Problem List Refactoring

Release 1.05

PricewaterhouseCoopers, LLC (PwC) on behalf of the
Department of Veterans Affairs (VA)

May 20, 2013

Abstract

As part of the Open Source EHR Refactoring project funded by the VA, we have worked on refactoring the Problem List code. This document is to describe what we have done in term of code changes for all interested stakeholders.

Contents

1. Introduction

The goal of our code changes is to make the VistA code more modular and readable without changing functionality. The Problem List package has been chosen because it has dependencies to a moderate number of other packages. This criterion allows our initial refactored code to be manageable, but not trivial, so that lessons learned here and tools developed can be used for future refactoring of other packages.

The Problem List refactoring is concentrated on providing a complete Problem List API that is used both by CPRS RPC tags and the scroll & roll interface. In fact, RPC tags that provide information for CPRS form a suitable API, however:

1. They have minimal error checking for input parameters.
2. Although they share some code with the scroll & roll interface, the commonalities are not clearly identified in the code; typically they have non-descriptive assumed variables.
3. There are a number of places where code is copied and pasted from the scroll & roll interface code and minimal changes are done; this resulted in the duplication of business logic/database access code.
4. There are functionalities in the scroll & roll interface that is not covered by this API.
5. Globals are accessed directly.
6. This API is actually in the Order/Entry Result Reporting Package (Routines ORQQPL*).

Scroll & roll interface code also has a good structure that is mostly based on List Manager and Protocol actions, however:

1. Business logic and database access is mixed with user interface elements such as write statements, user input, and List Manager Update calls.
2. There are a number of places where there is duplicated business logic/database access code again due to copying and pasting.
3. Globals are accessed directly.
4. Text messages are hardcoded in the code.
5. There are direct M Read and Write commands to and from console.

Our main goals for this phase of refactoring were:

1. To define a complete Problem List API that can be used by both scroll & roll interface and RPC tags and accessible by other packages and applications.
2. Only allow business logic/database access for scroll & roll interface to be through this API; this excludes Fileman supplied user interface that directly updates File items.
3. Remove direct global access and replace them with Fileman DBS calls.
4. Minimize duplicated code.
5. Make the code readable and document where necessary.
6. Remove all M Read and Write commands and replace them with calls to appropriate FileMan API entry points (^DIR and EN^DDIOL).
7. Move all hardcoded text messages to the .84 (^DIALOG) file so they could be easily changed or localized.

2. Code Walkthrough

Several APIs were developed during the refactoring effort. They are described below.
The API's here use either ICD9 code or Expressions File (757.01) record to identify a problem. Lexicon Utility package already provides API methods to find either by entering a problem description. Please see Technical Manual Developer's guide on http://www.va.gov/vdl/application.asp?appid=76 and in particular API methods LOOK^LEXA and ICD^LEXU.

Unless otherwise noted all date formats are internal FileMan format (e.g. 3120121 = Jan 21, 2012).   Various utility methods to handle internal date format, including conversions from/to human readable forms, are available in http://www.va.gov/vdl/application.asp?appid=10 Kernel Developer's Guide and http://www.va.gov/vdl/application.asp?appid=5 Programmer Manual.

2.1.   Problem API

$$ACTIVE^GMPLAPI2() – Is problem active
This extrinsic function verifies if a problem is active or not.

Format
        $$ACTIVE^GMPLAPI2(.RETURN,GMPIFN)

Input Parameters
.RETURN   [Required,Boolean] Set to 1 if the problem is active, 0 otherwise

GMPIFN   [Required,Numeric] Problem IEN (pointer to file 9000011)

Output
A Boolean value signaling if the call was successful or not.

Error Codes Returned
INVPARAM   Invalid parameter value

PRBNFND   Problem not found

$$CODESTS^GMPLAPI2() – Check code status
This extrinsic function checks the status of the code associated with a problem.

Format
        $$ CODESTS^GMPLAPI2(.RETURN,GMPIFN,ADATE)

Input Parameters
.RETURN   [Required,Boolean] Set to 1 if the code is active on the date passed in ADATE, 0 otherwise.

GMPIFN   [Required,Numeric] Problem IEN (pointer to file 9000011)

ADATE   [Optional,DateTime] The date on which to check the status of ICD9 code. Default is current date.

Output

A Boolean value signaling if the call was successful or not

Error Codes Returned

INVPARAM   Invalid parameter value (GMPIFN,ADATE)

PRBNFND   Problem not found

$$DELETE^GMPLAPI2() – Deletes a problem

This extrinsic function deletes an existing problem from the list.

Format

$$DELETE^GMPLAPI2(.RETURN,GMPIFN,GMPROV,REASON)

Input Parameters

.RETURN   [Required,Boolean] Set to 0 if the delete failed, 1 otherwise. If it failed, RETURN will be will hold an array of error descriptions.

GMPIFN   [Required,Numeric] Problem IEN (pointer to file 9000011)

GMPROV   [Required,Numeric] Provider IEN (pointer to file 200)

REASON   [Optional,String] Comment describing the reason for deleting this problem.

Output

A Boolean value signaling if the call was successful or not

Error Codes Returned

INVPARAM   Invalid parameter passed (invalid GMPIFN or GMPROV)

PRBNFND   Problem not found

PROVNFND   Provider not found

PRBDLTD   Problem already deleted

$$DELETED^GMPLAPI2() – Is problem deleted

This extrinsic function verifies if a problem is deleted or not.

Format

$$DELETED^GMPLAPI2(.RETURN,GMPIFN)

Input Parameters

.RETURN   [Required,Boolean] Set to 1 if the problem is deleted, 0 otherwise

GMPIFN   [Required,Numeric] Problem IEN (pointer to file 9000011)

Output

A Boolean value signaling if the call was successful or not

Error Codes Returned

INVPARAM   Invalid parameter passed (GMPIFN)

PRBNFND   Problem not found

$$DETAIL^GMPLAPI2() – Detailed problem information

This extrinsic function returns detailed information on a problem.

Format

       $$DETAIL^GMPLAPI2(.RETURN,GMPIFN, GMPROV)

Input Parameters

.RETURN   [Required,Array] Array passed by reference that will receive the data. The output format is RETURN(FIELD)=internal_format^external_format (the data types shown below are for the internal format, external format is String):

RETURN(.01)   [Numeric] ICD9 diagnosis IEN (pointer to file 80)

RETURN(.02)   [Numeric] Patient IEN (pointer to file 9000001, same internal number as file 2)

RETURN(.03)   [DateTime] Date last modified

RETURN(.05)   [Numeric] Provider narrative IEN (pointer to file 9999999.27)

RETURN(.06)   [Numeric] Facility IEN (facility where this problem was originally observed and documented; pointer to file 9999999.06, same internal number as file 4)

RETURN(.07)   [Numeric] A number which, together with the Patient (#.02) and Facility (#.06) fields, serves as a unique identifier for this problem.

RETURN(.08)   [DateTime] The date this problem was entered into file.

