	
LST	Last entry reviewed	
NAR	User input Narrative	X, XTLKX
NUM	Number of matches found	^TMP("XTLKHITS", \$J)
OVR	Overwrite user defaults flag	
RES	Last user response	
SCH	Search string	
SCT	Shortcut preference	
SVC	User's Service	
TOL	Top of the List flag	
UNR	Unresolved Narrative flag	LEXUN
USR	User ID	DUZ
VDT	Code Set Version Date	
VOC	Subset (vocabulary)	LEXSUB

15.2.4 Local Arrays

There is only one local array, LEX. It contains the following segments:

LEX("ERR",#)	Errors
LEX("EXC")	Location of an Exact Match Major Concept
LEX("EXM")	Location of an Exact Match Major Concept
LEX("HLP",#)	Help Text to Display
LEX("LIST",#)	Selection List to Display
LEX("NAR")	User Narrative
LEX("MAT")	Matches Found String

LEX("MAX")	The Maximum allowable Selection
LEX("MIN")	The minimum allowable Selection
LEX("RES")	Last Response from the User
LEX("UNR")	Unresolved Narrative flag
LEX("SEL",SEG,#)	The Results of a User Selection

15.2.5 Error Array

LEX("ERR")

This segment only exists if an exact match is found during the lookup. It provides the location where the exact match is stored. There are two forms of this segment:

1. The list still exists and the user is reviewing the entries on the list for selection:

LEX("EXM")=<position on the list>^<term>

In this case, the position on the list is set to 1 (exact matches are placed on the top of the selection list). A calling application could use the position on the list (the first piece) as a default value (formerly DIC("B")) when offering the user a choice.

22 matches found

1. Exact match
2. Exact match Major Concept - see LEX("EXC") below
3. Other match
4. Yet another match
5.

Select 1-5: 1//

2. The list no longer exists because the user has either made a selection from the entries on the list or has reviewed all the entries on the list without making a selection:

LEX("EXM")=<IEN>^<term>

In this case, the position on the list has been replaced with the internal entry number in the Lexicon Expression file (757.01) and remains available for further use (i.e., Unresolved Narratives).

15.2.6 Exact Match Concept

LEX("EXC")

This segment only exists if an exact match is found during the lookup (see EXM above) and the exact match is not a Major Concept (i.e., synonym or lexical variant to a Major Concept). It provides the location where the Major Concept of the exact match is stored. Like EXM, there are two forms of this segment, they are:

1. The list still exists and the user is reviewing the entries on the list for selection:

LEX("EXC")=<position on the list>^<term>

Example: User searches the Lexicon for "CHF"

5 matches found

1. CHF (exact match, synonym to Major Concept)
2. Congestive Heart Failure (Major Concept of exact match)
3. Other match
4. Yet another match
5.

Select 1-5: 1//

- The list no longer exists because the user has either made a selection from the entries on the list or has reviewed all the entries on the list without making a selection:

LEX("EXC")=<IEN>^<term>

15.2.7 Help Array

LEX("HLP")

Help text to be displayed (or term definition) when the user's response contains a "?"

15.2.8 List Array

LEX("LIST")

Contains only those entries that should be displayed to the user for selection. It differs from ^TMP("LEXFND") which contains all matches found and ^TMP("LEXHIT") which contains all entries reviewed by the user. It can be thought of as a single page of the selection list with a page length defined by the calling application at the time the search is initiated (the third input parameter of LOOK^LEXA). The default page length of the displayable list is 5, displaying 5 entries at a time until the user has reviewed all the entries on the list or made a selection from the list.

15.2.9 User Narrative

LEX("NAR")

This is the text string that the user inputs to the lookup. It only exists if the calling application uses the Unresolved Narrative function of the Lexicon.

15.2.10 Matches Found String

LEX("MAT")

This text string indicates the total number of entries found during the search, and it is only available during the initial review of the list and when the user is at the top of the list. Examples might be 1 match found or 36 matches found, and could be used as:

22 matches found

1. Condition
2. Condition without mention of complications
3. Condition in late stages of development
4. Condition
5. Condition

Select 1-5:

15.2.11 Maximum Selection

LEX("MAX")

This segment only exists if a selection from the list is possible. When it exists, it sets to the number of the last entry on the list that the user has reviewed, not the total number of entries found. The total number of entries found is stored at LEX. LEX is frequently greater than LEX("MAX") until the user has reached the end of the list, then they are the same. If the last entry on the list that a user has reviewed was 30, and the user jumps backwards on the list (jumps from entry 30 to entry 8), LEX("MAX") remains at 30.

We suggest that both LEX("MIN") and LEX("MAX") may be used to build a selection prompt (formerly DIC("A")) for the user. For example:

Select _LEX("MIN")_ - _LEX("MAX")_:

15.2.12 Minimum Selection

LEX("MIN")

This segment only exists if a selection from the list is possible. When it exists, it should always be set to 1.

15.2.13 Response from the User

LEX("RES")

This segment contains the last response from the user. It only exists if the global array ^TMP("LEXSCH") exists.

15.2.14 Selection Array

LEX("SEL")

Returned information about the user's selection (formerly a non-negative Y in Version 1.0). The absence of this array segment at the conclusion of the user's review of the list implies that no selection was made or that the user up-arrowed out of the selection process (implied -Y). The calling application must extract from the array the information needed and delete the array. The following is an example of the SEL array:

```
LEX("SEL", "EXP", 0)=5
LEX("SEL", "EXP", 1)=22600^Sexual Abuse of Child
LEX("SEL", "EXP", 2)=22601^Child Molestation, Sexual
LEX("SEL", "EXP", 3)=22604^Abuses, Child Sexual
LEX("SEL", "EXP", 4)=22608^Child Sexual Abuses
LEX("SEL", "EXP", 5)=22610^Sexual Abuses, Child
LEX("SEL", "EXP", "B", 22600, 1)=
LEX("SEL", "EXP", "B", 22601, 2)=
LEX("SEL", "EXP", "B", 22604, 3)=
LEX("SEL", "EXP", "B", 22608, 4)=
LEX("SEL", "EXP", "B", 22610, 5)=
LEX("SEL", "EXP", "C", "LEX", 3)=
LEX("SEL", "EXP", "C", "LEX", 4)=
LEX("SEL", "EXP", "C", "LEX", 5)=
LEX("SEL", "EXP", "C", "MAJ", 1)=
LEX("SEL", "EXP", "C", "SYN", 2)=
LEX("SEL", "SIG", 0)=1
LEX("SEL", "SIG", 1)=Sexual maltreatment of the child or
minor.
LEX("SEL", "SRC", 0)=3
LEX("SEL", "SRC", 1)=ICD-9-CM^995.5^22600
LEX("SEL", "SRC", 2)=ICD-9-CM^V61.21^22600
LEX("SEL", "SRC", 3)=DSM-IV^V61.21^22600
LEX("SEL", "SRC", "B", "DSM-IV", 3)=
LEX("SEL", "SRC", "B", "ICD-9-CM", 1)=
LEX("SEL", "SRC", "B", "ICD-9-CM", 2)=
LEX("SEL", "SRC", "C", 995.5, 1)=
LEX("SEL", "SRC", "C", "V61.21", 2)=
LEX("SEL", "SRC", "C", "V61.21", 3)=
LEX("SEL", "SRC", "D", 22600, 1)=
LEX("SEL", "SRC", "D", 22600, 2)=
LEX("SEL", "SRC", "D", 22600, 3)=
LEX("SEL", "STY", 0)=1
LEX("SEL", "STY", 1)=Diseases/Pathologic Processes^Mental or
Behavioral Dysfunction
LEX("SEL", "VAS", 0)=11
LEX("SEL", "VAS", 1)=80^11656; ICD9 (^V61.21^22600
LEX("SEL", "VAS", 2)=80^7571; ICD9 (^995.5^22600
```



```

LEX("SEL", "VAS", 3) = 627.7^1055; YSD(627.7, ^V61.21^22600
LEX("SEL", "VAS", "B", 80, 1) =
LEX("SEL", "VAS", "B", 80, 2) =
LEX("SEL", "VAS", "B", 627.7, 3) =
LEX("SEL", "VAS", "C", 995.5, 2) =
LEX("SEL", "VAS", "C", "V61.21", 1) =
LEX("SEL", "VAS", "C", "V61.21", 3) =
LEX("SEL", "VAS", "D", 22600, 1) =
LEX("SEL", "VAS", "D", 22600, 2) =
LEX("SEL", "VAS", "D", 22600, 3) =
LEX("SEL", "VAS", "V", "1055; YSD(627.7, ", 3) =
LEX("SEL", "VAS", "V", "11656; ICD9(", 1) =
LEX("SEL", "VAS", "V", "7571; ICD9(", 2) =

```

The LEX("SEL") array is in 5 segments:

Expressions

LEX("SEL", "EXP")

Contains the expressions selected by the user in the same format as FileMan's returned variable Y. This portion of the array includes the Major Concept and all Synonyms and Lexical Variants.

LEX("SEL", "EXP", 1) is always the expression selected by the user. This segment has two indexes:

B Internal Entry Point of the Expression file #757.01.

C Expression type; (MAJ)or concept, (SYN)onym, and (LEX)ical variants

Significance

LEX("SEL", "SIG")

Contains the definition of the Major Concept, if one exists.

Sources

LEX("SEL", "SRC")

Contains source codes for specified classification systems (i.e., ICD, CPT, DSM, etc.) for the expressions contained in LEX("SEL", "EXP"). Each entry contains the classification system nomenclature, the classification code, and the internal entry number to the expression in file 757.01 to which it is mapped.

All classification codes returned in this segment are active codes based on the versioning date provided. If no date is provided, then all codes returned in this array are active as of TODAY (default).

This segment has three indexes:

- B Classification System Nomenclature
- C Classification Code
- D Internal Entry Number to file 757.01

Semantics

LEX("SEL", "STY")

Contains the Semantic Class and the Semantic Type of the Major Concept contained in LEX("SEL", "EXP").

VA Sources

LEX("SEL", "VAS")

If one or more of the sources in LEX("SEL", "SRC") is found in one of the primary VA authoritative files, then this section contains the file number, variable pointer, the source code, an internal entry number to the Lexicon expression, the source abbreviation, and the source name. The primary VA authoritative files pointed to include file #80 (ICD Diagnosis), file #80.1 (ICD Procedures), file #81 (CPT), and file #627.7

(DSM-IV). There exist one exception, Title 38 disability codes for which the Lexicon CODES file #757.02 is the authoritative file.

```
LEX("SEL", "VAS", 1) = "80^2895; ICD9 (^530.6^270063^ICD^ICD-9-CM"
LEX("SEL", "VAS", 2) = "757.02^317612; LEX(757.02, ^7205^270063^SCC^TITLE 38"
LEX("SEL", "VAS", #) = File #^Variable Pointer^Code^IEN^SAB^Source
```

For each entry, an activation history is provided including the effective date, the status, and a comment.

```
LEX("SEL", "VAS", 1, 1) = "2781001^1^Activated"
LEX("SEL", "VAS", #, #) = effective date^status^comment
```

All classification codes returned in this segment are active codes based on the versioning date provided. If no date is provided, then all codes returned in this array are active as of TODAY (default).

This segment has five indexes:

- B VA authoritative file number
- C Classification Code
- D Internal Entry Number to file 757.01
- S Source Abbreviation
- V Variable pointer to the authoritative file

15.3 Controlling the View

15.3.1 View by Semantic Class and Types

The Lexicon provides for filtering the search to view only those terms that semantically fit into a class and type, or a group of classes and types. We listed these classes and types in Appendix A of this document. The Lexicon uses a string of identifiers indicating the classes and types to either include or exclude in a search. This string is in two parts (delimited by a ";") of those classes and types to include in the search and those classes and types to exclude. The 'include' portion of the string has precedence over the 'exclude' portion. The insertion of a class into the string represents all of the types belonging to that class; consequently, it is not necessary to repeat all of the types with the class. The absence of a class/type in the include portion of the string automatically excludes it from the search. Semantic classes are represented by a 3-character mnemonic and semantic types are represented numerically (provided by NLM UMLS). For example:

The string BEH/DIS/44/45/49/167/4/5/7/PHY/PRO;50 translates to:

INCLUDE:	EXCLUDE:	
Behaviors	Activities	Nucleic Acid
Diseases/Pathologic	Anatomy	Nucleoside or Nucleotide
Process		
Physiology	Chemicals and Drugs	Amino Acid, Peptide or Protein
Procedures	Concepts and Ideas	Gene Product
Fungus	Geographic Areas	Plant
Virus	Groups	Alga

Bacterium	Physical Objects	Rickettsia or Chlamydia
Molecular Functions	Occupations/Organizations	Animal
Genetic Functions	Macromolecular Structure	Invertebrate
Cell/Molecular	Gene or Genome	Vertebrate
Dysfunctions		
INCLUDE:	EXCLUDE:	
Substances	Molecular Function	Amphibian
	Genetic Function	Bird
	Research Technique	Fish
	Molecular Sequence	Reptile
	Nucleotide Sequence	Mammal
	Amino Acid Sequence	Human
	Carbohydrate Sequence	

15.3.2 View by Classification System

The Lexicon lets you filter the search to view only those terms linked to a specified classification system. These classification systems (provided by both the NLM and the VA) are represented by a 3 character mnemonic and are listed in Appendix B of this document. The Lexicon uses a string of mnemonic identifiers indicating the classification systems to include in a search. For example:

The string: "ICD/CPT/DS4" translates to:

Include terms linked to:

ICD-9 International Classification of Diseases

CPT-4 Current Procedural Terminology

DSM-IV Diagnostic and Statistical Manual of Mental Disorders

15.3.3 View by both Semantics and Classification Systems

This is a combination of the two previous views. In this scenario, if the search encounters a term which is to be excluded from the search by virtue of the semantics, but the term is found to be linked to one of the specified classification systems, then the term is included in the search, ignoring the instruction to exclude the term based on semantics. An example of this type of string would be:

"BEH/DIS/44/45/49/4/5/7/PHY/PRO;50;ICD/CPT"

15.3.4 View by Subset

This is not a filtered view in the sense of including/excluding terms from a selection list. A sub-set is a group of terms based on a common theme (e.g., specialty, function, etc.) which are indexed separately from the main word index in the Lexicon. This type of view has two distinct advantages over filtering: 1) it is significantly faster since it does not have the additional burden of deciding whether to include or exclude a term, and 2) it imposes a limit on the search, making it impossible to find a term not contained in the sub-set (e.g., Diabetes Mellitus could not be found within the Dental sub-set).

15.3.5 Other Views

The Lexicon is always open to new methods of changing the view to suit the client application using the existing structures and fields. New fields can be created to support new functionality; however, it should be done with great care and thought (an 8-character mandatory field adds 2 megabytes to the Lexicon). Generally, if the view can be described, then it can be created.

16. Searching the Lexicon: Building and Re-ordering the List

The Lexicon reorders the results of a search beginning with the exact match (if found) followed by other matches in descending order of frequency of use. The reordering of the search results occurs after the search has been completed and while the user is reviewing the matches found. In order to do this, the Lexicon must build three lists. These lists include:

16.1 Matches Found

^TMP("LEXFND",\$J)

This list is built while the search is in progress. Each time a match is found, that term is placed on this list in an order based on a term's frequency of use and Internal Entry Number (IEN) from the Expression file (#757.01). When the search is completed, this list contains all of the matches found.

As the user reviews the matches found, entries are taken off this list and placed on the review list ^TMP("LEXHIT") until the user either selects an entry, terminates the selection process by entering an up-arrow (^), or reaches the end of the list. As the user continues to review the matches found, this list continues to shrink until it no longer exists.

16.2 Matches Reviewed

^TMP("LEXHIT",\$J)

The Lexicon begins to build this list only after the search has completed. This list is initially populated with the first few entries to be reviewed by the user (the exact number is determined by the third input parameter of LOOK^LEXA). Entries on this list are ordered sequentially from one to the total number of matches reviewed by the user.

As the user reviews the matches found, entries are added to this list from the list of matches found in ^TMP("LEXFND") until the user either selects an entry, terminates the selection process by entering an up-arrow ("^"), or reaches the end of the list. As the user continues to review the matches found, this list continues to grow until it contains all of the matches found.

16.3 Matches Displayed

LEX("LIST")

This list contains only those entries to be displayed. The length of this list does not exceed the list length as specified by the calling application in the third input parameter of LOOK^LEXA. If the list length is not specified by the calling application, then the default list length is set to 5.

16.4 Example Search

The user searches the Lexicon with the following results:

Matches found	20
List Length (specified by the calling application)	5

Initially the list of matches found in ^TMP("LEXFND") would contain 20 entries; however, when the search is completed and the selection process begins, the first five (5) entries are taken off the list of matches found in ^TMP("LEXFND") and placed on both the review list in ^TMP("LEXHIT") and the display list in LEX("LIST"). The calling application should display the contents of the display list LEX("LIST") for the user to review.

If the user does not select one of the first five (5) entries on the display list in LEX("LIST") and presses <Return> to review the next five, then an additional five entries are taken from the list of matches found in ^TMP("LEXFND") and placed on the review list in ^TMP("LEXHIT") with only the current five entries being placed on the display list in LEX("LIST"). The calling application should again only display the five entries on the display list LEX("LIST").

As long as the user does not make a selection, and keeps pressing <Return>, entries are taken from the list of matches found in ^TMP("LEXFND") and placed on the review list in ^TMP("LEXHIT") with the current five entries on the display list in LEX("LIST"). Once the user gets to the end of the list, the list of matches found in ^TMP("LEXFND") is depleted, and the list of entries reviewed in ^TMP("LEXHIT") has 20 entries. The display list in LEX("LIST") always has the number of entries specified by the calling application (in this case, five).

If the user has reviewed some or all of the matches found and decides to jump backwards on the list, then the display list in LEX("LIST") is populated from the list of entries reviewed in ^TMP("LEXHIT").

17. Unresolved Narratives

It is possible for users and applications to provide feedback from the sites regarding the content of the Lexicon. This is done either by a user through a calling application (user unresolved narratives) or by the calling application (application unresolved narratives).

Which applications should use Unresolved Narratives? Chances are if the vocabulary which the targeted users are employing is subject to a myriad of synonyms and lexical variants (e.g. plurals, singular form, etc.), then the application should use the Unresolved Narrative functionality. Applications which would not want to use Unresolved Narratives are those which use an extremely controlled vocabulary where a single concept has only one acceptable form or if adding terminology would disrupt the content and purpose of the controlled vocabulary.

17.1 User Unresolved Narratives

There are three prerequisites the calling application must meet to use and return User Unresolved Narratives:

1. The calling application must be able to store the text within the calling application or store the text in an alternate file (i.e., the Provider Narrative file) and point to the text.
2. The calling application must be defined in the Subset Definition file (#757.2).
3. The Unresolved Narrative flag in the Subset Definition file for the calling application must be set to 1.

There are two prerequisites the Lexicon must meet to save and return the User Unresolved Narrative:

1. The lookup must have completed with no selection made. This is determined by the absence of the following arrays:

^TMP("LEXFND", \$J)

^TMP("LEXHIT", \$J)

LEX("LIST")

LEX("SEL")

2. The Lexicon must have knowledge of the conditions under which the User Unresolved Narrative occurred. This is determined by the presence of the array:

^TMP("LEXSCH", \$J)

When a User Unresolved Narrative occurs, and the conditions above are met, the calling application may store and return the User Unresolved Narrative by calling the entry point:

EN^LEXAR(<user unresolved narrative text>)

When this is done, the User Unresolved Narrative is temporarily stored in the Unresolved Narratives file (757.06). Periodically the Lexicon Utility packs the entries in this file into a mail message and sends them to G.LEXICON@ISC-SLC.VA.GOV for consideration for inclusion in the Lexicon. After the Lexicon Utility sends this message, it deletes the entries in this file.

The following information about the narrative and the conditions of the search may be returned to the OI Field Office:

Narrative	Mandatory	User Input
Date-Time	Mandatory	When the search was conducted
Search String	Mandatory	Actual search string
Matches	Mandatory	Number of matches found
Application	Mandatory	Name of the calling application
Service	Optional	Service of the user
File	Mandatory	Number of file searched
Index	Mandatory	Name of the index used
Shortcuts	Optional	Name of the Shortcut set used
Screen	Optional	Screen used (MUMPS code)

User Unresolved Narratives received at the IRM Field Office are reviewed and classified as:

1. A valid expression to be linked (e.g., synonym or lexical variant).
2. A valid expression to be added (no equivalent concept in the current version).
3. A valid expression in the current version containing a spelling error, acronym, or abbreviation not previously defined. Only the spelling error, acronym, or abbreviation is linked to the existing expression while the remainder of the expression is ignored.
4. A valid expression in the current version.
5. An invalid expression is ignored (e.g., XXXX?).

If the User Unresolved Narrative is included in a future release of the Lexicon and exported to the site, it becomes the responsibility of the calling application to resolve the entry at the site. The Problem List application is the only exception at this time. When a new release of the Lexicon Utility is installed at a site, the Problem List is updated by a series of routines (LEXPL*) called by the Lexicon Utility's Post-Install.

17.2 Application Unresolved Narratives

The purpose of this type of unresolved narrative is to permit the calling application to return a comment about an existing term in the Lexicon. This occurs when an application detects a problem with an expression in the Expression file (757.01). The application can return the Internal Entry Number (IEN) of that expression along with a short comment stating the problem. These commented, unresolved narratives are also temporarily stored and periodically packed up into a mail message that is sent to G.LEXICON@ISC-SLC.VA.GOV. However, instead of considering these narratives for inclusion in the Lexicon (since they already exist), the problem stated in the comment field is reviewed and action is taken where appropriate.

There are no prerequisites for the calling application; however, the lookup for the Lexicon must not be engaged (determined by the absence of all Lexicon variables and arrays).

The following information about the expression is returned to the IRM Field Office:

Narrative	Mandatory	User Input
Expression	Mandatory	Pointer to file #757.01
Comment	Mandatory	Brief description of the problem

When an Application Unresolved Narrative occurs, the calling application may store and return the Application Unresolved Narrative by calling the entry point:

EN^LEXAR(<IEN^Comment>)

An application comment is in the general format IEN^COMMENT, where IEN is a pointer to an expression in the Expression file (757.01) and the COMMENT is a text string comment about the expression.

Examples of application comments might be:

IEN^Diagnostic term maps to 799.9

This type of comment might be used by an application which requires a valid ICD with a diagnostic term (as is the case with Problem List). If the Lexicon returns the term without an ICD or with ICD code not found in the ICD-9 file (#80), then the application could take advantage of this entry point to instruct the developers of the Lexicon to have Medical Records Technicians look at, evaluate, and match the term to an appropriate ICD code for future iterations of the Lexicon.

IEN^RBBB suggested shortcut for Right bundle branch block

This type of comment might be used when the user input RBBB fails to return a selection list, and on a subsequent search the user entered bundle branch block and selected Right bundle branch block, implying that RBBB was intended to have found Right bundle branch block.

18. Re-indexing the Lexicon

For re-indexing, the Lexicon can be divided into two types of files: Those which can be re-indexed independently and those which are re-indexed conditionally. If there is a need to re-index the Lexicon, the files should be re-indexed as follows.

These files may be independently re-indexed.

757	Major Concept Map
757.011	Expression Type
757.014	Expression Form
757.03	Coding System
757.033	Character Positions
757.04	Excluded Words
757.06	Unresolved Narratives
757.11	Semantic Classes
757.12	Sub-Set Definitions
757.3	Lookup Screens
757.32	Mapping Definitions
757.33	Mappings
757.4	Shortcuts
757.41	Shortcut Context

These files have conditions placed on the re-indexing.

757.01	Expressions Immediately after re-indexing this file, re-index the Replacement Word file #757.05 (also see 757.05 listed below).
757.02	Codes Re-index the Coding Systems file #757.03 first.
757.05	Replacement Words Re-indexing of this file depends on indexes in the Expressions file #757.01. If the indexes in file #757.01 are intact and current, then proceed with re-indexing of the Replacement Word file; otherwise, re-index file #757.01 first.
757.1	Semantic Map Re-index both the Semantic Class file #757.11 and the Semantic Type file #757.12 first.
757.21	Sub-Sets Re-indexing of this file depends on indexes in the Sub-Set Definition file #757.2. If the indexes in file #757.2 are intact and current, then proceed with re-indexing of the Sub-Sets file; otherwise, re-index file #757.2 first.

19. Subsets

Subsets, also known as vocabularies, are a collection of terms from the Lexicon that serve a specific purpose or discipline. There are two types of Subsets:

19.1 Logical Subset

This is a collection of terms found in the Lexicon that are set apart from the main Lexicon content through the use of filters and screens similar to Fileman's DIC("S").

Example: The "CPT/HCPCS Procedures" subset is artificially created through the use of a filter which will not permit the selection of a term that is not linked to a valid CPT-4 or HCPCS procedure.

19.2 Physical Subset

This is a collection of terms found in the Lexicon that have been physically set apart from the main Lexicon content by storing the terms in the Subset file 757.21. A physical subset has the advantages of being faster... essentially, it is searching a shorter list. As a result, the search does not need to inspect hundreds of records to determine if the term is contained in a subset. A physical subset has two disadvantages: First, if the physical subset is large, it will significantly increase the disk space requirements for the Lexicon global. Secondly, a physical subset requires constant maintenance (any change made in the Expression file 757.01 needs to be reflected in the Subset file 757.21).

Example: The "Nursing" subset contains terminology from the North American Nursing Diagnosis Association (NANDA), the Nursing Intervention Classification (NIC) and the Omaha Nursing Diagnosis classification systems and is physically stored in the Subset file 757.21.

19.3 Application Subset

An application subset can be either a Logical or Physical Subset. It is developed specifically for an application. An application may have one (primary) or more (secondary) subsets. The application subset will contain the applications namespace on the primary subset and an abbreviated namespace on all secondary subsets. The primary subset will also contain the file number where the pointer to the Lexicon is stored. the Expression file 757.01 needs to be reflected in the Subset file 757.21).

Example: Problem List subset is an application subset created for the Problem List application. It contains the namespace of GMPL and the file number of 9000011. It has a primary subset (PL1) which filters on semantic classes and types and a secondary subset (PL2) which filters on coding system (ICD-9 only).

19.4 Creating an Application Subset

Applications requiring a subset would coordinate with a Lexicon developer for the creation, addition and export of the application subset to the field. The following information will be needed:

Field	Field Content	Comment
.01	Subset Name:	3-35 characters (required)
1	Index Mnemonic:	3 characters, only used for physical subsets. If this is a logical subset, leave this field blank.
2	Global Reference:	For a physical subset use LEX(757.21, and for a logical subset use LEX(757.01,
3/4	Help Routine:	XTLK^LEXHLP (Only used by Kernel Multi-Term Lookup Utility when the MTLU is called)
5/6	Display Routine:	XTLK^LEXPRNT(Only used by Kernel Multi-Term Lookup Utility when the MTLU is called)
7	Display Codes:	<p>This is a string containing a series of coding source abbreviations delimited by the slash "/" character. This string can be used by the display routine. Select from:</p> <ul style="list-style-type: none"> ICD ICD-9-CM Diagnosis ICP ICD-9 Procedures CPT CPT-4 Procedures CPC HCPCS Procedures DS3 DSM-III-R Diagnosis DS4 DSM-IV Diagnosis SNM SNOMED 2 NAN NANDA Nursing Diagnosis NIC Nursing Intervention NOC Nursing Outcomes HHC Home Health Care Diagnosis OMA Omaha Nursing Diagnosis SCC Title 38 Diagnosis/Disabilities ACR Radiological Diagnosis AIR AI/RHEUM Disease/Findings COS COSTAR Term File CST COSTART Adverse Reaction Terms CSP CRISP Scientific Terms DXP DXPLAIN Diagnosis MCM Glossary of Epidemiology Terms UMD Universal Medical Devices UWA Glossary of Neuronames 10D ICD-10-CM Diagnosis 10P ICD-10-PCS Procedures MSH MeSH Medical Subject Headings LCH Library of Congress Headings MTH UMLS Metathesaurus DOR Dorland's Medical Dictionary LNC LOINC RVC Reason for Visit Codes DMI DoD DMIS ID's MTF DoD Military Treating Facilities PRB Problem List Code Set SCT SNOMED CT (Clinical Terms)

Field	Field Content	Comment
		BIR BI-RADS
8	Application Mnemonic	3 characters that represent the application. The last character should be unique to the mnemonic if the application is to have multiple subsets, for example, the Problem List has PL1 and PL2.
9	Application Index	3 characters that represent the cross-reference to be used during lookup. For a logical subset this would be "WRD" for the "AWRD" cross-reference found in file 757.01. For a physical subset this can be any three character that when appended with a leading "A" character the cross-reference can be found in file 757.21. Normally this value is "WRD"
10	Application filter	For a logical subset, this is MUMPS code in the form of an IF statement that when evaluated produces a true (1) or false (0) value. It is the same as FileMan's DIC("S") screen.
11	Application Display Codes	This is a string containing a series of coding source abbreviations delimited by the slash "/" character. It is generally identical to the Display Codes used for Kernel's MTLU. (see field 7)
12	Application User Defaults	Set this value to 1 if the application is to allow the user to have default filters, vocabularies and display values. Set this value to 0 if the application will not allow user defaults. This should generally be set to 0 (zero)
13	Application File Number	This is the file number that points to the Lexicon.
14	Application Namespace	This is the applications namespace (from file #9.4)
15	Unresolved Narratives	This is unique for Problem List. If not the Problem list then enter a 0 (zero).
16	Override User Defaults	Set of Codes: Set this value to 1 if the application's filter, vocabulary and display will override the user's default values. Set this value to 0 if the application will not override the user defaults.
17	Shortcut Context	No longer used, leave blank
18	User Modifiers	No longer used, leave blank
100	Description	Free Text - This is a one or two sentence describing the purpose of the subset and its usage.

20. Integration Control Registrations (ICRs) Summary

20.1 ICRs with Lexicon as the Custodian

20.1.1 Retired/Withdrawn

File

ICR	File	Scope	Subscriber	Status	Date
457	^GMP(757.01,	Supported	N/A	Next Ver	APR 26,1994
5387	^LEX(757/03,	Private	RA	Withdrawn	MAR 16,2009

Routine

ICR	Routine	Scope	Subscriber	Status	Date
339	GMPTDUSR	Private	PL	Retired	MAY 19,2003
340	GMPTSET	Private	PL	Retired	MAY 19,2003
1512	LEXU	Private	AICS	Retired	MAR 8,1996
1577	LEXSET	Controlled	PL	Retired	AUG 8,1996
1578	GMPTSET	Controlled	PL	Withdrawn	AUG 8,1996
2288	LEXU	Supported	N/A	Withdrawn	FEB 3,1998
10148	GMPTU	Supported	N/A	Retired	NOV 22,2011

20.1.2 Active/Pending

File

ICR	File	Scope	Subscriber	Status	Date
1571	^LEX(757.01,	Supported	N/A	Active	AUG 7,1996

Routine

ICR	Routine	Scope	Subscriber	Status	Date
1511	GMPTU	Private	AICS	Active	MAR 8,1996
	ICDONE				
1573	LEXU	Supported	N/A	Active	AUG 7,1996
	\$\$ICDONE(IEN,DATE) \$\$ICD(IEN,DATE) \$\$CPTONE(IEN,DATE) \$\$DSMONE(IEN)				
1597	LEXA	Supported	N/A	Active	AUG 18,1996
	INFO(IEN,DATE)				
1599	LEXDFL	Private	PL	Active	AUG 19,1996
	EN1(LEXAP)				
1601	LEXDCC	Private	PL	Active	AUG 19,1996
	EN1(LEXAP)				
1603	LEXDVO	Private	PL	Active	AUG 19,1996
	EN1(LEXAP)				

ICR	Routine	Scope	Subscriber	Status	Date
1605	LEXDCX	Private	PL	Active	AUG 19,1996
	EN1(LEXAP)				
1607	LEXDDS	Private	PL	Active	AUG 19,1996
	EN1(LEXAP)				
1609	LEXSET	Supported	N/A	Active	AUG 19,1996
	CONFIG(LEXNS,LEXSS,DATE)				
1614	LEXCODE	Supported	N/A	Active	AUG 20,1996
	EN(LEXSO,DATE)				
2950	LEXA	Supported	N/A	Active	APR 16,2003
	LOOK(LEXX,LEXAP,LEXLL,LEXSUB,DATE)				
4083	LEXSRC2	Supported	N/A	Active	APR 14,2003
	\$\$STATCHK(CODE,DATE,LEX,SAB)				
4912	LEXTRAN	Supported	N/A	Active	OCT 5,2006
	\$\$CODE(CODE,SOURCE,DATE,ARRAY,IENS,ID,INC)				
4913	LEXTRAN	Supported	N/A	Active	OCT 5,2006
	\$\$TEXT(TEXT,DATE,SUBSET,SOURCE,ARRAY)				
4914	LEXTRAN	Supported	N/A	Active	OCT 5,2006
	\$\$TXT4CS(TEXT,SOURCE)				
5006	LEXTRAN1	Supported	N/A	Active	JUN 28,2007
	\$\$GETSYN(SRC,CODE,CDT,LEXARY,IENS,ID,INC)				
5007	LEXTRAN1	Supported	N/A	Active	JUN 28,2007
	GETFSN(LEXSRC,LEXCODE,LEXVDT)				
5008	LEXTRAN1	Supported	N/A	Active	JUN 28,2007
	\$\$GETPREF				
5009	LEXTRAN1	Supported	N/A	Active	JUN 28,2007
	\$\$GETDES(LEXSRC)				
5010	LEXTRAN1	Supported	N/A	Active	JUN 28,2007
	GETASSN(LEXCODE,LEXMAP,LEXVDT,LEXRAY)				
5011	LEXTRAN	Supported	N/A	Active	JUN 28,2007
	\$\$VERSION(LEXSRC,LEXCODE,LEXVDT)				
5386	LEXU	Supported	N/A	Active	MAR 13,2009
	\$\$SC(Y,STRING,DATE) \$\$SO(Y,STRING,DATE)				
5252	LEXASCD	Supported	N/A	Pending	AUG 8,2008
	\$\$SC(ICD,VBA,EFF,ARY) \$\$DI(ICD,EFF,ARY) \$\$DX(VBA,EFF,ARY)				
5547	LEXLR	Controlled	LR/DSS	Pending	JUL 23,2010
	\$\$CHKCODE(LEXCODE) \$\$GETCODE(LEXCIEN) GETNAME(LEXINPT,LEXINTY,LEXNAME) \$\$STATUS(LEXINPT,LEXINTY) GETREC(LEXINPT,LEXINTY,LEXREC) \$\$VERSION() COMLST(LEXCOM,LEXARR) DEPLST(LEXARR)				
5679	LEXU	Supported	N/A	Pending	JUN 3,2011
	\$\$D10ONE(IEN,DATE) \$\$D10(IEN,DATE) \$\$P10ONE(IEN,DATE) \$\$ONE(IEN,DATE,SAB) \$\$ALL(IEN,DATE,SAB) \$\$IMPDATE(SAB)				

ICR	Routine	Scope	Subscriber	Status	Date
	\$\$DX(IEN,DATE)				
5680	LEXCODE	Supported	N/A	Pending	JUN 3,2011
	\$\$EXP(<CODE>,<SAB>,<DATE>)				
5681	LEX10CS	Supported	N/A	Pending	JUN 6,2011
	\$\$ICDSRCH(TEXT,ARRAY,DATE,LEN,FILTER) \$\$DIAGSRCH(TEXT,ARRAY,DATE,LEN,FILTER) \$\$PCSDIG(FRAG,DATE) \$\$CODELIST(SYS,SPEC,ARRAY,DATE,LEN,FMT)				
5840	LEX10CX	Controlled	OR/IBD/PL	Pending	SEP 6, 2012
	EN EN2(CODE,SAB) EN3(CODE,SAB,ARY,MAX)				

20.2 ICRs with Lexicon as the Subscriber

20.2.1 Retired/Expired/Withdrawn

File

ICR	File	Scope	Custodian	Status	Date
321	^LEX(757.01,	Private	DI	Expired	APR 3,2007
3997	^AUTNPOV(Private	PL	Withdrawn	MAR 12,2003
4012	^DIC(9.8,	Private	XU	Withdrawn	MAR 18,2003

Options

ICR	Option	Scope	Custodian	Status	Date
857	XLTKUSER2	Private	XT	Retired	FEB 4, 1994

20.2.2 Active/Pending

File

ICR	File	Scope	Custodian	Status	Date
345	^DD(Private	DI	Active	FEB 2, 1994
346	^XT(8984.1,	Private	XT	Active	FEB 4, 1994
510	^DISV(Controlled	DI	Active	JUL 27, 1989
854	^XT(8984.2,	Private	XT	Active	FEB 4, 1994
855	^XT(8984.3,	Private	XT	Active	FEB 4, 1994
856	^XT(8984.4,	Private	XT	Active	FEB 4, 1994
872	^ORD(101,	Controlled	XU	Active	APR 28, 1994
888	^DD(8984.1,	Private	DI	Active	MAY 16, 1994
889	^DD(8984.2,	Private	DI	Active	MAY 16, 1994
890	^XT(8984.2,	Controlled	XT	Active	MAY 16, 1994
891	^XT(8984.3,	Controlled	XT	Active	MAY 16, 1994
916	^DIC(Controlled	DI	Active	JUL 25, 1994
1611	^AUPNPROB(Private	PL	Active	AUG 20, 1996
3779	^DIC(4.2,	Controlled	MM	Active	OCT 7, 2002

ICR	File	Scope	Custodian	Status	Date
4184	^XPD(9.7,	Private	XU	Active	OCT 22, 2004
4475	^DD(Private	FM	Active	JAN 25, 2006
4485	^ICD9(Private	ICD	Active	JUL 28, 2004
4486	^ICD0(Private	ICD	Active	JUL 28, 2004
4487	^ICD(Private	ICD	Active	JUL 28, 2004
4488	^ICM(Private	ICD	Active	JUL 28, 2004
4489	^ICPT(Private	ICPT	Active	JUL 28, 2004
4490	^DIC(81.1,	Private	ICPT	Active	JUL 28, 2004
4491	^DIC(81.2,	Private	ICPT	Active	JUL 28, 2004
4492	^DIC(81.3,	Private	ICPT	Active	JUL 28, 2004
4494	^LEX(757.01,"B",	Private	DI	Active	APR 3, 2007
4797	^XT(8984.4,	Controlled	XT	Pending	SEP 21, 2005
5038	^DD(D0,0,"IX"	Private	DI	Active	NOV 6, 2007
5749	^DD "VR"	Private	DI	Active	NOV 30,2011

20.3 ICRs Supporting Lexicon External References

20.3.1 External Global References

Global Reference	ICR	Comment
^%ZOSF("PROD"	10096	Production Account
^%ZOSF("TEST"	10096	Test for Routine
^%ZOSF("UCI"	10096	Get Account UCI
^AUPNPROB(1611	Unresolved Narratives
^AUTNPOV(1593	Unresolved Narratives
^DD(757*,FLD)	345	Get Field Location
^DD(8984.1)	888	MTLU
^DD(8984.2)	889	MTLU
^DD(757.02)	4475	Control SAB list
^DD(file,0,"VR")	5749	File Version
^DD(file,0,"VRpk")	5749	File Package
^DD(file,0,"VRrv")	5749	File Revision
^DD(757*,0,'IX')	5038	Get Cross-References
^DIC(19	10075	Option file
^DIC(49	10093	Service/Section Defaults
^DIC(81.3	4492	CPT Modifier file
^DIC(9.4	10048	Package file
^DIC(9.8		No longer used (LEXXST2/3)
^DISV(510	Special Lookup Save X
^ICD(4487	DRG
^ICD0(4486	ICD-9 PR
^ICD9(4485	ICD-9 DX

Global Reference	ICR	Comment
^ICM(4488	ICD Major Diagnostic Category
^ICPT(4489	CPT file
^ORD(101	872	Protocol file
^TMP("LEX*",\$J,	SACC 2.3.2.5.1	Temporary Storage
^TMP(\$J,"LEX*",	SACC 2.3.2.5.1	Temporary Storage
^UTILITY(\$J,	10011	Parsing with DIWP
^VA(200	10060	Grandfathered
^XPD(9.6		No longer used (LEXXST3)
^XPD(9.7		No longer used (LEXXST3)
^XT(8984.4	856	MTLU
^XTMP(SACC 2.3.2.5.2	Long Term Controlled Storage
^YSD(627.7	1612	Mental Health DSM file

20.3.2 External Routine References

External Call	ICR
^%DT	10003
NOW^%DTC	10000
%XY^%RCR	10022
^%ZIS	10086
HOME^%ZIS	10086
^%ZISC	10089
%ZTLOAD	10063
\$\$S^%ZTLOAD	10063
^DIC	10006
FIND^DIC	2051
IX^DIC	10006
MIX^DIC1	10007
FILE^DICN	10009
YN^DICN	10009
FILE^DID	2052
^DIE	10018
^DIK	10013
IX1^DIK	10013
IX2^DIK	10013
^DIM	10016
\$\$GET1^DIQ	2056
GETS^DIQ	2056
^DIR	10026
^DIWP	10011
\$\$\$STATCHK^ICDAPIU	3991
HIST^ICDAPIU	3991
\$\$CODEN^ICDCODE	3990
\$\$ICDD^ICDCODE	3990
\$\$ICDDX^ICDCODE	3990
\$\$ICDOP^ICDCODE	3990
ICDD^ICDCODE	3990
DRGD^ICDGTDRG	4052
\$\$\$STATCHK^ICDXAU	5685
HIST^ICDXAU	5685
\$\$DX^ICDXCD	5684
\$\$LD^ICDXCD	5684
\$\$PR^ICDXCD	5684
\$\$SD^ICDXCD	5684
LK^ICDXLK	5686
\$\$\$STATCHK^ICPTAPIU	1997
HIST^ICPTAPIU	1997

External Call	ICR
\$\$CPT^ICPTCOD	1995
\$\$CPTD^ICPTCOD	1995
CPTD^ICPTCOD	1995
\$\$MOD^ICPTMOD	1996
\$\$MODD^ICPTMOD	1996
MODA^ICPTMOD	1996
MODD^ICPTMOD	1996
\$\$DT^XLFD	10103
\$\$FMADD^XLFD	10103
\$\$FMDIFF^XLFD	10103
\$\$FMTE^XLFD	10103
\$\$NOW^XLFD	10103
\$\$LOW^XLFSTR	10104
\$\$UP^XLFSTR	10104
^XMD	10070
BMES^XPDUTL	10141
MES^XPDUTL	10141
EN^XQOR	10140
XTLKKWL	10122
\$\$DTIME^XUP	4409

21. Package Security

Package Security for the Lexicon Utility is maintained through option assignments and VA FileMan Security Codes. We recommend that options and menus be assigned as shown below:

Options recommended for all users:

Option Name	Menu	Routine
Lexicon Utility	LEX UTILITY	Menu
Lookup Term	LEX LOOK-UP	LEXLK
User Defaults	LEX USER DEFAULTS	Menu
Filter	LEX USER FILTER	EN^LEXDFL
Display	LEX USER DISPLAY	EN^LEXDCC
Vocabulary	LEX USER VOCABULARY	EN^LEXDVO
Shortcuts	LEX USER SHORTCUTS	EN^LEXDCX
List Defaults	LEX USER DEFAULT LIST	EN^LEXDDS

Options recommended for managers only:

Option Name	Menu	Routine
Lexicon Management Menu	LEX MGT MENU	Menu
Defaults	LEX MGR DEFAULTS	Menu
Edit User/User Group Defaults	LEX MGR USER DEFAULTS	LEXDMG
Edit User/User Group Defaults	LEX MGR LIST DEFAULTS	LEXDD1
Edit Lexicon	LEX MGR EDIT LEXICON	Menu
Edit Term Definition	LEX MGR EDIT DEFN	LEXEDF1
Edit Shortcuts by Context	LEX MGR EDIT SHORTCUTS	LEXSC

21.1 Use of data by Salt Lake City IRM Field Office Developers:

Unresolved Narratives:

To expand the Lexicon Utility's terms, synonyms, abbreviations, etc., the Salt Lake City IRM Field Office developers have created a program which captures and stores user-entered terminology that doesn't match existing Lexicon terminology.

When users conduct searches in the Lexicon Utility and a match is not found, the text that is entered is saved into the Unresolved Narratives file (#757.06). When the file contains 50 entries, a mail message is generated to transmit the contents of this file to the developers and then entries are purged from the file. This terminology is considered for inclusion in future releases of the Lexicon Utility.

Term Definitions:

When a site edits the content of the Definition field in the Expression file (#757.01), the changes are recorded and a mail message is generated sending the changes to the Salt Lake City IRM Field Office developers. The changes are considered for updating the Lexicon Utility.

VA FileMan Security Codes:

All files are exported with the following security codes:

Action	Security Code
DD	@
Delete	@
Read	@
LAYGO	@
Write	@

22. SACC Exemptions/Non-Standard Code

A SACC exemption was granted on May 9, 2013 to the Clinical Lexicon package (distribution package for ICD data) for the purpose of enabling unsubscribed global kills in the pre-install using FileMan DIU2 utility. This is used when a “full file” distribution is made (delete file 80/80.1 and replace). The exemption reads as follows:

Clinical Lexicon requests an exemption to use \$ZU in the pre and post install routines for future LEX patches. This exemption will expire with the release of LEX 3.0. Calling \$ZU(68,28,0) to enable an unsubscribed global kill prior to installing the latest ICD files leaves the possibility that a global will be killed by another process during a lengthy installation. Placing the call in the pre (or post) install, instead of making the call manually before and after the install, cuts this window down to a few seconds.

23. Appendix A: Classification Systems

ID	Nomenclature	Name	Total Codes	Total Unique
ICD	ICD-9-CM	ICD-9 Diagnosis Clinical Mod	22835	14846
ICP	ICD Proc	ICD-9 Procedures	1021	649
10D	ICD-10-CM	ICD-10 Diagnosis Clinical Mod	69833	69833
10P	ICD-10-PCS	ICD-10 Procedure Coding System	71918	71918
CPT	CPT-4	Current Procedural Terminology	12869	10603
CPC	HCPCS	Current Procedural Codes	9111	8208
DS3	DSM-IIIR	Diagnostic & Stat of Mental Disorders	247	187
DS4	DSM-IV	Diagnostic & Stat of Mental Disorders	404	269
SNM	SNOMED 2	Systematized Nomenclature of Medicine	11102	6815
NAN	NANDA	Classification of Nursing Diagnosis	111	106
NIC	NIC	Nursing Intervention Classifications	341	336
HHC	HHCC	Home Health Care Component	115	115
OMA	Omaha	Omaha Nursing Diagnosis	80	76
SCC	SCC	Service Connected Disabilities	758	758
ACR	ACR	Index for Radiological Diagnosis	119	118
AIR	AI/Rheum	Disease/Findings Knowledge Base	755	751
COS	COSTAR	Computer Stored Ambulatory Records	1391	1385
CST	COSTART	Coding Symbols Adverse Reaction Terms	1669	1123
CSP	CRISP	Computer Retrieval of Info. on Sci Proj	5121	4586
DXP	DxPlain	Diagnostic Prompting System	490	487

ID	Nomenclature	Name	Total Codes	Total Unique
MCM	McMaster	Glossary of Epidemiology Terms	18	18
UMD	UMDNS	Universal Med Device Nomenclature Sys	78	78
SCT	SNOMED CT	SNOMED Clinical Terms	407932	395033

24. Appendix B: Semantic Classes and Types

1. Activities	ACT
Event	51
A broad type for grouping activities, processes, and states. The children of this type are Activity and Phenomenon or Process.	
Activity	52
An operation or series of operations that an organism or machine carries out or participates in. The children of this type are Behavior, Daily or Recreational Activity, Occupational Activity, and Machine Activity. Examples include Development Planning, Expeditions, Information Distribution, Migration, and Voting.	
Daily or Recreational Activity	56
An activity carried out for recreation or exercise. Examples include Swimming, Camping, Child Care, and Exercise.	
Occupational Activity	57
An activity carried out as part of an occupation or job. The children of this type are Health Care Activity, Research Activity, Governmental or Regulatory Activity, and Educational Activity. Examples include Financial Management, Collective Bargaining, Commerce, and Book Classification.	
Health Care Activity	58
An activity of or relating to the practice of medicine or involving the care of patients. The children of this type are Diagnostic Procedure, Laboratory Procedure, and Therapeutic or Preventive Procedure. Examples include Preventive Health Services, Ambulatory Care, Clinic Activities, and Geriatric Nursing.	
Research Activity	62
An activity carried out as part of research or experimentation. This type has one child in the network, Molecular Biology Research Technique. Examples include Study Design, Animal Experimentation, Biomedical Research, and Cluster Analysis.	
Governmental or Regulatory Activity	64
An activity carried out by officially constituted governments, or an activity related to the creation or enforcement of the rules or regulations governing some field of endeavor. Examples include Facility Regulation and Control, Public Assistance, Credentialing, and Certification.	
Educational Activity	65

	An activity related to the organization and provision of education. Examples include Community Health Education, Preceptorship, Academic Training, and Family Planning Training.	
	Machine Activity	66
	An activity carried out primarily or exclusively by machines. Examples include Air Conditioning, Equipment Failure, Natural Language Processing, Computer Simulation, and Word Processing.	
	Phenomenon or Process	67
	A process or state which occurs naturally or because of an activity. The children of this type are Human-caused Phenomenon or Process, Natural Phenomenon or Process, and Injury or Poisoning. Examples include Disasters, Famine, and Noise.	
	Human-caused Phenomenon or Process	68
	A phenomenon or process that is a result of the activities of human beings. If the term refers to the activity itself, rather than the result of that activity, a type from the Activity hierarchy is assigned instead. This type has one child in the network, Environmental Effect of Humans. Examples include Social Change, Baby Boom, and International Cooperation.	
	Environmental Effect of Humans	69
	A change in the natural environment that is a result of the activities of human beings. Examples include Water Pollution, Acid Rain, Soil Degradation, and Smog.	
	Natural Phenomenon or Process	70
	A phenomenon or process that occurs irrespective of the activities of human beings. This type has one child in the network, Biologic Function. Examples include Lightning, Air Movements, Sunlight, Biological Phenomena, and Corrosion.	
2.	Anatomy	ANT
	Anatomical Structure	17
	A normal or pathological part of the anatomy or structural organization of an organism. If the term refers to a structure found only in non-humans, the Non-Human flag is assigned as well. Examples of this would be such terms as Feathers, Gills, and Horns. The children of this type are Embryonic Structure, Congenital Abnormality, Acquired Abnormality, and Fully Formed Anatomical Structure.	
	Embryonic Structure	18
	An anatomical structure that exists only before the organism is fully formed; in mammals, for example, a structure that exists only prior to the birth of the organism. This structure may be normal or abnormal. Examples include Neural Crest, Blastoderm, and Fetal Heart.	
	Congenital Abnormality	19
	An abnormal structure, or one that is abnormal in size or location, present at birth or evolving over time because of a defect in embryogenesis. Examples include Congenital cranial meningocele and Syndactylia.	
	Acquired Abnormality	20
	An abnormal structure or one that is abnormal in size or location, found in or deriving from a previously normal structure. Examples include Hernia, Fistula, Hemorrhoids, and Varicose Veins.	
	Fully Formed Anatomical Structure	21

	An anatomical structure in a fully formed organism; in mammals, for example, a structure in the body after the birth of the organism. The children of this type in the network are Body Part, Organ, or Organ Component, Tissue, Cell, Cell Component, and Macromolecular Structure. They are linked to each other by the part_of relationship in the network. Thus, a Macromolecular Structure is part of a Cell Component, which is part of a Cell, etc. A term is assigned to the most specific type available.	
	Body System	22
	A complex of anatomical structures that performs a common function. Examples include Renin-Angiotensin System, Limbic System, Skeleton, and Reticuloendothelial System.	
	Body Part, Organ, or Organ Component	23
	A collection of cells and tissues which are localized to a specific area or combine and carry out one or more specialized functions of an organism. This ranges from gross structures to small components of complex organs. These structures are relatively localized in comparison to tissues. Examples include Eye, Liver, Pulmonary Artery, and Laryngeal Mucosa.	
	Tissue	24
	An aggregation of similarly specialized cells and the associated intercellular substance. Tissues are relatively non-localized in comparison to body parts, organs, or organ components. Examples include Cartilage, Epidermis, Basophilic muscle fibers, and Endothelium.	
	Cell	25
	The fundamental structural and functional unit of living organisms. Examples include Erythrocytes, Dendritic Cells, and Histiocytes.	
	Cell Component	26
	A part of a cell or the intercellular matrix, generally visible by light microscopy. Examples include Golgi Apparatus, Microsomes, and Organelles.	
	Body Location or Region	29
	An area, subdivision, or region of the body demarcated for the purpose of topographical description. If the term refers to a body location or region found only in non-humans, the Non-Human flag is assigned as well. Examples include Abdomen, Thorax, Back, and Gluteal Region.	
	Body Space or Junction	30
	An area enclosed or surrounded by body parts or organs or the place where two anatomical structures meet or connect. If the term refers to a body space or junction found only in non-humans, the Non-Human flag is assigned as well. Examples include Synapses, Peritoneal Cavity, Neuromuscular Junction, and Knee Joint.	
	Body Substance	31
	Extracellular material, or mixtures of cells and extracellular material, produced, excreted, or accreted by the body. Included here are substances such as saliva, dental enamel, sweat, and gastric acid. If the term refers to a body substance found only in non-humans, the Non-Human flag is assigned as well. Examples include Saliva, Necrotic debris, Mucus, and Amniotic Fluid.	
3.	Behavior	BEH
	Behavior	53
	Any of the activities of humans or animals that can be observed directly by others or can be made systematically observable by the use of special strategies. If the term refers to a behavior	

exhibited only by non-humans, the Non-Human flag is assigned as well. The children of this type are Social Behavior and Individual Behavior.

Social Behavior 54

Behavior that is a direct result or function of the interaction of humans or animals with their fellows. Examples include Interpersonal Relations, Social Conformity, Acculturation, and Communication.

Individual Behavior 55

Behavior exhibited by a human or an animal that is not a direct result of interaction with other members of the species, but which may have an effect on others. Examples include Assertiveness, Self Disclosure, Nail Biting, and Risk-Taking.

4. Chemicals and Drugs CHM

Chemical 103

Chemicals are viewed from two distinct perspectives in the network, functionally and structurally. Almost every chemical term is assigned at least two types, one from the structure hierarchy and at least one from the function hierarchy. The children of this type are Chemical Viewed Functionally and Chemical Viewed Structurally.

Chemical Viewed Structurally 104

A chemical viewed from the perspective of its structural characteristics. Included here are terms which can mean a salt, an ion, or a compound (e.g., Bromates and Bromides). The children of this type are Inorganic Chemical and Organic Chemical. Examples include Free Radicals, Onium Compounds, Salts, and Sulfur Compounds.

Inorganic Chemical 105

The general class of substances including the elements, their ionic and isotopic counterparts, and any chemical compound whose molecules are bound together ionically rather than covalently. This includes all compounds which do not contain carbon as a principal component. The children of this type are Element or Ion, Isotope, and Inorganic Compound. Examples include Electrolytes, Dithionite, and Technetium Tc 99m Sulfur Colloid.

Element or Ion 106

One of the 109 presently known kinds of substance that comprise all matter at and above the atomic level. This includes elemental metals, rare gases, and naturally occurring radioactive elements, as well as the ionic counterparts of elements. This does not include the less abundant isotopic forms, for which the type Isotope is assigned. Examples include Aluminum, Carbon, Uranium, Beryllium, and Oxygen Ion.

Isotope 107

A form of element having the same atomic number (i.e., the same number of protons), but differing in atomic weight or mass due to the presence of one or more additional neutrons. Included here are both stable and radioactive isotopes. Examples include Radioisotopes, Chromium Isotopes, Cobalt Radioisotopes, Co-58 (8), and Deuterium.

Inorganic Compound 108

A single compound, generally with ionic bonding, not containing carbon as a principal component (except carbides, carbonates, cyanides, cyanates, and carbon disulfide). The bonding between elements in inorganic compounds is generally ionic. Included here are inorganic acids

and salts, alloys, alkalies, and minerals. Excluded are hydrocarbons. Examples include Ferrocyanide salt, Ammonia, and Aluminum Hydroxide.

Organic Chemical 109

The general class of carbon-containing compounds usually based on carbon chains or rings, and containing hydrogen (hydrocarbons), with or without nitrogen, oxygen, or other elements. The bonding between elements is generally covalent. The children of this type are Steroid, Eicosanoid, Lactam, Alkaloid, Nucleic Acid, Nucleoside, or Nucleotide, Organophosphorus Compound, Amino Acid, Peptide, or Protein, Carbohydrate, and Lipid. Examples include Busulfan, Carotene, Trinitrobenzene, and Metanephine.

Steroid 110

One of a group of polycyclic, 17-carbon-atom, fused-ring compounds occurring both in natural and synthetic forms. Included here are naturally occurring and synthetic steroids, bufanolides, cardanolides, homosteroids, norsteroids, and secosteroids. Examples include Bufanolides, Norandrostanes, 17-Hydroxycorticosteroids, and Prednisone.

Eicosanoid 111

A compound structurally related to arachidonic acid. Included here are arachidonic acid, eicosanoic acid, and saturated or unsaturated derivatives of each. Examples include Thromboxane B₂, n-Eicosanoic acid, 8,11,14-Eicosatrienoic Acid, and Leokotriene C-4.

Lactam 112

A cyclic amide, usually with 4- or 5-membered rings that may or may not be fused to other rings, as in compounds structurally related to the penicillins and cephalosporins. Examples include Penicillanic Acid, Caprolactam, Alloxan, and Ticarcillin.

Alkaloid 113

A basic, nitrogen-containing compound of plant origin. Included here are aporphines, cinchona, curare, ergot, opium, belladonna, rauwolfia, and vinca alkaloids, among others. Examples include Quinidine, Aconitine, 3-Hydroxy-N-Methylmorphinan, Vincamine, and Rauwolfia Alkaloids.

Organophosphorus Compound 115

An organic compound containing phosphorus as a constituent. Included here are organic phosphinic, phosphonic and phosphoric acid derivatives and their thiophosphorus counterparts. Excluded are phospholipids and sugar phosphates. Examples include Phosphonoacetic Acid, Phosphoric Acid Esters, Diphosponates, and Thiamine Triphosphate.

Carbohydrate 118

A compound consisting of carbon, hydrogen, and oxygen in which the hydrogen/oxygen ratio is the same as in water, and in which repeating units are joined through oxygen linkages. Carbohydrates are generally characterized as sugars and include mono-, di-, oligo-, and polysaccharides, glycosides, glycans, and starches. Included here are sugar phosphates. Excluded are glycolipids. Examples include Glycosides, Polysaccharides, Deoxyglucose, and Sepharose.

Lipid 119

A fat or fat-derived substance, such as fatty acids, fatty alcohols, and waxes. Included here are glyco- and phospholipids. Examples include Ceroid, Sphingolipids, Glycerides, and Calcifediol.

Chemical Viewed Functionally 120

A chemical viewed from the perspective of its functional characteristics or pharmacological activities. The children of this type are Pharmacologic Substance, Biomedical or Dental Material, Biologically Active Substance, Indicator or Reagent, and Hazardous or Poisonous Substance. Examples include Aerosol Propellants, Soaps, and Food Additives.

Pharmacologic Substance 121

A substance used in the treatment, diagnosis, prevention, or analysis of normal and abnormal body function. This includes substances that occur naturally in the body and are administered therapeutically. Examples include Codeine, Antipruritics, Ampicillin, Cardiovascular Agents, Insulin, and Ganglionic Blockaders.

Biomedical or Dental Material 122

A substance used in biomedicine or dentistry predominantly for its physical, as opposed to chemical, properties. Included here are biocompatible materials, tissue adhesives, bone cements, resins, etc. Examples include Anion Exchange Resins, Dental Casting Investment, Elastasil, Bone Cements, and Drug Implants.

Biologically Active Substance 123

A substance produced or required by an organism, of primary interest because of its role in the biologic functioning of the organism that produces it. The children of this type are Neuroreactive Substance or Biogenic Amine, Hormone, Enzyme, Vitamin, Prostaglandin, and Immunologic Factor. Examples include Myelin, Gastric Acid, Growth Substances, and Enzyme Precursors.

Neuroreactive Substance or Biogenic Amine 124

A biologic factor whose activities affect or play a role in the functioning of the nervous system. Included here are catecholamines, neuroregulators, neurophysins, etc. Examples include Catecholamine, Tryptamines, and Neurotensin.

Hormone 125

In animals, a chemical secreted by an endocrine gland that releases its products into the circulating fluid. Plant hormones or synthetic hormones that are used only to alter or control various physiologic processes, e.g., reproductive control agents, are assigned only to the type Pharmacologic Substance. Hormones act as chemical messengers and regulate various physiologic processes such as growth, reproduction, metabolism, etc. They usually fall into two broad classes, steroid hormones and peptide hormones. Examples include Gonadotropins, Epicortisol, Glucocorticoids, Pentagastrin, and MSH Release Inhibiting Hormone.

Enzyme 126

A complex protein that living cells produce and which catalyzes specific biochemical reactions. There are six main types of enzymes, oxidoreductases, transferases, hydrolases, lyases, isomerases, and ligases. Examples include ATP Citrate Lyase, Acetyl CoA Acetyltransferase, Complement Activating Enzymes, and Glucose Oxidase.

Vitamin 127

A substance, usually an organic chemical complex, present in natural products or made synthetically, which is essential in the diet of humans or other higher animals. Included here are vitamin precursors and provitamins. Examples include Vitamin A, Ascorbic Acid, Biotin, Riboflavin, and 25-Hydroxyvitamin D 2.

Prostaglandin 128

	A member of the group of physiologically active compounds derived from arachidonic acid. Members of the group play major roles in the reproductive process, smooth muscle stimulation, blood pressure levels, inflammation, etc. Included here are prostacyclins, thromboxanes, and leukotrienes. Examples include Alprostadil, Prostaglandins F, Thromboxane A2, and Rioprostil.	
	Immunologic Factor	129
	A biologic factor whose activities affect or play a role in the functioning of the immune system. Examples include Autocrine Motility Factor, Antilymphocyte Globulin, HIV Antigens, and Hepatitis surface antigen.	
	Indicator or Reagent	130
	A substance used in laboratory reactions, or laboratory or diagnostic tests and procedures to detect, measure, examine, or analyze other chemicals, processes, or conditions. Examples include Contrast Media, Buffers, Affinity Labels, and Dansyl Compounds.	
	Hazardous or Poisonous Substance	131
	A substance of concern because of its potentially hazardous or toxic effects. This would include most drugs of abuse, as well as agents that require special handling because of their toxicity. Most pharmaceutical agents, although potentially harmful, we exclude here and assign to the type Pharmacologic Substance. Examples include Paraquat, Crack Cocaine, Plant poison, Carcinogens, and Sodium Cyanide.	
5.	Concepts and Ideas	CON
	Conceptual Entity	77
	A broad type for grouping abstract entities or concepts. The children of this type in the network are Idea or Concept, Finding, Organism Attribute, Intellectual Product, Language, Occupation or Discipline, Organization, Group Attribute, and Group.	
	Idea or Concept	78
	An abstract concept, such as a social, religious, or philosophical concept. The children of this type are Temporal Concept, Qualitative Concept, Quantitative Concept, Functional Concept, and Spatial Concept. Examples include Civil Rights, Freedom, Ethics, Spiritualism, and Capitalism.	
	Temporal Concept	79
	A concept that pertains to time or duration. Examples include Half-Life, Postoperative Period, Puerperium, Birth Intervals, and Postimplantation Phase.	
	Qualitative Concept	80
	A concept that is an assessment of some quality, rather than a direct measurement. Examples include Clinical Competence, Quality of Health Care, Abuse of Health Services, and Consumer Satisfaction.	
	Quantitative Concept	81
	A concept that involves the dimensions, quantity or capacity of something using some unit of measure, or which involves the quantitative comparison of entities. Examples include Metric System, Body Height, Age Distribution, and Secretory Rate.	
	Spatial Concept	82
	A location, region, or space, generally having definite boundaries. The children of this type are Body Space or Junction, Body Location or Region, Molecular Sequence, and 'Geographic Area.	

Regulation or Law	89
An intellectual product resulting from legislative or regulatory activity. Examples include Building Codes, Criminal Law, Health Planning Guidelines, and Security Measures.	
Group Attribute	102
A conceptual entity that refers to the frequency or distribution of certain characteristics or phenomena in certain groups. Examples include Neonatal Mortality, Life Expectancy, Family Size, Population Characteristics, and Group Structure.	
Functional Concept	169
A concept that is of interest because it pertains to the carrying out of a process or activity. This type has one child in the network, Body System. Examples include Solar System.	
Intellectual Product	170
A conceptual entity resulting from human endeavor. Terms assigned to this type generally refer to information created by humans for some purpose. This type has one child in the network, 'Regulation or Law. Examples include Bayes Theorem, Information Systems, and Literature.	
Language	171
The system of communication used by a particular nation or people. Examples include Afrikaans, Greek, Modern, Braille, and Welsh.	
6. Diseases and Pathologic Processes	DIS
Pathologic Function	46
A disordered process, activity, or state of the organism as a whole, of a body system or systems, or of multiple organs or tissues. Included here are normal responses to a negative stimulus as well as pathologic conditions or states that are less specific than a disease. Pathologic functions frequently have systemic effects. The children of this type are Disease or Syndrome, Cell or Molecular Dysfunction, and Experimental Model of Disease. Examples include Shock, Infarction, Cerebral Anoxia, Inflammation, Anaphylaxis, and Acid-Base Imbalance.	
Disease or Syndrome	47
A condition that alters or interferes with a normal process, state, or activity of an organism. It is usually characterized by the abnormal functioning of one or more of the host's systems, parts, or organs. Included here is a complex of symptoms descriptive of a disorder. This type has one child in the network, Mental or Behavioral Dysfunction. Examples include Diabetes Mellitus, Brain Neoplasms, Nephrotic Syndrome, Dumping Syndrome, and Malabsorption Syndromes.	
Mental or Behavioral Dysfunction	48
A clinically significant dysfunction whose major manifestation is behavioral or psychological. These dysfunctions may have identified or presumed biological etiologies or manifestations. Examples include Memory Disorders, Agoraphobia, Hallucinations, Anxiety States, Neurotic, and Cyclothymic Disorder.	
Experimental Model of Disease	50
A representation in a non-human organism of a human disease for the purpose of research into its mechanism or treatment. Examples include Avian Leukosis, Streptozotocin Diabetes, Ehrlich Ascites Tumor, and Melanoma, Experimental.	
Finding	33

That which is discovered by direct observation or measurement of an organism attribute or condition, including the clinical history of the patient. The children of this type are Laboratory or Test Result, and Sign or Symptom. Examples include Occupational problem, Birth History, and Downward displacement of diaphragm.

Laboratory or Test Result 34

The outcome of a specific test to measure an attribute or to determine the presence, absence, or degree of a condition. Laboratory or test results are inherently quantitative and, thus, we do not assign the additional type Quantitative Concept. Examples include Apgar Score, Gastric acidity, Blood Volume, and Hybernemia.

Sign or Symptom 184

An observable manifestation of a disease or condition based on clinical judgment, or a manifestation of a disease or condition that the patient experiences and reports as a subjective observation. Examples include Pallor, Body Weight Changes, Echolalia, Hyperventilation, Pain, Toothache, Nausea, and Cough. Formerly semantic types Signs (# 35) and Symptoms (# 36).

Injury or Poisoning 37

A traumatic wound, injury, or poisoning caused by an external agent or force. Examples include Frostbite, Mushroom Poisoning, Acid burn, Snake Bites, and Ergotism.

7. Geographic Areas GEO

Geographic Area 83

A geographic location, generally having definite boundaries. Examples include Canada, Baltimore, Far East, Arctic Regions, and Cities.

8. Groups GRP

Group 96

A conceptual entity referring to the classification of individuals according to certain shared characteristics. The children of this type are Professional or Occupational Group, Population Group, Family Group, Age Group, and Patient or Disabled Group.

Professional or Occupational Group 97

An individual or individuals classified according to their vocation. Examples include Zoologist, Physicians, Hospital Volunteers, Clergy, Military Personnel, and Demographers.

Population Group 98

An individual or individuals classified according to their sex, racial origin, religion, common place of living, financial or social status, or some other cultural or behavioral attribute. Examples include Asian Americans, Ethnic Groups, Homeless Persons, and Low-Income Population.

Family Group 99

An individual or individuals classified according to their family relationships or relative position in the family unit. Examples include Only Child, Single Parent, Surrogate Mothers, and Twins.

Age Group 100

An individual or individuals classified according to their age. Examples include Adult, Infant, Premature, Adolescents, and Octogenarian.

Patient or Disabled Group 101

An individual or individuals classified according to a disability, disease, condition, or treatment. Examples include Amputees, Child, Institutionalized, and Inpatients.

9. Molecular Biology	MOL
Macromolecular Structure	27
A very large molecule whose structure contributes to the physiology of the cell. This type has one child in the network, Gene or Genome. Examples include Scleroproteins, Histone H5, and Collagen.	
Gene or Genome	28
A specific sequence, or in the case of the genome the complete sequence, of nucleotides along a molecule of DNA or RNA (in the case of some viruses) which represent the functional units of heredity. Examples include Alleles, Genes, Structural, Genome, Human, and c-Ha-ras Genes.	
Molecular Function	44
A physiologic function occurring at the molecular level. This type has one child in the network, Genetic Function. Examples include Electron Transport, Glycolysis, and Binding, Competitive.	
Genetic Function	45
Functions of or related to the maintenance, translation, or expression of the genetic material. Examples include Amino Acid Activation, Early Gene Transcription, Gene Amplification, and RNA Splicing.	
Cell or Molecular Dysfunction	49
A pathologic function inherent to cells, parts of cells, or molecules. Examples include Cellular necrosis, Wallerian Degeneration, Cell Transformation, Neoplastic, and DNA Damage.	
Molecular Biology Research Technique	63
Any of the techniques used in the study of or the directed modification of the gene complement of a living organism. Examples include Genetic Engineering, Heterozygote Detection, Sequence Homology Determination, and Blotting, Northern.	
Molecular Sequence	85
A broad type for grouping the collected sequences of amino acids, carbohydrates, and nucleotide sequences. Descriptions of these sequences are generally reported in the published literature and/or are deposited in and maintained by data banks such as GenBank, European Molecular Biology Laboratory (EMBL), National Biomedical Research Foundation (NBRF), or other sequence repositories. The children of this type are Nucleotide Sequence, Amino Acid Sequence, and Carbohydrate Sequence.	
Nucleotide Sequence	86
The sequence of purines and pyrimidines in nucleic acids and polynucleotides. Included here are nucleotide-rich regions, conserved sequence, and DNA transforming region. Examples include AT Rich Region, Base Sequence, Direct Repeat, and Exons.	
Amino Acid Sequence	87
The sequence of amino acids as arrayed in chains, sheets, etc., within the protein molecule. It is of fundamental importance in determining protein structure.	
Carbohydrate Sequence	88
The sequence of carbohydrates within polysaccharides, glycoproteins, and glycolipids.	

Nucleic Acid, Nucleoside, or Nucleotide	114
A complex compound of high molecular weight occurring in living cells. These are of two types, ribonucleic (RNA) and deoxyribo-nucleic (DNA) acids, both of which consist of nucleotides (nucleoside phosphates linked together by phosphate bridges). Examples include Adenosine, Dibutyryl Cyclic AMP, Deoxyadenosines, and Nicotinamide Mononucleotide.	
Amino Acid, Peptide, or Protein	116
Amino acids and chains of amino acids connected by peptide linkages. Examples include Glycoproteins, Myoglobin, Alanine, Sulfatase, and Acetylcysteine.	
Gene Product	<deleted>
Formerly semantic type # 117.	
10. Physical Objects	OBJ
Entity	71
A physical or conceptual entity. The children of this type are Physical Object and Conceptual Entity.	
Physical Object	72
An object perceptible to the sense of vision or touch. The children of this type in the network are Organism, Anatomical Structure, Manufactured Object, and Substance.	
Manufactured Object	73
A physical object made by human beings. The children of this type in the network are Medical Device and Research Device. Examples include Cooking and Eating Utensils, Bookplates, Adhesive tape, and Car Seats.	
Medical Device	74
A manufactured object used primarily in the diagnosis, treatment, or prevention of physiologic or anatomic disorders. Examples include Hip Prosthesis, Oxygenators, Syringes, and Obstetrical Forceps.	
Research Device	75
A manufactured object used primarily in carrying out scientific research or experimentation. Examples include Questionnaires, Atmosphere Exposure Chambers, and Cell-Free System.	
Substance	167
A material with definite or fairly definite chemical composition. The children of this type are Chemical, Body Substance, and Food. Examples include Charcoal, Foreign Bodies, Air, Fossils," and Electrons.	
Food	168
Any substance containing nutrients, such as carbohydrates, proteins, and fats that a living organism can ingest and metabolize into energy and body tissue. Some foods are naturally occurring; others are either partially or entirely synthetic. Examples include Egg Yolk, Nuts, Beverages, and Margarine.	
11. Occupations and Organizations	OCC
Occupation or Discipline	90

A vocation, academic discipline, or field of study, or a subpart of an occupation or discipline. If the term refers to the individuals who have the vocation, then we assign the type Professional or Occupational Group. This type has one child in the network, Biomedical Occupation or Discipline. Examples include Anthropology, Ecology, Linguistics, Air Microbiology, and Craniology.

Biomedical Occupation or Discipline 91

A vocation, academic discipline, or field of study related to biomedicine. Examples include Dermatology, Emergency Nursing, Dentistry, Family Practice, and Cellular Neurobiology.

Organization 92

The result of uniting for a common purpose or function. The continued existence of an organization is not dependent on any of its members, its location, or particular facility. Components or subparts of organizations are also included here. The children of this type are Health Care Related Organization, Professional Society, and Self-help or Relief Organization.' Examples include Universities, United Nations, United States Environmental Protection Agency, European Economic Community, and Labor Unions.

Health Care Related Organization 93

An established organization which carries out specific functions related to health care delivery or research in the life sciences. Terms for health care related professional societies are assigned the type Professional Society. Examples include American Cancer Society Health Care Coalitions, Ambulatory Care Facilities, and Pan American Health Organization.

Professional Society 94

An organization uniting those who have a common vocation or who are involved with a common field of study. Examples include American Medical Association, Library Associations, and International Council of Nurses.

Self-help or Relief Organization 95

An organization whose purpose and function is to provide assistance to the needy or to offer support to those sharing similar problems. Examples include Alcoholics Anonymous, Red Cross, Charities, and Tuberculosis Societies.

12. Organism ORG

Organism 1

Generally, a living individual, including all plants and animals. The children of this type are Plant, Fungus, Virus, Rickettsia or Chlamydia, Bacterium, and Animal. Examples include Plankton, Homozygote, and Radiation Chimera.

Plant 2

An organism having cellulose cell walls, growing by synthesis of inorganic substances, generally distinguished by the presence of chlorophyll, and lacking the power of locomotion. Plant parts are included here as well. This type has one child in the network, Alga. Examples include Potatoes, Pollen, and Vegetables.

Alga 3

A chiefly aquatic plant that contains chlorophyll, but does not form embryos during development and lacks vascular tissue. Examples include Chlorella, Laminaria, Seaweed, and Anabaena.

Fungus 4

A eukaryotic organism characterized by the absence of chlorophyll and the presence of a rigid cell wall. Included here are both slime molds and true fungi such as yeasts, molds, mildews, and mushrooms. Examples include *Blastomyces*, *Neurospora*, *Aspergillus clavatus*, and *Helminthosporium*.

Virus 5

An organism consisting of a core of a single nucleic acid enclosed in a protective coat of protein. A virus may replicate only inside a host living cell. A virus exhibits some but not all of the usual characteristics of living things. Examples include Parvoviridae, Foot-and-Mouth Disease Virus, and Echovirus 6.

Rickettsia or Chlamydia 6

An organism intermediate in size and complexity between a virus and a bacterium, and which is parasitic within the cells of insects and ticks. Included here are all the chlamydias, also called PLT for psittacosis- lymphogranuloma venereum-trachoma. Examples include *Anaplasma*, *Bartonella*, and *Chlamydia trachomatis*.

Bacterium 7

A small, typically one-celled, prokaryotic micro-organism. Examples include *Bacillus cereus*, *Acetobacter*, *Bordetella pertussis*, and *Cytophaga*.

Animal 8

An organism with eukaryotic cells, and lacking stiff cell walls, plastids and photosynthetic pigments. The children of this type are Invertebrate and Vertebrate. Examples include Animals, Poisonous; Animals, Newborn; and Animals, Laboratory.

Invertebrate 9

An animal which has no spinal column. This type has no children in the network and is assigned to all invertebrate animals. Examples include Helminths, Octopus, Wasps, and Protozoa.

Vertebrate 10

An animal which has a spinal column. The children of this type are Amphibian, Bird, Fish, Reptile, and Mammal.

Amphibian 11

A cold-blooded, smooth-skinned vertebrate which characteristically hatches as an aquatic larva, breathing by gills. When mature, the amphibian breathes with lungs. Examples include *Salamandra*, *Urodela*, and Frog.

Bird 12

A vertebrate having a constant body temperature and characterized by the presence of feathers. Examples include Canaries, Pigeons, and Quail.

Fish 13

A cold-blooded aquatic vertebrate characterized by fins and breathing by gills. Included here are fishes having either a bony skeleton, such as a perch, or a cartilaginous skeleton, such as a shark, or those lacking a jaw, such as a lamprey or hagfish. Examples include Bass, Eels, and Carp.