RETURN(.12)   [String] Status: A = active or I = inactive

RETURN(.13)   [DateTime] Date of onset

RETURN(1.01)   [Numeric] Lexicon term IEN (pointer to file 757.01)

RETURN(1.02)   [String] Condition. Can be one of T = transcribed by a clerk from a paper chart, P = permanent (entered or verified by a provider), H = hidden (marked as removed)

RETURN(1.03)   [Numeric] Entered by IEN(pointer to file 200)

RETURN(1.04)   [Numeric] Recording provider IEN (the provider who first recorded this problem, either on paper or online; pointer to file 200)

RETURN(1.05)   [Numeric] Responsible provider IEN (pointer to file 200)

RETURN(1.06)   [Numeric] Service IEN (the service primarily involved in the treatment of this problem; pointer to file 49)

RETURN(1.07)   [DateTime] Date resolved

RETURN(1.08)   [Numeric] Clinic IEN (the clinic in which this patient is being seen for this problem; pointer to file 44)

RETURN(1.09)   [DateTime] Date recorded (the date this problem was originally recorded either online or in paper chart)

RETURN(1.1)   [Boolean] Service connected (1 if service connected, 0 if not)

RETURN(1.11)   [Boolean] Agent orange exposure. External value is "AGENT ORANGE"

RETURN(1.12)   [Boolean] Ionizing radiation exposure. External value is "RADIATION"

RETURN(1.13)   [Boolean] Persian gulf exposure. External value is "ENV CONTAMINANTS"

RETURN(1.14)   [String] Priority: A= acute, C = chronic

RETURN(1.15)   [Boolean] Head and/or neck cancer. External value is "HEAD/NECK CANCER"

RETURN(1.16)   [Boolean] Military sexual trauma. External value is "MIL SEXUAL TRAUMA"

RETURN(1.17)   [Boolean] Combat veteran. External value is "COMBAT VET"

RETURN(1.18)   [Boolean] Shipboard hazard and defense. External value is "SHAD"

RETURN(10,0)   [Numeric] number of comments

RETURN(10,#)   [String] note_nmbr^facility^note_narrative^status^date_note_added^author

All values are internal format:

note_nmbr [Numeric]   Unique note identifier

facility [Numeric]   IEN of location at which this note originated (pointer to file 9999999.06, same IEN as file 4)

note_narrative [String]   Narrative

status [String]   A=active, ""=inactive

date_note_added[DateTime] The date this note was entered into file

author [Numeric]   IEN of the provider who authored this note (pointer to file 200)

GMPIFN   [Required,Numeric] Problem IEN (pointer to file 9000011)

GMPROV   [Optional,Numeric] Provider IEN (pointer to file 200). The comments returned will be filtered by this provider.

Output
A Boolean value signaling if the call was successful or not

Error Codes Returned

INVPARAM   Invalid parameter value

PRBNFND   Problem not found

PROVNFND   Provider not found

$$DETAILX^GMPLAPI2() Detailed problem information, formatted

This extrinsic function returns detailed information on a problem. It is similar to DETAIL^GMPLAPI2 but the values returned are in External Format only, and the fields have human readable names

Format

$$DETAILX^GMPLAPI2(.RETURN,GMPIFN, GMPMULTI)

Input Parameters

.RETURN   (Required) Array passed by reference that will receive the data. The output format is:

RETURN("DIAGNOSIS")	[String] ICD Code
RETURN("PATIENT")	[String] Patient Name
RETURN("MODIFIED")	[DateTime] Date Last Modified
RETURN("NARRATIVE")	[String] Provider Narrative
RETURN("ENTERED")	[String] Date Entered ^ Entered by
RETURN("STATUS")	[String] Status
RETURN("PRIORITY")	[String] Priority Acute/Chronic
RETURN("ONSET")	[DateTime] Date of Onset
RETURN("PROVIDER")	[String] Responsible Provider
RETURN("RECORDED")	[String] Date Recorded ^ Recorded by
RETURN("CLINIC")	[String] Hospital Location
RETURN("SC")	[String] Service Connected (SC/NSC/"")
RETURN("EXPOSURE")	[Numeric] Number of exposure factors returned
RETURN("EXPOSURE",#)	[String] one of the following options: "AGENT ORANGE", "RADIATION", "ENV CONTAMINANTS", "HEAD AND/OR NECK CANCER", "MILITARY SEXUAL TRAUMA", "COMBAT VET", "SHAD"
RETURN("COMMENT")	[Numeric] Number of comments
RETURN("COMMENT",#)	[String] date_note_added^author_name^note_narrative

GMPIFN   [Required,Numeric] Problem IEN (pointer to file 9000011)

GMPMULTI   [Optional,Boolean] Multidivisional.If it is set to 1 all comments will be returned. If it is set to 0 only comments originating from the current users's institution (defined by DUZ(2)) will be returned. Default: 1 – all comments will be returned

Output

A Boolean value signaling if the call was successful or not

Error Codes Returned

INVPARAM   Invalid parameter value

PRBNFND   Problem not found

$$DUPL^GMPLAPI2() – Check for duplicate entries

This extrinsic function finds duplicate problem entries.

Format

$$DUPL^GMPLAPI2(.RETURN,GMPDFN,TERM,TEXT,GMPBOTH)

Input Parameters

.RETURN   [Required,Numeric] If duplicate problem is found this will be set to its IEN, 0 otherwise

GMPDFN   [Required,Numeric] Patient IEN (pointer to file 2)

TERM   [Required,Numeric] Problem id. Pointer to the EXPRESSIONS file # 757.01

TEXT   [Optional,String] Provider narrative to look for.

GMPBOTH   [Optional,Boolean] Both Lexicon term (TERM) and provider narrative (TEXT) should match in order to flag a duplicate. Default: either TERM or TEXT will flag a duplicate entry.

Output

A Boolean value signaling if the call was successful or not

Error Codes Returned

INVPARAM   Invalid parameter passed

PATNFND   Patient not found

TERMNFND   Lexicon term not found

$$INACTV^GMPLAPI2() - Inactivate a problem

This extrinsic function inactivates an existing problem.

Format

$$INACTV^GMPLAPI2(.RETURN,GMPIFN,GMPROV,NOTE,RESOLVED)

Input Parameters

.RETURN   [Required,Boolean] Set to 0 if the inactivation failed, 1 otherwise. If it failed, RETURN will hold an array of error descriptions.

GMPIFN   [Required,Numeric] Problem IEN (pointer to file 9000011)

GMPROV   [Required,Numeric] Provider IEN (pointer to file 200)

NOTE   [Optional,String] Comment describing the reason for inactivating this problem.

RESOLVED   [Optional,DateTime] Resolved date.

Output

A Boolean value signaling if the call was successful or not

Error Codes Returned

INVPARAM   Invalid parameter passed

PRBNFND	Problem not found
PRBDLDT	Problem is deleted
PRBINACT	Problem is already inactive.
PROVNFND	Provider not found

$$NEW^GMPLAPI2() – Add new problem to list

This extrinsic function adds a new problem to the list of patient problems.

Format

$$NEW^GMPLAPI2(.RETURN,GMPDFN,GMPROV,.GMPFLD,GMPLUSER)

Input Parameters

.RETURN   [Required,Boolean] Set to the new problem IFN, 0 otherwise. If it failed, RETURN will hold an array of error descriptions.

GMPDFN   [Required,Numeric] Patient IEN (pointer to file 2)

GMPROV   [Required,Numeric] Provider IEN (pointer to file 200)

.GMPFLD   [Required,Array] Array passed by reference that holds the new data. It should be in the internal format:

GMPFLD (.01)   [Optional,Numeric] ICD9 IEN (pointer to file 80). If it is null, it will default to code 799.9 (Other unknown and unspecified cause of morbidity or mortality). If it is passed it will be checked to be active and if not an error will be returned.

GMPFLD (.05)   [Required,Numeric] Provider narrative IEN (pointer to file 9999999.27). As an alternative this field can receive the actual narrative text in the form: ^narrative_text. In this case the text will be searched in file 9999999.27 and if not found, it will be added.

GMPFLD (.12)   [Optional,String] Status, can be one of 'A'=active or 'I'=inactive. Default: 'A'=active

GMPFLD (.13)   [Required,DateTime] Date of onset

GMPFLD (1.01) [Optional,Numeric] Lexicon term IEN (pointer to file 757.01), Default: 1 (Unresolved)

GMPFLD (1.04) [Optional,Numeric] Recording provider IEN (pointer to file 200)

GMPFLD (1.05) [Optional,Numeric] Responsible provider IEN (pointer to file 200)

GMPFLD (1.06) [Optional,Numeric] Service IEN (pointer to file 49)

GMPFLD (1.07) [Optional,DateTime] Date resolved

GMPFLD (1.08) [Optional,Numeric] Clinic (pointer to file 44)

GMPFLD (1.09) [Optional,DateTime] Date recorded

GMPFLD (1.1)   [Optional,Boolean] Service connected

GMPFLD (1.11) [Optional,Boolean] Agent orange exposure

GMPFLD (1.12) [Optional,Boolean] Ionizing radiation exposure

GMPFLD (1.13) [Optional,Boolean] Persian Gulf exposure

GMPFLD (1.14) [Optional,String] Priority, can be one of 'A'=acute, 'C'=chronic

GMPFLD (1.15) [Optional,Boolean] Head and or neck cancer

GMPFLD (1.16) [Optional,Boolean] Military sexual trauma

GMPFLD (1.17) [Optional,Boolean] Combat veteran

GMPFLD (1.18) [Optional,Boolean] Shipoard hazard and defense

GMPLFLD(10,"NEW",#) [Optional,String] Notes

GMPLUSER   [Optional,Boolean] User is a provider (Problem List User) or a clerk (transcriptionist). If GMPLUSER is 0, then new problems entered will be flagged as transcribed.