Reptile 14

	A cold-blooded vertebrate having an external covering of scales or horny plates. Reptiles breathe by means of lungs and are generally egg-laying. Examples include Lizards, Snakes, Turtles, and Iguanas.	
	Mammal	15
	A vertebrate having a constant body temperature and characterized by the presence of hair, mammary glands, and sweat glands. This type has one child in the network, Human. Examples include Bears, Macaca, Hamsters, and Kangaroos.	
	Human	16
	Modern man, the only remaining species of the Homo genus. If a term describes a human being from the point of view of occupational, family, social status, etc., then a type from the Group hierarchy is assigned instead. A small number of terms have been assigned this type, e.g., Hominidae, Man, and Homo sapiens.	
13.	Physiology	PHY
	Biologic Function	38
	A state, activity, or process of the body or one of its systems or parts. If the term refers to a biologic function found only in non-humans, the Non-Human flag is assigned as well. The children of this type are Physiologic Function and Pathologic Function.	
	Physiologic Function	39
	A normal process, activity, or state of the body. The children of this type in the network are Organism Function, Organ or Tissue Function, Cell Function, and Molecular Function.	
	Organism Function	40
	A physiologic function of the organism as a whole, of multiple organ systems, or of multiple organs or tissues. This type has one child in the network, Mental Process. Examples include Growth, Sleep, Hibernation, and Homeostasis.	
	Mental Process	41
	A physiologic function involving the mind or cognitive processing. Examples include Avoidance Learning, Pattern Recognition, Anger, and Cognition."	
	Organ or Tissue Function	42
	A physiologic function of a particular organ, organ system, or tissue. Examples include Osteogenesis, Tooth Calcification, and Renal Circulation.	
	Cell Function	43
	A physiologic function inherent to cells or cell components. Examples include Cell Division, Cell Cycle, Erythrocyte Aggregation, and Lymphocyte Transformation.	
	Organism Attribute	32
	A property of the organism or its major parts. If the term refers to an attribute found only in non-humans, the Non-Human flag is assigned as well. Examples include Body Weight, Body Temperature, Ambidexterity, and Eye Color.	
14.	Procedures	PRO
	Laboratory Procedure	59

A procedure, method, or technique used to determine the composition, quantity, or concentration of a specimen, which is carried out in a clinical laboratory. Included here are procedures which measure the times and rates of reactions. Examples include Radioimmunoassay, Legionella titer, Blood Protein Electrophoresis, and Spectrophotometry.

Diagnostic Procedure 60

A procedure, method, or technique used to determine the nature or identity of a disease or disorder. This excludes procedures which are primarily carried out on specimens in a laboratory. Examples include Electrocardiography, Ultrasonography, Heart Auscultation, and Personality Assessment.

Therapeutic or Preventive Procedure 61

A procedure, method, or technique designed to prevent a disease or a disorder, or to improve physical function, or used in the process of treating a disease or injury. Examples include Cesarean Section, Counseling, Vaccine Therapy, and Cochlear Implant.

15. Unknown/Untyped UNK

Unknown/Untyped 999

A vocabulary concept where the semantic type is either unknown or by its recent addition to the vocabulary, remains untyped. Most untyped concepts acquire a semantic assignment by either further investigation or usage.

25. Appendix C: Integration Control Registrations Detailed

26. Lexicon as a Subscriber

321 MODIFY 'B' XREF OF 757.01

CUSTODIAL PACKAGE: VA FILEMAN
SUBSCRIBING PACKAGE: LEXICON UTILITY

USAGE: Private	ENTERED: DEC 1,1993
STATUS: Expired	EXPIRES: APR 3,2007
DURATION: Next Version	VERSION: LEXICON 1.0
FILE: 757.01	ROOT:

DESCRIPTION: TYPE: File

The FM team grants the request of the Clinical Lexicon package to modify the "B" index of file 757.01 as follows:

```
S ^GMP(757.01,"B",$(($SUP^XLFSTR(X),1,63),DA)=""  
K ^GMP(757.01,"B",$(($SUP^XLFSTR(X),1,63),DA)
```

It is further agreed that the following tools will not be used with this file: DIFROM, COMPARE/MERGE and TRANSFER. These tools rely on an unmodified 'B' index to function properly.

Using the modified 'B' index of file 757.01 along with any of the named tools may produce unexpected results.

345 Read ^DD(file)

CUSTODIAL PACKAGE: VA FILEMAN
SUBSCRIBING PACKAGE: LEXICON UTILITY
USAGE: Private ENTERED: FEB 2,1994
STATUS: Active EXPIRES:
DURATION: Till Otherwise Agr VERSION:
FILE: ROOT: DD(
DESCRIPTION: TYPE: File

Read ^DD(FN), where FN is a file number, to determine the existence of a file prior to initiating a look-up (GMPTA4).

Read ^DD(757*,FLD in indexing routines to obtain the location (node/piece) of data in Clinical Lexicon files 757-757.3 prior to eXecuting Set/Kill logic (GMPTNDX2).

346 Read/Write Access to ^XT(8984.*

CUSTODIAL PACKAGE: TOOLKIT
SUBSCRIBING PACKAGE: LEXICON UTILITY
USAGE: Private ENTERED: FEB 4,1994
STATUS: Active EXPIRES:
DURATION: Till Otherwise Agr VERSION:
FILE: 8984.1 ROOT: XT(8984.1,
DESCRIPTION: TYPE: File

Read only access to ^XT(8984.* globals for \$D checks in the Environment Check routine prior to installing the Clinical Lexicon (GMPTIENV).

i.e. I '\$D(^XT(8984.1)) W !,"Multi-Term Look-Up Utility not found" K DIFQ Q

Read/Write access to ^XT(8984.* global in Post-Init routines to setup the Multi-Term Look-Up Utility for the Clinical Lexicon (GMPTIPST).

i.e.,

- a. Seeding the Local Look-Up file #8984.4 with the Clinical Lexicon Expression file #757.01, the "AWRD" index and the XTLK^GMPTPRNT display routine.
- b. Seeding the Synonym file #8984.3 with Cancer as a sample synonym for Carcinoma
- c. Seeding the Short Cut file #8984.2 with DM II as a sample short cut for Diabetes Mellitus, Non-Insulin Dependent

510 DISV

CUSTODIAL PACKAGE: VA FILEMAN
SUBSCRIBING PACKAGE: LEXICON UTILITY
USAGE: Controlled Subscri ENTERED: JUL 27,1989
STATUS: Active EXPIRES:

DURATION: Till Otherwise Agr VERSION:
FILE: ROOT: DISV(
DESCRIPTION: TYPE: File
Used to process 'space-bar return' on user input.

854 Read/Write Access to ^XT(8984.*

CUSTODIAL PACKAGE: TOOLKIT
SUBSCRIBING PACKAGE: LEXICON UTILITY
USAGE: Private ENTERED: FEB 4,1994
STATUS: Active EXPIRES:
DURATION: Till Otherwise Agr VERSION:
FILE: 8984.2 ROOT: XT(8984.2,
DESCRIPTION: TYPE: File
Read only access to ^XT(8984.2,"B" and the associated
data node ^XT(8984.2,DA,0)

If the user input is found in the "B" cross-reference, and it is a valid "Short Cut" for the Clinical Lexicon - ^XT(8984.2,DA,0) [GMP(757.01 - then the preprocessing of the input string is disabled and the Multi-Term Look-Up Utility (MTLU) is called directly (GMPTA2).

Read only access to ^XT(8984.* globals for \$D checks in the Environment Check routine prior to installing the Clinical Lexicon (GMPTIENV).

i.e. I '\$D(^XT(8984.1)) W !,"Multi-Term Look-Up Utility not found" K DIFQ Q

Read/Write access to ^XT(8984.* global in Post-Init routines to setup the Multi-Term Look-Up Utility for the Clinical Lexicon (GMPTIPST).

i.e.,

Seeding the Local Look-Up file #8984.4 with the Clinical Lexicon Expression file #757.01, the "AWRD" index and the XTLK^GMPTPRNT display routine.

Seeding the Synonym file #8984.3 with Cancer as a sample synonym for Carcinoma

Seeding the Short Cut file #8984.2 with DM II as a sample short cut for Diabetes Mellitus, Non-Insulin Dependent

855 Read/Write Access to ^XT(8984.*

CUSTODIAL PACKAGE: TOOLKIT
SUBSCRIBING PACKAGE: LEXICON UTILITY
USAGE: Private ENTERED: FEB 4,1994
STATUS: Active EXPIRES:
DURATION: Till Otherwise Agr VERSION:
FILE: 8984.3 ROOT: XT(8984.3,

DESCRIPTION: TYPE: File
Read only access to ^XT(8984.* globals for \$D checks in the Environment Check routine prior to installing the Clinical Lexicon (GMPTIENV).

i.e.

I '\$D(^XT(8984.1)) W !,"Multi-Term Look-Up Utility not found" K DIFQ Q

Read/Write access to ^XT(8984.* global in Post-Init routines to setup the Multi-Term Look-Up Utility for the Clinical Lexicon (GMPTIPST).

i.e.,

- a. Seeding the Local Look-Up file #8984.4 with the Clinical Lexicon Expression file #757.01, the "AWRD" index and the XTLK^GMPTPRNT display routine.
- b. Seeding the Synonym file #8984.3 with Cancer as a sample synonym for Carcinoma
- c. Seeding the Short Cut file #8984.2 with DM II as a sample short cut for Diabetes Mellitus, Non-Insulin Dependent

856 Read/Write Access to ^XT(8984.*

CUSTODIAL PACKAGE: TOOLKIT
SUBSCRIBING PACKAGE: LEXICON UTILITY
USAGE: Private ENTERED: FEB 4,1994
STATUS: Active EXPIRES:
DURATION: Till Otherwise Agr VERSION:
FILE: 8984.4 ROOT: XT(8984.4,
DESCRIPTION: TYPE: File

Read only access to ^XT(8984.* globals for \$D checks in the Environment Check routine prior to installing the Clinical Lexicon (GMPTIENV).

i.e.

I '\$D(^XT(8984.1)) W !,"Multi-Term Look-Up Utility not found" K DIFQ Q

Read/Write access to ^XT(8984.* global in Post-Init routines to setup the Multi-Term Look-Up Utility for the Clinical Lexicon (GMPTIPST).

i.e.,

- a. Seeding the Local Look-Up file #8984.4 with the Clinical Lexicon Expression file #757.01, the "AWRD" index and the XTLK^GMPTPRNT display routine.
- b. Seeding the Synonym file #8984.3 with Cancer as a sample synonym for Carcinoma
- c. Seeding the Short Cut file #8984.2 with DM II as a sample short cut for Diabetes Mellitus, Non-Insulin Dependent

857 XTLK Namespace Option

CUSTODIAL PACKAGE: TOOLKIT
SUBSCRIBING PACKAGE: LEXICON UTILITY
USAGE: Private ENTERED: FEB 4,1994
STATUS: Retired EXPIRES:
DURATION: Till Otherwise Agr VERSION:
FILE: ROOT:
DESCRIPTION: TYPE: Other
Agreement is to add XTLK name-spaced Option XTLKUSER2 to the GMPT
CLINICAL LEXICON MGT MENU so managers can add keywords, short-cuts and
synonyms to the ^XT(8984.* files without leaving the Clinical Lexicon
Manager menu.

872 File 101

CUSTODIAL PACKAGE: KERNEL
SUBSCRIBING PACKAGE: LEXICON UTILITY
USAGE: Controlled Subscri ENTERED: APR 28,1994
STATUS: Active EXPIRES:
DURATION: Till Otherwise Agr VERSION:
FILE: 101 ROOT: ORD(101,
DESCRIPTION: TYPE: File
This file may be referenced by packages to maintain protocols within their
namespace. This file may also be pointed to.

888 MTLU setup 8984.1

CUSTODIAL PACKAGE: VA FILEMAN
SUBSCRIBING PACKAGE: LEXICON UTILITY
USAGE: Private ENTERED: MAY 16,1994
STATUS: Active EXPIRES:
DURATION: Till Otherwise Agr VERSION:
FILE: 8984.1 ROOT: DD(8984.1
DESCRIPTION: TYPE: File
The Clinical Lexicon Utility needs to write to the DD of the Kernel
Toolkit Multi-Term Look-up Utility (MTLU) during the Post-Init.

```
^DD(8984.1,.02,'V',D0,0)
.01 FILE 0;1 Both R/W w/Fileman
.02 MESSAGE 0;2 Both R/W w/Fileman
.03 ORDER 0;3 Both R/W w/Fileman
.05 SHOULD ENTRIES BE SC 0;5 Both R/W w/Fileman
.06 SHOULD USER BE ALLOW 0;6 Both R/W w/Fileman
.04 PREFIX 0;4 Both R/W w/Fileman
```

889 MTLU setup 8984.2

CUSTODIAL PACKAGE: VA FILEMAN
SUBSCRIBING PACKAGE: LEXICON UTILITY
USAGE: Private ENTERED: MAY 16,1994
STATUS: Active EXPIRES:
DURATION: Till Otherwise Agr VERSION:
FILE: 8984.2 ROOT: DD(8984.2,
DESCRIPTION: TYPE: File
The Clinical Lexicon Utility needs to write to the DD of Kernel Toolkit
Multi-Term Look-up Utility (MTLU) during the Post-Init.

```

^DD(8984.2,.02,'V',D0,0)
.01      FILE                0;1      Both R/W w/Fileman
.02      MESSAGE             0;2      Both R/W w/Fileman
.03      ORDER               0;3      Both R/W w/Fileman
.05      SHOULD ENTRIES BE SC 0;5      Both R/W w/Fileman
.06      SHOULD USER BE ALLOW 0;6      Both R/W w/Fileman
.04      PREFIX              0;4      Both R/W w/Fileman

```

ROUTINE:

890 MTLU setup 8984.2

```

CUSTODIAL PACKAGE: TOOLKIT
SUBSCRIBING PACKAGE: LEXICON UTILITY
      USAGE: Controlled Subscri  ENTERED: MAY 16,1994
      STATUS: Active              EXPIRES:
      DURATION: Till Otherwise Agr VERSION:
      FILE: 8984.2                ROOT: XT(8984.2,
DESCRIPTION:  TYPE: File
The Clinical Lexicon Utility needs to write to the Kernel Toolkit
Multi-Term Look-up Utility's (MTLU) files/DDs during the Post-Init.

```

```

^XT(8984.2,D0,0)
.01      FREQUENTLY USED NARR 0;1      Both R/W w/Fileman
.02      ENTRY                0;2      Both R/W w/Fileman
.03      CODE                  0;3      Both R/W w/Fileman

```

891 MTLU setup 8984.3

```

CUSTODIAL PACKAGE: TOOLKIT
SUBSCRIBING PACKAGE: LEXICON UTILITY
      USAGE: Controlled Subscri  ENTERED: MAY 16,1994
      STATUS: Active              EXPIRES:
      DURATION: Till Otherwise Agr VERSION:
      FILE: 8984.3                ROOT: XT(8984.3,
DESCRIPTION:  TYPE: File
The Clinical Lexicon Utility needs to write to the Kernel Toolkit
Multi-Term Look-up Utility's (MTLU) files/DDs during the Post-Init.

```

```

^XT(8984.3,D0,0)
.01      TERM                 0;1      Both R/W w/Fileman
.02      ASSOCIATED FILE      0;2      Both R/W w/Fileman
.03      SYNONYM              1;0      Both R/W w/Fileman

```

Multiple

```

^XT(8984.3,D0,1,D1,0)
.01      SYNONYM              0;1      Both R/W w/Fileman

```

916 Read Access to ^DD(file,0,"GL"

```

CUSTODIAL PACKAGE: VA FILEMAN
SUBSCRIBING PACKAGE: LEXICON UTILITY
Read only access for ^DIC(FN,0,"GL"), where FN is a file number,
to verify the value of DIC prior to initiating the lookup (GMPTA4).
      USAGE: Controlled Subscri  ENTERED: JUL 25,1994
      STATUS: Active              EXPIRES:
      DURATION:VERSION: Fileman 20

```


FILE: ROOT: DIC(
DESCRIPTION: TYPE: File
The current packages subscribing to this IA are expected to migrate
to use DID calls. NO NEW FUTURE SUBSCRIBERS WILL BE ADDED.

^DIC(FILE_NO.,0,"GL")
1 GLOBAL NAME Direct Global Read
A direct global read is performed on this node to determine the global
root or a file.

1593 PATIENT CARE ENCOUNTER ^AUTNPOV

CUSTODIAL PACKAGE: PCE PATIENT CARE ENCOUNTER
SUBSCRIBING PACKAGE: LEXICON UTILITY
added 6/8/2011

USAGE: Controlled Subscri ENTERED: AUG 9,1996
STATUS: Active EXPIRES:
DURATION: Till Otherwise Agr VERSION:
FILE: 9999999.27 ROOT: AUTNPOV(
DESCRIPTION: TYPE: File
The purpose of this IA is to allow access to the ^AUTNPOV(global for
purposes of gathering information specific to a problem.

^AUTNPOV(D0,0)
.01 NARRATIVE 0;1 Direct Global Read & W

1611 PROBLEM FILE ^AUPNPROB(

CUSTODIAL PACKAGE: PROBLEM LIST
SUBSCRIBING PACKAGE: LEXICON UTILITY

USAGE: Private ENTERED: AUG 20,1996
STATUS: Active EXPIRES:
DURATION: Till Otherwise Agr VERSION:
FILE: 9000011 ROOT: AUPNPROB(
DESCRIPTION: TYPE: File
This gives the Lexicon Utility the ability to update the ICD codes and the
Lexicon pointer (Problem) in the Problem List application with each new
release of the Lexicon.

^AUPNPROB(D0,0)
.01 DIAGNOSIS 0;1 Both R/W w/Fileman
Pointer to ICD-9 file #80.
^AUPNPROB(D0,1)
1.01 PROBLEM 1;1 Both R/W w/Fileman
Pointer to Expressions file
#757.01.

3779 Access to Domain file 4.2

CUSTODIAL PACKAGE: MAILMAN
SUBSCRIBING PACKAGE: LEXICON UTILITY

USAGE: Controlled Subscri ENTERED: OCT 7,2002
STATUS: Active EXPIRES:
DURATION: Till Otherwise Agr VERSION:
FILE: 4.2 ROOT: DIC(4.2
DESCRIPTION: TYPE: File
Permission is granted to:

1. Perform a FileMan lookup on file #4.2, DOMAIN, using the B and C cross
references.

2. Read the FLAGS field #1, using either direct global access or FileMan read.

```
^DIC(4.2,D0)
  1      FLAGS                0;2      Read w/Fileman
Both direct global reads and
read with FileMan are OK.
  .01    NAME                 0;1      Read w/Fileman
It's OK to look up a domain
name using a FileMan call and
the B and C cross references.
```

3997 Access to File 9999999.27

```
CUSTODIAL PACKAGE: PROBLEM LIST
SUBSCRIBING PACKAGE: LEXICON UTILITY
      USAGE: Private          ENTERED: MAR 12,2003
      STATUS: Withdrawn      EXPIRES:
      DURATION: Till Otherwise Agr VERSION:
      FILE: 9999999.27      ROOT: AUTNPOV(
DESCRIPTION:   TYPE: File
```

4012 Access to File 9.8

```
CUSTODIAL PACKAGE: KERNEL
SUBSCRIBING PACKAGE: LEXICON UTILITY
      USAGE: Private          ENTERED: MAR 18,2003
      STATUS: Withdrawn      EXPIRES:
      DURATION: Till Otherwise Agr VERSION:
      FILE: 9.8              ROOT: DIC(9.8,
DESCRIPTION:   TYPE: File
```

4184 KIDS Install Start/Complete Times

```
CUSTODIAL PACKAGE: KERNEL
SUBSCRIBING PACKAGE: LEXICON UTILITY
The Lexicon needs to use the Kernel's KIDS variable
XPDA to retrieve the Installation Start and
Completion times from the Install File #9.7 to
include in a post-install status message from the
install sites to the Lexicon developers. This
message is used to trouble-shoot various problems in
the field.
```

```
      USAGE: Private          ENTERED: OCT 22,2004
      STATUS: Active          EXPIRES:
      DURATION: Till Otherwise Agr VERSION:
      FILE: 9.7              ROOT: XPD(9.7,
DESCRIPTION:   TYPE: File
^XPD(9.7,XPDA,1)
  11     INSTALL START TIME   1;1     Read w/Fileman
This is the time the install
started
  17     INSTALL COMPLETE TIM 1;3     Read w/Fileman
This is the time the install
finished
```

This file contains the installation information for a site from the Kernel Installation & Distribution System (KIDS). This file is

updated when a KIDS Distribution is installed at a site.

4475 Code Set DD Fixes

CUSTODIAL PACKAGE: VA FILEMAN
SUBSCRIBING PACKAGE: LEXICON UTILITY

USAGE: Private ENTERED: JAN 25,2006
STATUS: Active EXPIRES:
DURATION: Till Otherwise Agr VERSION:
FILE: ROOT: DD(
DESCRIPTION: TYPE: File

During the SQA of patch LEX*2.0*39, several anomalies were discovered with the Lexicon, CPT and ICD data files stemming from the Code Set Versioning and Code Text Descriptors projects. There were several identical fields identified by the cross-references, and a field that points to a non-existing file.

Rather than delete the DD and refresh it, potentially wiping out local mods, the Lexicon team is requesting a one-time permission to write and delete directly from the Data Dictionary.

The code is as follows:

1 File #757.28, Index "ACT" has duplicate Fields

Field .01 ACTIVATION EFFECTIVE DATE
Field 1 ACTIVATION STATUS

```
S ^DD(757.02,1,1,2,0)="757.02^ACT1^MUMPS"  
S ^DD(757.28,.01,1,2,0)="757.02^ACT2^MUMPS"  
S ^DD(757.28,1,1,1,0)="757.02^ACT3^MUMPS"  
K ^DD(757.02,0,"IX","ACT",757.02,1)  
K ^DD(757.02,0,"IX","ACT",757.28,.01)  
K ^DD(757.02,0,"IX","ACT",757.28,1)  
S ^DD(757.02,0,"IX","ACT1",757.02,1)=""  
S ^DD(757.02,0,"IX","ACT2",757.28,.01)=""  
S ^DD(757.02,0,"IX","ACT3",757.28,1)=""
```

2 File #757.02, Index "APCODE" has duplicate Fields

Field 1 EXPRESSION
Field 4 PREFERENCE FLAG

```
S ^DD(757.02,1,1,4,0)="757.02^APCODE1^MUMPS"  
S ^DD(757.02,4,1,1,0)="757.02^APCODE2^MUMPS"  
K ^DD(757.02,0,"IX","APCODE",757.02,1)  
K ^DD(757.02,0,"IX","APCODE",757.02,4)  
S ^DD(757.02,0,"IX","APCODE1",757.02,1)=""  
S ^DD(757.02,0,"IX","APCODE2",757.02,4)=""
```

3 File #81.02, Index "ACT" has duplicate Fields

Field .01 EFFECTIVE DATE
Field .02 STATUS

```
S ^DD(81,.01,1,5,0)="81^ACT1^MUMPS"  
S ^DD(81.02,.01,1,2,0)="81^ACT2^MUMPS"  
S ^DD(81.02,.02,1,1,0)="81^ACT3^MUMPS"
```

```

K ^DD(81,0,"IX","ACT",81,.01)
K ^DD(81,0,"IX","ACT",81.02,.01)
K ^DD(81,0,"IX","ACT",81.02,.02)
S ^DD(81,0,"IX","ACT1",81,.01)=""
S ^DD(81,0,"IX","ACT2",81.02,.01)=""
S ^DD(81,0,"IX","ACT3",81.02,.02)=""

```

4 File #81.33, Index "ACT" has duplicate Fields

```

Field .01 EFFECTIVE DATE
Field .02 STATUS

```

```

S ^DD(81.3,.01,1,3,0)="81.3^ACT1^MUMPS"
S ^DD(81.33,.01,1,2,0)="81.3^ACT2^MUMPS"
S ^DD(81.33,.02,1,1,0)="81.3^ACT3^MUMPS"
K ^DD(81.3,0,"IX","ACT",81.3,.01)
K ^DD(81.3,0,"IX","ACT",81.33,.01)
K ^DD(81.3,0,"IX","ACT",81.33,.02)
S ^DD(81.3,0,"IX","ACT1",81.3,.01)=""
S ^DD(81.3,0,"IX","ACT2",81.33,.01)=""
S ^DD(81.3,0,"IX","ACT3",81.33,.02)=""

```

5 File #80.066, Index "ACT" has duplicate Fields

```

Field .01 EFFECTIVE DATE
Field .02 STATUS

```

```

S ^DD(80,.01,1,4,0)="80^ACT1^MUMPS"
S ^DD(80.066,.01,1,2,0)="80^ACT2^MUMPS"
S ^DD(80.066,.02,1,1,0)="80^ACT3^MUMPS"
K ^DD(80,0,"IX","ACT",80,.01)
K ^DD(80,0,"IX","ACT",80.066,.01)
K ^DD(80,0,"IX","ACT",80.066,.02)
S ^DD(80,0,"IX","ACT1",80,.01)=""
S ^DD(80,0,"IX","ACT2",80.066,.01)=""
S ^DD(80,0,"IX","ACT3",80.066,.02)=""

```

4485 ICD DIAGNOSIS file 80

```

CUSTODIAL PACKAGE: DRG GROUPER
SUBSCRIBING PACKAGE: LEXICON UTILITY
      USAGE: Private          ENTERED: JUL 28,2004
      STATUS: Active         EXPIRES:
      DURATION: Till Otherwise Agr  VERSION:
      FILE: 80ROOT: ICD9(
DESCRIPTION:  TYPE: File
Lexicon Utility has all privileges as though it were the custodial
package.

```

4486 ICD OPERATION/PROCEDURE file 80.1

```

CUSTODIAL PACKAGE: DRG GROUPER
SUBSCRIBING PACKAGE: LEXICON UTILITY
      USAGE: Private          ENTERED: JUL 28,2004
      STATUS: Active         EXPIRES:
      DURATION: Till Otherwise Agr  VERSION:
      FILE: 80.1            ROOT: ICD0(
DESCRIPTION:  TYPE: File
Lexicon Utility has all privileges as though it were the custodial
package.

```

4487 DRG file 80.2

CUSTODIAL PACKAGE: DRG GROUPER
SUBSCRIBING PACKAGE: LEXICON UTILITY
 USAGE: Private ENTERED: JUL 28,2004
 STATUS: Active EXPIRES:
 DURATION: Till Otherwise Agr VERSION:
 FILE: 80.2 ROOT: ICD(
DESCRIPTION: TYPE: File
Lexicon Utility has all privileges as though it were the custodial package.

4488 MAJOR DIAGNOSTIC CATEGORY file 80.3

CUSTODIAL PACKAGE: DRG GROUPER
SUBSCRIBING PACKAGE: LEXICON UTILITY
 USAGE: Private ENTERED: JUL 28,2004
 STATUS: Active EXPIRES:
 DURATION: Till Otherwise Agr VERSION:
 FILE: 80.3 ROOT: ICM(
DESCRIPTION: TYPE: File
Lexicon Utility has all privileges as though it were the custodial package.

4489 CPT file 81

CUSTODIAL PACKAGE: CPT/HCPCS CODES
SUBSCRIBING PACKAGE: LEXICON UTILITY
 USAGE: Private ENTERED: JUL 28,2004
 STATUS: Active EXPIRES:
 DURATION: Till Otherwise Agr VERSION:
 FILE: 81ROOT: ICPT(
DESCRIPTION: TYPE: File
Lexicon Utility has all privileges as though it were the custodial package.

4490 CPT CATEGORY file 81.1

CUSTODIAL PACKAGE: CPT/HCPCS CODES
SUBSCRIBING PACKAGE: LEXICON UTILITY
 USAGE: Private ENTERED: JUL 28,2004
 STATUS: Active EXPIRES:
 DURATION: Till Otherwise Agr VERSION:
 FILE: 81.1 ROOT: DIC(81.1,
DESCRIPTION: TYPE: File
Lexicon Utility has all privileges as though it were the custodial package.

4491 CPT COPYRIGHT file 81.2

CUSTODIAL PACKAGE: CPT/HCPCS CODES
SUBSCRIBING PACKAGE: LEXICON UTILITY
 USAGE: Private ENTERED: JUL 28,2004
 STATUS: Active EXPIRES:
 DURATION: Till Otherwise Agr VERSION:
 FILE: 81.2 ROOT: DIC(81.2,
DESCRIPTION: TYPE: File
Lexicon Utility has all privileges as though it were the custodial package.

4492 CPT MODIFIER file 81.3

CUSTODIAL PACKAGE: CPT/HCPCS CODES
SUBSCRIBING PACKAGE: LEXICON UTILITY
 USAGE: Private ENTERED: JUL 28,2004
 STATUS: Active EXPIRES:
 DURATION: Till Otherwise Agr VERSION:
 FILE: 81.3 ROOT: DIC(81.3,
DESCRIPTION: TYPE: File
Lexicon Utility has all privileges as though it were the custodial package.

4494 MODIFY 'B' XREF OF 757.01

CUSTODIAL PACKAGE: VA FILEMAN
SUBSCRIBING PACKAGE: LEXICON UTILITY
 USAGE: Private ENTERED: APR 3,2007
 STATUS: Active EXPIRES:
 DURATION: Till Otherwise Agr VERSION:
 FILE: 757.01 ROOT:
DESCRIPTION: TYPE: File
The FM team grants the request of the Clinical Lexicon package to modify the "B" index of file 757.01 as follows:

```
S ^LEX(757.01,"B", $E($SUP^XLFSTR(X),1,63),DA)=""  
K ^LEX(757.01,"B", $E($SUP^XLFSTR(X),1,63),DA)
```

It is further agreed that the following tools will not be used with this file: DIFROM, COMPARE/MERGE and TRANSFER. These tools rely on an unmodified 'B' index to function properly. Using the modified 'B' index of file 757.01 along with any of the named tools may produce unexpected results.

4797 MTLU Setup for Code Sets

CUSTODIAL PACKAGE: TOOLKIT
SUBSCRIBING PACKAGE: LEXICON UTILITY
 This IA supersedes previous IA #346, #856 and 887, originally written to support the Clinical Lexicon Utility v 1.0 (in the GMPT namespace). The updated agreement will support Code Sets as implemented in the Lexicon Utility v 2.0 (in the LEX namespace). The Lexicon Utility is responsible for the ICD Code Sets and the CPT Code Sets in files 757.01, 80, 80.1, and 81.
 USAGE: Controlled Subscri ENTERED: SEP 21,2005
 STATUS: Pending EXPIRES:
 DURATION: Till Otherwise Agr VERSION:
 FILE: 8984.4 ROOT: XT(8984.4,
DESCRIPTION: TYPE: File
The Lexicon Utility needs to write to the Kernel Toolkit Multi-Term Look-up Utility's (MTLU) files during a KIDS install/post-init.

```
^XT(8984.4,<file>,0)  
  .01        NAME                   0;1        Both R/W w/Fileman  
  .03        INDEX                  0;3        Both R/W w/Fileman  
^XT(8984.4,<file>,1)  
  .02        DISPLAY PROTOCOL       1;E1,20   Both R/W w/Fileman
```

ROUTINE:

5038 LEXICON Read of ^DD(D0,0,'IX')

CUSTODIAL PACKAGE: VA FILEMAN

SUBSCRIBING PACKAGE: LEXICON UTILITY

Lexicon needs to be able to obtain the field number that contains a classic Fileman cross-reference by a direct global read of the Lexicon's Data Dictionary (DD) 0 node, 'IX' subscripts.

^DD(<file>,0,'IX',<file/sub-file>,<field>)

The Lexicon has become so large that conventional Re-indexing by Kill/Set logic needs to be replaced with Index Repair logic, avoiding the killing of a cross-reference and allowing users to stay on the system without loss of access to the Lexicon package.

USAGE: Private

ENTERED: NOV 6,2007

STATUS: Active

EXPIRES:

DURATION: Till Otherwise Agr VERSION:

FILE: ROOT: DD(

DESCRIPTION: TYPE: File

^DD(D0,0,'IX')

DD(D0,0,'IX') - Where D0 is the number of a Lexicon file, and the 'IX' subscript contains a listing of Lexicon fields that are cross-referenced.

5747 ICD Data Extraction ^ICDEX

CUSTODIAL PACKAGE: DRG GROUPER

SUBSCRIBING PACKAGE: LEXICON UTILITY

The LEXICON UTILITY has access to all APIs listed in this ICR as if it were the Custodial Package.

ACCOUNTS RECEIVABLE

The ACCOUNT RECEIVABLE (PRCA) package will use the following APIs:

\$\$CODEC^ICDEX to return the ICD code based on a file number and an Internal Entry Number (IEN).

\$\$CODECS^ICDEX to return the coding system based on a code in a file on a specified date.

INTEGRATED BILLING

INTEGRATED BILLING will use the following APIs

\$\$SYS^ICDEX to return the current diagnosis code coding system

\$\$CODEABA^ICDEX to return the IEN for a code in a specified coding system

\$\$STATCHK^ICDEX to return the status (active or inactive) of a code and the code's IEN for a code in

a specified coding system

FEE BASIS

The FEE BASIS package will use the following APIs:

\$\$GETDRG^ICDEX to return the DRGs associated with a code and the fiscal year.

\$\$STATCHK^ICDEX to return the status (active or inactive) of a code and the code's IEN for a code in a specified coding system

PROSTHETICS

PROSTHETICS (RMPR) will use the following APIs:

\$\$SINFO^ICDEX to return basic information about the active coding system based on a Date of Service.

\$\$CSI^ICDEX to return the coding system for an Internal Entry Number (IEN) to filter searches by coding system

\$\$STATCHK^ICDEX to return the code status, IEN and the effective date of the status.

\$\$ICDDX^ICDEX to return data about an ICD Diagnosis code.

\$\$VLT^ICDEX to return the versioned long text (description) for either an ICD diagnosis or procedure.

\$\$LS^ICDEX to return the last status based on date regardless of coding system.

\$\$CODEC^ICDEX to return an ICD code based on file number and Internal Entry Number (IEN).

SCHEDULING

SCHEDULING (SD) will use the following APIs

\$\$IMP^ICDEX to return the date a coding system was implemented. This date will be used as parameter of an input transformation when entering ICD codes.

\$\$CSI^ICDEX to return the coding system based on an Internal Entry Number (IEN) used to identify commonly used ICD codes as either ICD-9 or ICD-10.

\$\$VER^ICDEX to return either the current version, the next version or the previous version of the ICD coding system.

\$\$SYS^ICDEX to return the current coding system based on a file number (80/80.1) and date.

\$\$LS^ICDEX to return the last status based on date

regardless of coding system

\$\$ICDDX^ICDEX to return information about a diagnosis code

REGISTRATION

REGISTRATION (DG) will use the following APIs:

\$\$CSI^ICDEX to return the coding system based on an Internal Entry Number (IEN) used to identify ICD codes as either ICD-9 or ICD-10.

\$\$CODEC^ICDEX to return an ICD code based on file number and Internal Entry Number (IEN).

\$\$CODEN^ICDEX to return the Internal Entry Number (IEN) of a code based on a code and file number.

\$\$CODEABA^ICDEX to return the IEN for a code in a specified coding system (this is the preferred method of obtaining an IEN for a code)

\$\$LS^ICDEX to return the last status for an Internal Entry Number (IEN) based on a date.

\$\$NOT^ICDEX to return a list of codes not to be use with a code passed as a parameter.

\$\$REQ^ICDEX to return a list of codes required with a code passed as a parameter.

\$\$SYS^ICDEX to return a coding system based on the contents of the Coding System file 80.4, a mnemonic, a pattern match or a code

\$\$VLT^ICDEX to return the versioned long text (aka description) based on a file number, Internal Entry Number (IEN) and date.

\$\$SINFO^ICDEX to return basic information about the active coding system based on a Date of Service

CLINICAL REMINDERS

CLINICAL REMINDERS (PXR) will use the following APIs:

\$\$CODEN^ICDEX to return the Internal Entry Number (IEN) of a code based on a code and file number

\$\$CODEABA^ICDEX to return the IEN for a code in a specified coding system (this is the preferred method of obtaining an IEN for a code)

\$\$ICDDX^ICDEX to return information about an ICD diagnosis code.

\$\$ICDOP^ICDEX to return information about an ICD

procedure code.

\$\$NEXT^ICDEX to return the next code in a coding system.

\$\$PREV^ICDEX to return the previous code in a coding system.

\$\$IMP^ICDEX to return the implementation date of a coding system.

\$\$ROOT^ICDEX to return the global root of a coding system.

\$\$HDR^ICDEX to return the header nodes of files 80 and 80.1.

\$\$CODEC^ICDEX to return the ICD code based on a file number and an Internal Entry Number (IEN)

\$\$CSI^ICDEX to return the coding system for an Internal Entry Number (IEN) to filter searches by coding system

\$\$SINFO^ICDEX to return basic information about the active coding system based on a Date of Service

PHARMACY BENEFITS MANAGEMENT

PHARMACY BENEFITS MANAGEMENT (PSU) will use the following APIs:

\$\$CSI^ICDEX to return the coding system based on an Internal Entry Number (IEN) used to identify commonly used ICD codes as either ICD-9 or ICD-10.

\$\$ICDDX^ICDEX to return information about an ICD diagnosis code.

\$\$ICDOP^ICDEX to return information about an ICD procedure code.

CLINICAL CASE REGISTRIES

The CLINICAL CASE REGISTRIES (ROR) will use the following APIs:

\$\$CSI^ICDEX to return the coding system for an Internal Entry Number (IEN) to filter searches by coding system

\$\$VSEX^ICDEX to return the sex designation for a code.

\$\$UPDX^ICDEX to determine if a code is unacceptable as a principle diagnosis.

\$\$CODEC^ICDEX to return an ICD code based on file number and Internal Entry Number (IEN).

\$\$CODEABA^ICDEX to return the IEN for a code in a specified coding system

\$\$VSTD^ICDEX to return the Versioned Short Text for a diagnosis on a specified date.

\$\$VLTD^ICDEX to return the Versioned Long Text for a diagnosis on a specified date.

\$\$VSTP^ICDEX to return the versioned Short Text for a procedure on a specified date.

\$\$VLTP^ICDEX to return the versioned Long Text for a procedure on a specified date.

\$\$FILE^ICDEX to return a file number based on a coding system, global root, filename or file identifier.

\$\$VLT^ICDEX to return the versioned long text (aka description) based on a file number, Internal Entry Number (IEN) and date.

\$\$VST^ICDEX to return the versioned short text based on a file number, Internal Entry Number (IEN) and date.

\$\$CODEN^ICDEX to return the Internal Entry Number (IEN) of a code based on a code and file number.

\$\$ICDDX^ICDEX to return information about an ICD diagnosis code.

\$\$ICDOP^ICDEX to return information about an ICD procedure code.

\$\$SNAM^ICDEX to return the name of the coding system.

CLINICAL PROCEDURES

CLINICAL PROCEDURES (MD) will use the following APIs:

\$\$ICDDX^ICDEX to return data about an ICD Diagnosis code.

\$\$CSI^ICDEX to return the coding system for an Internal Entry Number (IEN) to filter searches by coding system

\$\$IMP^ICDEX to return the implementation date of a coding system.

\$\$SINFO^ICDEX to return basic information about the active coding system based on a Date of Service.

SPINAL CORD DYSFUNCTION

The SPINAL CORD DYSFUNCTION (SPN) package will use

the following APIs:

\$\$OBA^ICDEX to replace cross-reference access to a specified code file.

\$\$CODEBA^ICDEX to return the IEN of a code found in a specified code file.

\$\$CSI^ICDEX to return the coding system based on an Internal Entry Number (IEN) used to identify commonly used ICD codes as either ICD-9 or ICD-10.

\$\$CODEABA^ICDEX to return the IEN for a code in a specified coding system

\$\$VLT^ICDEX to return the Versioned Long Text for an diagnosis or procedure on a specified date.

\$\$VST^ICDEX to return the Versioned Short Text for an diagnosis or procedure on a specified date.

HOSPITAL BASED HOME CARE

Home-Based Primary Care (HBH) will use the following API:

\$\$SYS^ICDEX to return a coding system based on the contents of the Coding System file 80.4, a mnemonic, a pattern match or a code

\$\$CODEC^ICDEX to return an ICD code based on file number and Internal Entry Number (IEN).

\$\$VSTD^ICDEX to return the Versioned Short Text for a diagnosis on a specified date.

\$\$SAI^ICDEX to return the status, active date and inactive date based on a file, an Internal Entry Number (IEN) an a date.

\$\$CSI^ICDEX to return the coding system for an Internal Entry Number (IEN) to filter searches by coding system

EVENT CAPTURE

The EVENT CAPTURE (EC) package will use the following APIs:

\$\$SINFO^ICDEX to return basic information about the active coding system based on a Date of Service.

\$\$ICDDX^ICDEX to return data about an ICD Diagnosis code.

AUTOMATED INFO COLLECTION SYS

The AUTOMATED INFO COLLECTION SYS (IBD) package will use the following APIs:

\$\$SINFO^ICDEX to return basic information about the active coding system based on a Date of Service.

LAB SERVICE

The LAB SERVICES (LR) will use the following APIs:

\$\$CODEC^ICDEX to return the ICD code based on a file number and an Internal Entry Number (IEN).

\$\$ICDDX^ICDEX to return information about an ICD diagnosis code.

\$\$ICDOP^ICDEX to return information about an ICD procedure code.

\$\$IMP^ICDEX to return the implementation date of a coding system.

\$\$SINFO^ICDEX to return basic information about the active coding system based on a Date of Service.

\$\$CSI^ICDEX to return the coding system based on the internal entry number.

QUASAR

QUASAR (ACKQ) will use the following APIs:

\$\$CODEC^ICDEX to return the ICD code based on a file number and an Internal Entry Number (IEN).

\$\$CSI^ICDEX to return the coding system for an Internal Entry Number (IEN) to filter searches by coding system

EMERGENCY DEPARTMENT

The EMERGENCY DEPARTMENT (EDP) package will use the following APIs:

\$\$ICDDX^ICDEX to return information about an ICD diagnosis code.

\$\$ICDOP^ICDEX to return information about an ICD procedure code.

PROBLEM LIST

PROBLEM LIST (GMPL) will use the following APIs:

\$\$CODEC^ICDEX to return the ICD code based on a file number and an Internal Entry Number (IEN).

\$\$SINFO^ICDEX to return basic information about the active coding system based on a Date of Service.

\$\$SDH^ICDEX to return a history of short descriptions for a file number and Internal Entry Number (IEN)

\$\$LDH^ICDEX to return a history of long descriptions

for a file number and Internal Entry Number (IEN)

USAGE: Controlled Subscri ENTERED: NOV 6,2011

STATUS: Pending EXPIRES:

DURATION: Till Otherwise Agr VERSION:

DESCRIPTION: TYPE: Routine

Application Programmer Interfaces (APIs) in this routine were developed to remove the need for direct global access to either the DIAGNOSIS file 80 or OPERATIONS/PROCEDURE file 80.1.

These entry points are meant to replace the following active/retired ICRs:

48	Private	YS File 80.2	Weight (2)
280	Private	HBH File 80	Code (.01)
365	Private	QAM File 80	Code (.01)
368	Private	IB File 80	Retired Nov 15, 2008
369	Private	IB File 80.1	Retired Nov 15, 2008
370	Private	IB/DSS 80.2	DRG Name (.01)
582	Private	IMR File 80	Code (.01)
647	Private	IB File 80	Retired Nov 15, 2008
1161	Private	VAM File 80	Retired Nov 15, 2008
1275	Private	GMTS File 80	Retired Nov 15, 2008
1276	Private	GMTS File 80.1	Retired Nov 15, 2008
1294	Subscription	PCE/TIU/OR File 80	Retired Nov 15, 2008
1487	Private	ACKQ File 80	Retired Nov 15, 2008
1586	Subscription	AICS/PCE File 80.3	MDC Name (.01)
2435	Private	PXRM File 80 Hdr	^ICD9(0)
2436	Private	PXRM File 80.1 Hdr	^ICD0(0)
3990	Supported	Routine ICDCODE	
3991	Supported	Routine ICDAPIU	
4052	Supported	Routine ICDGTDRG	
5028	Subscription	GMPL File 80	
5388	Supported	File 80	Code (.01), AB/BA/D/AST/ACT
5404	Supported	File 80.1	Code (.01), BA/ACT
5699	Supported	Routine ICDXCODE	
5757	Supported	Routine ICDSAPI	
10082	Supported	File 80	Retired Nov 15, 2008
10083	Supported	File 80.1	Retired Nov 15, 2008

ROUTINE: ICDEX

COMPONENT: HELP

This is an interactive help entry point for the input and output variables for the APIs contained in the routine ICDEX.

COMPONENT: \$\$ICDDX(CODE,CDT,SYS,FMT)

This entry point extracts data for an ICD-9 or ICD-10 code in the DIAGNOSIS file 80.

This entry point is intended to replace the ICD-9 Legacy API \$\$ICDDX^ICDCODE (ICR 3990) and \$\$ICDDATA^ICDXCODE (ICR 5699), providing a single point of entry for ICD diagnostic data.

VARIABLES: Input CODE

This is an ICD diagnosis code in either the external or internal format. If the internal

format is used, then the input variable FMT must be set to "I" (Required).

- VARIABLES: Input CDT
This is the Code Set Versioning date (Fileman format) used to identify the code and text that was appropriate for the date passed in this input parameter. (Optional, if not supplied, TODAY will be used)
- VARIABLES: Input SYS
This is an ICD coding system identifier (taken from file 80.4). The following coding systems are found in file 80:
- 1 = ICD-9 Diagnosis
 - 30 = ICD-10 Diagnosis
- (Optional, but highly encouraged)
- VARIABLES: Input FMT
This variable tells the API if the CODE is in External or Internal format.
- "E" = External (default)
 - "I" = Internal Entry Number
- (Conditional, required if CODE is in internal format)
- VARIABLES: Output \$\$ICDDX
This is a 20 piece string delimited by "^"
- 1 IEN of code in ^ICD9(
 - 2 ICD-9 Dx Code (#.01)
 - 3 Identifier (#1.2)
 - 4 Versioned Dx (67 multiple)
 - 5 Unacceptable as Principal Dx (#1.3)
 - 6 Major Dx Cat (72 multiple)
 - 7 MDC13 (#1.4)
 - 8 Compl/Comorb (103 multiple)
 - 9 ICD Expanded (#1.7)
 - 10 Status (66 multiple)
 - 11 Sex (10 multiple)
 - 12 Inactive Date (66 multiple)
 - 13 MDC24 (#1.5)
 - 14 MDC25 (#1.6)
 - 15 Age Low (11 multiple)
 - 16 Age High (12 multiple)
 - 17 Activation Date (66 multiple)
 - 18 Message
 - 19 Complication/Comorbidity (103 multiple)
 - 20 Coding System (#1.1)
 - 21 Primary CC Flag (103 multiple)
 - 22 PDX Exclusion Code (#1.11)
- or

-1^Error Description

COMPONENT: \$\$ICDOP(CODE,CDT,SYS,FMT)
This entry point extracts data for an ICD-9 or ICD-10 code in the OPERATIONS/PROCEDURE file 80.1

This entry point is intended to replace the ICD-9 Legacy API \$\$ICDOP^ICDCODE (ICR 3990) and \$\$ICDDATA^ICDXCODE (ICR 5699), providing a single point of entry for ICD procedural data.

VARIABLES: Input CODE
This is an ICD operation/procedure code in either the external or internal format. If the internal format is used, then the input variable FMT must be set to "I" (Required)

VARIABLES: Input CDT
This is the Code Set Versioning date (Fileman format) used to identify the code and text that was appropriate for the date passed in CDT. (Optional, if not supplied, TODAY will be used)

VARIABLES: Input SYS
This is an ICD coding system identifier (taken from file 80.4). The following coding systems are found in file 80.1:

2 = ICD-9 Procedures
31 = ICD-10 Procedures

(Optional, but highly encouraged)

VARIABLES: Input FMT
This variable tells the API if the CODE is in External or Internal format.

"E" = External (default)
"I" = Internal Entry Number

(Conditional, required if CODE is in internal format)

VARIABLES: Output \$\$ICDOP
This is a 15 piece string delimited by "^"

1	IEN of code in ^ICD0(
2	ICD procedure code	(#.01)
3	Identifier	(#.1.2)
4	MDC24	(#.1.5)
5	Versioned Oper/Proc	(67 multiple)
6	<null>	
7	<null>	
8	<null>	
9	ICD Expanded	(#.1.7)
10	Status	(66 multiple)

11	Use with Sex	(10 multiple)
12	Inactive Date	(66 multiple)
13	Activation Date	(66 multiple)
14	Message	
15	Coding System	(#1.1)

or

-1^Error Description

COMPONENT: \$\$ICDD(CODE, .ARY, CDT, SYS, LEN)

This API returns the long description of either an ICD-9 or ICD-10 code.

This entry point is intended to replace the ICD-9 Legacy API \$\$ICDD^ICDCODE (ICR 3990) and \$\$ICDDESC^ICDXCODE (ICR 5699), providing a single point of entry for ICD diagnosis/procedure descriptions.

VARIABLES: Input CODE
This is an ICD-9 or ICD-10 code in external format only (Required).

VARIABLES: Input .ARY
This is the name of a local array, passed by reference that will contain the output of this API. (Required)

VARIABLES: Input CDT
This is the Code Set Versioning date (Fileman format) used to identify the text that was appropriate for the date passed in this input parameter. (Optional, if not supplied, TODAY will be used)

VARIABLES: Input SYS
This is an ICD coding system identifier (taken from file 80.4). The following coding systems are found in files 80 and 80.1:

- 1 = ICD-9 Diagnosis file 80
- 2 = ICD-9 Procedure file 80.1
- 30 = ICD-10 Diagnosis file 80
- 31 = ICD-10 Procedure file 80.1

(Optional, but highly encouraged)

VARIABLES: Input LEN
This is the text string length of the description placed in array .ARY. (Optional, if passed it must be greater than 27 based on the longest word found in a diagnosis or procedure description and not greater than 245. If not passed it defaults to 245 characters based in the input transformation)

VARIABLES: Output \$\$ICDD

This is the number of lines in the output array
.ARY or if an error occurs, -1^Error Message

VARIABLES: Output ARY

This is a local array, passed by reference, containing the long description of an ICD code with string lengths defined by LEN when passed or 245 characters. If there is a warning message about text accuracy (ICD-9 only) it will be appended to the end of the message preceded by a blank line.

ARY(1) - Description (length of LEN)
ARY(n) - Description (continued if necessary)

If there is a warning message (ICD-9 only):

ARY(n+1) - blank
ARY(n+2) - message: CODE TEXT MAY BE INACCURATE

COMPONENT: \$\$CODEN(CODE,FILE)

This API returns the Internal Entry Number (IEN) of a ICD code.

This entry point is intended to replace the ICD-9 Legacy API \$\$CODEN^ICDCODE (ICR 3990). It is also intended to replace the need for direct global access of the 'BA' cross-reference in ICRs 5388 and 5404.

VARIABLES: Input CODE

This is an ICD-9 or ICD-10 code in external format only (Required).

VARIABLES: Input FILE

This is the file number where the CODE is stored, either 80 or 80.1 (Required)

VARIABLES: Output \$\$CODEN

This is the Internal Entry Number (IEN) of CODE in file FILE appended by a tilde "~" and the global root FILE:

IEN~^ROOT

or -1^Error Message on error

COMPONENT: \$\$CODEC(FILE,IEN)

This entry point returns the ICD-9 or ICD-10 code from a specified ICD file and Internal Entry Number (IEN).

This entry point is intended to replace the ICD-9 Legacy API \$\$CODEC^ICDCODE (ICR 3990). It is also intended to replace the need for direct global access in ICRs 280, 365, 582, 5388, and 5404.

VARIABLES: Input FILE

This is the ICD file number used to retrieve the

code (Required)

80 = ICD Diagnosis file

80.1 = ICD Operation/Procedure file

VARIABLES: Input IEN
This is the internal entry number in FILE were the code to be retrieved is stored (Required)

VARIABLES: Output \$\$CODEC
This is either the ICD code stored at the Internal Entry Number IEN in the file identified by the FILE input parameter, or upon error:

-1 ^ Error Message

COMPONENT: \$\$CODEBA (CODE, ROOT)
This entry point returns the internal entry number (IEN) of a code found in the 'BA' cross-reference in the file specified.

This entry point is provided in lieu of ICD-9 Legacy entry point \$\$CODEN^ICDCODE (ICR 3990) which will crash with a <MAXNUMBER> error if the code passed has the letter 'E' in the middle of the code (example, ICD-10 procedure code 041E499 would be interpreted as scientific notation). \$\$CODEBA^ICDEX is much safer.

If you already know the coding system, please use \$\$CODEABA^ICDEX instead.

This entry point replaces the need for direct global read access of the 'BA' cross-reference allowed by ICRs 5388 and 5404.

VARIABLES: Input CODE
This is either an ICD Diagnosis code or ICD Procedure code (Required)

VARIABLES: Input ROOT
This is the global root (or file number) where the code is stored (Required)

VARIABLES: Output \$\$CODEBA
This is the internal entry number (IEN) in the specified file where the code is stored or -1 if not found.

COMPONENT: \$\$CODEABA (CODE, ROOT, SYS)
This entry point returns the internal entry number (IEN) of a code found in the system specific 'ABA' cross-reference in the file specified.

This entry point is provided in lieu of ICD-9 Legacy entry point \$\$CODEN^ICDCODE (ICR 3990) and new entry point \$\$CODEABA^ICDEX.

Entry point Comparison:

\$\$CODEN^ICDCODE will crash if the code has the letter 'E' in the middle of the code. Do not use it.

\$\$CODEBA^ICDEX is safer but it will fail to return the correct IEN if ICD-9 and ICD-10 ever have a similar code.

\$\$CODEABA^ICDEX will neither crash or fail to return the correct IEN.

- VARIABLES: Input CODE
This is either an ICD Diagnosis code or ICD Procedure code (Required)
- VARIABLES: Input ROOT
This is the global root (or file number) where the code is stored (Optional if SYS is supplied)
- VARIABLES: Input SYS
This is an ICD coding system identifier (taken from file 80.4). The following coding systems are found in files 80 and 80.1:
- 1 = ICD-9 Diagnosis file 80
 - 2 = ICD-9 Procedure file 80.1
 - 30 = ICD-10 Diagnosis file 80
 - 31 = ICD-10 Procedure file 80.1
- This API will look for the code on one of the system specific cross-references:
- ^ICD9("ABA",1,CODE,IEN) ICD-9 Diagnosis
 - ^ICD9("ABA",30,CODE,IEN) ICD-10 Diagnosis
 - ^ICD0("ABA",2,CODE,IEN) ICD-9 Procedure
 - ^ICD0("ABA",31,CODE,IEN) ICD-10 Procedure
- If not supplied, the API will attempt to determine the system based on code and file.
- (Optional, but highly encouraged)
- VARIABLES: Output \$\$CODEABA
This is the internal entry number (IEN) in the specified file where the code is stored or -1 if not found.
- COMPONENT: \$\$CODEFI(CODE)
This entry point tries to resolve which file has an ICD code on file.
- VARIABLES: Input CODE
This is either an ICD Diagnosis code or ICD Procedure code (Required)
- VARIABLES: Output \$\$CODEFI

This is the ICD file number where the specified code was found:

80 = ICD Diagnosis file
80.1 = ICD Operation/Procedure file

or NULL if not found or could not resolve to a single file.

COMPONENT: \$\$CODECS (CODE, FILE, CDT)

This entry point tries to resolve the Coding System based on a code, a file and a date.

VARIABLES: Input CODE

This is either an ICD Diagnosis code or ICD Procedure code (Required)

VARIABLES: Input FILE

This is the ICD file number used to resolve the coding system:

80 = ICD Diagnosis file
80.1 = ICD Operation/Procedure file

(Optional, but encouraged) If not supplied, an attempt to resolve the input variable FILE will be made using the entry point \$\$CODEFI(CODE).

VARIABLES: Input CDT

This is the Code Set Versioning date (Fileman format) used to resolve the coding system.

This date is ONLY used if a code is found in both ICD-9 and ICD-10 systems. If that ever happens, the date passed will determine the coding system. If the date passed is before the ICD-10 implementation date then it will be considered an ICD-9 code and if it is on or after the ICD-10 implementation date then it will be considered ICD-10.

VARIABLES: Output \$\$CODECS

This is a 2 piece "^" delimited string containing:

1 Coding System (pointer to file 80.4)
2 Coding Nomenclature (commonly used name)

Example output values:

1^ICD-9-CM
30^ICD-10-CM
2^ICD-9 Proc
31^ICD-10-PCS

NULL if the API cannot resolve the coding system based on code, file and date.

COMPONENT: \$\$CSI(FILE,IEN)
This entry point returns the Coding System for an Internal Entry Number (IEN).

VARIABLES: Input FILE
This is the ICD file number used to retrieve the coding system (Required):
80 = ICD Diagnosis file
80.1 = ICD Operation/Procedure file

VARIABLES: Input IEN
This is an Internal Entry Number (IEN) in the file specified (Required).

VARIABLES: Output \$\$CSI
This is a pointer to the ICD CODING SYSTEMS file #80.4

COMPONENT: \$\$VMDC(IEN,CDT,FMT)
This entry point retrieves the versioned Major Diagnostic Category (MDC) for a diagnostic code in the DIAGNOSIS file 80.

VARIABLES: Input IEN
This is an Internal Entry Number (IEN) in the DIAGNOSIS file 80 (Required)

VARIABLES: Input CDT
This is the Code Set Versioning date (Fileman format) used to identify the Major Diagnostic Category that was appropriate for the date passed (Optional, if not passed TODAY is used).

VARIABLES: Input FMT
This is a flag used to determine the output format. Acceptable values are 0 and 1 (Optional, default value is 0).
FMT = 0 Major Diagnostic Category (MDC)
FMT = 1 MDC^Effective Date

VARIABLES: Output \$\$VMDC
This is the Major Diagnostic Category (MDC) that was appropriate for the date passed and the diagnosis code identified by input parameter IEN. The output may also have a second "^" delimited piece containing the MDC Effective Date if the input parameter FMT is set to 1.

COMPONENT: \$\$VAGEL(IEN,CDT,FMT)
This entry point retrieves the versioned Age Low value for a diagnostic code in the DIAGNOSIS file 80. Age Low is the minimum age value for an age range for which the diagnostic code can be applied.

VARIABLES: Input IEN
This is an Internal Entry Number (IEN) in the

DIAGNOSIS file 80 (Required)

VARIABLES: Input CDT
This is the Code Set Versioning date (Fileman format) used to identify the Age Low value that was appropriate for the date passed (Optional, if not passed TODAY is used).

VARIABLES: Input FMT
This is a flag used to determine the output format. Acceptable values are 0 and 1 (Optional, default value is 0).

FMT = 0 Age Low
FMT = 1 Age Low^Effective Date

VARIABLES: Output \$\$VAGEL
This is the Age Low that was appropriate for the date passed and the diagnosis code identified by the input parameter IEN. The output may also have a second "^" delimited piece containing the Age Low Effective Date if the input parameter FMT is set to 1. Null if Age Low not found for date.

COMPONENT: \$\$VAGEH (IEN, CDT, FMT)
This entry point retrieves the versioned Age High value for a diagnostic code in the DIAGNOSIS file 80. Age High is the maximum age value for an age range for which the diagnostic code can be applied.

VARIABLES: Input IEN
This is an Internal Entry Number (IEN) in the DIAGNOSIS file 80 (Required)

VARIABLES: Input CDT
This is the Code Set Versioning date (Fileman format) used to identify the Age High value that was appropriate for the date passed (Optional, if not passed TODAY is used).

VARIABLES: Input FMT
This is a flag used to determine the output format. Acceptable values are 0 and 1 (Optional, default value is 0).

FMT = 0 Age High
FMT = 1 Age High^Effective Date

VARIABLES: Output \$\$VAGEH
This is the Age High that was appropriate for the date passed and the diagnosis code identified by the input parameter IEN. The output may also have a second "^" delimited piece containing the Age High Effective Date if the input parameter FMT is set to 1. Null if Age High is not found for date.

COMPONENT: \$\$VCC (IEN, CDT, FMT)

This entry point retrieves the versioned Complication Comorbidity (CC) designation for a diagnostic code in the DIAGNOSIS file 80. A diagnostic code can be designated as:

Non-Complication Comorbidity (Non-CC)
Complication Comorbidity (CC)
Major Complication Comorbidity (MCC)

VARIABLES: Input IEN
This is an Internal Entry Number (IEN) in the DIAGNOSIS file 80 (Required)

VARIABLES: Input CDT
This is the Code Set Versioning date (Fileman format) used to identify the CC designation value that was appropriate for the date passed (Optional, if not passed TODAY is used).

VARIABLES: Input FMT
This is a flag used to determine the output format. Acceptable values are 0 and 1 (Optional, default value is 0).

FMT = 0 CC designation
FMT = 1 CC designation^Effective Date

VARIABLES: Output \$\$VCC
This is the CC designation that was appropriate for the date passed and the diagnosis code identified by the input parameter IEN. The output may also have a second "^" delimited piece containing the CC designation Effective Date if the input parameter FMT is set to 1.

0 = Non-Complication Comorbidity (Non-CC)
1 = Complication Comorbidity (CC)
2 = Major Complication Comorbidity (MCC)

Null if not found for date

COMPONENT: \$\$VSEX(FILE, IEN, CDT, FMT)
This entry point retrieves the versioned sex designation for a diagnostic or procedure code in either the ICD DIAGNOSIS file 80 or the ICD OPERATION/PROCEDURE file 80.1. If a sex designation exist then the diagnosis or procedure should be applied only to that sex.

VARIABLES: Input FILE
This is the ICD file number used to retrieve the sex designation:

80 = ICD Diagnosis file
80.1 = ICD Operation/Procedure file

VARIABLES: Input IEN
This is an Internal Entry Number (IEN) in either the DIAGNOSIS file 80 or OPERATION/PROCEDURE file

80.1 (Required)

VARIABLES: Input CDT
This is the Code Set Versioning date (Fileman format) used to identify the sex designation value that was appropriate for the date passed (Optional, if not passed TODAY is used).

VARIABLES: Input FMT
This is a flag used to determine the output format. Acceptable values are 0 and 1 (Optional, default value is 0).

FMT = 0 Sex designation
FMT = 1 Sex designation^Effective Date

VARIABLES: Output \$\$VSEX
This is the sex designation that was appropriate for the date passed and the code identified by the input parameter IEN. The output may also have a second "^" delimited piece containing the sex designation Effective Date if the input parameter FMT is set to 1.

M = Male
F = Female
Null if sex is N/A or not found for date

COMPONENT: \$\$\$SAI (FILE, IEN, CDT)
This entry point retrieves the Status, Activation date and Inactivation date for a diagnosis or procedure on a specified date.

VARIABLES: Input FILE
This is the ICD file number used to retrieve the status and effective dates:

80 = ICD Diagnosis file
80.1 = ICD Operation/Procedure file

VARIABLES: Input IEN
This is an Internal Entry Number (IEN) in either the DIAGNOSIS file 80 or OPERATION/PROCEDURE file 80.1 (Required)

VARIABLES: Input CDT
This is the Code Set Versioning date (Fileman format) used to retrieve the status and effective dates that were appropriate for the date passed (Optional, if not passed TODAY is used).

VARIABLES: Output \$\$\$SAI
This is a 6 piece "^" delimited string

1 Status
2 Activation Date
3 Inactivation Date

- 4 IEN
- 5 Code
- 6 Short Text

If the status is active, the short text will be the most recent.

If the status is inactive, the short text will be the text in use on the date it was inactivated.

Null if no status for date.

COMPONENT: \$\$VST(FILE, IEN, CDT)
This entry point retrieves the Versioned Short Text for an diagnosis or procedure on a specified date.

VARIABLES: Input FILE
This is the ICD file number used to retrieve the Versioned Short Text:

- 80 = ICD Diagnosis file
- 80.1 = ICD Operation/Procedure file

VARIABLES: Input IEN
This is an Internal Entry Number (IEN) in either the DIAGNOSIS file 80 or OPERATION/PROCEDURE file 80.1 (Required)

VARIABLES: Input CDT
This is the Code Set Versioning date (Fileman format) used to retrieve the Versioned Short Text that was appropriate for the date passed (Optional, if not passed TODAY is used).

VARIABLES: Output \$\$VST
This is the Versioned Short Text from either file 80 (DIAGNOSIS) or 80.1 (OPERATION/PROCEDURE) that was appropriate for the date passed and the code identified by the input parameter IEN. Null if not found.

COMPONENT: \$\$VLT(FILE, IEN, CDT)
This entry point retrieves the Versioned Long Text (description) for a diagnosis or procedure on a specified date.

VARIABLES: Input FILE
This is the ICD file number used to retrieve the Versioned Long Text (description):

- 80 = ICD Diagnosis file
- 80.1 = ICD Operation/Procedure file

VARIABLES: Input IEN
This is an Internal Entry Number (IEN) in either the DIAGNOSIS file 80 or OPERATION/PROCEDURE file

80.1 (Required)

VARIABLES: Input CDT
This is the Code Set Versioning date (Fileman format) used to retrieve the Versioned Long Text (description) that was appropriate for the date passed (Optional, if not passed TODAY is used).

VARIABLES: Output \$\$VLT
This is the Versioned Long Text (description) from either file 80 or 80.1 that was appropriate for the date passed and the code identified by the input parameter IEN. Null if not found.

COMPONENT: \$\$VSTD(IEN,CDT)
This entry point retrieves the Versioned Short Text for a diagnosis on a specified date.

VARIABLES: Input IEN
This is an Internal Entry Number (IEN) in the DIAGNOSIS file 80 (Required)

VARIABLES: Input CDT
This is the Code Set Versioning date (Fileman format) used to retrieve the Versioned Short Text that was appropriate for the date passed (Optional, if not passed TODAY is used).

VARIABLES: Output \$\$VSTD
This is the Versioned Short Text from file 80 that was appropriate for the date passed and the code identified by the input parameter IEN. Null if not found.

COMPONENT: \$\$VSTP(IEN,CDT)
This entry point retrieves the Versioned Short Text for a procedure on a specified date.

VARIABLES: Input IEN
This is an Internal Entry Number (IEN) in the OPERATION/PROCEDURE file 80.1 (Required)

VARIABLES: Input CDT
This is the Code Set Versioning date (Fileman format) used to retrieve the Versioned Short Text that was appropriate for the date passed (Optional, if not passed TODAY is used).

VARIABLES: Output \$\$VSTP
This is the Versioned Short Text from file 80.1 that was appropriate for the date passed and the code identified by the input parameter IEN. Null if not found.

COMPONENT: \$\$VLTD(IEN,CDT)
This entry point retrieves the Versioned Long Text (description) for a diagnosis on a specified date.

VARIABLES: Input IEN
This is an Internal Entry Number (IEN) in the
DIAGNOSIS file 80 (Required)

VARIABLES: Input CDT
This is the Code Set Versioning date (Fileman
format) used to retrieve the Versioned Long Text
(description) that was appropriate for the date
passed (Optional, If not passed TODAY is used).

VARIABLES: Output \$\$VLTD
This is the Versioned Long Text (description) from
file 80 that was appropriate for the date passed
and the code identified by the input parameter
IEN. Null if not found.

COMPONENT: \$\$VLTP(IEN,CDT)
This entry point retrieves the Versioned Long Text
(description) for a procedure on a specified date.

VARIABLES: Input IEN
This is an Internal Entry Number (IEN) in the
OPERATION/PROCEDURE file 80.1 (Required)

VARIABLES: Input CDT
This is the Code Set Versioning date (Fileman
format) used to retrieve the Versioned Long Text
(description) that was appropriate for the date
passed (Optional, If not passed TODAY is used).

VARIABLES: Output \$\$VLTP
This is the Versioned Long Text (description) from
file 80.1 that was appropriate for the date passed
and the code identified by the input parameter
IEN. Null if not found.

COMPONENT: \$\$\$SD(FILE,IEN,CDT,.ARY,LEN)
This entry point retrieves the Versioned Short Text for a
procedure on a specified date. This entry point is similar to
\$\$VST except you can elect to have the Short Text returned in
a local array and you can specify the string lengths of the
text in the array.

VARIABLES: Input FILE
This is the ICD file number used to retrieve the
Versioned Short Text (Required):

80 = ICD Diagnosis file
80.1 = ICD Operation/Procedure file

VARIABLES: Input IEN
This is an internal entry number (IEN) in either
file 80 or 80.1 (Required)

VARIABLES: Input CDT
This is the Code Set Versioning date (Fileman

format) used to retrieve the Versioned Short Text that was appropriate for the date passed (Optional, If not passed TODAY is used).

VARIABLES: Input .ARY
This is a local array name passed by reference that will contain the Short Text output.

VARIABLES: Input LEN
This is a number greater than 27 and less than 246 representing the desired text string lengths for the Short Text output. If specified, the output will be parsed into strings not to exceed the length specified (Optional, default 245)

VARIABLES: Output \$\$\$SD
This is the Versioned Short Text from either file 80 or 80.1 that was appropriate for the date passed and the code identified by the input parameter IEN. If not found:

-1^Error Message

VARIABLES: Output ARY
If passed, this is a local array containing the number of text lines, the effective date of the Short Text and the text. If the input parameter LEN (length) is specified and the length is shorter than the Short Text, then the Short Text will be parsed into test strings not to exceed LEN.

ARY(0)=# lines ^ effective date
ARY(1)=Short Text

LEN is defined shorter than text

ARY(0)=# lines ^ effective date
ARY(1)=String length not to exceed LEN
ARY(n)=String length not to exceed LEN

Null if not found

COMPONENT: \$\$LD(FILE,IEN,CDT,.ARY,LEN)
This entry point retrieves the Versioned Long Text (description) for a procedure on a specified date. This entry point is similar to \$\$VLT except you can elect to have the Long Text (description) returned in a local array and you can specify the string lengths of the text in the array.

VARIABLES: Input FILE
This is the ICD file number used to retrieve the Versioned Long Text (description) (Required):

80 = ICD Diagnosis file
80.1 = ICD Operation/Procedure file

VARIABLES: Input IEN
This is an internal entry number (IEN) in either file 80 or 80.1 (Required)

VARIABLES: Input CDT
This is the Code Set Versioning date (Fileman format) used to retrieve the Versioned Long Text (description) that was appropriate for the date passed (Optional, If not passed TODAY is used).

VARIABLES: Input .ARY
This is a local array name passed by reference that will contain the Long Text (description) output.

VARIABLES: Input LEN
This is a number greater than 27 and less than 246 representing the desired text string lengths for the Long Text (description) output. If specified, the output will be parsed into strings not to exceed the length specified (Optional, default 245)

VARIABLES: Output \$\$LD
This is the Versioned Long Text (description) from either file 80 or 80.1 that was appropriate for the date passed and the code identified by the input parameter IEN. If not found:

-1^Error Message

VARIABLES: Output ARY
If passed, this is a local array containing the number of text lines, the effective date of the Long Text (description) and the text. If the input parameter LEN (length) is specified and the length is shorter than the Long Text (description), then the Long Text (description) will be parsed into test strings not to exceed LEN.

ARY(0)=# lines ^ effective date
ARY(1)=Long Text (description)

LEN defined shorter than text

ARY(0)=# lines ^ effective date
ARY(1)=String length not to exceed LEN
ARY(n)=String length not to exceed LEN

COMPONENT: PAR(.ARY,LEN)
This entry point takes text in a local array (passed by reference) and parses it into string lengths not to exceed the length specified.

VARIABLES: Input .ARY
This is a local array name passed by reference and

contains the text to be parsed into strings not to exceed the length specified.

ARY(1) = Unparsed Text

VARIABLES: Input LEN
This is a number representing the desired text string lengths for the text found in ARY().
(Optional, default length 79)

VARIABLES: Output ARY
This is a local array containing the input text parsed so that each text string length does not exceed the length specified.

ARY(1)=Parsed Text length not to exceed LEN
ARY(n)=Parsed Text length not to exceed LEN

COMPONENT: \$\$STATCHK(CODE, CDT, SYS)
This entry point is used to determine the status (active or inactive) of a ICD code.

VARIABLES: Input CODE
This is either an ICD diagnosis or procedure code (external format) (Required)

VARIABLES: Input CDT
This is the Code Set Versioning date (Fileman format) used to retrieve the code's status, internal entry number (IEN) and effective date that was appropriate for the date passed
(Optional, If not passed TODAY is used)

VARIABLES: Input SYS
This is an ICD coding system identifier (taken from file 80.4). The following coding systems are found in files 80 and 80.1:

1 = ICD-9 Diagnosis
30 = ICD-10 Diagnosis
2 = ICD-9 Procedures
31 = ICD-10 Procedures

(Optional, but encouraged, if doesn't exist it will try to determine coding system by input parameter CODE)

VARIABLES: Output \$\$STATCHK
This is a three piece "^" delimited string

1 Status 1 = Active, 0 = Inactive
2 IEN or -1 on error
3 Effective Date or error message

Error 0 ^ -1 ^ Error message
Active Code 1 ^ IEN ^ Effective Date
Inactive Code 0 ^ IEN ^ Effective Date

COMPONENT: \$\$DTBR (CDT,STD,SYS)
This entry point returns the business rule date for a coding system. This is in earliest date possible for a coding standard and/or a coding system.

VARIABLES: Input CDT
This is the Code Set Versioning date (Fileman format) used to resolved the business rule date. (Optional, if not passed TODAY is used)

VARIABLES: Input STD
This is a coding standard from a Standards Development Organization (SDO). A standard may have one or more coding systems. (Optional, default is 0)

0 = ICD (Default)
1 = CPT/HCPCS
2 = DRG

VARIABLES: Input SYS
This is an ICD coding system identifier (taken from file 80.4). (Optional, there is no default value for this parameter, if it does not exist then it is not used)

The following coding systems are found in files 80 and 80.1:

1 = ICD-9 Diagnosis
30 = ICD-10 Diagnosis
2 = ICD-9 Procedures
31 = ICD-10 Procedures

VARIABLES: Output \$\$DTBR
Date adjusted by business rules:

If Standard (SDT) = 0 (ICD)

If CDT < 2781001 use 2781001
If CDT < 3131001 and SYS=30, use 3131001
If CDT < 3131001 and SYS=31, use 3131001

If Standard (SDT) = 1 (CPT/HCPCS)

If CDT < 2890101 use 2890101

If Standard (SDT) = 2 (DRG)

If CDT < 2821001 use 2821001

If CDT is year only, use first of the year If CDT is year and month only, use first of the month

COMPONENT: \$\$IMP (SYS,CDT)
This entry point returns the date a coding system was

implemented (taken from file 80.4).

VARIABLES: Input SYS
 This is a coding system (taken from file 80.4) or
 a coding system identifier that can be resolved to
 a coding system.

 1 = ICD-9-CM
 2 = ICD-9-PCS
 30 = ICD-10-CM
 31 = ICD-10-PCS

 DX, DIAG, 80, ^ICD9(
 1 = ICD-9-CM if CDT is before the ICD-10
 implementation date
 30 = ICD-10-CM if CDT is on or after the ICD-10
 implementation date

 PR, PROC, OPER, 80.1, ^ICD0(
 2 = ICD-9-CM if CDT is before the ICD-10
 implementation date
 31 = ICD-10-CM if CDT is on or after the ICD-10
 implementation date

VARIABLES: Input CDT
 This is the Code Set Versioning date (Fileman
 format) used to resolve the coding system
 parameter SYS (Optional, if not passed TODAY is
 used)

VARIABLES: Output \$\$IMP
 This is the date that a coding system identified
 by the input parameters SYS and CDT was
 implemented in Fileman format or on error:

 -1 ^ Error message

COMPONENT: \$\$MSG(CDT,STD,SYS)
 This entry point returns a warning message that the text may
 be inaccurate for the date specified. It applies only to
 ICD-9 Diagnosis and Procedures.

VARIABLES: Input CDT
 This is the Code Set Versioning date (Fileman
 format) used to determine the accuracy of the text
 being returned (Optional, if not passed TODAY is
 used)

VARIABLES: Input STD
 This is a coding standard from a Standards
 Development Organization (SDO). A standard may
 have one or more coding systems. (Optional,
 default is 0)

 0 = ICD (Default)

1 = CPT/HCPCS
2 = DRG

VARIABLES: Input SYS
 This is an ICD coding system identifier (taken from file 80.4). (Optional, there is no default value for this parameter, if it does not exist then it is not used)

 The following coding systems are found in files 80 and 80.1:

1 = ICD-9 Diagnosis
30 = ICD-10 Diagnosis
2 = ICD-9 Procedures
31 = ICD-10 Procedures

VARIABLES: Output \$\$MSG
 If coding system is not ICD-10 and the date passed is before the Code Set Versioning project Oct 1, 2002, then this variable is set to the warning message, "CODE TEXT MAY BE INACCURATE" otherwise it is null.

COMPONENT: \$\$SEL(FILE,IEN)
 This entry point determines if an entry in a file is selectable by calling applications.

VARIABLES: Input FILE
 This is an ICD file number:

 80 = ICD Diagnosis file
 80.1 = ICD Operation/Procedure file

VARIABLES: Input IEN
 This is an Internal Entry Number (IEN) in the file specified.

VARIABLES: Output \$\$SEL
 This is a Boolean value:

 1 Entry IEN in file FILE is Selectable
 0 Entry IEN in file FILE is NOT Selectable

 or

 -1 on error

COMPONENT: \$\$NEXT(CODE,SYS,CDT)
 This entry point returns the Next code in a sequence of codes in a coding system.

VARIABLES: Input CODE
 This is either an ICD diagnosis, an ICD procedure code or null to retrieve the first code in a sequence.

VARIABLES: Input SYS
This is an ICD coding system identifier (taken from file 80.4). (Optional)

The following coding systems are found in ICD files 80 and 80.1:

- 1 = ICD-9 Diagnosis
- 30 = ICD-10 Diagnosis
- 2 = ICD-9 Procedures
- 31 = ICD-10 Procedures

VARIABLES: Input CDT
This is the Code Set Versioning date (Fileman format) used to determine the next code being returned (Optional, there is no default value for this parameter)

If CDT date is not passed then this entry point will return the next code, regardless of status (active or inactive)

If CDT date is passed then this entry point will return the next active code.

VARIABLES: Output \$\$NEXT
This is the next code in a sequence of codes. If the input code is null, then it will return the first code of the sequence of codes. If a date is passed in the input parameter CDT, then it will return the next active code in a sequence of codes.

COMPONENT: \$\$PREV(CODE, SYS, CDT)
This entry point returns the Previous code in a sequence of codes in a coding system.

VARIABLES: Input CODE
This is either an ICD diagnosis, an ICD procedure code or null to retrieve the last code in a sequence.

VARIABLES: Input SYS
This is an ICD coding system identifier (taken from file 80.4). (Optional)

The following coding systems are found in ICD files 80 and 80.1:

- 1 = ICD-9 Diagnosis
- 30 = ICD-10 Diagnosis
- 2 = ICD-9 Procedures
- 31 = ICD-10 Procedures

VARIABLES: Input CDT
This is the Code Set Versioning date (Fileman format) used to determine the Previous code being

returned (Optional, there is no default value for this parameter)

If CDT date is not passed then this entry point will return the previous code, regardless of status (active or inactive)

If CDT date is passed then this entry point will return the previous active code.