Output
A Boolean value signaling if the call was successful or not

Error Codes Returned

ICDINACT   Inactive ICD9 code

INVPARAM   Invalid parameter passed (GMPDFN, GMPROV,GMPFLD,GMPLUSER)

PATNFND   Patient not found

PROVNFND   Provider not found

$$ONSET^GMPLAPI2() – Returns onset date
This extrinsic function returns the onset date of a problem.

Format
$$ONSET^GMPLAPI2(.RETURN,GMPIFN)

Input Parameters

.RETURN   [Required,DateTime] Set to the onset date

GMPIFN   [Required,Numeric] Problem IEN (pointer to file 9000011)

Output
A Boolean value signaling if the call was successful or not

Error Codes Returned

INVPARAM   Invalid parameter passed (GMPIFN)

PRBNFND   Problem not found

$$UPDATE^GMPLAPI2() – Update an existing problem

This extrinsic function updates a problem's data.

Format

$$UPDATE^GMPLAPI2(.RETURN,GMPIFN,.GMPFLD,GMPLUSER,GMPROV)

Input Parameters

.RETURN   [Required,Boolean] Set to 0 if the save failed, 1 otherwise. If it failed, RETURN will hold an array of error descriptions.

GMPIFN   [Required,Numeric] Problem IEN (pointer to file 9000011)

.GMPFLD   [Required,Array] Array of modified values. See NEW^GMPLAPI2 for format.

GMPLUSER   [Optional,Boolean] User is a provider (Problem List User) or a clerk (transcriptionist). If GMPLUSER is 0, then new problems entered will be flagged as transcribed.

GMPROV   [Required,Numeric] Provider IEN

Output

A Boolean value signaling if the call was successful or not

Error Codes Returned

INVPARAM   Invalid parameter passed (GMPIFN)

PRBNFND   Problem not found

PROVNFND   Provider not found

$$VERIFIED^GMPLAPI2() – Is problem verified

This extrinsic function checks if a problem is verified or not.

Format

$$VERIFIED^GMPLAPI2(.RETURN,GMPIFN)

Input Parameters

.RETURN   [Required,Boolean] Set to 1 if the problem is verified, 0 otherwise

GMPIFN   [Required,Numeric] Problem IEN (pointer to file 9000011)

Output

A Boolean value signaling if the call was successful or not

Error Codes Returned

INVPARAM   Invalid parameter passed (GMPIFN)

PRBNFND   Problem not found

$$VERIFY^GMPLAPI2() – Verify a transcribed problem

This extrinsic function marks a transcribed problem as permanent

Format
    $$VERIFY^GMPLAPI2(.RETURN,GMPIFN)

Input Parameters
.RETURN   [Required,Boolean] Set to 0 if the save failed, 1 otherwise. If it failed, RETURN will
               hold an array of error descriptions.

GMPIFN   [Required,Numeric] Problem IEN (pointer to file 9000011)

Output
A Boolean value signaling if the call was successful or not

Error Codes Returned
ICDINACT   Inactive ICD9 code.

INVPARAM   Invalid parameter passed (GMPIFN)

PRBNFND   Problem not found

PRBVRFD   Problem already verified

FILELOCKED   File is in use. Try again later.

$$BUILDLST^GMPLAPI4() – Return list of problems

This extrinsic function returns a list of detailed information on problem IENs passed in GMPLIST.

Format
    $$BUILDLST^GMPLAPI4(.RETURN,.GMPLIST)

Input Parameters
.RETURN   [Required,Array] Array passed by reference that will receive the data. The output format is

       RETURN(0)=number of problems returned

       RETURN(#)=problem_IEN^status^problem^ICD9^onset^last_modified^sc^exposures^condition^

               location^loc_type^provider^service^priority^has_comments^

               date_recorded^sc_condition^icd_inactive

       problem_IEN   [Numeric] IEN from file 9000011

       status   [String] 'A'=active, 'I'=inactive

       problem   [String] display text

       ICD9   [String] ICD9 code

       onset   [DateTime] onset date

	last_modified	[DateTime] date when the problem was last modified
	sc	[String] 'SC'=service connected, 'NSC'=not service connected
	exposures	[String] a string of exposure factors separated by a forward slash (AO=agent orange, IR=ionizing radiation, EC=environment contaminants, HNC=head and/or neck cancer, MST=military sexual trauma, CV=combat veteran, SHD=shipboard hazard and defense)
	condition	[String] 'T'=transcribed, 'P'=permanent, 'H'=removed
	location	[String] format as clinic_IEN;clinic_name where clinic_IEN is a pointer to file 9999999_06 (same IEN as in file 4)
	loc_type	[String] one of hospital location types from file 44
	provider	[String] responsible provider formatted as provider_IEN;provider_name
	service	[Numeric] service primarily involved in the treatment of this problem (pointer to file 49)
	priority	[String] 'A'=acute, 'C'=chronic
	has_comments	[Boolean] 1 if there are notes attached to this problem
	date_recorded	[DateTime] date when the problem was originally recorded
	sc_condition	[String] a string concatenating the values of sc and exposures fields above
	icd_inactive	[String] set to '#' if the ICD9 code is inactive
.GMPLIST		[Required,Array] List of problem IENs in the following format:
		GMPLIST(0)=number of records
		GMPLIST(1)=problem_IEN 1
		…
		GMPLIST(n)=problem_IEN n

Output

A Boolean value signaling if the call was successful or not

Error Codes Returned

INVPARAM   Invalid parameter (GMPLIST)

$$DEFAULT^GMPLAPI4() – Return default values

This extrinsic function initializes an array with the default problem values. They could be used to edit a new problem and the saved using NEW^GMPLAPI2

Format

$$DEFAULT^GMPLAPI4(.RETURN,GMPROB,GMPICD,GMPTERM,GMPROV,GMPCLIN,GMPELIG)

Input Parameters

.RETURN   [Required,Array] An array passed by reference that will be initialized with the default values. It has the same structure as GMPFLD array described in NEW^GMPLAPI2 above.

GMPROB   [Required,String] Provider narrative

GMPICD   [Required,String] The ICD9 code associated to this problem

GMPTERM   [Required,Numeric] Lexicon term IEN (pointer to file 757.01)

GMPROV   [Optional,Numeric] Provider IEN (pointer to file 200)

GMPCLIN   [Optional, Numeric] Clinic IEN (pointer to file 44)

GMPELIG   [Optional,Array] An array of eligibilities for the environmental exposures. If any of the following nodes are defined it will set the corresponding entry to 1:

              GMPELIG("SC")=service connected

              GMPELIG("AO")=agent orange

              GMPELIG("IR")=ionizing radiation

              GMPELIG("EC")=environment contaminants

Output

A Boolean value signaling if the call was successful or not

Error Codes Returned

INACTICD   Inactive ICD9 code

$$DIAG^GMPLAPI4() – Return ICD code

This extrinsic function returns ICD code for a problem.

Format

$$DIAG^GMPLAPI4(.RETURN,GMPIFN)

Input Parameters

.RETURN   [Required,String] ICD code in the following format: pointer_to_icd_file^icd_code

GMPIFN   [Required,Numeric] Problem IEN (pointer to file 9000011)

Output

A Boolean value signaling if the call was successful or not

Error Codes Returned

INVPARAM   Invalid parameter (GMPIFN)

PRBNFND   Problem not found

$$GETPLIST^GMPLAPI4() – Return list of problem IENs

This extrinsic function returns a list of problems IENs given the patient IEN.

Format

$$GETPLIST^GMPLAPI4(.RETURN,GMPDFN,GMPSTAT,GMPREV,GMPROV,GMPVIEW, GMPIDX)

Input Parameters

.RETURN   [Required,Array] Array passed by reference that will receive the data. The output format is RETURN=Total number of patient's problems that would be returned if GMPROV and GMPVIEW were not specified. Depends on GMPSTAT value:if GMPSTAT="A", RETURN will be set to the number of patient's active problems, if GMPSTAT="I" will be set to the number of patient's inactive problems, etc.

RETURN(0)=number of problems returned

RETURN(#)=IEN #

GMPDFN   [Required,Numeric] Patient IEN (pointer to file 2)

GMPSTAT   [Optional,String] Status of problems to be returned. Can be any combination of (A)ctive, (I)nactive and (R)emoved. Default: "AI" – returns both active and inactive problems (but not removed ones).

GMPREV   [Optional,Boolean] Reversed order. The problems will be sorted in reversed order of recorded date.

GMPROV   [Optional,Numeric] Responsible provider IEN (pointer to file 200). If passed, the problems returned will be filtered by this provider. Default: "" – return all problems

GMPVIEW   [Optional,String] Filter by service location (inpatient problems) or clinic (outpatient problems). Format "S/facility_ien/facility_ien/…/" or "C/clinic_ien/clinic_ien/…/". Note: the string should end in a forward slash. Default: "" - returns all problems.

GMPIDX   [Optional,Boolean] Create "B" index. If set to 1 will append a "B" index to the output array.

RETURN("B",problem_ien)=#. Default - 0

Output

A Boolean value signaling if the call was successful or not

Error Codes Returned

INVPARAM   Invalid parameter passed

PATNFND   Patient not found

PROVNFND   Provider not found

$$HASPRBS^GMPLAPI4() – Are any problems assigned to the patient

This extrinsic function returns a Boolean value signaling if there are any problems in the patient file.

Format

$$HASPRBS^GMPLAPI4(.RETURN,GMPDFN,GMPSTAT)

Input Parameters

.RETURN   [Required,Boolean] Set to 1 if patient file contains problems with status GMPSTAT

GMPDFN   [Required,Numeric] Patient IEN (pointer to file 2)

GMPSTAT   [Optional,String] Problem status: Any combination of (A)ctive and (I)nactive. Default: AI = both active and inactive.

Output

A Boolean value signaling if the call was successful or not

Error Codes Returned

INVPARAM   Invalid parameter (GMPDFN,GMPSTAT)

PATNFND   Patient not found

$$LASTMOD^GMPLAPI4() – Last modified date

This extrinsic function returns the last modified date for a problem.

Format

$$LASTMOD^GMPLAPI4(.RETURN,GMPIFN)

Input Parameters

.RETURN   [Required,DateTime] Set to last modified date

GMPIFN   [Required,Numeric] Problem IEN (pointer to file 9000011)

Output

A Boolean value signaling if the call was successful or not

Error Codes Returned

INVPARAM   Invalid parameter (GMPIFN)

PRBNFND   Problem not found

$$LIST^GMPLAPI4() – Return list of problems

This extrinsic function returns a filtered list of patient problems.

Format

$$LIST^GMPLAPI4(.RETURN,GMPDFN,GMPSTAT, GMPROV,GMPVIEW, GMPREV,GMPIDX)

Input Parameters

.RETURN   [Required,Array] Array passed by reference that will receive the data. The output format is
                  RETURN=Total number of problems in patient's file. Depends on GMPSTAT value. See
                     GETPLIST^GMPLAPI4 above for details

                  RETURN(0)=number of problems returned

RETURN(I)=problem_IEN^status^problem^ICD9^onset^last_modified^sc^exposures^condition^

location^loc_type^provider^service^priority^has_comments^

date_recorded^sc_condition^inactive . See BUILDLST^GMPLAPI4 above for details.

GMPDFN   [Required,Numeric] Patient IEN (pointer to file 2)

GMPSTAT   [Optional,String] Status of problems to be returned. Can be any combination of (A)ctive,
            (I)nactive and (R)emoved. Default: A = returns active problems only

GMPROV   [Optional,Numeric] Provider IEN (pointer to file 200). If present, the problems returned
            will be filtered by this provider. Default – return all problems

GMPVIEW   [Optional,String] Filter by service location or clinic. Format "S/facility_ifn/facility_ifn/…" or
            "C/clinic_ifn/clinic_ifn/…". If facility IFN's are note passed, returns inpatient problems
            when GMPVIEW="S" or outpatient ones when it is set to "C". Default – returns all
            problems.

GMPREV   [Optional,Boolean] Reversed order. The problems will be sorted in reversed order of
            recorded date.

GMPIDX   [Optional,Boolean] Create "B" index. If set to 1 will append a "B" index to the output array.

            RETURN("B",ien#)=#. Default - 0

Output
A Boolean value signaling if the call was successful or not

Error Codes Returned
INVPARAM   Invalid parameter (GMPDFN,GMPSTAT,GMPROV)

PATNFND   Patient not found

PROVNFND   Provider not found

$$PATIENT^GMPLAPI4() – Get patient IEN
This extrinsic function returns the patient IEN given a problem IEN.

Format
$$PATIENT^GMPLAPI4(.RETURN,GMPIFN)

Input Parameters
.RETURN   [Required,Numeric] Set to patient IEN (pointer to file 2)

GMPIFN   [Required,Numeric] Problem IEN (pointer to file 9000011)

Output
A Boolean value signaling if the call was successful or not

Error Codes Returned

INVPARAM   Invalid parameter (GMPIFN)

PRBNFND   Problem not found

$$PROBNARR^GMPLAPI4() – Get provider narrative

This extrinsic function returns the provider narrative for a given problem IEN.

Format

        $$PROBNARR^GMPLAPI4(.RETURN,GMPIFN)

Input Parameters

.RETURN   [Required,String] Set to provider_narrative_ien^problem_narrative

GMPIFN   [Required,Numeric] Problem IEN (pointer to file 9000011)

Output

A Boolean value signaling if the call was successful or not

Error Codes Returned

INVPARAM   Invalid parameter (GMPIFN)

PRBNFND   Problem not found

$$REPLACE^GMPLAPI4() – Replace diagnosis

This extrinsic function replaces the diagnosis code for a given problem IEN.

Format

        $$REPLACE^GMPLAPI4(.RETURN,GMPIFN,NEWDIAG)

Input Parameters

.RETURN   [Required,Boolean] Set 1 if the call succeeded

GMPIFN   [Required,Numeric] Problem IEN (pointer to file 9000011)

NEWDIAG   [Required,Numeric] ICD9 code IEN (pointer to file 80)

Output

A Boolean value signaling if the call was successful or not

Error Codes Returned

INVPARAM   Invalid parameter (GMPIFN,NEWDIAG)

PRBNFND   Problem not found

ICDNFND   ICD9 code not found

ICDINACT   ICD9 code is inactive

$$UNDELETE^GMPLAPI4() – Undeletes problem

This extrinsic function marks a deleted problem as permanent.

Format

$$UNDELETE^GMPLAPI4(.RETURN,GMPIFN)

Input Parameters

.RETURN   [Required,Boolean] Set to 1 if the call succeeded

GMPIFN   [Required,Numeric] Problem IEN (pointer to file 9000011)

Output

A Boolean value signaling if the call was successful or not

Error Codes Returned

INVPARAM   Invalid parameter (GMPIFN)

INVREC   Invalid record (problem already active)

PRBNFND   Problem not found

$$VALID^GMPLAPI4() – Is problem IEN valid

This extrinsic function returns a boolean value signaling if the problem IEN corresponds to a valid record.