VARIABLES: Output \$\$PREV
This is the previous code in a sequence of codes. If the input code is null, then it will return the last code of the sequence of codes. If a date is passed in the input parameter CDT, then it will return the previous active code in a sequence of codes.

COMPONENT: \$\$HIST(CODE, .ARY, SYS)
This entry point returns a code's activation history.

VARIABLES: Input CODE
This is an ICD diagnosis or procedure code.

VARIABLES: Input .ARY
This is a local array name passed by reference that will contain the code's activation history.

VARIABLES: Input SYS
This is an ICD coding system identifier (taken from file 80.4). (Optional)

The following coding systems are found in ICD files 80 and 80.1:

- 1 = ICD-9 Diagnosis
- 30 = ICD-10 Diagnosis
- 2 = ICD-9 Procedures
- 31 = ICD-10 Procedures

VARIABLES: Output \$\$HIST
This is set equal to the number of history entries in the local array ARY or -1 if there is an error or the code is not found.

VARIABLES: Output ARY
This is a local array containing the history records

- ARY(0) = Number of History Entries
- ARY(<effective date>) = status
- ARY("IEN") = <ien>

COMPONENT: \$\$PERIOD(CODE, .ARY, SYS)
This entry point returns all the activation periods for a code. An activation period is defined as the period of time between the beginning activation effective date and the ending

inactivation effective date. If the code is still active the period will have an activation date without an inactivation date.

VARIABLES: Input CODE
This is either an ICD diagnosis or procedure code.

VARIABLES: Input .ARY
This is a local array name passed by reference that will contain the code's activation periods.

VARIABLES: Input SYS
This is an ICD coding system identifier (taken from file 80.4). (Optional)

The following coding systems are found in ICD files 80 and 80.1:

- 1 = ICD-9 Diagnosis
- 30 = ICD-10 Diagnosis
- 2 = ICD-9 Procedures
- 31 = ICD-10 Procedures

VARIABLES: Output \$\$PERIOD
This is a 2 piece "^" delimited string if successful and 3 piece "^" delimited string if unsuccessful or error.

- 1 IEN of code
- 2 Code is selectable (boolean 1/0)

or on error

-1 ^ 0 ^ Error Message

VARIABLES: Output ARY
This is a local array containing the Periods of activation for the code

ARY(0)

This is a 2 piece "^" delimited string if successful and a 3 piece "^" delimited string if unsuccessful or error.

- 1 IEN of code
- 2 Code is selectable (boolean 1/0)

or on error

-1^0^Error Message

ARY(Activation Date) = Inactivation Date^Short Name

Where the Short Name is the Versioned text, and the

text is versioned as follows:

Period is active - Text for TODAY's date
Period is inactive - Text for inactivation date

COMPONENT: \$\$OBA(FILE, CODE, SYS, REV)
This entry point is used to \$ORDER through the BA or ABA cross-references and replaces the need to access the BA/ABA cross-references in a FOR loop. This entry point is meant to replace BA cross-reference in ICRs 5388 and 5404.

\$\$OBA(<file>, <code>, <system>) replaces:

\$O(^ICD9("BA", (<code>_ " "))) and
\$O(^ICD0("BA", (<code>_ " ")))

Examples:

```
F S CODE=$$OBA(80, CODE, 1) Q: '$L(CODE) D
F S CODE=$$OBA(80, CODE, 30) Q: '$L(CODE) D
F S CODE=$$OBA(80.1, CODE, 2) Q: '$L(CODE) D
F S CODE=$$OBA(80.1, CODE, 31) Q: '$L(CODE) D
```

VARIABLES: Input FILE
This is the ICD file number used to determine the global root to \$ORDER through (Required):

80 = ICD Diagnosis file
80.1 = ICD Operation/Procedure file

VARIABLES: Input CODE
This is either an ICD diagnosis or procedure code to \$ORDER from (required):

\$O(^ROOT("BA", (CODE_ " ")))
\$O(^ROOT("ABA", SYS, (CODE_ " ")))

VARIABLES: Input SYS
This is either an ICD diagnosis or procedure

This is an ICD coding system identifier (taken from file 80.4). (Optional)

The following coding systems are found in ICD files 80 and 80.1:

1 = ICD-9 Diagnosis
30 = ICD-10 Diagnosis
2 = ICD-9 Procedures
31 = ICD-10 Procedures

If the coding system can be identified then the "BA" cross-reference is ignored and the \$ORDER will be performed on the "ABA" cross-reference:

\$O(^ROOT("ABA", SYS, (CODE_ " ")))

The "ABA" cross-reference is a coding system specific cross-reference.

VARIABLES: Used REV
This is a Reverse \$ORDER flag, if set to 1, the \$ORDER operation will be in the reverse direction of "BA" or "ABA" cross-reference (Optional, default is 0, \$ORDER forward)

If equal to 1

```
$O(^ROOT("BA", (CODE_ " ")), -1)
$O(^ROOT("ABA", SYS, (CODE_ " ")), -1)
```

VARIABLES: Output \$\$OBA
This is the Next or Previous Code in the "BA" or "ABA" cross-reference depending on the \$ORDER direction established by the input parameter REV.

COMPONENT: \$\$OD(FILE,WORD,SYS,REV)
This entry point is used to \$ORDER through the "D" or "AD" cross-reference and replaces the need to access the D/AD cross-references in a FOR loop. This entry point is meant to replace the D cross-reference in ICRs 5388 and 5404.

\$\$OD(<file>,<word>,<system>) replaces:

```
$O(^ICD9("D", (<word>_ " ")) and
$O(^ICD0("D", (<word>_ " "))
```

Examples:

```
F S WORD=$$OD(80,WORD,1) Q:'$L(WORD) D
F S WORD=$$OD(80,WORD,30) Q:'$L(WORD) D
F S WORD=$$OD(80.1,WORD,2) Q:'$L(WORD) D
F S WORD=$$OD(80.1,WORD,31) Q:'$L(WORD) D
```

VARIABLES: Input FILE
This is the ICD file number used to determine the global root to \$ORDER through (Required):

```
80 = ICD Diagnosis file
80.1 = ICD Operation/Procedure file
```

VARIABLES: Input WORD
This is a one or two piece "^" delimited string

- 1 WORD This is a single word parsed from the codes description.
- 2 IEN This is the internal entry number where the description can be found that contains the parsed word

WORD and IEN can be null.

\$\$OD \$ORDER through "WORD^IEN" on either the D or

AD cross-references

Coding System unknown: \$O(^ROOT("D",WORD,IEN))
Coding System known:
\$O(^ROOT("AD",SYS,WORD,IEN))

VARIABLES: Input

SYS

This is an ICD coding system identifier (taken from file 80.4). (Optional)

The following coding systems are found in ICD files 80 and 80.1:

1 = ICD-9 Diagnosis
30 = ICD-10 Diagnosis
2 = ICD-9 Procedures
31 = ICD-10 Procedures

If the coding system can be identified then the "D" cross-reference is ignored and the \$ORDER will be performed on the "AD" cross-reference:

\$O(^ROOT("AD",SYS,(CODE_" ")))

The "AD" cross-reference is a coding system specific cross-reference.

VARIABLES: Input

REV

This is a Reverse \$ORDER flag, if set to 1, the \$ORDER operation will be in the reverse direction of "D" or "AD" cross-reference (Optional, default is 0, \$ORDER forward)

If equal to 1

\$O(^ROOT("D",WORD)),-1)
\$O(^ROOT("AD",SYS,WORD)),-1)

VARIABLES: Output

\$\$OD

This is a 2 piece "^" delimited string containing the Next or Previous Word in the "D" or "AD" cross-reference and accompanying IEN depending on the \$ORDER direction established by the input parameter REV.

WORD^IEN taken from cross-references

^ROOT("D",WORD,IEN) or
^ROOT("AD",SYS,WORD,IEN)

COMPONENT: \$\$DLM(FILE,IEN,FIELD,CDT)

This entry point returns the date a record or field was last modified. If the field number is passed, then the date last modified (based on date) for the field is returned. If the field is not passed, then the date last modified (based on date) for the record at IEN is returned. The following are valid versioned fields:

File 80

10	Sex	5;0
11	Age Low	6;0
12	Age High	7;0
66	Status	66;0
67	Diagnosis	67;0
68	Description	68;0
71	DRG Grouper	3;0
72	Major Diagnostic Category	4;0
103	Complication/Comorbidity	69;0

File 80.1

10	Sex	3;0
66	Status	66;0
67	Operation/Procedure	67;0
68	Description	68;0
71	DRG Grouper	2;0

VARIABLES: Input FILE
This is the ICD file number used to determine the global root to \$ORDER through (Required):

80 = ICD Diagnosis file
80.1 = ICD Operation/Procedure file

VARIABLES: Input IEN
This is an Internal Entry Number (IEN) in the file specified (Required)

VARIABLES: Input FIELD
This is the field number of a versioned data element in the file specified. (Optional, with no default value)

If the field number is provided then this API will return the date that the field was last modified.

If the field number is not provided then this API will return the date that the record was last modified.

VARIABLES: Input CDT
This is the Code Set Versioning date (Fileman format) used to determine the date last modified (Optional, if not provided then TODAY is used)

VARIABLES: Output \$\$DLM
This is the date last modified for the record identified by the input parameters FILE and IEN. If the input parameter FIELD is set to a valid versioned field then this will be the date that the field was last modified.

or -1 ^ message on error

COMPONENT: \$\$CS (FILE, FMT, CDT)
This is an interactive entry point to select a coding system.

VARIABLES: Input FILE
This is the ICD file number used to select a coding system (Optional, if not provided you will be prompted for an ICD file Number):

80 = ICD Diagnosis file
80.1 = ICD Operation/Procedure file

VARIABLES: Input FMT
This is a flag to determine the display format for the prompts:

E Display External only (default)
I Display External with Internal

Prompt using External only, default:

FMT=E 1 ICD-9-CM
2 ICD-10-CM

Prompt using External with Internal:

FMT=I 1 ICD-9-CM (#1)
2 ICD-10-CM (#30)

VARIABLES: Input CDT
This is an optional date to use in selecting a coding system. If passed, only coding systems with an implementation date on or before the date passed are selectable (optional)

VARIABLES: Output \$\$CS
This is a 2 piece "^" delimited string

1 Coding System (internal) 2 Coding System (external)

or -1 on error or non-selection or ^^ double up-arrows or ^ timeout or single up-arrow

COMPONENT: \$\$EFF (FILE, IEN, CDT)
This entry point returns a codes status, inactivation date and activation date (replaces EFF^ICDSUPT)

VARIABLES: Input FILE
This is an ICD file number (Required):

80 = ICD Diagnosis file
80.1 = ICD Operation/Procedure file

VARIABLES: Input IEN
This is an Internal Entry Number (IEN) in the file

specified (Required)

VARIABLES: Input CDT
This is the Code Set Versioning date (Fileman format) used to determine the status and effective dates on the date specified (Optional, if not provided then TODAY is used)

VARIABLES: Output \$\$EFF
This is a 3 piece "^" delimited string

- 1 Status
 - 1 - Active
 - 0 - Inactive
- 2 Inactivation Date
- 3 Activation Date

or
-1^error message

COMPONENT: \$\$LA (FILE, IEN, CDT)
This entry point returns the last activation effective date based on a date passed.

VARIABLES: Input FILE
This is an ICD file number (Required):

- 80 = ICD Diagnosis file
- 80.1 = ICD Operation/Procedure file

VARIABLES: Input IEN
This is an Internal Entry Number (IEN) in the file specified (Required)

VARIABLES: Input CDT
This is the Code Set Versioning date (Fileman format) used to determine the last activation date based on the date specified (Optional, if not provided then TODAY is used)

VARIABLES: Output \$\$LA
This is the last activation date (Fileman format)
or
-1^Not activated on or before date specified

COMPONENT: \$\$LI (FILE, IEN, CDT)
This entry point returns the last inactivation effective date based on a date passed.

VARIABLES: Input FILE
This is an ICD file number (Required):

- 80 = ICD Diagnosis file
- 80.1 = ICD Operation/Procedure file

VARIABLES: Input IEN

This is an Internal Entry Number (IEN) in the file specified (Required)

VARIABLES: Input CDT
This is the Code Set Versioning date (Fileman format) used to determine the last inactivation date based on the date specified (Optional, if not provided then TODAY is used)

VARIABLES: Output \$\$LI
This is the last inactivation date (Fileman format) or

-1^Not inactivated on or before date specified

COMPONENT: \$\$LS(FILE,IEN,CDT)
This entry point returns the last code status based on a date passed.

VARIABLES: Input FILE
This is an ICD file number (Required):

80 = ICD Diagnosis file
80.1 = ICD Operation/Procedure file

VARIABLES: Input IEN
This is an Internal Entry Number (IEN) in the file specified (Required)

VARIABLES: Input CDT
This is the Code Set Versioning date (Fileman format) used to determine the last code status based on the date specified (Optional, if not provided then TODAY is used)

VARIABLES: Output \$\$LS
This is the last code status based on the date passed.

1 - Active
0 - Inactive

or

-1^No status on or before date specified

COMPONENT: \$\$NUM(CODE)
This entry point converts a code to a numeric representation (found on the AN cross-reference)

VARIABLES: Input CODE
This is either an ICD diagnosis or procedure code (Required) (This is the opposite of \$\$COD)

VARIABLES: Output \$\$NUM
This is a numeric representation of a code.

COMPONENT: \$\$COD(NUM)
This entry point converts a numeric representation of a code to a code (found on the AN cross-reference)

VARIABLES: Input NUM
This is a numeric representation of an ICD diagnosis or procedure code (This is the opposite of \$\$NUM)

VARIABLES: Output \$\$COD
This is an ICD diagnosis or procedure code.

COMPONENT: \$\$IE(CODE)
This entry point determines if a code is in an external or internal format without plusing (+) the code.

If you have an ICD-10 code with the letter "E in the center and plus it you will receive a MAXNUMBER error.

Example: If you plus (+) the ICD-10 procedure code "041E499" it will be interpreted as a scientific notation (E499 is a really big number). Applications that plus the ICD code can use this entry point to safely determine a code's format.

VARIABLES: Input CODE
This is either an ICD diagnosis or procedure code (Required)

VARIABLES: Output \$\$IE
This is a set of codes as follows:

I CODE is in an internal format (IEN)
E CODE is in an external format (Code)

or

Null on error

COMPONENT: \$\$FILE(SYS)
This entry point will return an ICD file number.

VARIABLES: Input SYS
This is a coding system, a global root or a file identifier.

Global roots ^ICD9(and ^ICD0(are acceptable
Coding Systems can be found in file 80.4 File Identifier: DX or PR
DIAG or PROC or OPER

VARIABLES: Output \$\$FILE
This is an ICD file number 80 or 80.1

or -1 on error

COMPONENT: \$\$ROOT(SYS)
This entry point will return an ICD global root.

VARIABLES: Input SYS
This is a coding system, file number, a file identifier or even an ICD code, provided the code is unique to a file.

Coding Systems can be found in file 80.4 File Number 80 or 80.1 File Identifier: DX or PR
DIAG or PROC or OPER

VARIABLES: Output \$\$ROOT
This is a global root ^ICD9(or ^ICD0(or Null on error

COMPONENT: \$\$SYS(SYS,CDT,FMT)
This entry point will return a coding system.

VARIABLES: Input SYS
This can be either a Coding System name, Abbreviation, system identifier (uses date) or a code.

Coding System Names: ICD-9-CM, ICD-9 Proc, ICD-10-CM or ICD-10-PCS

Coding System Abbreviations: ICD, ICP, 10D or 10P

System Identifier (with date CDT)

Date is before the ICD-10 implementation date

DIAG, ICD9, 80, DX = 1
PROC, OPER, ICD0, ICP9, 80.1, PR = 2

Date is on or after the ICD-10 implementation date

DIAG, ICD9, 80, DX = 30
PROC, OPER, ICD0, ICP9, 80.1, PR = 31

An ICD code

If an ICD code is unique to an ABA cross-reference then the Coding System can be determined from a code

^ICD9("ABA",1,(CODE_" ")) = 1
^ICD9("ABA",30,(CODE_" ")) = 30
^ICD9("ABA",2,(CODE_" ")) = 2
^ICD9("ABA",31,(CODE_" ")) = 31

VARIABLES: Input CDT
This is the Code Set Versioning date (Fileman format) used to determine the coding system based on a system identifier (Optional, if not provided then TODAY is used)

VARIABLES: Input FMT
This is a single character identifying the desired output format (Optional, default is "I"):

I Internal (default)
E External
B Both Internal ^ External

VARIABLES: Output \$\$\$SYS
This is the Coding System in the format specified by the input parameter FMT:

FMT=I	FMT=E	FMT=B
Internal	External	Both
1	ICD-9-CM	1^ICD-9-CM
2	ICD-9 Proc	2^ICD-9 Proc
30	ICD-10-CM	30^ICD-10-CM
31	ICD-10-PCS	31^ICD-10-PCS

or

-1 on error

COMPONENT: \$\$\$INFO(SYS,CDT)
This entry point returns coding system information taken from file 80.4.

VARIABLES: Input SYS
This can be either a Coding System name, Abbreviation, system identifier, file number or a code. (system identifier and code uses date).

Coding System Names:

ICD-9-CM
ICD-9 Proc
ICD-10-CM or
ICD-10-PCS

Coding System Abbreviations:

ICD, ICP, 10D or 10P

System Identifier/File Number (with date CDT)

Date is before the ICD-10 implementation date

DIAG, ICD9, 80, DX = 1
PROC, OPER, ICD0, ICP9, 80.1, PR = 2

Date is on or after the ICD-10 implementation date

DIAG, ICD9, 80, DX = 30
PROC, OPER, ICD0, ICP9, 80.1, PR = 31

VARIABLES: Input CDT
This is the Code Set Versioning date (Fileman format) used to determine the coding system based on a system identifier (Optional, if not provided then TODAY is used)

VARIABLES: Output \$\$\$INFO
This is a 6 piece "^" delimited string

1	IEN to file 80.4
2	Coding System
3	Coding System Nomenclature
4	Coding system Abbreviation
5	File where the Coding System is stored
6	Implementation Date

or

-1 on error

COMPONENT: \$\$\$NAM(SYS)
This entry point returns the coding system name.

VARIABLES: Input SYS
This is a pointer to the coding system file 80.4

VARIABLES: Output \$\$\$NAM
This the coding system name, file 80.4 (.01)

ICD-9-CM
ICD-9 Proc
ICD-10-CM
ICD-10-PCS

Or -1 on error

COMPONENT: \$\$\$SAB(SYS,CDT)
This entry point returns the coding system abbreviation.

VARIABLES: Input SYS
This can be either a Coding System name, Abbreviation, system identifier (uses date) or a code.

Coding System Names: ICD-9-CM, ICD-9 Proc, ICD-10-CM or ICD-10-PCS

Coding System Abbreviations: ICD, ICP, 10D or 10P

System Identifier (with date CDT)

Date is before the ICD-10 implementation date

DIAG, ICD9, 80, DX = 1
PROC, OPER, ICD0, ICP9, 80.1, PR = 2

Date is on or after the ICD-10 implementation

date

DIAG, ICD9, 80, DX = 30
PROC, OPER, ICD0, ICP9, 80.1, PR = 31

VARIABLES: Input CDT
This is the Code Set Versioning date (Fileman format) used to determine the source abbreviation based on a system identifier (Optional, if not provided then TODAY is used)

VARIABLES: Output \$\$\$SAB
3 Character Coding System abbreviation, file 80.4 (.02)

ICD
ICP
10D
10P

Or -1 on error

COMPONENT: \$\$EXC(FILE,IEN)
This entry point returns a boolean value indicating if an entry in the specified file is to be excluded from lookup. If it is to be excluded, then the entry will not be placed on the selection list for a user to select from. Used primarily for the special lookup.

VARIABLES: Input FILE
This is an ICD file number:
80 = ICD Diagnosis file
80.1 = ICD Operation/Procedure file

VARIABLES: Input IEN
This is an Internal Entry Number (IEN) in the file specified.

VARIABLES: Output \$\$EXC
Boolean value
1 = Yes, exclude from lookup
0 = No, include in the lookup

COMPONENT: \$\$ISA(IEN1,IEN2,FIELD)
This entry point returns a boolean value indicating that one code is a "condition" of another. Conditions include:

Code 1 is Not Used With Code 2
Code 1 is Required With Code 2
Code 1 is Not Considered CC With Code 2

VARIABLES: Input IEN1
This is the internal entry number (IEN) of a code in file 80 that has a relationship with the code at IEN2 IEN1 is equivalent to Fileman's DA and

identifies a code stored in a multiple in field 20, 30, 40 or pointed to by field 1.11.

VARIABLES: Input IEN2
This is the internal entry number (IEN) of a code in file 80 that may have other codes (IEN1) associated with it. IEN2 is equivalent to Fileman's DA(1) and identifies the code in the .01 field.

VARIABLES: Input FIELD
This is a field number in file 80 that contains one or more ICD codes that have a relationship to the main entry. Acceptable field numbers and the type of relationships to check include:

Field	Relationship
20	Code 1 Not Used With Code 2
30	Code 1 Required With Code 2
40 or 1.11	Code 1 Not Considered CC With Code 2

VARIABLES: Output \$\$ISA
This is a Boolean value

1 Yes/The relationship is True
0 No/The relationship is False

Field	Answers the Question
20	Code 1 (identified by IEN1) is not used with Code 2 (identified by IEN2)
30	Code 1 (identified by IEN1) is required with Code 2 (identified by IEN2)
40 or 1.11	Code 1 (identified by IEN1) is not considered Complication Comorbidity (CC) with Code 2 (identified by IEN2)

COMPONENT: \$\$EXIST(IEN, FIELD)
This entry point determines if special condition ICD codes exist.

VARIABLES: Input IEN
This is an Internal Entry Number (IEN) in the DIAGNOSIS file 80 (Required)

VARIABLES: Input FIELD
This is a field number in file 80 that contains one or more ICD codes that have a relationship to the main entry (Required) Acceptable field numbers to check include:

20 Code Not Used With

30 Code Required With
40 Code Not Considered CC With

VARIABLES: Output \$\$EXIST
Boolean value

1 Yes/True, codes exist
0 No/False, codes do not exist

Field Answers the Question

20 Are there any codes that should
not be used with this code (IEN)
30 Are there any codes required
with this code (IEN)
40 Are there any codes that are not
considered CC with this code
(IEN)

COMPONENT: \$\$GETDRG (FILE, IEN, CDT, MDC)
This entry point returns a string of DRGs for an ICD Diagnosis
or Procedure code.

VARIABLES: Input FILE
This is the ICD file number used to retrieve the
DRGs (Required):

80 = ICD Diagnosis file
80.1 = ICD Operation/Procedure file

VARIABLES: Input IEN
This is an Internal Entry Number (IEN) in the file
specified (Required)

VARIABLES: Input CDT
This is the Code Set Versioning date (Fileman
format) used to identify the DRGs that were
appropriate on that date (Optional, if not passed
then TODAY is used)

VARIABLES: Input MDC
This is a Major Diagnostic Category (pointer to
file 80.3) used as a screen to limit the DRGs to
an MDC. This input parameter only applies to the
ICD OPERATIONS/PROCEDURE file 80.1 which has
multiple MDCs, each with a possibility of multiple
DRGs (Conditional)

VARIABLES: Output \$\$GETDRG
3 piece semi-colon ";" delimited string

1 DRGs delimited by ^
2 Fiscal Year
3 Status flag
0 inactive
1 active

Example output:

907^908^909^;3071001;1

On Error:

-1;No DRG level;0

COMPONENT: MD(FILE,IEN,CDT,.ARY,FLAG)

This entry point returns an array of Major Diagnostic Categories (MDCs) and Diagnosis Related Groups (DRGs)

VARIABLES: Input FILE

This is the ICD file number used to retrieve the Major Diagnostic Categories (Required):

80 = ICD Diagnosis file

80.1 = ICD Operation/Procedure file

VARIABLES: Input IEN

This is an Internal Entry Number (IEN) in the file specified (Required)

VARIABLES: Input CDT

This is the Code Set Versioning date (Fileman format) used to identify the MDCs that were appropriate on that date (Optional, if not passed then TODAY is used) NOTE: If no Fiscal Year is found for the input date then the first (earliest) Fiscal Year is used.

VARIABLES: Input .ARY

This is a local array name passed by reference that will contain a list of MDCs by effective date

VARIABLES: Input FLAG

This is a flag that determines the output format:

I = Internal (default) Internal values are always returned

E = Include External values with Internal values

VARIABLES: Output ARY

ICD Procedures file 80.1 (multiple MDC)

ARY(<fiscal year>,<MDC>)=DRG^;FY;STA

ARY(<fiscal year>,<MDC>)="DRG^DRG^;FY;STA

If Flag contains "E"

ARY(<fiscal year>,"E",<MDC>)=MDC Name

ARY(<fiscal year>,"E",<MDC>,<DRG>)=DRG Name

ARY(<fiscal year>,"E",<MDC>)=MDC Name

ARY(<fiscal year>,"E",<MDC>,<DRG>)=DRG Name

ARY(<fiscal year>,"E",<MDC>,<DRG>)=DRG Name

ARY(<fiscal year>,"E","FY")=External FY

ICD Diagnosis file 80 (single MDC)

ARY(<fiscal year>,<MDC>)="DRG^DRG^;FY;STA

If Flag contains "E"

ARY(<fiscal year>,"E",<MDC>)=MDC Name

ARY(<fiscal year>,"E",<MDC>,<DRG>)=DRG Name

ARY(<fiscal year>,"E",<MDC>,<DRG>)=DRG Name

ARY(<fiscal year>,"E","FY")=External FY

NOTE: If no Fiscal Year found for the input date then the first (earliest) Fiscal Year is used.

COMPONENT: \$\$EFM(CDT)

This entry point converts an external date to a Fileman internal date. This entry point replaces unsupported \$\$DGY2K^DGPTOD0 (X)

VARIABLES: Input CDT

External date (Required), examples of valid dates:

JAN 20 1957 or 20 JAN 57

1/20/57 or 012057

T (for TODAY)

T+1 (for TOMORROW), T+2, etc.

T-1 (for YESTERDAY)

T-3W (for 3 WEEKS AGO), etc.

VARIABLES: Output \$\$EFM

Internal Fileman Date

or -1 on error

COMPONENT: \$\$FY(CDT)

This entry point returns the 4 digit fiscal year for a specified date. This entry point replaces unsupported \$\$FY^DGPTOD0 (X)

VARIABLES: Input CDT

This is an internal Fileman date.

VARIABLES: Output \$\$FY

This is a 4 digit fiscal year (YYYY) for the date specified or null on error.

COMPONENT: \$\$VMDCDX(IEN,CDT)

This entry point returns the versioned Major Diagnostic Code for an ICD Diagnosis.

VARIABLES: Input IEN

This is an Internal Entry Number (IEN) in the DIAGNOSIS file 80 (Required)

VARIABLES: Input CDT

This is the Code Set Versioning date (Fileman format) used to identify the MDCs that was appropriate on that date (Optional, if not passed then TODAY is used)

VARIABLES: Output \$\$VMDCDX
 This is a single MDC (pointer to file 80.3) active on the date specified.

COMPONENT: \$\$VMDCOP (IEN, MDC, CDT)
 This entry point returns the versioned Major Diagnostic Codes for an ICD Procedure.

VARIABLES: Input IEN
 This is an Internal Entry Number (IEN) in the OPERATION/PROCEDURE file 80.1 (Required)

VARIABLES: Input MDC
 This is a Major Diagnostic Category (pointer to file 80.3) used as a screen to limit the results to a single MDC (Required)

VARIABLES: Input CDT
 This is the Code Set Versioning date (Fileman format) used to identify the MDC that was appropriate on that date (Optional, if not passed then TODAY is used)

VARIABLES: Output \$\$VMDCOP
 4 piece "^" delimited string

- 1 Fiscal Year, Fileman format
- 2 MDC, pointer to file 80.3
- 3 Fiscal Year, pointer to sub-file 80.171 (formerly DADRGFY)
- 4 MDC, pointer to sub-file 80.1711 (formerly DAMDC)

COMPONENT: MDCG (IEN, CDT, .ARY)
 This entry point sets up an array of MDCs (later used in \$\$MDCT)

VARIABLES: Input IEN
 This is an Internal Entry Number (IEN) in the DIAGNOSIS file 80 (Required)

VARIABLES: Input CDT
 This is the Code Set Versioning date (Fileman format) used to identify the MDCs that were appropriate on that date (Optional, if not passed then TODAY is used)

VARIABLES: Input .ARY
 This is a local array name passed by reference that will contain a list of MDCs (Required)

VARIABLES: Output ARY

This is an array listing MDCs for all DRGs associated with a diagnosis on the date specified.

```
ARY (MDC)=""  
ARY (MDC)=""
```

COMPONENT: \$\$MDCT(IEN,CDT,.ARY,FMT)
This entry point compares a single entry in the ICD OPERATIONS/PROCEDURE file 80.1 to an array of Major Diagnostic Categories to see if the ICD procedure is assigned to one or more of the MDCs in the array.

VARIABLES: Input IEN
This is an Internal Entry Number (IEN) in the OPERATIONS/PROCEDURE file 80.1 (Required)

VARIABLES: Input CDT
This is the Code Set Versioning date (Fileman format) used to identify the MDCs that were appropriate on that date (Optional, if not passed then TODAY is used)

VARIABLES: Input .ARY
This is a local array passed by reference containing a list of MDCs for comparison (Required)

VARIABLES: Input FMT
This is a flag defining the output format (optional):

```
0 Boolean value only (default)  
1 2 piece "^" delimited string  
  
1 Boolean value  
2 String of matching MDCs  
delimited by ";"
```

VARIABLES: Output \$\$MDCT
Boolean value

```
0 The ICD Procedure code identified by IEN  
does not include any of the MDCs passed  
in .ARY(MDC) on the date specified (CDT)  
  
1 The ICD Procedure code identified by IEN  
includes one or more of the MDCs passed  
in .ARY(MDC) on the date specified (CDT)
```

Assuming the following input parameters:

```
IEN=4  
CDT=3111110  
ARY(2)=""  
ARY(21)=""
```

Output format when input parameter FMT=0

(default)

\$\$MDCT(IEN,CDT,.ARY) = "1"

Output format when input parameter FMT=1

\$\$MDCT(IEN,CDT,.ARY) = "1^2;21"

COMPONENT: \$\$MDCD(IEN,MDC,CDT)

This entry point checks for a Major Diagnostic Category MDC in the ICD OPERATION/PROCEDURE file.

VARIABLES: Input IEN

This is an Internal Entry Number (IEN) in the OPERATIONS/PROCEDURE file 80.1 (Required)

VARIABLES: Input MDC

This is a Major Diagnostic Category (pointer to file 80.3) (Required)

VARIABLES: Input CDT

This is the Code Set Versioning date (Fileman format) used to identify the MDCs that were appropriate on that date (Optional, if not passed then the first FY is used)

VARIABLES: Output \$\$MDCD

Boolean value

0 MDC does not exist on date specified

1 MDC exist on date specified

COMPONENT: \$\$MOR(IEN)

This entry point returns the Major O.R. Procedure string

VARIABLES: Input IEN

This is an Internal Entry Number (IEN) in the OPERATIONS/PROCEDURE file 80.1 (Required)

VARIABLES: Output \$\$MOR

Major O.R. Procedure or Null if the procedure is not defined as a Major O.R. Procedure or is not found

Major O.R. Procedure definitions include:

1	Bowel	2	Chest	3	
	Lymphoma/Leukemia	4	Joint	5	Pancreas/Liver
	6	Pelvic	7	Shoulder/Elbow	8
	Thumb/Joint	9	Head/Neck	A	Cardio M
	Musculoskeletal	B	Spine		

COMPONENT: \$\$UPDX(IEN)

This entry point determines if a diagnosis is unacceptable as a principle diagnosis.

VARIABLES: Input IEN

This is an Internal Entry Number (IEN) in the
DIAGNOSIS file 80 (Required)

VARIABLES: Output \$\$UPDX
 Boolean value, answers the question:

 Is the diagnosis UNACCEPTABLE as a Principle
 DX?

 1 Yes Code is Unacceptable as Principle DX
 0 No Code is Acceptable as Principle DX

COMPONENT: \$\$NOT (IEN, SUB, FMT)
This entry point returns the number of ICD codes that cannot
be used with a specified code. It can also return a global
array containing a list of the codes that cannot be used with
the specified code.

VARIABLES: Input IEN
 This is an Internal Entry Number (IEN) in the
 DIAGNOSIS file 80 (Required)

VARIABLES: Input SUB
 This is a subscript name used in a ^TMP global
 array (Optional, if not provided, the subscript
 "ICDNOT" will be used)

 ^TMP (SUB, \$J)

VARIABLES: Input FMT
 This is a flag defining the output format.

 0 - Total number only (default)
 1 - Total number with global array

VARIABLES: Output \$\$NOT
 The number of ICD codes that cannot be used with
 the ICD code identified by IEN (FMT=0 or 1)

 TMP global array as follows (FMT=1):

 ^TMP (SUB, \$J, IEN)=CODE
 ^TMP (SUB, \$J, "B", (CODE_ " "), IEN)=""

COMPONENT: \$\$REQ (IEN, SUB, FMT)
This entry point returns the number of ICD codes that are
required when the specified code is used. It can also return
a global array containing a list of the codes that are
required when the specified code is used.

VARIABLES: Input IEN
 This is an Internal Entry Number (IEN) in the
 DIAGNOSIS file 80 (Required)

VARIABLES: Input SUB
 This is a subscript name used in a ^TMP global
 array (Optional, if not provided, the subscript

"ICDREQ" will be used)

^TMP (SUB, \$J)

VARIABLES: Input FMT
This is a flag defining the output format.

0 - Total number only (default)
1 - Total number with global array

VARIABLES: Output \$\$REQ
The number of ICD codes required when the ICD code identified by IEN is used. (FMT=0 or 1)

TMP global array as follows (FMT=1):

^TMP (SUB, \$J, IEN)=CODE
^TMP (SUB, \$J, "B", (CODE_ " "), IEN)=""

COMPONENT: \$\$NCC (IEN, SUB, FMT)
This entry point returns the number of ICD codes that are not considered CC with a specified code. It can also return a global array containing a list of the codes that are not considered CC with a specified code.

VARIABLES: Input IEN
This is an Internal Entry Number (IEN) in the DIAGNOSIS file 80 (Required)

VARIABLES: Input SUB
This is a subscript name used in a ^TMP global array (Optional, if not provided, the subscript "ICDNCC" will be used)

^TMP (SUB, \$J)

VARIABLES: Input FMT
This is a flag defining the output format.

0 - Total number only (default)
1 - Total number with global array

VARIABLES: Output \$\$NCC
The number of ICD codes not considered CC with the code identified by IEN. (FMT=0 or 1)

TMP global array as follows (FMT=1):

^TMP (SUB, \$J, IEN)=CODE
^TMP (SUB, \$J, "B", (CODE_ " "), IEN)=""

COMPONENT: LK
Special Lookup (called by DIC)

This is the Special Lookup program for files 80 and 80.1. Only the ^DIC call honors the special lookup routines. Those calls that allow the user to specify the indexes (IX^DIC and

MIX^DIC1), and the Data Base Server calls (FIND^DIC, \$\$FIND1^DIC, and UPDATE^DIE) all ignore the Special Lookup Program. Also, if DIC(0) contains an "I" then the Special Lookup program will be ignored.

This routine uses a majority of the variables used in calling Fileman ^DIC. In addition to the Fileman variables, there are three special variables that aid in controlling the lookup that can be set and killed by the calling application;

Versioning Date (Fileman format)

ICDVDT or
^TMP("ICDEXLK",\$J,"ICDVDT")=<versioning date>

Coding System (from file 80.4)

ICDSYS or
^TMP("ICDEXLK",\$J,"ICDSYS")=<coding system>

Display Format (numeric, 1-4) (new)

ICDFMT or
^TMP("ICDEXLK",\$J,"ICDFMT")=<display format>

VARIABLES: Input ICDVDT
Versioning Date (Fileman format)

ICDVDT or
^TMP("ICDEXLK",\$J,"ICDVDT")=<date>

This is a Code Set Versioning Date (in Fileman format). If set, it must also be killed by the calling application.

If supplied, it is assumed that the lookup is to be a versioned lookup and only active codes on that date will be included in the selection list.

If not supplied, the date will default to TODAY and all codes may be selected, active and inactive.

In both cases the display will be altered based on the date.

VARIABLES: Input ICDSYS
Coding System (from file 80.4)

ICDSYS or
^TMP("ICDEXLK",\$J,"ICDSYS")=<coding system>

This is the Coding System taken from file 80.4. If set, it must be killed by the calling application. It may be any of the following:

1 ICD ICD-9-CM

2	ICP	ICD-9 Proc
30	10D	ICD-10-CM
31	10P	ICD-10-PCS

If supplied, the lookup will only look in the cross-references specific for that coding system.

VARIABLES: Input

ICDFMT

Display Format (numeric, 1-4)

ICDFMT or

^TMP("ICDEXLK", \$J, "ICDFMT")=<display format>

This is a flag defining a Display Format (numeric, 1-4). If set, it must be killed by the calling application.

1 = Fileman format, code and short text (default)

250.00 DMII WO CMP NT ST UNCNTR

2 = Fileman format, code and description

250.00 DIABETES MELLITUS WITHOUT MENTION OF COMPLICATION, TYPE II OR UNSPECIFIED TYPE, NOT STATED AS UNCONTROLLED

3 = Lexicon format, short text followed by code

DMII WO CMP NT ST UNCNTR (250.00)

4 = Lexicon format, description followed by code

DIABETES MELLITUS WITHOUT MENTION OF COMPLICATION, TYPE II OR UNSPECIFIED NOT STATED AS UNCONTROLLED (250.00)

VARIABLES: Input

X

This is the user's input, if not available the user will be prompted for input.

VARIABLES: Input

FILEMAN

FileMan Variables used

DIC, DIC(0), DIC("A"), DIC("B"), DIC("S"), DIC("W"), DIC("?N", <file>)

FileMan Variables not used:

DIC("DR"), DIC("PTRIX", <fm>, <to>, <file>), DIC("T"), DIC("V"), DIC("?PARAM")

DIC(0) parameters applicable to a versioned file

A Ask the entry; if erroneous, ask again
B Only the B index is used
E Echo information
F Forget the lookup value
I Ignore the special lookup program
M Multiple-index lookup allowed
N Uppercase, IEN lookup allowed (not forced)
O Only find one entry if it matches exactly
S Suppresses display of .01
T Search until user selects or enters ^^
X EXact match required
Z Zero node in Y(0), external form in Y(0,0)

DIC(0) parameters NOT applicable to a versioned file and not used

C Versioned cross-references not turned off
K Primary Key not established
L Learning a new entry LAYGO not allowed
n ICD has no pure numeric entries
Q Input is pre-processed, ?? not necessary
U All values are external
V Verification is not optional

FileMan Variables KILLed:

DLAYGO
DINUM

VARIABLES: Output Y

Fileman Compliant:

Y IEN ^ Code
If DIC(0) containing "Z"
Y(0) 0 Node
Y(0,0) Code

Non-Fileman Compliant, DIC(0) contains "Z"

Y(0,1) \$\$ICDDX or \$\$ICDOP
Y(0,2) Long Description

COMPONENT: \$\$LKT(X,ROOT,CDT,SYS,VER,OUT)

This entry point is a lookup for text in either file 80 or 80.1 It is similar to the special lookup except there is no prompt for input or display for selection (silent) and intended for GUI applications.