Format

$$VALID^GMPLAPI4(.RETURN,GMPIFN)

Input Parameters

.RETURN   [Required,Boolean] Set to 1 if problem file contains GMPIFN

GMPIFN   [Required,Numeric] Problem IEN (pointer to file 9000011)

Output

A Boolean value signaling if the call was successful or not

Error Codes Returned

INVPARAM   Invalid parameter (GMPIFN)

PRBNFND   Problem not found

2.2.   Problem notes API

$$NEWNOTE^GMPLAPI3() – Add new comment to problem

This extrinsic function adds a new comment to a problem.

Format

$$NEWNOTE^GMPLAPI3(.RETURN,GMPIFN,GMPROV,.NOTES)

Input Parameters

.RETURN   [Required,Boolean] Set to 0 if the call failed, 1 otherwise. If it failed, RETURN will
                 hold an array of error descriptions.

GMPIFN   [Required,Numeric] Problem IEN (pointer to file 9000011)

GMPROV   [Required,Numeric] Provider IEN (pointer to file 200)

.NOTES [Required,Array] Array passed by reference that holds comment lines. It should be in the following format:

  NOTES(#)=comment

Output

A Boolean value signaling if the call was successful or not

Error Codes Returned

FILELOCK File is in use. Try again later.

ICDINACT Inactive ICD found. Edit diagnosis first.

INVPARAM Invalid parameter passed (GMPIFN,GMPROV, NOTES)

PRBDLTD Problem is deleted

PRBNFND Problem not found

PROVNFND Provider not found

$$NOTES^GMPLAPI3() – Return list of comments for problem

This extrinsic function returns the list of comments assigned to a problem.

Format

  $$NOTES^GMPLAPI3(.RETURN,GMPIFN,GMPACT)

Input Parameters

.RETURN [Required,Array] Array passed by reference that will receive the data. The output format depends on the value of GMPFMT

GMPIFN [Required,Numeric] Problem IEN (pointer to file 9000011)

GMPACT [Optional,Boolean] Active. If set to 1 only active comments will be returned. Default: 1

GMPFMT [Optional,Numeric] Format. Controls the output format. Default: 1. The following values are allowed:

  1 -- RETURN(#)=note_narrative
  2 -- RETURN(#)=date_note_added^author^note_narrative

  Values are in external format:

  date_note_added [String] The date this note was entered into file

  author [String] Name of the provider who authored this note

  note_narrative [String] Narrative

  3 -- RETURN(#)=note_nmbr^facility^note_narrative^status^date_note_added^author
  4 -- RETURN(facility,note_nmbr)=note_nmbr^^note_narrative^status^date_note_added^author

  For formats 3 and 4 all values returned are in internal format:

note_nmbr [Numeric]	Unique note identifier	
facility [Numeric]	IEN of location at which this note originated (pointer to file 9999999.06, same IEN as file 4)	
note_narrative [String]	Narrative	
status [String]	A=active, ""=inactive	
date_note_added[DateTime]	The date this note was entered into file	
author [Numeric]	IEN of the provider who authored this note (pointer to file 200)	

Output
A Boolean value signaling if the call was successful or not

Error Codes Returned
INVPARAM   Invalid parameter (GMPIFN,GMPFMT)

PRBNFND   Problem not found

$$UPDNOTE^GMPLAPI3() – Replaces an existing comment
This extrinsic function updates a problem's data.

Format
    $$UPDNOTE^GMPLAPI3(.RETURN,GMPIFN,NEWNOTE, GMPROV)

Input Parameters
.RETURN   [Required,Boolean] Set to 0 if the save failed, 1 otherwise. If it failed, RETURN will hold an array of error descriptions.

GMPIFN   [Required,Numeric] Problem IEN (pointer to file 9000011)

NEWNOTE   [Required,String] New comment formatted as: note_IEN^facility_IEN^Text

If Text is empty the comment will be deleted.

note_IEN [Numeric]	IEN in the "note" multiple (9000011.1111)	
facility_IEN [Numeric]	IEN of location at which this note originated (pointer to file 9999999.06, same IEN as file 4)	
Text [String]	new note narrative	

GMPROV   [Required,Numeric] Provider IEN (pointer to file 200)

Output
A Boolean value signaling if the call was successful or not

Error Codes Returned
INVPARAM   Invalid parameter

NOTENFND   Note not found

PRBNFND   Problem not found

PROVNFND   Provider not found

2.3. Audit history API

$$AUDET^GMPLHIST() – Return audit entry details
This extrinsic function returns detailed data for an audit entry.

Format
     $$AUDET^GMPLHIST(.RETURN,AIFN)

Input Parameters
.RETURN   [Required,Array] Array passed by reference that will receive the audit entry details. It has the following format (the data is intended for display so it is in external format):

     RETURN("DATE")   date modified

     RETURN("FLD")   modified_field_id^modified_field_name (ex. .12^STATUS)

     RETURN("NEW")   new value

     RETURN("OLD")   old value

     RETURN("PROV")   provider name

     RETURN("OLDDATE")   old note date (this subscript is returned only when field is 1101^NOTE)

     RETURN("OLDNOTE")   old note text (this subscript is returned only when field is 1101^NOTE)

AIFN   [Required,Numeric] Audit entry IEN (pointer to file 125.8)

Output
A Boolean value signaling if the call was successful or not

Error Codes Returned
None

$$AUDITX^GMPLHIST() – Return problemaudit data (external format)
This extrinsic function returns all audit entries for a given problem, ready for display (in external format).

Format
     $$AUDITX^GMPLHIST(.RETURN,GMPIFN,FIELD,VALUE)

Input Parameters
.RETURN   [Required,Array] An array passed by reference that will receive the audit data. It has the following format (all values are in external format):

     RETURN   number of audit entries returned

RETURN(#,"FIELD")	changed field id^changed field name (ex. .12^STATUS)
RETURN(#,"MODIFIED")	modified date
RETURN(#,"MODIFIEDBY")	user who modified the problem (provider name)
RETURN(#,"NEW")	new value
RETURN(#,"OLD")	old value
RETURN(#,"OLDPROBLEM")	the entire problem entry (internal format) as it existed before it was changed (this subscript is returned only if the changed field is 1101^NOTE)
RETURN(#,"REASON")	reason for change
RETURN(#,"REQUESTINGBY")	requesting provider (the provider who either changed the data or requested the change)

GMPIFN	[Required,Numeric]	Problem IEN (pointer to file 9000011)
FIELD	[Optional,Numeric]	If specified, the function will return audit data for this field change only
VALUE	[Optional,Numeric]	If specified along with FIELD the function will return changes made to that field having the new value of VALUE only.

Output
A Boolean value signaling if the call was successful or not

Error Codes Returned
PRBNFND   Problem not found

$$GETAUDIT^GMPLHIST() – Return problem audit data (internal format)
This extrinsic function returns all audit entries for a given problem in internal format.

Format
        $$GETAUDIT^GMPLHIST(.RETURN,GMPIFN,FIELD,VALUE)

Input Parameters
.RETURN   [Required,Array] An array passed by reference that will receive the audit data. It has the following format (all values are in external format):

        RETURN   number of audit entries returned

        RETURN(#,0)   audit data: field_number^field_name^date_modified^who_modified^old_value^

                        new_value^reason_for_change^requesting_provider

                        feld_number [Numeric]   Number of the modified field (data dictionary
                                                        number)

	field_name [String]	Human readadble name of the modified field
	date_modified [DateTime]	Date and time this field value has been modified (internal format)
	who_modified [Numeric]	IEN of the user who modified this problem (pointer to file 200)
	old_value [String]	Old field value (internal format)
	new_value [String]	New field value (internal format)
	reason_for_change [String]	Description of why the change was made
	requesting_provider [Numeric]	IEN of the provider who either changed this data, or directed it to be changed (pointer to file 200)

RETURN(#,1)		old problem entry (the entire problem entry as it existed before change)

GMPIFN	[Required,Numeric]	Problem IEN (pointer to file 9000011)
FIELD	[Optional,Numeric]	If specified, the function will return audit data for this field change only
VALUE	[Optional,Numeric]	If specified along with FIELD the function will return changes made to that field having the new value of VALUE only.