VARIABLES: Input X

This is a string of text to search for.

VARIABLES: Input ROOT

This is either a global root or file number to

indicate either the DIAGNOSIS file 80 or the OPERATIONS/PROCEDURE file 80.1

VARIABLES: Input CDT
This is the Code Set Versioning date (Fileman format) used to determine the status of a code (active or inactive) It normally represents the date that service was provided to the patient (HIPAA). However, it may also represent the date of onset, visit date or movement date depending on the application calling the lookup.

VARIABLES: Input SYS
This is a coding system identifier (pointer to file 80.4)

- 1 = ICD-9-CM
- 2 = ICD-9-PCS
- 30 = ICD-10-CM
- 31 = ICD-10-PCS

VARIABLES: Input VER
This is the versioned flag (boolean) to indicate if the lookup is to be versioned or not:

- 0 No Include all codes, active and inactive
- 1 Yes Include only Active codes for date specified

VARIABLES: Input OUT
This is a flag that defines the output format:

- 1 Fileman, Code and Short Text (default)
250.00 DMII WO CMP NT ST UNCNTR
- 2 Fileman, Code and Description
250.00 DIABETES MELLITUS WITHOUT MENTION OF COMPLICATION TYPE II OR UNSPECIFIED TYPE, NOT STATED AS UNCONTROLLED
- 3 Lexicon, Short Text and Code
DMII WO CMP NT ST UNCNTR (250.00)
- 4 Lexicon, Description and Code
DIABETES MELLITUS WITHOUT MENTION OF COMPLICATION, TYPE II OR UNSPECIFIED, NOT STATED AS UNCONTROLLED (250.00)

VARIABLES: Output \$\$LK
This is the number of entries found

The entries will be included in a ^TMP Global Array:

```
^TMP(ID,$J,"SEL")
^TMP(ID,$J,"SEL",0)=# of entries
^TMP(ID,$J,"SEL",#)=IEN ^ Display Text
```

Where ID is a package namespaced subscript:

```
ICD9 - for file #80
ICD0 - for file #80.1
```

COMPONENT: \$\$VER(SYS,REL)
This API returns the current Coding System version, the previous Coding System version or the next Coding System version based on input parameters.

VARIABLES: Input SYS
This is a pointer to the coding system file 80.4

VARIABLES: Input REL
This input parameter indicates the relationship of the output coding system to the input coding system (Optional)

0 N/A - Return the current version (default)
1 Return the next version
-1 Return the previous version

VARIABLES: Output \$\$VER
This is a 5 piece string containing:

1 Coding System (pointer to file 80.4)
2 Coding System Nomenclature
3 Coding System Abbreviation
4 File Number containing the Coding System
5 Date Coding System was Implemented or
-1 on error

COMPONENT: Y(ROOT,IEN,CDT,FMT)
Given the global root or file number, the Internal Entry Number (IEN) and a date, this API will return the equivalent of FileMan's output variable Y without having to perform the lookup.

VARIABLES: Input ROOT
This is either an ICD global root or file number.

VARIABLES: Input IEN
This is an Internal Entry Number in the file identified by the input parameter ROOT.

VARIABLES: Input CDT
This is a code set versioning date used to returned versioned (date sensitive) data from the ICD files.

VARIABLES: Input FMT
 This is a output format flag (optional, default 0).
 0 Return standard Fileman Y - IEN ^ CODE
 1 Return Expanded Y as if DIC(0) contained a "Z"

VARIABLES: Output Y
 Input parameter FMT = 0 or 1
 Y = IEN ^ Code
 Input parameter FMT = 1
 FileMan Compliant
 Y(0) = 0 Node (aka Code)
 Y(0,0) = .01 Field (aka Code)
 Non-FileMan Compliant
 Y(0,1) = \$\$ICDDX or \$\$ICDOP
 Y(0,2) = Versioned Long Description

COMPONENT: TOKEN(TEXT,ROOT,SYS,ARY)
 This API parses text into words/tokens and saves them in a local array for later processing. Words and tokens not found in the file and coding system identified by the input parameters are not included in the output array.

VARIABLES: Input TEXT
 This is a text string to parse.

VARIABLES: Input ROOT
 This is a global root or file number (required)
 ^ICD9(or 80
 ^ICD0(or 80.1

VARIABLES: Input SYS
 This is the coding system (Required)
 1 or ICD or ICD-9-CM
 2 or ICP or ICD-9 Proc
 30 or 10D or ICD-10-CM
 31 or 10P or ICD-10-PCS

VARIABLES: Both .ARY
 This is the output array passed by reference containing words parsed from the input string TEXT and arranged by frequency of use (Required)
 ARY(USE,SYS)=WORD
 Where USE is the number of times the word was used

in the file identified by ROOT and coding system
SYS and WORD is a single word found in designated
coding system

COMPONENT: \$\$WORD(WORD,ROOT,SYS)
This API determines if a word is found in a file or a coding
system identified by the input parameters

VARIABLES: Input WORD
This is a single word.

VARIABLES: Input ROOT
This is a global root or file number (optional)

^ICD9(or 80
^ICD0(or 80.1

VARIABLES: Input SYS
This is the coding system (Optional)

1 or ICD or ICD-9-CM
2 or ICP or ICD-9 Proc
30 or 10D or ICD-10-CM
31 or 10P or ICD-10-PCS

VARIABLES: Output \$\$WORD
This is a Boolean value indicating if a word is
contained in a set (file or system).

1 = Word was found

If ROOT is not supplied, the word was
found in either file 80 or 80.1

If SYS is not supplied, the word was
found in the file designated by ROOT
in any coding system in the file

If both ROOT and SYS are supplied,
the word was found in the specified
coding system

0 = Word was not found

COMPONENT: \$\$ICDIDS(FILE, CODE, ARY)
This API returns an array of Diagnosis or Procedure code
Identifiers used in the calculation of DRG groups.

VARIABLES: Input FILE
This is the ICD file number used to retrieve the
identifier codes (Required):

80 = ICD Diagnosis file
80.1 = ICD Operation/Procedure file

VARIABLES: Input CODE
This is an Internal Entry Number (IEN) in the file

specified (Required).

VARIABLES: Both ARY
This is a local array of identifiers found for the code identified input parameters FILE and CODE.

ARY(<identifier>)=""

VARIABLES: Output \$\$ICDIDS
This is the number of identifiers found for the code identified by the input parameters FILE and CODE, or upon error:

-1^error message

COMPONENT: \$\$ICDID(FILE, ID, CODE)
This API checks if a specified ICD identifier exist for a code identified by the input parameters FILE and CODE.

VARIABLES: Input FILE
This is the ICD file number used to retrieve the identifier codes (Required):

80 = ICD Diagnosis file
80.1 = ICD Operation/Procedure file

VARIABLES: Input ID
This is a Diagnosis or Procedure code identifier (required)

VARIABLES: Input CODE
This is an Internal Entry Number (IEN) in the file specified (Required).

VARIABLES: Output \$\$ICDID
Boolean value

1 if identifier was found for code
0 if identifier was not found for code

or upon error -1^error message

COMPONENT: \$\$ISOWNCC(IEN, CDT, FMT)
This API returns the Complication/Comorbidity (CC) value for an ICD Diagnosis code when the primary diagnosis is its own CC/MCC.

VARIABLES: Input IEN
This is the Internal Entry Number (IEN) of the ICD Diagnosis file #80.

VARIABLES: Input CDT
Date to use to extract CC (default TODAY)

VARIABLES: Input FMT
This is a flag that controls the output format:

0 = CC only (default)
1 = CC ^ Effective Date

VARIABLES: Output \$\$ISOWNCC
 Complication/Comorbidity (CC)

DX is Own CC	Format	Output
Yes	0	CC Value
Yes	1	CC Value ^ Effective Date
No	N/A	0 (zero)

or upon error -1^error message

COMPONENT: \$\$ICDRGCC (DRG, CDT)
This API returns the CC/MCC flag from DRG file #80.2

VARIABLES: Input DRG
 This is an Internal Entry Number for the DRG file
 80.2 (required)

VARIABLES: Input CDT
 Date to use to extract CC/MCC flag (default TODAY)

VARIABLES: Output \$\$ICDRGCC
 This is the Complication/Comorbidity/Major CC flag

0	No CC or MCC
1	CC present
2	MCC present
3	CC or MCC present

or upon error -1^error message

COMPONENT: \$\$DRG (CODE, CDT)
This API returns basic information about a DRG.

VARIABLES: Input CODE
 DRG code, internal or external format (Required)

VARIABLES: Input CDT
 Date to check status for, FileMan format (default
 = TODAY)

If CDT < 10/1/1978, use 10/1/1978
If CDT > DT, validate with In/Activation Dates
If CDT is year only, use first of the year
If CDT is year and month, use first of the
month

VARIABLES: Output \$\$DRG
 Returns an 22 piece string delimited by the
 up-arrow (^) the pieces are:

1	DRG name (field #.01)
2	Weight (field #2)
3	Low Trim (days) (field #3)

4 High Trim (days) (field #4)
 5 MDC (field #5)
 6 Surgery Flag (field #.06)
 7 <null>
 8 Avg Length of Stay (days) (field 10)
 9 Local Low Trim Days (field #11)
 10 Local High Trim Days (field #12)
 11 <null>
 12 Local Breakeven (field #13)
 13 Activation Date (.01 field, 66 multiple)
 14 Status (.03 field, 66 multiple)
 15 Inactivation Date (.01 field, 66 multiple)
 16 Effective date (.01 field, 66 multiple)
 17 Internal Entry Number (IEN)
 18 Effective date (.01 field, 66 multiple)
 19 Reference (field #900)
 20 Weight (Non Affil) (field #7)
 21 Weight (Int Affil) (field #7.5)
 22 Message

or

-1^Error Description

COMPONENT: \$\$DRGD(CODE,ARY,CDT)
 Returns an unformatted DRG Description.

VARIABLES: Input CODE
 ICD Code, Internal or External Format (required)

VARIABLES: Both ARY
 Input: Name of Output Array for description
 e.g. "ABC" or "ABC("TEST")"
 Default = ^TMP("DRGD",\$J)

Output: Description in array

@ARY(1:n) - Description (lines 1-n)
 @ARY(n+1) - Blank
 @ARY(n+1) - Warning Message

or

-1^Error Description

NOTE:

User must initialize ^TMP("DRGD",\$J)
 if used. The data is place in the
 array unformatted, exactly as it is in
 the DESCRIPTION multiple (sub-files
 #80.068 or #80.168)

SEE ALSO:

\$\$DRGDES^ICDEX(IEN,CDT,.ARY,LENGTH) to retrieve the description formatted into string lengths specified by input parameter for length.

VARIABLES: Input CDT
Date to screen against (default = TODAY)

If CDT < 10/1/1978, use 10/1/1978
If CDT > DT, use DT
If CDT = year only, use 01/01/yyyy
If CDT = year & month, use mm/01/yyyy

VARIABLES: Output \$\$DRGD
This is the number of lines in description output array.

COMPONENT: \$\$DRGDES(IEN,CDT,ARY,LEN)
This API returns the DRG Description formatted into string lengths specified by the calling application.

VARIABLES: Input IEN
Internal Entry Number of DRG file 80.2

VARIABLES: Input CDT
Date to screen against (default = TODAY)

VARIABLES: Both .ARY
This is a local array passed by reference containing the DRG description. The text is formatted into string lengths specified by the LEN input parameter.

VARIABLES: Input LEN
Length of line of the description in the output array

Missing Defaults to 79
Less than 25 Defaults to 25

VARIABLES: Output \$\$DRGDES
This is the number of lines in description output array.

COMPONENT: \$\$DRGN(CODE)
This API returns the Internal Entry Number (IEN) of the DRG specified by a DRG code.

VARIABLES: Input CODE
This is a DRG code.

VARIABLES: Output \$\$DRGN
This is the IEN of the DRG code specified.

COMPONENT: \$\$EFD(X)
This is an interactive API that will prompt the user for an effective date in a range of dates.

VARIABLES: Output \$\$EFD
This is a 3 piece "^" delimited string containing an effective date in both internal and external formats:

1	Date Fileman format	nnnnnnn
2	Date External Short Format	mm/dd/yyyy
3	Date External Long Format	Mmm dd, yyyy

or

"^^" if the user enters double up-arrows
"^" if the user enters a single up-arrow
"" if the user times out

The earliest possible date is Oct 1, 1978, the initial ICD implementation date in the VA.

If today's date is less than the implementation date of ICD-10, then the latest possible date is 3 years from the ICD-10 implementation date.

If today's date is greater than the implementation date of ICD-10, then the latest possible date is 3 years from today's date.

COMPONENT: \$\$GETDATE(IEN)
This API calculates the Effective Date to use retrieving ICD/DRG data based on a patient's treatment.

VARIABLES: Input IEN
This is an Internal Entry Number of the PTF file #45

VARIABLES: Output \$\$GETDATE
This is the correct "EFFECTIVE DATE" for a patient to be used retrieving DRG/ICD/CPT data (default TODAY)

"EFFECTIVE DATE" Derived from:

Census Date	^DGPT	0;13
Discharge Date	^DG(45.86	0;1
Surgery Date	^DGPT(D0,"S"	0;1
Movement Date	^DGPT(D0,"M"	0;10
Default	\$\$NOW^XLFDT	

COMPONENT: \$\$IA(FILE,IEN)
This API returns an codes Initial Activation Date based on a file number and the codes Internal Entry Number. The Initial Activation date may be different from the Last Activation date (see \$\$LA) if the code was re-used.

VARIABLES: Input FILE
This is a Global Root or File Number for either the ICD Diagnosis or ICD Procedure files (Required)

VARIABLES: Input IEN
This is an Internal Entry Number (IEN) in the specified file (Required)

VARIABLES: Output \$\$IA
Initial Activation Date

OR

-1 ^ Error Message

COMPONENT: \$\$IDSTR(FILE,IEN)
This API returns a string of ICD identifier associated with either an ICD Diagnosis or ICD Procedure code (supports legacy APIs)

VARIABLES: Input FILE
File Number or root (required)

80 or ^ICD9 = File #80
80.1 or ^ICD0 = File #80.1

VARIABLES: Input IEN
This is a Diagnosis/Procedure code IEN (required)

VARIABLES: Output \$\$IDSTR
This is a string of Identifiers delimited by a semi-colon

ID;ID;ID

COMPONENT: \$\$ISVALID(FILE,IEN,CDT)
This API determine is an ICD code is valid.

VARIABLES: Input FILE
This is a file number or global root for either the ICD Diagnosis file or the ICD Procedure file

VARIABLES: Input IEN
This is an Internal Entry Number (IEN) in the file specified.

VARIABLES: Input CDT
This is the date to use to determine if the code is valid for date (default TODAY)

VARIABLES: Output \$\$ISVALID
This is a Boolean value

1 if the code is valid
0 if the code is not valid

COMPONENT: \$\$PDXE(IEN)
This API returns the Primary Diagnosis Exclusion Code.

VARIABLES: Input IEN

This is an Internal Entry Number (IEN) for the ICD
Diagnosis file #80

VARIABLES: Output \$\$PDXE
 This is a pointer to DRG CC Exclusions file #82.13

COMPONENT: \$\$REF(IEN,CDT)
 This API returns the name of the DRG Reference Table.

VARIABLES: Input IEN
 This is an Internal Entry Number (IEN) of the DRG
 file #80.2

VARIABLES: Input CDT
 Effective date to use (default TODAY)

VARIABLES: Output \$\$REF
 Table reference associated with a DRG entry or
 null if not found

COMPONENT: \$\$VCCP(IEN,CDT,FMT)
 This API returns the CC Primary Flag for a diagnosis.

VARIABLES: Input IEN
 This is an Internal Entry Number (IEN) in the ICD
 Diagnosis file 80 (required)

VARIABLES: Input CDT
 This is the date to use to Extract CC Primary Flag
 (default TODAY)

VARIABLES: Input FMT
 Is is a flag to determine the output format
 (optional):

 0 = CC Primary Flag only (default)
 1 = CC Prim Flag^Effective Date^Value

VARIABLES: Output \$\$VCCP
 This the CC Primary Flag in one of two formats:

 CC Primary Flag only (FMT=0)
 CC Primary Flag^Effective Date^Value (FMT=1)

COMPONENT: \$\$DRGW(IEN)
 This API returns the DRG Weighted Work Unit (WWU)

VARIABLES: Input IEN
 This is an Internal Entry Number (IEN) of the DRG
 file 80.2

VARIABLES: Output \$\$DRGW
 This is the Weighted Work Unit (WWU) for a DRG

COMPONENT: \$\$DRGC(IEN)
 This API returns the DRG code.

VARIABLES: Input IEN
This is an Internal Entry Number (IEN) of the DRG file 80.2

VARIABLES: Output \$\$DRGC
This is a DRG Code (field .01)

COMPONENT: \$\$MDCN(IEN)
This API returns the name of a Major Diagnostic Category (MDC)

VARIABLES: Input IEN
This is the Internal Entry Number (IEN) for file 80.3

VARIABLES: Output \$\$MDCN
This is a Major Diagnostic Category Name

COMPONENT: \$\$HDR(FILE)
This API returns the header node of either file 80 or 80.1.

VARIABLES: Input FILE
This is a File Number or Global Root

80 or ^ICD9(
80.1 or ^ICD0(

VARIABLES: Output \$\$HDR
This is the header node of either the ICD Diagnosis file 80 or the Operation Procedure file 80.1

^ICD9(0)
^ICD0(0)

COMPONENT: \$\$IEN(CODE,ROOT,SYS)
This API returns an internal entry number for a code based on file/global root and coding system.

This API is similar to \$\$CODEABA^ICDEX except it will also return IENs for codes excluded from lookup and VA Local Codes. Its primary purpose to to support file maintenance. Use with great caution.

DO NOT USE in any application that requires codes and text to be versioned (date sensitive).

VARIABLES: Input CODE
This is an ICD Diagnosis or Procedure Code from either the ICD-9 or ICD-10 coding systems (required)

VARIABLES: Input ROOT
This is a file number or global root (optional)

^ICD9(or 80
^ICD0(or 80.1

VARIABLES: Input SYS
 This is a coding system (optional)

 1 = ICD-9 Diagnosis
 2 = ICD-9 Procedure
 30 = ICD-10 Diagnosis
 31 = ICD-10 Procedure

VARIABLES: Output \$\$IEN
 Returns the Internal Entry Number (IEN) for a CODE
 or -1 if not found

COMPONENT: \$\$SDH(FILE,IEN,ARY)
 This API returns a history of Short Description changes by
 date.

VARIABLES: Input FILE
 This is an ICD file number:

 80 = ICD Diagnosis file
 80.1 = ICD Operation/Procedure file

VARIABLES: Input IEN
 This is an Internal Entry Number (IEN) in the file
 specified.

VARIABLES: Input .ARY
 This is a local array name passed by reference
 that will contain the code's short description
 history.

VARIABLES: Output \$\$SDH
 This is a three piece "^" delimited string
 containing:

 1 The number of short descriptions found
 2 The earliest date found
 3 The latest date found

VARIABLES: Output ARY
 This is a local array containing a history of
 Short Descriptions by date:

 ARY(0)= # ^ Earliest Date ^ Latest Date
 ARY(1)=Long Description

COMPONENT: \$\$LDH(FILE,IEN,ARY)
 This API returns a history of Long Description changes by
 date.

VARIABLES: Input FILE
 This is an ICD file number:

 80 = ICD Diagnosis file
 80.1 = ICD Operation/Procedure file

VARIABLES: Input IEN

This is an Internal Entry Number (IEN) in the file specified.

VARIABLES: Input .ARY
This is a local array name passed by reference that will contain the code's long description history.

VARIABLES: Output \$\$LDH
This is a three piece "^" delimited string containing:

- 1 The number of long descriptions found
- 2 The earliest date found
- 3 The latest date found

OR -1 ^ Error Message

VARIABLES: Output ARY
This is a local array containing a history of Long Descriptions by date:

ARY(0)= # ^ Earliest Date ^ Latest Date
ARY(DATE)=Long Description

5749 Updating DD 'VR' Nodes

CUSTODIAL PACKAGE: VA FILEMAN

SUBSCRIBING PACKAGE: LEXICON UTILITY

The Lexicon needs to be able to update the DD "VR" nodes during data updates. The Lexicon exports data in the export global ^LEXM. This export global is created by comparing the development account with a gold account and recording the changes in ^LEXM in the form of executable SET and KILL statements. Also recorded are the DD "VR" nodes.

Example of export global for patch LEX*2.0*80 for file 757.11:

```
^LEXM(757.11,4)=S ^DD(757.11,0,"VR")="2.0"  
^LEXM(757.11,5)=S ^DD(757.11,0,"VRPK")="LEX"  
^LEXM(757.11,6)=S ^DD(757.11,0,"VRRV")="80^3131001"
```

Example of export global for patch ICD*18.0*57 for file 80:

```
^LEXM(80,5391580)=S ^DD(80,0,"VR")="18.0"  
^LEXM(80,5391581)=S ^DD(80,0,"VRPK")="ICD"  
^LEXM(80,5391582)=S ^DD(80,0,"VRRV")="57^3131001"
```

Data installation is done by \$ORDERing through the ^LEXM export global and executing the MUMPS code found in the global. When the above export global is installed at a site, the version number for file 757.11 is updated to 80 (for LEX*2.0*80) and given the effective date of the the ICD-10 implementation

date. The effective date is not the date released, it is the date that the data becomes effective, and in this case it is the date the ICD-10 data is effective.

This activity only occurs during the post-install of an ICD, CPT or Lexicon KIDS Installation containing data. It is this method of exporting only the changes in a series of SET and KILL statements that allows for the distribution of large quantities of data without forcing users off the system.

```
        USAGE: Private                ENTERED: NOV 30,2011
        STATUS: Active                EXPIRES:
        DURATION: Till Otherwise Agr  VERSION:
        FILE:   ROOT: DD(
DESCRIPTION:  TYPE: File
^DD(FILE,0,'VR')
^DD(FILE,0,'VRPK')
^DD(FILE,0,'VRRV')
```

ROUTINE:

5755 ICD CODING SYSTEMS

```
CUSTODIAL PACKAGE: DRG GROUPER
SUBSCRIBING PACKAGE: LEXICON UTILITY
        USAGE: Private                ENTERED: DEC 24,2011
        STATUS: Pending              EXPIRES:
        DURATION: Till Otherwise Agr  VERSION:
        FILE: 80.4                   ROOT: ICDS(
DESCRIPTION:  TYPE: File
Lexicon Utility has all privileges as though it were the custodial
package.
```

27. Lexicon as a Custodian

457 Expression file #757.01

```
CUSTODIAL PACKAGE: LEXICON UTILITY
SUBSCRIBING PACKAGE:
        USAGE: Supported              ENTERED: APR 26,1994
        STATUS: Active                EXPIRES:
        DURATION: Next Version        VERSION: 1.0
        FILE: 757.01                 ROOT: GMP(757.01,
DESCRIPTION:  TYPE: File
The Clinical Lexicon Utility will maintain static internal entry numbers
(IENs) for the Expression file (#757.01). As a result, this file may be
pointed to to retrieve the Display Text (.01) for both current Expressions
and formerly used (deleted) Expressions.
^GMP(757.01,D0,0)
.01      DISPLAY TEXT                0;1      Direct Global Read & w
```

The Display Text contained in the Clinical Lexicon is the text which will be used in all display/print routines.

1511 GMPTU

CUSTODIAL PACKAGE: LEXICON UTILITY
SUBSCRIBING PACKAGE: AUTOMATED INFO COLLECTION SYS
USAGE: Private ENTERED: MAR 8,1996
STATUS: Active EXPIRES:
DURATION: Till Otherwise Agr VERSION:
FILE: ROOT:
DESCRIPTION: TYPE: Routine

The Automated Information Collection System has the ability to print lists of terms based on the Clinical Lexicon on Encounter Forms. When the forms are scanned and data is passed the PCE, the ICD9 diagnosis code associated with the term is required to populate the Purpose of Visit. This agreement is to allow AICS to use the call ICDONE^GMPTU (and its successor) ICDONE^LEXU to determine the correct, or best ICD9 Diagnosis code associated with the selected term. Input variable is the pointer to the clinical lexicon entry in file 757.01. Output is the ICD9 code, or null if none is found. This will be coded in such a way as when Clinical Lexicon converts to the LEX namespace that no changes will be required.

ROUTINE: GMPTU
COMPONENT: ICDONE
VARIABLES: INPUT Type: Input

The input to this function is the pointer to the Clinical Lexicon file (757.01) as the only parameter. This value is retrieved for other call to the clinical lexicon.

OUTPUT Type: Output

The function returns the ICD9 Diagnosis most appropriate for the term, or null if none exists.

Returns the best ICD9 code to associate with a clinical lexicon entry.

1571 Expression file 757.01

CUSTODIAL PACKAGE: LEXICON UTILITY
SUBSCRIBING PACKAGE:
USAGE: Supported ENTERED: AUG 7,1996
STATUS: Active EXPIRES:
DURATION: Till Otherwise Agr VERSION:
FILE: 757.01 ROOT: LEX(757.01,
DESCRIPTION: TYPE: File

The Lexicon Utility (version 2.0 and greater) will maintain static internal entry numbers (IENs) for the Expression file (#757.01). As a result, this file may be pointed to to retrieve the Display Text (.01) for both current Expressions and deactivated Expressions (Deactivation Flag 757.01;9 1;5 set to 1). This agreement is a follow-on to DBIA 457 (version 1.0) and is re-issued to include the package name, namespace and global root changes occurring in version 2.0. This is not an amendment to

457.

	Version 1.0	Version 2.0
Package name	Clinical Lexicon Utility	Lexicon Utility
Namespace	GMPT	LEX Expression
File Global Root	^GMP(757.01, ^LEX(757.01,D0,0)	^LEX(757.01,
.01	Display Text	0;1
		Direct Global Read & w This Display Text contained in the Lexicon Utility is the text which will be used in all display/print routines.

1573 LEXU

CUSTODIAL PACKAGE: LEXICON UTILITY
SUBSCRIBING PACKAGE:

USAGE: Supported ENTERED: AUG 7,1996
STATUS: Active EXPIRES:
DURATION: Till Otherwise Agr VERSION:

FILE: ROOT:
DESCRIPTION: TYPE: Routine

LEXU is a utility routine for the Lexicon Utility which contains functions useful in retrieving classification code(s) for a term. This agreement is a follow-on to DBIA 10148 (version 1.0) and is re-issued to include the package name, namespace, routine name and global root changes occurring in version 2.0. This is not an amendment to 10148.

ROUTINE: LEXU
COMPONENT: \$\$ICDONE(IEN,DATE)
VARIABLES: IEN Type: Input

Internal Entry Number in the Expression file ^LEX(757.01).

DATE Type: Input

This is a date in Fileman format used to check if a code is active or inactive on a specified date. If not supplied, it will default to TODAY.

Returns either a single active ICD-9 code linked to the Lexicon expression or Null if no ICD-9 code is found.

COMPONENT: \$\$ICD(IEN,DATE)
VARIABLES: IEN Type: Input

Internal Entry Number in the Expression file ^LEX(757.01).

DATE Type: Input

This is a date in Fileman format used to check if a code is active or inactive on a specified date. If not supplied, it will default to TODAY.

Returns either a string of active ICD-9 codes linked to an expression (separated by semicolon, i.e., ICD;ICD;ICD) or Null if no ICD-9 codes are found.

COMPONENT: \$\$CPTONE(IEN,DATE)
VARIABLES: IEN Type: Input

Internal Entry Number in the Expression file ^LEX(757.01).

DATE Type: Input

This is a date in Fileman format used to check if a code is active or inactive on a specified date. If not supplied, it will default to TODAY.

Returns either a single active CPT-4 code linked to the Lexicon expression or Null if no CPT-4 code is found.

COMPONENT: \$\$DSMONE(IEN)

VARIABLES: IEN Type: Input

Internal Entry Number in the Expression file ^LEX(757.01).

Returns either a single DSM-IV code linked to the Lexicon expression or Null if no DSM-IV code is found.

COMPONENT: \$\$CPCONE(IEN,DATE)

VARIABLES: IEN Type: Input

Internal Entry Number in the Expressions file ^LEX(757.01).

DATE Type: Input

This is a date in Fileman format used to check if a code is active or inactive on a specified date. If not supplied, it will default to TODAY.

Returns either a single active HCPCS code linked to the Lexicon expression or Null if no HCPCS codes exist. HCPCS stands for Healthcare Financing Administration (HCFA) Common Procedure Coding System.

1597 INFO^LEXA Expression Information

CUSTODIAL PACKAGE: LEXICON UTILITY

SUBSCRIBING PACKAGE:

USAGE: Supported

ENTERED: AUG 18,1996

STATUS: Active

EXPIRES:

DURATION: Till Otherwise Agr

VERSION:

FILE:

ROOT:

DESCRIPTION:

TYPE: Routine

LEXA is used by the Lexicon Utility to perform a silent look-up and return an array of the expression found.

ROUTINE: LEXA

COMPONENT: INFO(IEN,DATE)

VARIABLES: IEN Type: Input

Internal Entry Number in the Expression file #757.01.

LEX Type: Output

The local array LEX("SEL") contains the major concept, synonyms, lexical variants, associated codes (i.e., ICD, CPT, DSM, etc.), the expression definition (if one exists), the semantic class, the semantic type, and all VA classification sources. See the Lexicon Utility's Technical Manual for a detailed description of this array.

DATE Type: Input

This is a date in Fileman format used to check if a code is active or inactive on

a specified date. If not supplied, it will default to TODAY. Active codes will be retrieved and displayed.

This entry point allows applications to retrieve information about an expression without conducting a search.

1599 LEXICON USER DEFAULTS - Filter LEXDFL

CUSTODIAL PACKAGE: LEXICON UTILITY

SUBSCRIBING PACKAGE: PROBLEM LIST

USAGE: Private

ENTERED: AUG 19,1996

STATUS: Active

EXPIRES:

DURATION: Till Otherwise Agr VERSION:

FILE:

ROOT:

DESCRIPTION:

TYPE: Routine

The entry point EN1^LEXDFL will be used to setup user default filter for conducting searches in the Lexicon Utility. This entry point, along with EN1^LEXDCC, EN1^LEXDVO, EN1^LEXDCX and EN1^LEXDDS replaces ^GMPTDUSR used in verion 1.0 of the Clinical Lexicon Utility (see DBIA 339).

ROUTINE: LEXDFL

COMPONENT: EN1(LEXAP)

VARIABLES: LEXAP Type: Input

LEXAP is the Internal Entry Number of the Subset Definition file (#757.2) where the application definition is located.

This entry point allows a single user to edit their default look-up filter for the Lexicon Utility.

1601 LEXICON USER DEFAULTS - Display LEXDCC

CUSTODIAL PACKAGE: LEXICON UTILITY

SUBSCRIBING PACKAGE: PROBLEM LIST

USAGE: Private

ENTERED: AUG 19,1996

STATUS: Active

EXPIRES:

DURATION: Till Otherwise Agr VERSION:

FILE:

ROOT:

DESCRIPTION:

TYPE: Routine

The entry point EN1^LEXDCC(LEXAP) will be used to setup user default display (classification codes) for conducting searches in the Lexicon Utility. This entry point along with EN1^LEXDFL, EN1^LEXDVO, EN1^LEXDCX and EN1^LEXDDS replaces ^GMPTDUSR used in verion 1.0 of the Clinical Lexicon Utility (see DBIA 339).

ROUTINE: LEXDCC

COMPONENT: EN1(LEXAP)

VARIABLES: LEXAP Type: Input

LEXAP is the Internal Entry Number of the Subset Definition file (#757.2) where the application definition is located.

This entry point allows a single user to edit their default look-up display for the Lexicon Utility.

1603 LEXICON USER DEFAULTS – Vocabulary LEXDVD

CUSTODIAL PACKAGE: LEXICON UTILITY

SUBSCRIBING PACKAGE: PROBLEM LIST

USAGE: Private

ENTERED: AUG 19,1996

COMPONENT: EN1 (LEXAP)
VARIABLES: LEXAP Type: Input

LEXAP is the Internal Entry Number of the Subset Definition file (#757.2) where the application definition is located.

This entry point allows a single user to list their Lexicon Utility defaults to a device (terminal or printer).

1609 Lexicon Setup LEXSET

CUSTODIAL PACKAGE: LEXICON UTILITY
SUBSCRIBING PACKAGE:

USAGE: Supported ENTERED: AUG 19,1996
STATUS: Active EXPIRES:
DURATION: Till Otherwise Agr VERSION:
FILE: ROOT:
DESCRIPTION: TYPE: Routine

The Lexicon Utility uses LEXSET to setup search parameters based on applications definitions, subset definitions and user defaults stored in the Subsets Definition file (#757.2). These search parameters are stored in the global array ^TMP("LEXSCH",\$J).

ROUTINE: LEXSET
COMPONENT: CONFIG (LEXNS, LEXSS, DATE)
VARIABLES: LEXNS Type: Input

LEXNS is an application identifier (formerly namespace) which tells the setup routines which application definition in file 757.2 to use to retrieve application defaults (i.e., global, display, filter, etc.) Acceptable values for LEXNS are found in file 757.2 in the "AN" index:

^LEXT(757.2,"AN",LEXNS)

LEXSS Type: Input

LEXSS is a subset identifier which tells the setup routines which subset definition in file 757.2 to use to retrieve subset and user defaults (i.e., global, display, filter, etc.). Acceptable values for LEXSS may be found in file 7 57.2 in either the "AA" or the "AB" indexes:

^LEXT(757.2,"AA",LEXSS)
^LEXT(757.2,"AB",LEXSS)

TMP(LEXSCH Type: Output

^TMP("LEXSCH",\$J) is a global array used by the Lexicon Utility to control how a search of the Lexicon is to be conducted. It contains the following segments:

APP Application (from LEXNS)
DIS Display format

FIL Filter
 FLN File Number
 GBL Global (Fileman DIC)
 IDX Index used during the search
 LEN Length of list to display
 OVR Overwrite User Defaults flag
 SCT Shortcuts
 SVC Service
 UNR Unresolved Narrative flag
 USR User (DUZ)
 VDT Version Date Check (for
 classification codes)
 VOC Vocabulary

A detailed description of this global array may be found in the Lexicon Utility's Technical Manual.

DATE Type: Input

This is a date in Fileman format used to check classification code codes to determine if they are active or inactive on the specified date. If not supplied, it will default to TODAY.

This entry point may be used by other applications to setup parameters for conducting a search of the Lexicon Utility.

1614 Lexicon Expressions for Codes LEXCODE

CUSTODIAL PACKAGE: LEXICON UTILITY
 SUBSCRIBING PACKAGE:

USAGE: Supported ENTERED: AUG 20,1996
 STATUS: Active EXPIRES:
 DURATION: Till Otherwise Agr VERSION:

FILE: ROOT:
 DESCRIPTION: TYPE: Routine

The Lexicon Utility uses the LEXCODE routine to extract expressions (terms) in the form of Fileman's output variable "Y" based on a classification code.

ROUTINE: LEXCODE
 COMPONENT: EN(LEXSO,DATE)
 VARIABLES: LEXSO Type: Input

LEXSO is a classification code from one of several sources (i.e., ICD, CPT, DSM). A complete list of these sources can be found in the Lexicon Utility's Technical Manual.

LEXS(SAB,# Type: Output

LEXS(SAB,#)=IEN^TERM is a local array containing references to expressions linked to the classification code. SAB refers to the three-character source abbreviation of the classification system (i.e., ICD-9-CM = ICD). A description of this array and a list of the source

abbreviations can be found in the Lexicon Utility's Technical Manual.

DATE Type: Input

This is a date in Fileman format used to check if a code is active or inactive on a specified date. If not supplied, it will default to TODAY.

This entry point builds a local array containing expressions linked to an active classification code.

2288 Lexicon Utilities - LEXU

CUSTODIAL PACKAGE: LEXICON UTILITY

SUBSCRIBING PACKAGE:

USAGE: Supported

ENTERED: FEB 3,1998

STATUS: Withdrawn

EXPIRES:

DURATION: Till Otherwise Agr

VERSION:

FILE:

ROOT:

DESCRIPTION:

TYPE: Routine

LEXU is a utility routine for the Lexicon Utility which contains functions useful in retrieving classification codes for a term. This agreement is an amendment to DBA #1573.

ROUTINE: LEXU

COMPONENT: \$\$CPCONE (IEN)

VARIABLES: IEN Type: Input

Internal Entry Number in the Expressions file ^LEX(757.01).

Returns either a single HCPCS code linked to the Lexicon expression or Null if no HCPCS codes exist. HCPCS stands for Healthcare Financing Administration (HCFA) Common Procedure Coding System.

2950 Lexicon Lookup LOOK^LEXA

CUSTODIAL PACKAGE: LEXICON UTILITY

SUBSCRIBING PACKAGE: ORDER ENTRY/RESULTS REPORTING

USAGE: Supported

ENTERED: APR 16,2003

STATUS: Active

EXPIRES:

DURATION: Till Otherwise Agr

VERSION:

FILE:

ROOT:

DESCRIPTION:

TYPE: Routine

This entry point is silent and intended to support Graphical User Interface (GUI) development. The lookup returns an array of information on the expressions found. The lookup includes reordering the selection list with the most frequently used at the top, and places any exact match at the top of the list.

ROUTINE: LEXA

COMPONENT: LOOK (LEXX, LEXAP, LEXLL, LEXSUB, DATE)

VARIABLES: LEXX Type: Input

Equivalent to Fileman's variable X and contains the text to search for.

LEXAP Type: Input

This is the application identification and may be in the form of a name, namespace, or a pointer (Internal Entry Number - IEN) from an application

definition in the Subset Definition file (#757.2).

The default value for this parameter, if it is not supplied, is the one (1), pointing to the Lexicon application definition.

Included in this application definition are a number of application defaults which assist in searching the Lexicon. Application defaults included the global root, index, filter, display format, vocabulary, shortcuts, user default flag, overwrite user default flag, and the unresolved narrative flag. These are described in the Special Variable section of the Lexicon Utility Technical Manual.

At this time, there are six (6) application definitions.

Name	Namespace	IEN
----	-----	---
Lexicon	LEX	1
Problem List	GMPL	4
ICD Diagnosis	ICD	12
CPT Procedures	CPT	13
Mental Health	DSM	14
ICD, CPT, and DSM	VAC	15

To conduct a search of the Lexicon using the application defaults for the Problem List, you may pass this parameter as:

Name "PROBLEM LIST" - This form is not case sensitive, and can be found either the "B" or "C" index of file 757.2.

Namespace "GMPL" - Namespace - This form is not case sensitive, and can be found in the "AN" index of file 757.2.

Pointer 4 - This form is numeric, and is an Internal Entry Number (IEN) of file 757.2.

LEXLL Type: Input

This is a numeric value which controls the returning list length in the local array LEX("LIST"). The default value for this parameter when not supplied is five (5).