Output
A Boolean value signaling if the call was successful or not

Error Codes Returned
INVPARAM   Invalid parameter (GMPIFN)

$$GETHIST^GMPLHIST() – Return audit entries list
This extrinsic function returns a list of audit entry IENs for a given problem.

Format
$$GETHIST^GMPLHIST(.RETURN,GMPIFN)

Input Parameters
.RETURN	[Required,Array]	An array passed by reference that will receive the list of audit entry IENs.
RETURN	[Boolean]	1 if the call succeeded, 0 otherwise
RETURN(#)	[Numeric]	Audit entry IEN (file 125.8)
GMPIFN	[Required,Numeric]	Problem IEN (pointer to file 9000011)

Output
A Boolean value signaling if the call was successful or not

Error Codes Returned

PRBNFND   Problem not found

2.4.   Problem selection lists API

$$CATUSED^GMPLAPI1() – Is category used

Verifies if a problem category is used by any problem selection list.

Format

        CATUSED^GMPLAPI1(.RETURN,GMPLGRP)

Input Parameters

.RETURN   [Required,Boolean] Set to 1 if problem category is used by at least one list, 0 otherwise

GMPLGRP   [Required,Numeric] Problem category IEN (pointer to file 125.11)

Output

A Boolean value signaling if the call was successful or not.

Error Codes Returned

INVPARAM   Invalid parameter passed

CTGNFND   Problem category not found

$$DELCAT^GMPLAPI1() – Delete category

Delete a problem category if it is not used by any problem selection list.

Format

        DELCAT^GMPLAPI1(.RETURN,GMPLGRP)

Input Parameters

.RETURN   [Required,Boolean] Set to 1 if the operation succeeds, 0 otherwise

GMPLGRP   [Required,Numeric] Problem category IEN (pointer to file 125.11)

Output

A Boolean value signaling if the call was successful or not.

Error Codes Returned

INVPARAM   Invalid parameter passed

CTGNFND   Problem category not found

CATUSED   There is at least one problem selection list that contains the category problem

$$DELLST^GMPLAPI1() – Delete list

Delete a problem selection list if this list is not assigned to any users.

Format

        DELLST^GMPLAPI1(.RETURN,GMPLLST)

Input Parameters

.RETURN   [Required,Boolean] Set to 1 if operation succeed, 0 otherwise

GMPLLST   [Required,Numeric] Problem selection list IEN (pointer to file 125)

Output

A Boolean value signaling if the call was successful or not.

Error Codes Returned

FILELOCK   Record in use. Try again in a few moments.

INVPARAM   Invalid parameter passed (GMPLLST)

LISTNFND   Problem selection list not found

LISTUSED   List is assigned to at least one user

$$GETCAT^GMPLAPI1() – Detailed category information
Returns specified problem selection category.

Format
$$GETCAT^GMPLAPI1(.RETURN,GMPLGRP)

Input Parameters

.RETURN   [Required,Array] Array passed by reference that will receive the data. The output format is
RETURN(problem_IEN)=sequence#^lexicon_term_IEN(757.01)^display_text^ICD9_code

RETURN(problem_IEN, "CODE")=ICD9_code^inactive_flag

RETURN("CAT","MODIFIED")=date last modified in internal^external format

RETURN("CAT","NAME")=category name

RETURN("SEQ",sequence#)=problem_IEN

RETURN("PROB",lexicon_term_IEN(757.01))=problem_IEN

GMPLGRP   [Required,Numeric] Problem category IEN (pointer to file 125.11)

Output
A Boolean value signaling if the call was successful or not.

Error Codes Returned
INVPARAM   Invalid parameter passed (invalid GMPLGRP)

CTGNFND   Problem category not found

$$GETLIST^GMPLAPI1() – Detailed problem list information
Returns problem selection list details.

Format
$$GETLIST^GMPLAPI1(.RETURN,GMPLLST,CODLEN,MINIM)

Input Parameters

.RETURN   [Required,Array] Array passed by reference that will receive the data. The output format is

RETURN("LST","CLINIC")=clinic_IEN^clinic_name – the clinic associated with this list (clinic_IEN is pointer to file 44)

RETURN("LST","NAME") =name – the problem selection list name

RETURN("LST","MODIFIED")= date last modified in internal^external format, if the list is new this parameter will be valued: ^<new list>

RETURN(0) - number of categories

RETURN(selection_list_IEN)=sequence#^category_IEN^subheader^show_problems_flag

RETURN("GRP", category_IEN)=selection_list_IEN

RETURN("SEQ",sequence#)=selection_list_IEN

RETURN("GRP", category_IEN,#)=problem_name^ICD9_code^inactive_flag (1 for inactive code, 0 for active)

GMPLLST   [Required,Numeric] Problem selection list IEN (pointer to file 125)

CODLEN   [Optional,Numeric] A number that specifies the maxim length of the returned problem text

MINIM   [Optional,Boolean] When set to 1 will return minimal information (problem selection list name, date last modified and clinic), otherwise returns full info. Default: 0

Output
A Boolean value signaling if the call was successful or not.

Error Codes Returned

INVPARAM   Invalid parameter passed (GMPLLST,CODLEN)

LISTNFND   Problem selection list not found

$$LOCKCAT^GMPLAPI1() – Lock category
Lock specified problem category.

Format
$$LOCKCAT^GMPLAPI1(.RETURN,GMPLGRP)

Input Parameters

.RETURN   [Required,Boolean] Set to 1 if operation succeed, 0 otherwise

GMPLGRP   [Required,Numeric] Problem category IEN (pointer to file 125.11)

Output
A Boolean value signaling if the call was successful or not.

Error Codes Returned

INVPARAM   Invalid parameter passed (invalid GMPLGRP)

CTGNFND   Problem category not found

FILELOCK   Problem category is already locked by another process

$$LOCKLST^GMPLAPI1() – Lock list

Lock specified problem selection list.

Format

      LOCKLST^GMPLAPI1(.RETURN,GMPLLST)

Input Parameters

.RETURN   [Required,Boolean] Set to 1 if the operation succeeds, 0 otherwise

GMPLLST   [Required,Numeric] Problem selection list IEN (pointer to file 125)

Output

A Boolean value signaling if the call was successful or not.

Error Codes Returned

INVPARAM   Invalid parameter passed (invalid GMPLLST)

LISTNFND   Problem selection list not found

FILELOCK   Problem selection list is already locked by another process

$$LSTUSED^GMPLAPI1() – Is list used?

Returns the number of users this list is assigned to.

Format

      LSTUSED^GMPLAPI1(.RETURN,GMPLLST)

Input Parameters

.RETURN   [Required,Numeric] Set to the number of users this list is assigned to.

GMPLLST   [Required,Numeric] Problem selection list IEN (pointer to file 125)

Output

A Boolean value signaling if the call was successful or not.

Error Codes Returned

INVPARAM   Invalid parameter passed (invalid GMPLLST)

LISTNFND   Problem selection list not found

$$NEWCAT^GMPLAPI1() – Add new category

Add new problem category.

Format

      NEWCAT^GMPLAPI1(.RETURN,GMPGRP,DUPLIC)

Input Parameters

.RETURN   [Required,Numeric] Set to the new problem category IEN if the call succeeds, 0 otherwise

GMPLGRP   [Required,String] The problem category name. Category name must be 3-30 characters, not numeric or starting with punctuation

DUPLIC   [Optional,Boolean] Allow duplicate category names or not. Default: duplicate names are not allowed.

Output

A Boolean value signaling if the call was successful or not.

Error Codes Returned

INVPARAM   Invalid parameter passed (GMPGRP)

CTGEXIST   Another problem category with the same name already exists

$$NEWLST^GMPLAPI1() – Add new list

Add new problem selection list.

Format

    NEWLST^GMPLAPI1(.RETURN,GMPLLST,GMPLLOC)

Input Parameters

.RETURN   [Required,Numeric] Set to the new problem selection list IEN if the call succeeds, 0 otherwise

GMPLLST   [Required,String] The problem selection list name. List name must be 3-30 characters, not numeric or starting with punctuation

GMPLLOC   [Optional,Numeric] IEN of location which will be assigned to the new problem selection list

Output

A Boolean value signaling if the call was successful or not.

Error Codes Returned

INVPARAM   Invalid parameter passed (GMPLLST,GMPLLOC)

LOCNFND   Location does not exist

LISTXST   Another list with the same name already exists

$$SAVGRP^GMPLAPI1() – Save category

Save changes to existing problem category.

Format

    $$SAVGRP^GMPLAPI1(.RETURN,GMPLGRP,SOURCE)

Input Parameters

.RETURN   [Required,Boolean] Set to 1 if operation succeed, 0 otherwise

GMPLGRP [Required,Numeric] The category IEN (pointer to file 125.11)

SOURCE   [Required,Array] A collection of problems that will be assigned to the problem category

    SOURCE(n)=sequence^ lexicon_term_IEN ^display_text^ICD9_code

     'n' can have one of the following values:

- problem_IEN (from file 125.12) – in this case the corresponding entry will be updated or, will be removed if SOURCE(n)="@"
- a sequence number followed by 'N' (e.g. 1N,2N etc.) – in this case a new problem entry will be added to this category

    sequence [Numeric,Optional] a number which determines the order this problem will appear within this group

    lexicon_term_IEN [Numeric,Required] IEN from file 757.01

    display_text [String,Optional] display text

    ICD9_code [String,Optional] ICD9 code to be displayed with the text of this problem

Output

A Boolean value signaling if the call was successful or not.

Error Codes Returned

ICDINACT Inactive ICD code

ICDNFND ICD9 code not found

INVPARAM Invalid parameter passed (invalid GMPLGRP)

ITEMNFND Item not found (problem_IEN not found in file 125.12)

CTGNFND Problem category not found

$$SAVLST^GMPLAPI1() – Save list

Save changes to existing list.

Format

   $$SAVLST^GMPLAPI1(.RETURN,GMPLLST,SOURCE)

Input Parameters

.RETURN   [Required,Boolean] Set to 1 if operation succeed, 0 otherwise

GMPLLST   [Required,Numeric] Problem selection list IEN (pointer to file 125)

SOURCE   [Required,Array] A collection of problem categories that will be assigned to the selection list. The array should have the following format:

    SOURCE(n)=sequence#^category_IEN^subheader^show_problems_flag

    'n' can have one of the following values:

- problem selection list contents IEN (from file 125.1) – in this case the

corresponding entry will be updated or, will be removed if SOURCE(n)="@"
- a sequence number followed by 'N' (e.g. 1N,2N etc.) – in this case a new category entry will be added to this category

Output
A Boolean value signaling if the call was successful or not.

Error Codes Returned
INVPARAM   Invalid parameter passed (GMPLLST,SOURCE)

LISTNFND   Problem selection list not found

UNLKCAT^GMPLAPI1() – Unlock category
Unlock specified problem category.

Format
        UNLKCAT^GMPLAPI1(GMPLGRP)

Input Parameters
GMPLGRP   [Required,Numeric] Problem category IEN (pointer to file 125.11)

Output
None

Error Codes Returned
None

UNLKLST^GMPLAPI1() – Unlock list
Unlock specified problem selection list.

Format
        UNLKLST^GMPLAPI1(GMPLLST)

Input Parameters
GMPLLST   [Required,Numeric] Problem selection list IEN (pointer to file 125)

Output
None

Error Codes Returned
None

$$ADDLOC^GMPLAPI5() – Add location
Assigns a problem selection list to a clinic.

Format
        $$ADDLOC^GMPLAPI5(.RETURN,GMPLLST,GMPLLOC)

Input Parameters
.RETURN   [Required,Boolean] Set to 1 if operation succeed, 0 otherwise

GMPLLST   [Required,Numeric] The problem selection list IEN (pointer to file 125)

GMPLLOC   [Required,Numeric] The clinic location IEN (pointer to file 44)

Output
A Boolean value signaling if the call was successful or not.

Error Codes Returned
FILELOCK   Record in use. Try again in a few moments.

INVPARAM   Invalid parameter passed (invalid GMPLLST or GMPLLOC)

LISTNFND   Problem selection list not found

LOCNFND   Location not found

$$GETASUSR^GMPLAPI5() – Get users that own a problem selection list
This function returns the users assigned to a specific problem selection list.

Format
        $$GETASUSR^GMPLAPI5(.RETURN,GMPLLST)

Input Parameters
.RETURN   [Required,Array] Array passed by reference that will receive the data. The output format is

        RETURN(0)=number of currently assigned users

        RETURN(#,"ID")=user IEN (pointer to file 200)

        RETURN(#,"NAME")=user name

GMPLLST   [Required,Numeric] The problem selection list IEN.

Output
A Boolean value signaling if the call was successful or not.

Error Codes Returned
INVPARAM   Invalid parameter passed (GMPLLST)

LISTNFND   Problem selection list not found

$$GETCATD^GMPLAPI5() – Detailed problem selection category information
Returns detailed problem selection category information.

Format
        $$GETCATD^GMPLAPI5(.RETURN,GMPLGRP,CODLEN)

Input Parameters
.RETURN   [Required,Array] Array passed by reference that will receive the data. The output format is

        RETURN("GRP",category_IEN,#)=problem_name^ICD9_code^inactive_flag^lexicon_term_IEN(757.
                01)

GMPLGRP   [Required,Numeric] The problem category IEN

CODLEN   [Optional,Numeric] A number that specifies the maximum length of the returned problem text

Output
A Boolean value signaling if the call was successful or not.

Error Codes Returned
INVPARAM   Invalid parameter passed (GMPLGRP,CODLEN)

CTGNFND   Problem category not found

$$GETCATS^GMPLAPI5() – Get categories
This function returns all existing problems categories, or those that match the search criteria.

Format
       $$GETCATS^GMPLAPI5(.RETURN,SEARCH,START,NUMBER)

Input Parameters
.RETURN   [Required,Array] Array passed by reference that will receive the data. The output format is

       RETURN(0)=number_of_entries_found^maximum_requested^any_more?

       The number of entries found will be equal to or less than the maximum requested

       The maximum requested should be equal the NUMBER parameter or if NUMBER not passed, "*".

       The any more? Is 1 if there are more matching entries, or 0 if not.

       RETURN(#,"ID")=problem category IEN

       RETURN(#,"NAME")=problem category name

SEARCH   [Optional,String] The partial match restriction.

.START   [Optional,String] The index from which to begin the list. It can be used for pagination, passed by reference will be set to the last entry returned, if NUMBER parameter is specified and there are more matching entries. Subsequent calls will use this parameter to know where to start the next list. Simillar to .FROM parameter to LIST^DIC.

NUMBER   [Optional,Numeric] The number of entries to return.

Output
A Boolean value signaling if the call was successful or not.

Error Codes Returned
INVPARAM   Invalid parameter (START,NUMBER)

$$GETLSTS^GMPLAPI5() – Get lists
This function returns a filtered list of problem selection lists.

Format

$$GETLSTS^GMPLAPI5(.RETURN,SEARCH,START,NUMBER)

Input Parameters

.RETURN   [Required,Array] Array passed by reference that will receive the data. The output format is

RETURN(0)=number_of_entries_found^maximum_requested^any_more?

The number of entries found will be equal to or less than the maximum requested

The maximum requested should be equal the NUMBER parameter or if NUMBER not passed, "*".

The any more? Is 1 if there are more matching entries, or 0 if not.

RETURN(#,"ID")=problem selection list IEN

RETURN(#,"NAME")=problem selection list name

SEARCH   [Optional,String] The partial match restriction.

.START   [Optional,String] The index from which to begin the list. It can be used for pagination, passed by reference will be set to the last entry returned, if NUMBER parameter is specified and there are more matching entries. Subsequent calls will use this parameter to know where to start the next list. Simillar to .FROM parameter to LIST^DIC.

NUMBER   [Optional,Numeric] The number of entries to return.