LEXSUB Type: Input

This parameter represents the vocabulary subset to use during the search. These subsets are defined in the Subset Definition file (#757.2). This parameter may be in one of three forms. To use the "Nursing" subset you may pass the parameter as:

Name "NURSING" - This form is not case sensitive and may be found in either the "B" or "C" index of file 757.2.

Mnemonic "NUR" - This form is not case sensitive and the mnemonic may be found in either the "AA" or "AB" index of file 757.2.

Pointer 2 - This form is numeric, and is an Internal Entry Number (IEN) of file 757.2.

TMP Type: Output

`^TMP("LEXFND", $J, <freq>, <ien>)`

This global array contains all of the entries found during the search. The <freq> is a negative number based on the frequency of use for a given term. <ien> is the internal entry number in the Lexicon Expression file (757.01).

`^TMP("LEXHIT", $J, <seq>)`

This global array contains the entries reviewed by the user. The Lexicon Utility reorders the list based on frequency of use and assigns a sequence number representing where on the list this entry is located.

LEX Type: Output

`LEX("LIST")`

This local array contains only those entries on the list which are currently being reviewed by the user. The third parameter to the look-up defines the length of this list.

DATE Type: Input

This is a date in Fileman format used to check if a code is active or inactive on a specified date. If not supplied, it will default to TODAY. Only active codes can be displayed and returned during a lookup.

4083 Lexicon Code Status - LEXSRC2

CUSTODIAL PACKAGE: LEXICON UTILITY

SUBSCRIBING PACKAGE:

USAGE: Supported

ENTERED: APR 14,2003

STATUS: Active

EXPIRES:

DURATION: Till Otherwise Agr VERSION:

FILE:

ROOT:

DESCRIPTION:

TYPE: Routine

ROUTINE: LEXSRC2

COMPONENT: \$\$STATCHK (CODE, DATE, .LEX, SAB)

VARIABLES: CODE Type: Input

This is a code taken from a classification system contained in the Lexicon (i.e., ICD, CPT, etc.)

DATE Type: Input

This is the date used to determine if a code was either active or inactive on a specific date. If not supplied, TODAY will be used as the date.

.LEX Type: Input

(Optional) This is a local array, passed by reference. When passed it will return information about the code.

SAB Type: Input

(Optional) This is the source of the code. It is either a pointer to the CODING SYSTEMS file 757.03 or the source abbreviation expressed as the first 3 characters of the source in file 757.03.

\$\$STATCHK Type: Output

This is a two piece "^" delimited string in the following formats:

RETURNS

INDICATES

1 ^ IEN ^ Date

The code is active on the date returned and stored in
^LEX(757.02,IEN,0)

0 ^ IEN ^ Date

The code is inactive on the date returned and stored in
^LEX(757.02,IEN,0)

0 ^ -1

Code is not found in the Lexicon

LEX Type: Output

(Optional) This is a local array passed by reference. If passed it will contain information about the code in the following formatted subscripts:

LEX(0) = <ien 757.02> ^ <code>

2-piece String containing the IEN of the code and the code

LEX(1) = <ien 757.01> ^ <expression>
2-piece String containing the
IEN of the code's expression
and the expression

LEX(2) = <ien 757.03> ^ <abbr> ^ <nomen>
^ <name>
4-piece String containing the
IEN of the code's classification
system, the source
abbreviation, Nomenclature and the name
of

the classification system

This entry point is used to check the activation status of
a code in the Lexicon Utility.

4306 LEXICAL SERVICES UPDATE Protocol

CUSTODIAL PACKAGE: LEXICON UTILITY

SUBSCRIBING PACKAGE: DRG GROUPER

The subscribing protocol is: ICD CODE UPDATE EVENT
CPT/HCPCS CODES

The subscribing protocol is: ICPT CODE UPDATE EVENT

USAGE: Controlled Subscri ENTERED: DEC 3,2003

STATUS: Active EXPIRES:

DURATION: Till Otherwise Agr VERSION:

FILE: ROOT:

DESCRIPTION: TYPE: Other

This protocol is used to notify other applications and processes when the
Lexicon Utility or the Lexicon Change file is updated.

The Lexicon is updated using a temporary maintenance global, ^LEXM. This
global is processed by the routine LEXXGI. Once processed, this protocol
is triggered and the global ^LEXM is deleted.

Required Variable LEXSCHG Array contains a listing of those Lexicon Files
(#757 - 757.41) that were updated as a result of a recent install. In the
case of the CHANGE LOG (file #757.9), new changes to SDO controlled files
will be indicated by file number and the internal entry number to the
CHANGE LOG.

The variable LEXSCHG is created while processing the Lexicon Maintenance
global ^LEXM. It will indicate what files were updated.

Example:

```
LEXSCHG(757,0)=""  
LEXSCHG(757.001,0)=""  
LEXSCHG(757.01,0)=""  
LEXSCHG(757.02,0)=""  
LEXSCHG(757.1,0)=""  
LEXSCHG(757.11,0)=""  
LEXSCHG(757.9,0)=""  
LEXSCHG(757.9,2)=80  
LEXSCHG(757.9,3)=80.1  
LEXSCHG(757.9,4)=81
```



```

LEXSCHG(757.9,"B",80,2)=""
LEXSCHG(757.9,"B",80.1,3)=""
LEXSCHG(757.9,"B",81,4)=""

```

If ICD-9-CM and/or CPT-4 changes are included in the ^LEXM global, then the following entries will be found in the local array LEXSCHG:

```

LEXSCHG(80,0)=""
LEXSCHG(80.1,0)=""
LEXSCHG(81,0)=""

```

4912 Concept Data for Code - LEXTRAN

CUSTODIAL PACKAGE: LEXICON UTILITY

SUBSCRIBING PACKAGE:

USAGE: Supported ENTERED: OCT 5,2006

STATUS: Active EXPIRES:

DURATION: Till Otherwise Agr VERSION:

DESCRIPTION: TYPE: Routine

This API will return an array of data for a given code, code source, optional date, and optional return array name. The data returned will include:

```

code
hierarchy or subset (if available)
version (if available)
legacy code (if available)
code status
fully specified name (if available)
preferred term
any applicable synonyms

```

If any of the data in the passed parameters data is incorrect or unrecognizable, the API will return an error message indicating the nature of the error. If no date is specified, then the date will default to the current system date. This API was developed specifically for the SNOMED CT code system in support of the LDSI project, but is applicable to any code system.

ROUTINE: LEXTRAN

COMPONENT: \$\$CODE(CODE, SRC, VDT, ARRAY, IENS, ID, INC)

VARIABLES: Input CODE
This is a code of a classification system that is stored in the Lexicon. Classification systems include SNOMED CT, ICD, CPT, HCPCS, etc.

VARIABLES: Input SRC
This is the mnemonic for a code system (mandatory). The allowable code system mnemonics are those that exist in the "B" index of the coding systems file (757.03) This is code system source abbreviation Lexicon.

VARIABLES: Input VDT
This is the effective date; the default if no date is specified is the current system date (optional).

VARIABLES: Both

ARRAY

This is the name of the output array. The default, if no array name is specified, is 'LEX' (optional)
The format of the output is as follows:

Output

```
        if call finds an active code for the source
        "1^LEXCODE"
        LEX      - an array containing
information about the code
        LEX(0)   - a five piece string:
                    1. code
                    2. hierarchy
                    3. version
                    4. legacy code
                    5. code status
        LEX("F") fully specified name
        LEX("P") preferred term
        LEX("S",n) synonyms (n is the nth synonym)

        if call cannot find specified code on file
        "-2^"_LEXSCNM_" code "_LEXCODE_" not on file"
        where LEXSCNM is the source name
        LEXCODE is the code

        if call finds an inactive code for the source
        "-4^"_LEXSCNM_" code "_LEXCODE_" not
active for "_LEXVDT"
        LEX      - an array containing
information about the code
        LEX(0)   - a five piece string:
                    1. code
                    2. hierarchy
                    3. version
                    4. legacy code
                    5. code status

        otherwise
        "-1^error text"
```

example of LEX array:

```
LEX(0)="67922002^Substance^20050701^T-C2500^1"
LEX("F")="Serum (Substance)"
LEX("P")="Serum"
```

VARIABLES: Input

IENS

Include expression IENS in output array (optional)
1 return IENS (2nd piece)
0 do not return IENS (default)

VARIABLES: Input

ID

Designation Identifiers (optional)
1 return Designation IDs (3rd piece)
0 do not return Designation IDs (default)

VARIABLES: Input

INC

Include Deactivated Terms (optional)
1 return Deactivated Terms
0 do not return Deactivated Terms (default)

4913 Concept Data for Text - LEXTRAN

CUSTODIAL PACKAGE: LEXICON UTILITY
SUBSCRIBING PACKAGE:

USAGE: Supported ENTERED: OCT 5,2006
STATUS: Active EXPIRES:
DURATION: Till Otherwise Agr VERSION:
FILE: ROOT:
DESCRIPTION: TYPE: Routine

This API will return an array of data for a given text, optional code source, optional date, optional subset, and optional return array name. The API will display a pick list based on the parameters passed and allow a user to select an item from the list. The API will then return the array for the item selected. The data returned will include:

- code
- hierarchy or subset (if available)
- version (if available)
- legacy code (if available)
- code status
- fully specified name (if available)
- preferred term
- any applicable synonyms

If any of the data in the 'passed' parameters data is incorrect or unrecognizable, the API will return an error message indicating the nature of the error. If no date is specified, then the date will default to the current system date. This API was developed specifically for the SNOMED CT code system in support of the LDSI project, but is applicable to any code system.

ROUTINE: LEXTRAN
COMPONENT: \$\$TEXT(TEXT,DATE,SUBSET,SOURCE,ARRAY)
VARIABLES: TEXT Type: Input

This is the search text string
(mandatory).

DATE Type: Input

This is the effective date (optional);
the default, if no date is specified, is
the current system date.

SUBSET Type: Input

This is any code system subset mnemonic
(optional). The allowable subset
mnemonics are those that exist in the
"AA" index of the subset definitions file
(757.2).

SOURCE Type: Input

This is the mnemonic for a code system
(mandatory). The allowable code system
mnemonics are those that exist in the "B"
index of the coding systems file (757.
03).

ARRAY Type: Both

This is the name of the output array. The default if no array name is specified is 'LEX' (optional) The format of the output is as follows:

Output

```
if call finds an active code for
the source
    "1^LEXCODE"
    LEX      -   an array containing
                information about the
                code
    LEX(0)  -   a five piece string:
                1. code
                2. hierarchy
                3. version
                4. legacy code
                5. code status
    LEX("F") fully specified name
    LEX("P") preferred term
    LEX("S",n) synonyms (n is
                the nth synonym)
```

```
if call cannot find specified
code on file
    "-2^"_LEXSCNM_" code
"_LEXCODE_" not on file"
    "-2^"_LEXSCNM_" code
"_LEXCODE_" not on file"
    where LEXSCNM is the source
name
                LEXCODE is the code
```

```
if call finds an inactive code
for the source
    "-4^"_LEXSCNM_" code
"_LEXCODE_" not active for "_LEXVDT"
    LEX      -   an array containing
                information about the code
    LEX(0)  -   a five piece
string:
                1. code
                2. hierarchy
                3. version
                4. legacy code
                5. code status
```

```
otherwise
    "-1^error text"
```

example of LEX array:

```
LEX(0)="67922002^Substance^20050701^T-
```

```
C2500^1"
```

```
LEX("F")="Serum (Substance)"
LEX("P")="Serum"
```

4914 Validate Code for Source - LEXTRAN

CUSTODIAL PACKAGE: LEXICON UTILITY
SUBSCRIBING PACKAGE:

USAGE: Supported ENTERED: OCT 5,2006
STATUS: Active EXPIRES:
DURATION: Till Otherwise Agr VERSION:
FILE: ROOT:
DESCRIPTION: TYPE: Routine

This API will return an array for a given text and code system indicating whether the text is valid for the specified code system. The data array returned will include the following:

An indicator of whether the text is valid for the code system

The code in the code system to which the text, if valid for code system,

belongs. If any of the passed parameters are incorrect or unrecognizable, the API will return an error message indicating the nature of the error.

ROUTINE: LEXTRAN
COMPONENT: \$\$TXT4CS (TEXT, SOURCE)
VARIABLES: TEXT Type: Input

This is the search text string
(mandatory).

SOURCE Type: Input

This is the mnemonic for a code system
(mandatory). The allowable code system
mnemonics are those that exist in the "B"
index of the coding systems file (757.
03).

Type: Output

This API returns the following output:

1^code
or
-1^error message

5006 Obtain Synonyms for Code – LEXTRAN1

CUSTODIAL PACKAGE: LEXICON UTILITY
SUBSCRIBING PACKAGE:

USAGE: Supported ENTERED: JUN 28,2007
STATUS: Active EXPIRES:
DURATION: VERSION:
DESCRIPTION: TYPE: Routine

This API will return an array for a given code and coding system. The array will contain all synonyms for the concept including the preferred term and the fully specified name. If any of the passed parameters are incorrect or unrecognizable, the API will return an error message indicating the nature of the error.

ROUTINE: LEXTRAN1
COMPONENT: \$\$GETSYN (SRC, CODE, VDT, ARRAY, IENS, ID, INC)
VARIABLES: Input SRC

This is the mnemonic for a code system
(mandatory). The allowable code system mnemonics
are those that exist in the "B" index of the

coding systems file (757.03) This is code system source abbreviation Lexicon.

VARIABLES: Input CODE
This is a code of a classification system that is stored in the Lexicon. Classification systems include SNOMED CT, ICD, CPT, HCPCS, etc.

VARIABLES: Input VDT
This is the effective date; the default if no date is specified is the current system date (optional).

VARIABLES: Both ARRAY
This is the name of the output array. The default, if no array name is specified, is 'LEX' (optional)

The format of the output is as follows:
If valid code and source are passed
"1^no of synonyms"
LEX("P") = preferred term or major concept name^IEN
LEX("F") = fully specified name^IEN (if one exists)
LEX("S",n) = the nth synonym found^IEN (if they exist)
The presence of IEN in the return array is determined by the
LEXIENS parameter. If the call does not find the code for the specified source it will return
"-2^"_LEXSCNM_" code "_LEXCODE_" not on file"
where LEXCSNM is the source name
LEXCODE is the code If an invalid source is passed the call will return
"-1^source not recognized"

VARIABLES: Input IENS
If this parameter is set to 1 the expression IEN will be included in the return array. Default is 0 - exclude IENS from return array.

VARIABLES: Input ID
Designation Identifiers (optional)
1 return Designation IDs (3rd piece)
0 do not return Designation IDs (default)

VARIABLES: Input INC
Include Deactivated Terms (optional)
1 return Deactivated Terms
0 do not return Deactivated Terms (default)

5007 Obtain Fully Specified Name – LEXTRAN1

CUSTODIAL PACKAGE: LEXICON UTILITY
SUBSCRIBING PACKAGE:

USAGE: Supported	ENTERED: JUN 28,2007
STATUS: Active	EXPIRES:
DURATION:	VERSION:
FILE:	ROOT:
DESCRIPTION:	TYPE: Routine

This API returns the fully specified name for a given coding system and

code. If any of the passed parameters are incorrect or unrecognizable, the API will return an error message indicating the nature of the error.

ROUTINE: LEXTRAN1

COMPONENT: GETFSN (LEXSRC, LEXCODE, LEXVDT)

VARIABLES: LEXSRC Type: Input

This is the mnemonic for a coding system (mandatory). The allowable code system mnemonics are those that exist in the "B" index of the coding systems file (757.03). This is code system source abbreviation Lexicon.

LEXCODE Type: Input

This is a code that belongs to a coding system that is stored in the Lexicon. Coding systems include SNOMED CT, ICD, CPT, HCPCS, etc.

LEXVDT Type: Input

This is the effective date; the default if no date is specified is the current system date (optional).

5008 Obtain Preferred Term – LEXTRAN1

CUSTODIAL PACKAGE: LEXICON UTILITY

SUBSCRIBING PACKAGE:

USAGE: Supported

ENTERED: JUN 28, 2007

STATUS: Active

EXPIRES:

DURATION:

VERSION:

FILE:

ROOT:

DESCRIPTION:

TYPE: Routine

This API returns the preferred term for a given coding system and code. If any of the passed parameters are incorrect or unrecognizable, the API will return an error message indicating the nature of the error.

ROUTINE: LEXTRAN1

COMPONENT: \$\$GETPREF

VARIABLES: LEXSRC Type: Input

This is the mnemonic for a code system (mandatory). The allowable code system mnemonics are those that exist in the "B" index of the coding systems file (757.03). This is the Lexicon code system source abbreviation.

LEXCODE Type: Input

This is a code belonging to a coding system that is stored in the Lexicon. Coding systems include SNOMED CT, ICD-9-CM, CPT, HCPCS, etc.

LEXVDT Type: Input

This is the effective date; the default if no date is specified is the current system date (optional).

5009 Obtain Designation Code – LEXTRAN1

CUSTODIAL PACKAGE: LEXICON UTILITY

SUBSCRIBING PACKAGE:

system that is stored in the Lexicon. Coding systems include SNOMED CT, ICD, CPT, HCPCS, etc.

LEXRAY Type: Both

This is the name of the output array. The default, if no array name is specified, is 'LEX' (optional) The output array will have the following format:

```
LEX(n, CODE)=""
```

where n is the nth mapped

code

code is the code which is

mapped to

e.g.

```
LEXVFL>S
```

```
V=$$GETASSN(15250008,"SCT2ICD") ZW LEX
```

```
LEX=2
```

```
LEX(1,"371.30")=""
```

```
LEX(2,"371.40")=""
```

which shows that SNOEMD CT code 15250008 is mapped to two ICD-9-CM codes.

5011 Obtain Version Identifier - LEXTRAN

CUSTODIAL PACKAGE: LEXICON UTILITY

SUBSCRIBING PACKAGE:

USAGE: Supported

ENTERED: JUN 28,2007

STATUS: Active

EXPIRES:

DURATION:

VERSION:

FILE:

ROOT:

DESCRIPTION:

TYPE: Routine

This API returns the SDO version identifier for a given coding system, code, and date. If any of the passed parameters are incorrect or unrecognizable, the API will return an error message indicating the nature of the error.

ROUTINE: LEXTRAN

COMPONENT: \$\$VERSION(LEXSRC,LEXCODE,LEXVDT)

VARIABLES: LEXSRC Type: Input

This is the mnemonic for a coding system (mandatory). The allowable coding system mnemonics are those that exist in the "B" index of the coding systems file (757.03). This is the Lexicon coding system source abbreviation.

LEXCODE Type: Input

This is a code belonging to a coding system that is stored in the Lexicon. Coding systems include SNOMED CT ICD-9-CM, CPT, HCPCS, etc.

LEXVDT Type: Input

This is the effective date; the default if no date is specified is the current system date (optional).

5252 Lexicon/VBA APIs

CUSTODIAL PACKAGE: LEXICON UTILITY
SUBSCRIBING PACKAGE:

USAGE: Supported ENTERED: AUG 8,2008
STATUS: Pending EXPIRES:
DURATION: Till Otherwise Agr VERSION:
FILE: ROOT:
DESCRIPTION: TYPE: Routine
LEXASCD contains APIs for supporting the Automated Service Connected Designation (ASCD) project.

ROUTINE: LEXASCD
COMPONENT: \$\$SC(ICD,VBA,EFF,.ARY)
VARIABLES: ICD Type: Input

(Required) ICD-9-CM Diagnosis Code

VBA Type: Input

(Required) VA Disability code (Title 38)

EFF Type: Input

(Optional) Effective Date - This is the date that service was provided to the patient (aka, encounter date) and is used to check to see if the ICD code was mapped to the Disability code on that date.

.ARY Type: Input

(Optional) This is a local array, passed by reference. When passed it will return information about the ICD code and Disability codes.

\$\$SC Type: Output

If the ICD code is mapped to a VA disability, then the return value is a 5 piece "^" delimited string as follows:

#	Content	Value
1	Service Connected	1=Yes
2	Connection (Match)	1=Full 0=Partial
3	Mapping Status	1=Active 0=Inactive
4	ICD Code Status	1=Active 0=Inactive
5	Code Status	1=Active 0=Inactive

If the ICD Code is not mapped to a VA disability, then the return value is a negative 2 piece "^" delimited string as follows:

-1 ^ Not Found or Error Message

ARY Type: Output

(Optional) This is a local array passed by reference. If passed it will contain detailed information about the ICD code and Disability code. The local array will contain the following 2 subscripts:

```
ARY(1)=<ICD status>^<Date>^<Code>^<Term>
ARY(2)=<Disability
status>^<Date>^<Code>^<Term>
```

Where status is either a 1 (active) or 0 (inactive) and date is the effective date the code became either active or inactive.

This function determines if there is a partial or full service connection for an ICD code based on the ICD codes and disability codes in the Lexicon.

COMPONENT: \$\$DI(ICD,EFF,ARY)

VARIABLES: ICD Type: Input

(Required) ICD-9-CM Code

EFF Type: Input

(Optional) Effective Date (default TODAY)

.ARY Type: Input

(Optional) Local array passed by reference, returns a list of Disability codes mapped to the ICD code.

\$\$DI Type: Output

Returns the number of Disability codes mapped to an ICD code.

ARY Type: Output

(Optional) Local Array of Disability Codes passed by reference

```
ARY(0) = 5 Piece String detailing input
        Diagnosis code
```

```
1  Number of Disability Codes found
2  ICD Code
3  Status of ICD Code  1 = Active 0 =
Inactive
4  Effective Date of ICD Code Status
5  Versioned Text of ICD Code
```

```
ARY(#) = 6 Piece String detailing output
        Disability codes
```

```
1  Mapping 1 = Full 0 = Partial
2  Effective Date of Mapping
```

3 Disability Code
 4 Status of Disability Code
 5 Effective Date of Disability Code
 Status
 6 Versioned Text of Disability Code

 ARY("B",MAP,#)="" Index of Local Array

 MAP Mapping 1 = Full 0 = Partial
 # Entry Number in Array

Return the number of Disability codes an ICD code is mapped to. Optionally return an array of Disability codes an ICD code is mapped to in a local array passed by reference.

COMPONENT: \$\$DX(VBA, EFF, ARY)
 VARIABLES: VBA Type: Input

(Required) Disability Code (Title 38)

EFF Type: Input

(Optional) Effective Date (default TODAY)

.ARY Type: Input

(Optional) Local array passed by reference, return a list of ICD codes mapped to a Disability code.

\$\$DX Type: Output

The number of Diagnosis codes mapped to a Disability code.

ARY Type: Output

(Optional) Local Array of Diagnosis Codes passed by reference

ARY(0) = 5 Piece String detailing input Disability code

1 Number of Diagnosis Codes found
 2 Disability Code
 3 Status of Code 1 = Active 0 = Inactive
 Status
 4 Effective Date of Disability Code
 Status
 5 Versioned Text of Disability Code

ARY(#) = 6 Piece String detailing output Diagnosis codes

1 Mapping 1 = Full 0 = Partial
 2 Effective Date of Mapping
 3 ICD-9-CM Code
 4 Status of ICD-9-CM Code
 5 Effective Date of ICD-9-CM Code
 Status
 6 Versioned Text of ICD-9-CM Code

ARY("B",MAP,#)=" " Index of Local Array

MAP Mapping 1 = Full 0 = Partial

Entry Number in Array

Return the number of ICD Diagnosis codes a Disability code is mapped to. Optionally return an array of ICD codes a Disability code is mapped to in a local array passed by reference.

5386 Lexicon Lookup Screens - LEXU

CUSTODIAL PACKAGE: LEXICON UTILITY

SUBSCRIBING PACKAGE:

USAGE: Supported

ENTERED: MAR 13,2009

STATUS: Active

EXPIRES:

DURATION: Till Otherwise Agr VERSION:

FILE:

ROOT:

DESCRIPTION:

TYPE: Routine

This agreement includes common entry points for filtering Lexicon searches. Similar to DIC("S") screens.

ROUTINE: LEXU

COMPONENT: \$\$SC(Y, STRING, DATE)

VARIABLES: Y Type: Input

This is an Internal Entry Number (IEN) of the Lexicon's EXPRESSION file 757.01.

STRING Type: Input

This is a three piece ";" delimited string used by the filter/screen logic. The first piece is called the "inclusion string" and list the Semantic Classes and Types to include in the search. The second piece is called the "exclusion string" and list the Semantic Types to exclude from a search. The third piece is called the "source string" and list classification sources to include in the search.

Detailed Example: Problems and Diagnosis (including ICD, CPT and DSM) looks like this:

I

\$\$SC^LEXU(Y, "BEH/DIS;999/64/66/73/74/77/82
/169/170/171;ICD/CPT/CPC/DS4", DATE)

The full explanation:

Piece 1: BEH/DIS Include expressions which relate to Behaviors and Diseases or Pathologic Processes.

Piece 2:
999/64/66/73/74/77/82/169/170/171 Exclude expressions which relate to Unknown or Untyped, Governmental or Regulatory Activity, Machine Activity, Manufactured Object, Medical Device or Supplies, Conceptual Entity, Spatial Concept, Functional Concept, Intellectual Product and Language.

Piece 3: ICD/CPT/CPC/DS4 Also include expressions which are linked to ICD-9-CM, CPT-4, HCPCS and coding systems.

In the filter string, Semantic Classes are identified by a 3 character mnemonic which can be found in the "B" cross-reference of the SEMANTIC CLASS file 757.11 and the Semantic Type is identified by internal entry number of the SEMANTIC TYPE file 757.12. The coding systems are identified by a 3 character mnemonic which can be found in the "ASAB" cross-reference of the CODING SYSTEMS file 757.03.

DATE Type: Input

If applicable, this is the date service was provided to the patient and passed in Fileman format. Default is TODAY.

This entry point filters Lexicon searches based on Semantic Class/Types and Classification Codes.

DIC("S")/Screen Usage: I \$\$SC^LEXU(Y,STRING,DATE)

COMPONENT: \$\$SO(Y,STRING,DATE)

VARIABLES: Y Type: Input

This is an Internal Entry Number (IEN) of the Lexicon's EXPRESSION file 757.01.

STRING Type: Input

This string is called the "source string" and is a list classification coding systems to include in the search. The classification coding systems are identified by a 3 character mnemonic which can be found in the "ASAB" cross-reference of the CODING SYSTEMS file 757.03.

Example: ICD/CPT/CPC/DS4 Means:
Search the Lexicon and include terms that

are
linked to ICD-9-CM, CPT-4,
HCPCS and DSM-4
coding systems.

DATE Type: Input

If applicable, this is the date service
was provided to the patient and passed in
Fileman format. Default is TODAY.

This entry point filters Lexicon searches based on
Classification Codes.

DIC("S")/Screen Usage: I \$\$SO^LEXU(Y,STRING,DATE)

5547 LAB LOINC File #95.3 APIs - LEXLR

CUSTODIAL PACKAGE: LEXICON UTILITY
SUBSCRIBING PACKAGE: DSS EXTRACTS
LAB SERVICE

USAGE: Controlled Subscri ENTERED: JUL 23,2010
STATUS: Pending EXPIRES:
DURATION: VERSION:
FILE: ROOT:
DESCRIPTION: TYPE: Routine

These API(s) support the custodial transition of the LAB LOINC file
(#95.3) from Legacy LAB to Standards and Terminology Services (STS). These
API(s) provide Read Access to the LAB LOINC file (#95.3) and should be
used when accessing the file. The API(s) support Legacy LAB's
encapsulation efforts and STS's LOINC Deployment efforts.

ROUTINE: LEXLR
COMPONENT: \$\$CHKCODE(LEXCODE)
VARIABLES: LEXCODE Type: Input
LOINC Code
\$\$CHKCODE Type: Output
LOINC File IEN or Null
Check if LOINC Code exists

Example:

>W \$\$CHKCODE^LEXLR("38553-4")

38553

COMPONENT: \$\$GETCODE(LEXCIEN)
VARIABLES: LEXCIEN Type: Input
LOINC file IEN
\$\$GETCODE Type: Output
LOINC Code or Null
Get LOINC Code by IEN

Example:

>W \$\$GETCODE^LEXLR(38553)

38553-4

```

COMPONENT: GETNAME (LEXINPT, LEXINTY, .LEXNAME)
VARIABLES: LEXINPT  Type: Input
                                LOINC Code or IEN
            LEXINTY  Type: Input
                                Input Type (Optional- Default "C")
                                "C"=LOINC Code
                                "I"=LOINC IEN
            LEXNAME  Type: Output
                                LOINC Name Array subscripts:
                                ("FULLNAME")=Fully Specified Name field
                                (#80)
                                ("SHORTNAME")=Short Name filed (#81)
            Get LOINC Name Array by Code or IEN

```

Example:

```

>D GETNAME^LEXLR("38553-4",, .LEXNAME)
  ZW LEXNAME
  LEXNAME ("FULLNAME")="NARCOLEPSY ASSOCIATED
  AG:ACNC:PT:SER/PLAS:ORD"
  LEXNAME ("SHORTNAME")="Narcolepsy Assoc Ag SerPl Q1"

```

```

COMPONENT: $$STATUS (LEXINPT, LEXINTY)
VARIABLES: LEXINPT  Type: Input
                                LOINC Code or IEN
            LEXINTY  Type: Input
                                Input Type (Optional- Default "C")
                                "C"=LOINC Code
                                "I"=LOINC IEN
            $$STATUS Type: Output
                                Internal^External Status or Null
            Get LOINC Code Status by Code or IEN

```

Example:

```

>W $$STATUS^LEXLR("38340-6")
  1^DEL

```

```

COMPONENT: GETREC (LEXINPT, LEXINTY, .LEXREC)
VARIABLES: LEXINPT  Type: Input
                                LOINC Code or IEN
            LEXINTY  Type: Input
                                Input Type (Optional- Default "C")
                                "C"=LOINC Code
                                "I"=LOINC IEN
            LEXREC   Type: Output
                                LOINC Record Array subscripts:
                                RECORD ("ADJUSTMENT")
                                RECORD ("CHALLENGE")
                                RECORD ("CHANGETYPE")
                                RECORD ("CLASS")
                                RECORD ("CLASSTYPE")
                                RECORD ("CODE")
                                RECORD ("COMPONENT")
                                RECORD ("DATELASTCHANGED")
                                RECORD ("EXAMPLEUNITS")
                                RECORD ("FULLNAME")
                                RECORD ("MAPTO")
                                RECORD ("METHODTYPE")

```



```

RECORD ("PROPERTY")
RECORD ("SCALETYPE")
RECORD ("SHORTNAME")
RECORD ("STATUS")
RECORD ("SYSTEM")
RECORD ("TIME")
RECORD ("VACODE")
RECORD ("VUID")

```

Get LOINC Record Array by Code or IEN

Example:

```

>D GETREC^LEXLR("38553-4",,.LEXREC)
ZW LEXREC
LEXREC ("ADJUSTMENT")=""
LEXREC ("CHALLENGE")=""
LEXREC ("CHANGETYPE")="ADD"
LEXREC ("CLASS")="SERO"
LEXREC ("CLASSTYPE")="1^LABORATORY"
LEXREC ("CODE")="38553-4"
LEXREC ("COMPONENT")="NARCOLEPSY ASSOCIATED AG"
LEXREC ("DATELASTCHANGED")="3041103^NOV 03, 2004"
LEXREC ("EXAMPLEUNITS")=""
LEXREC ("FULLNAME")="NARCOLEPSY ASSOCIATED
AG:ACNC:PT:SER/PLAS:ORD"
LEXREC ("MAPTO")=""
LEXREC ("METHODTYPE")=""
LEXREC ("PROPERTY")="ACNC"
LEXREC ("SCALETYPE")="Ordinal"
LEXREC ("SHORTNAME")="Narcolepsy Assoc Ag SerPl Q1"
LEXREC ("STATUS")=""
LEXREC ("SYSTEM")="SER/PLAS"
LEXREC ("TIME")="POINT"
LEXREC ("VACODE")=""
LEXREC ("VUID")=4681780

```

```

COMPONENT: $$VERSION()
VARIABLES: $$VERSION  Type: Output
                                LOINC Version or Null
Get LOINC Version

```

Example:

```

>W $$VERSION^LEXLR()
2.14

```

```

COMPONENT: COMLIST (LEXCOM, LEXARR)
VARIABLES: LEXCOM  Type: Input
                                Component field (#100)
LEXARR  Type: Input
                                Component List Array (Full Global
                                Reference)

```

```

Note: LEXARR is not initialized (ie
KILLED) on input
      The calling application is
      responsible for
      initializing the array.

```

```

@LEXARR@L  Type: Output

```

Component List Array

@LEXARRAY@(LEXCODE)=Fully Specified Name
field (#80)

Get List by Component

Example:

```
>D COMLST^LEXL("VIRUS IDENTIFIED", "LEXARRAY")
ZW LEXARRAY
LEXARRAY("10736-7")="VIRUS
IDENTIFIED:PRID:PT:CSF:NOM:MICROSCOPY ELECTRON"
LEXARRAY("10737-5")="VIRUS
IDENTIFIED:PRID:PT:STL:NOM:MICROSCOPY ELECTRON"
LEXARRAY("10738-3")="VIRUS
IDENTIFIED:PRID:PT:TISS:NOM:MICROSCOPY ELECTRON"
LEXARRAY("10739-1")="VIRUS
IDENTIFIED:PRID:PT:XXX:NOM:MICROSCOPY ELECTRON"
LEXARRAY("11484-3")="VIRUS
IDENTIFIED:PRID:PT:AMN:NOM:VIRUS CULTURE"
LEXARRAY("12272-1")="VIRUS IDENTIFIED:PRID:PT:XXX:NOM:IF"
LEXARRAY("14451-9")="VIRUS
IDENTIFIED:PRID:PT:EYE:NOM:VIRUS CULTURE"
LEXARRAY("14452-7")="VIRUS
IDENTIFIED:PRID:PT:CVX:NOM:VIRUS CULTURE"
LEXARRAY("14453-5")="VIRUS
IDENTIFIED:PRID:PT:GENV:NOM:VIRUS CULTURE"
LEXARRAY("14454-3")="VIRUS
IDENTIFIED:PRID:PT:NOSE:NOM:VIRUS CULTURE"
LEXARRAY("14455-0")="VIRUS
IDENTIFIED:PRID:PT:PLR:NOM:VIRUS CULTURE"
LEXARRAY("14456-8")="VIRUS
IDENTIFIED:PRID:PT:PRT:NOM:VIRUS CULTURE"
LEXARRAY("14457-6")="VIRUS IDENTIFIED:PRID:PT:UR:NOM:VIRUS
CULTURE"
LEXARRAY("14458-4")="VIRUS
IDENTIFIED:PRID:PT:SPT:NOM:VIRUS CULTURE"
```

COMPONENT: DEPLST (LEXARR)

VARIABLES: LEXARR Type: Input

Deprecated List Array (Full Global
Reference)

Note: LEXARR is not initialized (ie
KILLed) on input
The calling application is
responsible for
initializing the array.

@LEXARR@ (L Type: Output

Deprecated List Array

@LEXARR@(LEXCODE)=Fully Specified Name
Field (#80)

Get Deprecated List

5679 Lexicon Utilities - LEXU (ICD-10 UPDATE)

CUSTODIAL PACKAGE: LEXICON UTILITY
SUBSCRIBING PACKAGE:

USAGE: Supported ENTERED: JUN 3,2011

STATUS: Pending EXPIRES:

DURATION: Till Otherwise Agr VERSION:

DESCRIPTION: TYPE: Routine

This is an addendum to ICR 1573 and contains functions added to LEXU during the implementation of ICD-10 Coding system. The APIs in this ICR become effective on the date of release of patches ICD*18.0*57 and LEX*2.0*80.

ROUTINE: LEXU

COMPONENT: \$\$D10ONE (IEN,DATE)

Returns either a single active ICD-10 Diagnosis code linked to the Lexicon expression or Null if no ICD-10 Diagnosis code is found.

VARIABLES: Input IEN
Internal Entry Number in the Expression file
^LEX(757.01).

VARIABLES: Input DATE
This is a date in Fileman format used to check if a code is active or inactive on a specified date. If not supplied, it will default to TODAY.

VARIABLES: Output \$\$D10ONE
A single active ICD-10 Diagnosis code or Null if no ICD-10 Diagnosis code is found.

COMPONENT: \$\$D10 (IEN,DATE)
Returns either a string of active ICD-10 Diagnosis codes linked to an expression (separated by semicolon, i.e., ICD10;ICD10;ICD10) or Null If no ICD-10 Diagnosis codes are found.

VARIABLES: Input IEN
Internal Entry Number in the Expression file
^LEX(757.01).

VARIABLES: Input DATE
This is a date in Fileman format used to check if a code is active or inactive on a specified date. If not supplied, it will default to TODAY.

VARIABLES: Output \$\$D10
A string of active ICD-10 Diagnosis codes linked to an expression (separated by semicolon, i.e., ICD.10;ICD.10;ICD.10) or Null if no ICD-10 Diagnosis codes are found.

COMPONENT: \$\$P10ONE (IEN,DATE)
Returns either a single active ICD-10 Procedure code linked to the Lexicon expression or Null if no icd-10 Procedure code is found.

VARIABLES: Input IEN
Internal Entry Number in the Expression file
^LEX(757.01).

VARIABLES: Input DATE
This is a date in Fileman format used to check if
a code is active or inactive on a specified date.
If not supplied, it will default to TODAY.

VARIABLES: Output \$\$P10ONE
A single active ICD-10 Procedure code or Null if
no ICD-10 Procedure code is found.

COMPONENT: \$\$ONE(IEN,DATE,SAB)
Returns a single code for a given internal entry number (IEN)
for a specified date and source.

VARIABLES: Input IEN
Internal Entry Number in the Expression file
^LEX(757.01).

VARIABLES: Input DATE
This is a date in Fileman format used to check if
a code is active or inactive on a specified date.
If not supplied, it will default to TODAY.

VARIABLES: Input SAB
Source, this is an internal entry number in file
757.03 or the 3 character source mnemonic (found
on the ASAB cross-reference in file 757.03) or the
SOURCE ABBREVIATION (.01 field in file 757.03)

VARIABLES: Output \$\$ONE
A single code belonging to the specified coding
system by the source abbreviation that is active
on the dated provided and assigned to the
expression indicated by the internal entry number
(IEN).

COMPONENT: \$\$ALL(IEN,DATE,SAB)
Returns all classification codes for a given internal entry
number (IEN) for a specified date and source.

VARIABLES: Input IEN
Internal Entry Number in the Expression file
^LEX(757.01).

VARIABLES: Input DATE
This is a date in Fileman format used to check if
a code is active or inactive on a specified date.
If not supplied, it will default to TODAY.

VARIABLES: Input SAB
Source, this is an internal entry number in file
757.03 or the 3 character source mnemonic (found
on the ASAB cross-reference in file 757.03) or the
SOURCE ABBREVIATION (.01 field in file 757.03)

VARIABLES: Output \$\$ALL
 A string of codes for the source provided (one or more) delineated by a semi-colon or null if no codes are found for the source.

COMPONENT: \$\$IMPDATE(SAB)
 This entry point (extrinsic function) returns the implementation date for a specified source.

VARIABLES: Input SAB
 Source, this is an internal entry number in file 757.03 or the 3 character source mnemonic (found on the ASAB cross-reference in file 757.03) or the SOURCE ABBREVIATION (.01 field in file 757.03)

VARIABLES: Output \$\$IMPDATE
 The date that a coding system was implemented in VistA in Fileman format.

COMPONENT: \$\$CSYS(SYS,VDT)
 This entry point returns information about a coding system on file in the Coding System file #757.03.

VARIABLES: Input SYS
 Coding system identification system and can be in any of the following formats:

A nickname if one exist, i.e. HCPCS, DSM, NANDA, BIRADS

First three characters of source abbreviation from file 757.03, field .01

Source Abbreviation (file 757.03, field .01), i.e., ICD9, CPT4, SNM2

Nomenclature (file 757.03, field 1), i.e., ICD-9-CM, ICD-10-PCS, NANDA

Type (only for ICD), i.e., "DIAG" or "PROC" (requires date)

VARIABLES: Input VDT
 Versioning date in Fileman format used to determine coding system if only the type is known (DIAG or PROC) and to determine the correct SDO version if one exists. If the date is not passed, then TODAY is used.

VARIABLES: Output \$\$CSYS
 A 13 piece caret (^) delimited string

1 IEN

2 SAB (3 character source abbreviation)

3 Source Abbreviation (3-7 char) (#.01)

4 Nomenclature (2-11 char) (#1)

5 Source Title (2-52 char) (#2)
 6 Source (2-50 char) (#3)
 7 Entries (numeric) (#4)
 8 Unique Entries (numeric) (#5)
 9 Inactive Version (1-20 char) (#6)
 10 HL7 Coding System (2-40 char) (#7)
 11 SDO Version Date (date) (757.08 #.01)
 12 SDO Version Id (1-40 char) (757.08 #1)
 13 Implementation Date (date) (#11)

COMPONENT: \$\$HIST(CODE, SYS, .ARY)
 This entry point returns a codes activation history in an array passed by reference.

VARIABLES: Input CODE
 This is a classification code found in the CODES file 757.02 (Required)

VARIABLES: Input SYS
 This is a coding system found in the CODING SYSTEMS file 757.03. It can be in the form of a pointer, a source abbreviation or the name of a coding nomenclature (Required)

VARIABLES: Both .ARY
 This is an array of status effective dates and activation status passed by reference (Required)

ARY(0) = Number of Activation History
 ARY(<date>, <status>) = Comment

Status

0 = Inactive
 1 = Active

Comments include:

Activated
 Inactivated
 Re-activated
 Revised
 Reused

VARIABLES: Output \$\$HIST
 This is the number of activation history entries found

or

-1 ^ error message

COMPONENT: \$\$PERIOD(CODE, SYS, ARY)
 This entry point returns the activation periods (active from and to) of a code in an array passed by reference.

VARIABLES: Input CODE

This is a classification code found in the CODES file 757.02 (Required)

VARIABLES: Input

SYS

This is a coding system found in the CODING SYSTEMS file 757.03. It can be in the form of a pointer, a source abbreviation or the name of a coding nomenclature (Required)

VARIABLES: Both

.ARY

This is an array of activation periods (including active on date and inactive on date when inactive) passed by reference (Required)

ARY(0) 6 piece "^" delimited string

- 1 Number of Activation Periods found
- 2 Coding System (pointer to file

775.03)

- 3 Coding System Abbreviation
- 4 Coding System Nomenclature
- 5 Coding System Full Name
- 6 Coding System Source

or

-1 ^ Message (no period or error)

ARY(Activation Date) = 4 piece "^" delimited string

- 1 Inactivation Date (conditional)
- 2 Pointer to Expression file 757.01 for the code in piece #2 above (required)
- 3 Variable Pointer IEN;Root of a national file (see below) Include when the code exist in an national file (conditional)

CPT Procedure code IEN;ICPT(
ICD Diagnosis code IEN;ICD9(
ICD Procedure code IEN;ICD0(

- 4 Short Description from the SDO file (CPT or ICD)

ARY(Activation Date,0) = Lexicon Expression

VARIABLES: Output

\$\$PERIOD

This is the number of activation periods found:

Same as output variable ARY(0)

or

-1 ^ error message

COMPONENT: \$\$DX(IEN,VDT)

This entry point is to be used as a screen Lexicon searches. It will screen out all terms not linked to either an ICD-9 or ICD-10 code. The code type (ICD-9 or ICD-10) is determined by date. If the date passed in is before the ICD-10 implementation date then it will screen on ICD-9 codes. If the date is on or after the ICD-10 implementation date then it will screen on ICD-10 codes.

Assuming the variable VDT is a valid Fileman format date:

Screen on ICD Diagnosis

```
S DIC("S")="I $$DX^LEXU(+Y,VDT)"
```

VDT is before the ICD-10 implementation date = ICD-9

VDT is on or after the ICD-10 implementation date = ICD-10

If the date is not passed, then TODAY is used.

VARIABLES: Input IEN

This is an internal entry number in the expression file 757.01. When performing Fileman lookups, set it to the variable +Y. (Required)

VARIABLES: Input VDT

This is the versioning date against which the codes found by the search will be compared in order to determine whether the code is active or inactive. Additionally if the date passed is earlier than the ICD-10 implementation date then the screen will only consider ICD-9 codes. If the date is on or after the ICD-10 implementation date then the screen will only consider ICD-10 codes. If the date is not passed, then TODAY's date will be used. (Optional)

VARIABLES: Output \$\$DX

This is a Boolean value:

\$\$DX = 1 (true) if the Lexicon entry is linked to an active ICD code of the type specified by the input parameter TYPE.

\$\$DX = 0 (false) if the Lexicon entry is not linked to an active ICD code of the type specified by the input parameter TYPE.

COMPONENT: \$\$CSDATA(CODE,CSYS,VDT,.ARY)

This entry point returns information about a code from a specified coding system. It is intended to be similar to ICDDATA^ICDXCODE except it is not limited to ICD coding systems.

VARIABLES: Input CODE
This is a code found in file 757.02 (CODES file).

VARIABLES: Input CSYS
This is a pointer to the CODING SYSTEMS file 757.03 that identifies the coding system that CODE belongs to. It is important to specify the coding system because some codes overlap various coding systems.

VARIABLES: Input VDT
This is the date that will be used to determine the status of the code in the CODE input parameter. The status will either be Inactive or Active.

VARIABLES: Both .ARY
This is the name of a local array passed by reference that will contain the output.

ARY()

Lexicon Data

ARY("LEX",1)	IEN ^ Preferred Term
ARY("LEX",2)	Status ^ Effective Date
ARY("LEX",3)	IEN ^ Major Concept Term
ARY("LEX",4)	IEN ^ Fully Specified Name
ARY("LEX",5)	Hierarchy (if it exists)
ARY("LEX",6,0)	Synonyms/Other Forms
ARY("LEX",6,1)	Synonym #1
ARY("LEX",6,n)	#n
ARY("LEX",7,0)	Semantic Map
ARY("LEX",7,1,1)	Class ^ Type (internal)
ARY("LEX",7,1,2)	Class ^ Type (external)
ARY("LEX",7,1,n)	#n
ARY("LEX",7,1,n)	#n
ARY("LEX",8)	Deactivated Concept Flag

Coding System Data

ARY("SYS",1)	IEN
ARY("SYS",2)	Short Name
ARY("SYS",3)	Age High
ARY("SYS",4)	Age Low
ARY("SYS",5)	Sex
ARY("SYS",6,0)	MDC/DRG Pairing
ARY("SYS",6,1,1)	MDC
ARY("SYS",6,1,2)	DRGs
ARY("SYS",6,n,1)	#n
ARY("SYS",6,n,2)	#n
ARY("SYS",7)	Complication/Comorbidity
ARY("SYS",8)	MDC13
ARY("SYS",9)	MDC24
ARY("SYS",10)	MDC24
ARY("SYS",11)	Unacceptable as Principal Dx

```

ARY("SYS",12)      Major O.R. Procedure
ARY("SYS",13)      Procedure Category
ARY("SYS",14,0)    Description
ARY("SYS",14,1)    Text 1
ARY("SYS",14,n)    #n

```

Each data element will be in the following format:

```

ARY(ID,SUB) = DATA
ARY(ID,SUB,"N") = NAME

```

Where

ID Identifier, may be:

```

"LEX" for Lexicon data
"SYS" for Coding System data

```

SUB Numeric Subscript

DATA This may be:

- A value if it applies and is found
- Null if it applies but not found
- N/A if it does not apply

NAME This is the common name given to the data element

Example:

```
S X=$$CSDATA^LEXU("C18.6",30,3141010,.ARY)
```

X=1

```

ARY("LEX",1)="267081^Malignant neoplasm of
descending colon" ARY("LEX",1,"N")="IEN ^
Preferred Term" ARY("LEX",2)="1^3131001"
ARY("LEX",2,"N")="Status ^ Effective Date"
ARY("LEX",3)="267081^Malignant neoplasm of
descending colon" ARY("LEX",3,"N")="IEN ^ Major
Concept Term" ARY("LEX",4)=""
ARY("LEX",4,"N")="IEN ^ Fully Specified Name"
ARY("LEX",5)="" ARY("LEX",5,"N")="Hierarchy (if
exists)" ARY("LEX",6,0)=0
ARY("LEX",6,0,"N")="Synonyms and Other Forms"
ARY("LEX",7,0)=1 ARY("LEX",7,0,"N")="Semantic Map"
ARY("LEX",7,1,1)="6^47"
ARY("LEX",7,1,1,"N")="Semantic Class ^ Semantic
Type (internal)"
ARY("LEX",7,1,2)="Diseases/Pathologic
Processes^Disease or Syndrome"
ARY("LEX",7,1,2,"N")="Semantic Class ^ Semantic
Type (external)" ARY("LEX",8)=""
ARY("LEX",8,"N")="Deactivated Concept Flag"

```

```

ARY("SYS",1)=501148 ARY("SYS",1,"N")="IEN"
ARY("SYS",2)="Malignant neoplasm of descending
colon" ARY("SYS",2,"N")="Short Name"
ARY("SYS",3)="" ARY("SYS",3,"N")="Age High"
ARY("SYS",4)="" ARY("SYS",4,"N")="Age Low"
ARY("SYS",5)="" ARY("SYS",5,"N")="Sex"
ARY("SYS",6,0)=0 ARY("SYS",6,0,"N")="MDC/DRG"
ARY("SYS",7)=""
ARY("SYS",7,"N")="Complication/Comorbidity"
ARY("SYS",8)="" ARY("SYS",8,"N")="MDC13"
ARY("SYS",9)="" ARY("SYS",9,"N")="MDC24"
ARY("SYS",10)="" ARY("SYS",10,"N")="MDC24"
ARY("SYS",11)="" ARY("SYS",11,"N")="Unacceptable
as Principal Dx" ARY("SYS",12)="N/A"
ARY("SYS",13)="N/A" ARY("SYS",14,0)=1
ARY("SYS",14,0,"N")="Description"
ARY("SYS",14,1)="MALIGNANT NEOPLASM OF DESCENDING
COLON"

```

VARIABLES: Output \$\$CSDATA
This is a boolean value:

1 if the API is successful (fully or partial)
0 if the API is unsuccessful

or

-1 ^ Error Message

COMPONENT: \$\$FREQ(TEXT)
This API checks the frequency of use of keywords contained in
a text string in the Lexicon.

VARIABLES: Input TEXT
This is a text string intended as the input for a
Lexicon search.

VARIABLES: Output \$\$FREQ
This is the maximum number of records that must be
inspected during a Lexicon search to find matching
entries for the input search text.

If this number is too high, applications can
prompt the user to either continue with the search
or to further refine the search.

COMPONENT: \$\$MAX(SYS)
This API returns the SEARCH THRESHOLD field #12 of the CODING
SYSTEMS file #757.03.

VARIABLES: Input SYS
This is a pointer to the CODING SYSTEM file
#757.03.

VARIABLES: Output \$\$MAX
This is the value stored in the SEARCH THRESHOLD
field #12 of the CODING SYSTEMS file 757.03. This

value, along with the value of `$$FREQ^LEXU`, can be used to evaluate if a search should continue or be further refined.

`$$FREQ` The maximum number or records to inspect during a search based on the input text string.

`$$MAX` The maximum number of records to consider for a coding system before refining the search.

COMPONENT: `$$CAT(CODE)`

This API returns the category (i.e., header) of an ICD Diagnosis code.

VARIABLES: Input `CODE`

This is a valid ICD Diagnosis code.

VARIABLES: Output `$$CAT`

This is the category (or header) to which the ICD Diagnosis code belongs.

COMPONENT: `$$ISCAT(CODE)`

This API determines if an ICD-10 string is an ICD category.

VARIABLES: Input `CODE`

This is a string used to determine if it is an ICD-10 code or a category.

VARIABLES: Output `$$CODE`

This is a 4 piece "^" delimited string contains the following:

- 1 Category flag
1 `CODE` is a Category
0 `CODE` is not a Category
- 2 Number of Sub-Categories belonging to the Category
- 3 Number of Codes belonging to the Category
- 4 Parent Category
Parent Category
Null if no Parent Category

COMPONENT: `$$PFI(FRAG,CDT,.ARY)`

This API returns a local array containing information about an ICD-10 procedure code fragment.

VARIABLES: Input `FRAG`

This is a string representing a fragment of an ICD-10 procedure code. An ICD-10 code is 7 characters long and a code fragment is a portion of the code starting at character position #1 and

not to exceed 6 characters in length.

VARIABLES: Input CDT
This is the versioning date used to select an entry that was appropriate on the date passed. If no date is passed, TODAY is used. Business rules apply, if the date passed is before the implementation date for ICD-10, then the implementation date is used.

VARIABLES: Both .ARY
This is a local array passed by reference that will contain information about a code fragment.

ARY(0) 5 piece "^" delimited string
1 Unique Id
2 Code Fragment
3 Date Entered
4 Source
5 Details

ARY(1) 4 piece "^" delimited string
1 Effective Date
2 Status
3 Effective Date External
4 Status External

ARY(2) Name/Title
ARY(3) Description
ARY(4) Explanation
ARY(5,0) # of synonyms included
ARY(5,n) Included synonyms

VARIABLES: Output \$\$PFI
This is a success flag

1 on success
-1 ^ error message on error

COMPONENT: \$\$NXSAB(SAB,REV)
This API returns the next Source Abbreviation found in the CODING SYSTEMS file 757.03 using the ASAB cross-reference. It is the equivalent of \$O(^LEX(757.03,"ASAB",SAB)).

VARIABLES: Input SAB
This is either a Source Abbreviation (SAB) from the .01 field of file 757.03 or null value to find the first SAB.

VARIABLES: Input REV
This is a reverse flag (optional). If set to 1 the API will find the next Source Abbreviation in the reverse order (aka, previous SAB)

VARIABLES: Output \$\$NXSAB
This is either the next Source Abbreviation (SAB) previous SAB (when reverse flag set to 1) or null

if the input parameter SAB has no next SAB.

COMPONENT: \$\$RECENT(SAB)

This API returns a boolean valued flag to indicate if the coding system identified by the source abbreviation has been recently updated with in a 90 day window (looking forward by 30 days and to the past by 60 days). This is to evaluate if a coding system was updated by a quarterly patch, and may be used to trigger an code set update protocol.

VARIABLES: Input SAB

This is either a 3 character source abbreviation taken from the .01 field of the CODING SYSTEM file 757.03 or a pointer to the CODING SYSTEM file 757.03.

VARIABLES: Output \$\$RECENT

This is a Boolean valued flag.

1 indicates the Coding System has been recently updated by a quarterly update by looking 30 days into the future and 60 days for a change made to the coding system.

0 indicate the Coding System has NOT been recently updated by a quarterly update.

COMPONENT: \$\$RUPD(SAB)

This API returns a date the coding system identified by the source abbreviation has been updated based on a recent date (TODAY+3). This is to evaluate if a coding system was updated by a quarterly patch, and may be used to trigger an code set update protocol.

VARIABLES: Input SAB

This is either a 3 character source abbreviation taken from the .01 field of the CODING SYSTEM file 757.03 or a pointer to the CODING SYSTEM file 757.03.

VARIABLES: Output \$\$RUPD

This is date found for the last update to a coding system based on a recent date (TODAY+30)

COMPONENT: \$\$LUPD(SAB,DATE)

This API returns the last date the coding system identified by the source abbreviation has been updated based on the date supplied (optional). If no date is supplied, the last date will be returned.

VARIABLES: Input SAB

This is either a 3 character source abbreviation taken from the .01 field of the CODING SYSTEM file 757.03 or a pointer to the CODING SYSTEM file 757.03.

VARIABLES: Input DATE
This is a date to use to retrieve the last update for a coding system (optional)

VARIABLES: Output \$\$LUPD
This is date found for the last update to a coding system based on a recent passed or the last date updated if a date is not passed.

COMPONENT: \$\$PAR(TXT,.ARY)
This API takes a string of text and parses the string into words using the parsing logic used by the Lexicon search engine.

VARIABLES: Input TXT
This is a text string intended as the input for a Lexicon search and will be parsed into words and placed in a local array (Required)

VARIABLES: Input .ARY
Local array, passed by reference

VARIABLES: Output \$\$PAR
This is the number of words parsed from the text

VARIABLES: Output ARY
This is a local array containing the words parsed from the input text. The words are arranged in the order they are found in the text; in alphabetical order; and in the order they are used in the Lexicon search (frequency order)

Total words found

ARY(0)=#

Words listed in the order they appear in the input variable TXT

ARY(1)=WORD1
ARY(n)=WORDn

Words listed alphabetically with the frequency of occurrence in the Lexicon

ARY("B",WORDA)=# (Frequency of Use)
ARY("B",WORDB)=#

Words listed in the frequency order. This is the order the words will be used by the Lexicon search engine, starting with the least frequently used word and ending with the most frequently used word.

ARY("L",1)=SEARCHWORD1
ARY("L",n)=SEARCHWORDn

COMPONENT: \$\$\$SCT(IEN,DATE)
This entry point is a screen used during searches to return terms with SNOMED CT codes that are not classified as Veterinary.

VARIABLES: Input IEN
Internal Entry Number in the Expression file ^LEX(757.01).

VARIABLES: Input DATE
This is a date in Fileman format used to check if a code is active or inactive on a specified date. If not supplied, it will default to TODAY.

VARIABLES: Output \$\$\$SCT
This is a Boolean value:

\$\$\$SCT = 1 (true) if the Lexicon entry is

Linked to an active SNOMED CT code
and the term is not a Veterinary term

\$\$\$SCT = 0 (false) if the Lexicon entry is

Not linked to an SNOMED CT code
or the SNOMED CT code is inactive
or the term is a Veterinary term

Excludes terms semantically typed as a Veterinary term.

5680 Lexicon Expression - LEXCODE

CUSTODIAL PACKAGE: LEXICON UTILITY
SUBSCRIBING PACKAGE:

USAGE: Supported ENTERED: JUN 3,2011
STATUS: Pending EXPIRES:
DURATION: Till Otherwise Agr VERSION:
FILE: ROOT:
DESCRIPTION: TYPE: Routine

This is an addendum to ICR 1614 and contains functions added to LEXCODE during the implementation of ICD-10 Coding system.

ROUTINE: LEXCODE
COMPONENT: \$\$EXP(<CODE>,<SAB>,<DATE>)
VARIABLES: CODE Type: Input Code taken from the Codes file 757.02.
(Required)

SAB Type: Input Source, this is an internal entry number in file 757.03 or the 3 character source mnemonic (found on the ASAB cross-reference in file 757.03) or the

SOURCE ABBREVIATION (.01 field in file 757.03) It is used to distinguish between different coding systems with the same code (i.e., the code 300.01 occurs in both the ICD-9 CM and DSM IV coding systems). (Required)

DATE Type: Input

This is a date in Fileman format used to check if a code is active or inactive on a specified date. If not supplied, it will default to TODAY.

\$\$EXP Type: Output

This is a 2 Piece "^" delimited string containing

Either:

- 1 Pointer to Expression file #757.01
- 2 Display Text (Expression)

or:

- 1 -1
- 2 Error Message

This entry point allows an application to retrieve an active preferred term for a coding system on the date provided.

5681 Lexicon ICD-10 APIs - LEX10CS

CUSTODIAL PACKAGE: LEXICON UTILITY

SUBSCRIBING PACKAGE:

USAGE: Supported

ENTERED: JUN 6,2011

STATUS: Pending

EXPIRES:

DURATION: Till Otherwise Agr VERSION:

DESCRIPTION:

TYPE: Routine

Supported APIs for the implementation of ICD-10. The APIs in this ICR become effective on the date of release of patches ICD*18.0*57 and LEX*2.0*80.

ROUTINE: LEX10CS

COMPONENT: \$\$ICDSRCH(TEXT, .ARRAY, DATE, LEN, FILTER)

This entry point searches for an ICD code and returns active ICD codes found up to the number defined by the input parameter Length. If the date is before the ICD-10 implementation date the search will be conducted for ICD-9 codes. If the date passed is on or after the ICD-10 implementation date the search will be conducted for ICD-10 codes.

VARIABLES: Input

TEXT

Text or Code to search for. (Required)

VARIABLES: Input .ARRAY
This is a local output array passed by reference.
(Required)

VARIABLES: Input DATE
The date against which the codes found by the search will be compared in order to determine whether the code is active or inactive. If not passed, TODAY's date will be used. (Optional, but when used must be in FileMan format)

VARIABLES: Input LEN
This specifies the length of the list of codes. Default value is 30. (Optional)

VARIABLES: Input FILTER
This is a filter to apply to the search to screen out unwanted entries. It is MUMPS code in the form of a valid IF statement. It is the same as Fileman's DIC("S"). (Optional)

VARIABLES: Output \$\$ICDSRCH
2 Piece "^" delimited string the success/error conditions

A Positive number for successful search not exceeding the Length of the list.

A Negative number for an unsuccessful search or a search condition

-1^No codes found

No codes found and local array not returned

-2^Too many items found, please refine search

The list exceeds the number indicated by

LEN,
however, the first LEN of the Array is returned and the list is marked as a pruned list

VARIABLES: Output ARRAY
Output Array passed by reference containing the codes found

ARRAY(0)=# found ^ Pruning Indicator
ARRAY(1)=CODE ^status effective date
ARRAY(1,"IDL ")=ICD Dx long description (if code)

ARRAY(1,"IDL ,1)=ICD Dx IEN ^ effective date
ARRAY(1,"IDS ")=ICD Dx short description (if code)

ARRAY(1,"IDS ,1)=ICD Dx IEN ^ effective date
ARRAY(1,"LEX ")=Lexicon expression

```

ARRAY(1,"LEX",1)=Lexicon IEN ^ effective date
ARRAY(1,"SYN",1)=synonym #1
ARRAY(1,"SYN",m)=Synonym #m

```

Pruning Indicator: If the second piece of ARY(0) is greater than 0, then the list has been pruned, limiting the list to the length specified by the input parameter <Len>.

COMPONENT: \$\$DIAGSRCH(TEXT, .ARRAY, DATE, LEN, FILTER)
This entry point searches for an ICD code and returns active ICD-10 codes found up to the number defined by the input parameter Length. This search is similar to \$\$ICDSRCH^LEX10CS except it only searches ICD-10 codes.

VARIABLES: Input TEXT
Text or Code to search for. (Required)

VARIABLES: Input .ARRAY
This is a local output array passed by reference. (Required)

VARIABLES: Input DATE
The date against which the codes found by the search will be compared in order to determine whether the code is active or inactive. (Optional, but when used must be in FileMan format)

VARIABLES: Input LEN
This specifies the length of the list of codes. Default value is 30. (Optional)

VARIABLES: Input FILTER
This is a filter to apply to the search to screen out unwanted entries. It is MUMPS code in the form of a valid IF statement. It is the same as Fileman's DIC("S"). (Optional)

VARIABLES: Output \$\$DIAGSRCH
2 Piece "^" delimited string the success/error conditions

A Positive number for successful search not exceeding the Length of the list.

A Negative number for an unsuccessful search or a search condition

-1^No codes found

No codes found and local array not returned

-2^Too many items found, please refine search

The list exceeds the number indicated by

LEN,

however, the first LEN of the Array is

list returned and the list is marked as a pruned

VARIABLES: Output ARRAY
Output Array passed by reference containing the ICD-10 codes found

ARRAY(0)=# found ^ Pruning Indicator
ARRAY(1)=CODE or Category ^ status effective

date (code only)
ARRAY(1,"CAT")=Category Name
ARRAY(1,"IDL ")=ICD Dx long description (if
code)
ARRAY(1,"IDL ,1)=ICD Dx IEN ^ effective date
ARRAY(1,"IDS)=ICD Dx short description (if
code)
ARRAY(1,"IDS ,1)=ICD Dx IEN ^ effective date
ARRAY(1,"LEX)=Lexicon expression
ARRAY(1,"LEX ,1)=Lexicon IEN ^ effective date
ARRAY(1,"SYN ,1)=synonym #1
ARRAY(1,"SYN",m)=Synonym #m
ARRAY(1,"MENU")=Menu Text
ARRAY(1,"MSG")=Message (unversioned only)

Notes:

Pruning Indicator: If the second piece of ARRAY(0) is greater than 0, then the list has been pruned, limiting the list to the length specified by the input parameter

LEN.

If there is a message, it can be either:

Inactive mm/dd/yyyy
Pending mm/dd/yyyy

COMPONENT: \$\$PCSDIG (FRAG,DATE)
This entry point takes an ICD-10-PCS code, full or a partial (code fragment), and returns a list of all possibilities for the next character, with any definitions and examples available. If a full code is passed (7 characters) it will return the code's long description, and status.

VARIABLES: Input FRAG
This is an ICD-10-PCS Code (7 characters) or a fragment of an ICD-10-PCS Code (less than 7 characters) (Required)

VARIABLES: Input DATE
The date against which the codes found by the search will be compared in order to determine whether the code is active or inactive. (Optional, defaults to TODAY)

VARIABLES: Output LEXPCDAT
This is both a variable and an array. If the

length of the FRAG is less than 7, then the array will contain the next level of choices and associated data. If the length of the FRAG is equal to 7, then a fully specified code has been passed and the array will contain the long description, status and effective date of the code.

If the string FRAG is a valid code fragment or null, the return value Of LEXPCDAT will be 1 and the following array will be returned.

description LEXPCDAT("NEXLEV ,char1, DESC)=char1
description LEXPCDAT("NEXLEV ,char2, DESC)=char2
description LEXPCDAT("NEXLEV ,charn, DESC)=charn

If the string FRAG is a valid code the return value Of LEXPCDAT will be 1 and the following array will be returned.

LEXPCDAT("PCSDESC)=long description for code
LEXPCDAT("STATUS)=status_ ^ _effective date

If the string FRAG is a not valid code fragment or null and it is not a valid code, the return value Of LEXPCDAT will be 0 and no array will be returned.

COMPONENT: \$\$CODELIST(SYS, SPEC, SUB, DATE, LEN, FMT)
This entry point creates a list of active codes based on an input code mask and date and places the list in a temporary global array with a subscript specified by the calling application.

VARIABLES: Input SYS
Coding system the Coding Systems file 757.03. This can be a pointer, the .01 field or the abbreviated 3 character mnemonic (found on the ASAB cross-reference (Required)

VARIABLES: Input SPEC
This is a code from the coding system or a code mask. Any character position can be occupied by a question mark "?" to allow any value in that character position. The trailing character may be an asterisk indicating any characters that follow are allowable. The following are all valid; C71.0, C71.*, C7?.0 or 02V?0* (Required)

VARIABLES: Input SUB
This is a string, preferably in the calling applications namespace, that will be used as a subscript in a temporary global array (optional, if not passed CODELIST will be used as a

subscript).

^TMP (SUB, \$J)

VARIABLES: Input DATE
The date against which the codes found by the search will be compared in order to determine whether the code is active or inactive. (Optional, but when used, must be in FileMan format)

VARIABLES: Input LEN
This specifies the length of the list of codes. Default value is 30. (Optional)

VARIABLES: Input FMT
List Format (Optional)

0 - (zero) returns a brief listing in the global array (codes only) (DEFAULT)

1 - returns a detailed listing in the global array, includes the code, a variable pointer the code in a code set file (i.e., ICD-9, CPT, etc), the code's effective date, the expression and the expression IEN from file #757.01.

VARIABLES: Output \$\$CODELIST
2 Piece "^" delimited string containing

Either:

Piece	Meaning
1	Positive value for success
2	Number of Codes Found

or:

Piece	Meaning
1	Negative number for error or condition
2	Error Message or Condition

Example errors/conditions

- 1 Coding system not specified (First parameter is missing)
- 2 Invalid coding system/source abbreviation (First parameter not valid)
- 3 No search specification (Second parameter missing)
- 4 Insufficient search specification" (Second parameter too short)
- 5 Invalid search specification (Second parameter invalid)
- 6 Number of matches exceeds specified limit (More matches found, only the

number specified will be returned)

VARIABLES: Output

TMP(SUB,\$J,

This is a global array subscripted as specified by the calling application, input parameter SUB. It contains a list of codes found in either a brief or detailed output.

Brief output array (FMT = 0)

^TMP(SUB,\$J,0)=Total n
^TMP(SUB,\$J,1)=Code 1
^TMP(SUB,\$J,2)=Code 2
^TMP(SUB,\$J,n)=Code n

Detailed output array (FMT = 1)

^TMP(SUB,\$J,0)=Total n
^TMP(SUB,\$J,1)=Code 1
^TMP(SUB,\$J,1,1)=Variable Pointer 1 ^
Code 1 ^ date
^TMP(SUB,\$J,1,2)=Term 1 IEN ^ Term 1
^TMP(SUB,\$J,1,"MSG")=Message (unversioned only)
^TMP(SUB,\$J,2)=Code 2
^TMP(SUB,\$J,2,1)=Variable Pointer 2 ^
Code 2 ^ date
^TMP(SUB,\$J,2,2)=Term 2 IEN ^ Term 2
^TMP(SUB,\$J,2,"MSG")=Message (unversioned only)
^TMP(SUB,\$J,n)=Code n
^TMP(SUB,\$J,n,1)=Variable Pointer n ^
Code n ^ date
^TMP(SUB,\$J,n,2)=Term n IEN ^ Term n
^TMP(SUB,\$J,n,"MSG")=Message (unversioned only)

Notes:

If the code is found in one of the Vista Code Set files controlled by a Standards Development Organization (SDO) then a variable pointer will be provided for that code in that file. Example of SDO controlled files include:

ICD DIAGNOSIS file #80
ICD OPERATION/PROCEDURE file #80.1
CPT file #81
DSM file #627.7

If there is a message, it can be either:

Inactive mm/dd/yyyy
Pending mm/dd/yyyy

COMPONENT: \$\$TAX(TEXT, SRC, CDT, SUB, VER)

This API returns codes that qualify for building a taxonomy.

Originally designed for ICD-10, but modified to include any coding system (DSM, ICD, SNOMED CT, CPT, etc.)

- VARIABLES: Input TEXT
This is the text or code to search for.
- VARIABLES: Input SRC
This is a string of coding systems delimited by an "^" up arrow to limit the search to the desired coding systems. The string can consist of pointers to the CODING SYSTEM file 757.03 or source abbreviations.
- Using source abbreviations
"ICD^ICP^10D^10P"
- Using source pointers to file 757.03
"1^2^30^31"
- VARIABLES: Input CDT
This a date used processing versioned data. Also, when a versioned list is required (input parameter VER=1) it is used to suppress inactive codes from the list.
- VARIABLES: Input SUB
This is the name of a subscript to use in the ^TMP global (optional). This allows for applications to put the data in their own namespace. It also allows for multiple search results to exist.
- ^TMP (LEXSUB, \$J,
^TMP ("LEXTAX", \$J, Default
- VARIABLES: Input VER
This is a boolean flag that indicates if the search is to be versioned. If set to 1, only active codes will be returned based on the date in the CDT input parameter. If no date, then TODAY is used.
- VER = 0 Return active and inactive codes
VER = 1 Version, return active codes only
- VARIABLES: Output \$\$TAX
This the number of codes found by the search or -1 ^ with an error message.
- VARIABLES: Output TMP (SUB, \$J)
This is the results of the search saved in the ^TMP global with the specified subscript:
- ^TMP (SUB, \$J, SRC, (CODE_ " "), #)
5 piece "^" delimited string
1 Activation Date (can be a future date)

- 2 Inactivation Date (can be a future date)
- 3 Lexicon IEN to Expression File 757.01
- 4 Variable Pointer to a National file
- 5 Short Name from a National file

^TMP(SUB,\$J, SRC, (CODE_ " "), #, 0)

2 piece "^" delimited string

- 1 Code (no spaces)
- 2 Lexicon Expression

Example:

Search for "DIFFICULTY IN WALKING" For sources "ICD^10D" (ICD-9/10 Diagnosis)

```

^TMP("TAX", $J, 0) = 3 ^TMP("TAX", $J, 1, "719.7 ", 1) =
    2781001^2791001^329945^4611; ICD9(^
    DIFFICULTY IN WALKING
^TMP("TAX", $J, 1, "719.7 ", 1, 0) =
    719.7^Difficulty in Walking
^TMP("TAX", $J, 1, "719.7 ", 2) =
    3031001^3131001^329945^4611; ICD9(^
    DIFFICULTY IN WALKING
^TMP("TAX", $J, 1, "719.7 ", 2, 0) =
    719.7^Difficulty in Walking
^TMP("TAX", $J, 1, "781.2 ", 1) =
    2781001^3131001^48820^5419; ICD9(^
    ABNORMALITY OF GAIT
^TMP("TAX", $J, 1, "781.2 ", 1, 0) =
    781.2^Abnormality of Gait
^TMP("TAX", $J, 30, "R26.2 ", 1) =
    3131001^^5019306^521502; ICD9(^
    Difficulty in walking, not
    elsewhere classified
^TMP("TAX", $J, 30, "R26.2 ", 1, 0) =
    R26.2^Difficulty in Walking, not
    elsewhere classified

```

5781 Mixed Case - LEXXM

5781 NAME: Mixed Case
 CUSTODIAL PACKAGE: LEXICON UTILITY
 SUBSCRIBING PACKAGE: DRG GROUPER

This API is used by the special lookup ICDEXLK* to display entries to the user for selection.

USAGE: Controlled Subscri ENTERED: MAR 8, 2012
 STATUS: Pending EXPIRES:
 DURATION: Till Otherwise Agr VERSION:
 DESCRIPTION: TYPE: Routine

ROUTINE: LEXXM
 COMPONENT: \$MIX(TEXT)
 This API converts text from any form to a modified mix text

format. Example:

Input:

arthropathy in behcet's syndrome involving
other specified sites

Traditional Mixed Case (FileMan):

Arthropathy In Behcet's Syndrome Involving
Other Specified Sites

Lexicon Mixed Case (\$\$MIX^LEXXM):

Arthropathy in Behcet's Syndrome involving
other specified sites

VARIABLES: Input TEXT
This is a text string to be converted to mix text.

VARIABLES: Output \$\$MIX
This is a string of text in mixed case.

5840 Lexicon ICD-10 Suggestions - LEX10CX

5840 NAME: ICD-10 Suggestions
CUSTODIAL PACKAGE: LEXICON UTILITY
SUBSCRIBING PACKAGE: ORDER ENTRY/RESULTS REPORTING
AUTOMATED INFO COLLECTION SYS
PROBLEM LIST
USAGE: Controlled Subscri ENTERED: SEP 6,2012
STATUS: Pending EXPIRES:
DURATION: Till Otherwise Agr VERSION:
DESCRIPTION: TYPE: Routine

ROUTINE: LEX10CX
COMPONENT: EN

This entry point is an interactive lookup where the input coding system and code are not known. There is no input variables for this API, however, the variable LEXSAB can be preset to a coding system (.01 field in file 757.03), else wise the user will be prompted for a coding system. This API will display a selection list of suggested ICD-10 codes that have a similar textual content as the user selected code and coding system. If no suggestions are available or the user does not make a selection, then the user will be prompted for a ICD-10 code. There are two output variables, X and Y.

Example Output:

ICD-9 to ICD-10

X="119899^Tobacco Use Disorder^305.1^ICD-9-CM"
Y="5003360^Nicotine Dependence, unspecified,
Uncomplicated^F17.200^ICD-10-CM"

SNOMED CT to ICD-10

```
X="7078519^Diabetes mellitus type 2^44054006^SNOMED CT"  
Y="5002666^Type 2 Diabetes Mellitus without  
Complications^E11.9^ICD-10-CM"
```

VARIABLES: Output X
X is a 4 piece "^" delimited string representing the source code.

- 1 Lexicon IEN for file 757.02
- 2 Expression
- 3 Code in selected Coding System
- 4 Coding System nomenclature

or null if search fails

VARIABLES: Output Y
Y is a 4 piece "^" delimited string representing the target ICD-10 code.

- 1 Lexicon IEN for file 757.02
- 2 Expression
- 3 ICD-10 Diagnostic Code
- 4 ICD-10-CM

or -1 if search fails

VARIABLES: EN2
COMPONENT: EN2 (CODE, SAB)

This entry point is an interactive lookup where the input coding system and code are known and supplied as input parameters CODE and SAB. This API will display a selection list of suggested ICD-10 codes that have a similar textual content as the specified code (CODE) and coding system (SAB). If no suggestions are available or the user does not make a selection, then the user will be prompted for a ICD-10 code. The output for EN2 is the same as entry point EN.

VARIABLES: Input CODE
This is a code from the specified coding system.

VARIABLES: Input SAB
This is the coding system abbreviation (a three character representation of the coding system taken from the .01 field of the CODING SYSTEMS file 757.03)

VARIABLES: Output X
X is a 4 piece "^" delimited string representing the source code.

- 1 Lexicon IEN for file 757.02
- 2 Expression
- 3 Code in selected Coding System
- 4 Coding System nomenclature

or null if search fails

VARIABLES: Output Y
Y is a 4 piece "^" delimited string representing the target ICD-10 code.

- 1 Lexicon IEN for file 757.02
- 2 Expression
- 3 ICD-10 Diagnostic Code
- 4 ICD-10-CM

or -1 if search fails

COMPONENT: EN3(CODE, SAB, .ARY, MAX)
This entry point is a silent lookup for suggested ICD-10 codes for a code in another coding system. The code (CODE) and coding system abbreviation (SAB) are passed as input parameters.

This API will create an array of suggested ICD-10 codes that have a similar textual content as the specified code (CODE) and coding system (SAB).

VARIABLES: Input CODE
This is a code in the coding system identified by the input parameter SAB.

VARIABLES: Input SAB
This is the coding system abbreviation (a three character representation of the coding system taken from the .01 field of the CODING SYSTEMS file 757.03)

VARIABLES: Both ARY
This is a local array, passed by reference:

ARY("X")	Input	
ARY("Y",0)	Output	Number of Suggestions
ARY("Y",1)	Output	First Suggestion
ARY("Y",n)	Output	nth Suggestion

ARY("E") Error Message

Both ARY("X") and ARY("Y",#) are 4 piece "^" delimited strings:

- 1 Internal Entry Number (IEN) file 757.01
- 2 Expression (file 757.01, field .01)
- 3 Code (file 757.02, field 1)
- 4 Nomenclature (file 757.03, field 1)
i.e., SNOMED CT, ICD-9-CM or ICD-10-CM

VARIABLES: Input MAX
This is the maximum number of suggestions to return in the array (optional, default 100)

KEYWORDS: ICD-10

ICD10
LEX10CS