Output

A Boolean value signaling if the call was successful or not.

Error Codes Returned

INVPARAM   Invalid parameter (START,NUMBER)

$$ASSUSR^GMPLAPI6() – Assign list to users

Assign a problem selection list to one or more users.

Format

ASSUSR^GMPLAPI6(.RETURN,GMPLLST,GMPLUSER)

Input Parameters

.RETURN   [Required,Boolean] Set to 1 if operation succeed, 0 otherwise

GMPLLST   [Required,Numeric] Problem selection list IEN (pointer to file 125)

GMPLUSER   [Required,String] Users IFN (pointer to file 200) list separated by "^"

Output

A Boolean value signaling if the call was successful or not.

Error Codes Returned

INACTICD   This Selection List contains problems with inactive ICD9 codes associated with them.

INVPARAM   Invalid parameter passed (invalid GMPLLST or GMPLUSER)

LISTNFND   Problem selection list not found

PROVNFND   Provider not found

$$GETCLST^GMPLAPI6() – Get first problem selection list assigned to the clinic

This function returns the first problem selection list assigned to the clinic.

Format

$$GETCLST^GMPLAPI6(.RETURN,GMPCLIN)

Input Parameters

.RETURN   [Required,String] Passed by reference, will receive the data. The output format is

            problem_selection_list_IEN^problem_selection_list_name

GMPCLIN   [Required,Numeric] Clinic IEN (pointer to file 44).

Output

A Boolean value signaling if the call was successful or not.

Error Codes Returned

INVPARAM   Invalid parameter passed (GMPCLIN)

LOCNFND   Location not found

$$GETULST^GMPLAPI6() – Get problem selection list assigned to the user

This function returns the problem selection list assigned to a specific user.

Format

$$GETULST^GMPLAPI6(.RETURN,USER)

Input Parameters

.RETURN   [Required,String] Passed by reference, will receive the data. The output format is

            problem_selection_list_IEN^problem_selection_list_name

USER   [Required,Numeric] User IEN (pointer to file 200).

Output

A Boolean value signaling if the call was successful or not.

Error Codes Returned

INVPARAM   Invalid parameter passed (USER)

PROVNFND   Provider not found

$$REMUSR^GMPLAPI6() – Remove list from users

Remove assigned problem selection list from one or more users.

Format

REMUSR^GMPLAPI6(.RETURN,GMPLST,GMPLUSER)

Input Parameters

.RETURN   [Required,Boolean] Set to 1 if the operation succeeds, 0 otherwise

GMPLLST   [Required,Numeric] The problem selection list IEN (pointer to file 125)

GMPLUSER   [Required,String] Users IEN (pointer to file 200) list separated by "^"

Output

A Boolean value signaling if the call was successful or not.

Error Codes Returned

INVPARAM   Invalid parameter passed (invalid GMPLLST or GMPLUSER)

LISTNFND   Problem selection list not found

PROVNFND   Provider not found

$$VALGRP^GMPLAPI6() – Check category for inactive codes

This function checks all problems in the category for inactive codes.

Format

$$VALGRP^GMPLAPI6(.RETURN,GMPLGRP)

Input Parameters

.RETURN   [Required,Boolean] Passed by reference, will be set to 1 if category has no problems with inactive codes, 0 if has one or more.

GMPLGRP   [Required,Numeric] Problem selection category IEN.

Output

A Boolean value signaling if the call was successful or not.

Error Codes Returned

INVPARAM   Invalid parameter passed (GMPLGRP)

CTGNFND   Category not found

$$VALLIST^GMPLAPI6() – Check selection list for inactive codes

This function checks all categories in the list for problems with inactive codes.

Format

$$VALLIST^GMPLAPI6(.RETURN,GMPLLST)

Input Parameters

.RETURN   [Required,Boolean] Passed by reference, will be set to 1 if list has no problems with inactive codes, 0 if has one or more.

GMPLLST   [Required,Numeric] Problem selection list IEN (pointer to file 125)

Output

A Boolean value signaling if the call was successful or not.

Error Codes Returned

INVPARAM   Invalid parameter passed (GMPLLST)

LISTNFND   Problem selection list not found

2.5.   Site parameters API

$$GET^GMPLSITE() – Get site parameters

Gets the Problem List site parameters.

Format

      $$GET^GMPLSITE(.RETURN)

Input Parameters

.RETURN   [Required,Array] Set to site parameters.

    RETURN("VER")   [Boolean] Automatically verify problems

    RETURN("PRT")   [Boolean] Prompt to print a chart copy on exit

    RETURN("CLU")   [Boolean] Use Clinical Lexicon

    RETURN("REV")   [String] Problem display order: C = chronological, R = reverse chronological

    RETURN("SDP")   [Boolean] Screen duplicate ICD9 entries.

Output

None.

Error Codes Returned

None

$$SET^GMPLSITE() – Set site parameters

Sets the site Problem List parameters.

Format

      $$SET^GMPLSITE(.RETURN,.PARAM)

Input Parameters

RETURN   [Required,String] Set to error messages if there are errors.

PARAM   [Required,Array] Input values

    PARAM("VER")   [Boolean] Automatically verify problems

    PARAM ("PRT")   [Boolean] Prompt to print a chart copy on exit

    PARAM ("CLU")   [Boolean] Use Clinical Lexicon

| PARAM ("REV") | [String] Problem display order: C = chronological, R = reverse chronological |
| PARAM ("SDP") | [Boolean] Screen duplicate ICD9 entries. |

Output
A Boolean value signaling if the call was successful or not.

Error Codes Returned
INVPARAM   Invalid parameter passed (PARAM)

2.6.   Reports

PPRBSPEC^GMPLAPI7() – List patients having specified problems on file
This function returns a list of patients having the specified problems on file.

Format
        $$PPRBSPEC ^GMPLAPI7(TARGET,GMPTERM,GMPTEXT,STATUS)

Input Parameters
TARGET   [Required,String] Root name of a local or global array that will receive data.

          @TARGET(patient_name)   [String] problem_status (can be one of "active", "inactive" or "active, inactive")

GMPTERM   [Optional,Numeric] Lexicon term IEN. If specified, only patients having this particular problem will be included.

GMPTEXT   [Optional,String] Provider narrative. If specified, only patients whose problems match this particular provider narrative will be included.

STATUS   [Optional,String] Problem status. Can be any combination of A – active and I - inactive

Output
Number of records returned. If the parameters passed are invalid this function will return -1 as the number of records.

Error Codes Returned
None

PPROBCNT^GMPLAPI7() – List patients having problems on file
This function returns a list of patients having active or inactive problem on file.

Format
        $$PPROBCNT^GMPLAPI7(TARGET)

Input Parameters
TARGET   [Required,String] Root name of a local or global array that will receive data.

          @TARGET   [Numeric]   Number of patients.

@TARGET(patient_name) [String] number_of_active_problems^number_of_inactive_problems

Output

Number of records returned. If the parameters passed are invalid this function will return -1 as the number of records.

Error Codes Returned

None

3. Routines changed in other packages

During the refactoring effort, some routines that belong to packages other than Problem List were modified. A summary of the interactions between those routines and the Problem List package and what has changed is described below.

In most cases, direct access to globals owned by Problem List were replaced with calls to their API equivalents. The RPC Broker entries in Order Entry/Results Reporting packaged served as a basis for the refactoring process, so they were changed to go through the new API's.

We also added a new Mumps code item HIDE (54) to PROTOCOL file. Using this item you can specify condition to hide a Protocol item in List Manager based entry screens. GMPL Verify (1532) now has this item specified so that it is hidden when "Verify Transcribed Problems" is false and vice versa. Protocol record update during changing "Verify Transcribed Problems" is thus removed.

Tests have been developed to exercise these changes. Changes in Order Entry/Results Reporting package are covered by Sikuli CPRS tests as well as M-Unit tests (^ZZRGUTRB). For the other packages, M-Unit tests are available (^ZZRGUTEX).

Automated Info Collection Sys

IBDFBK3 - AICS broker Utilities

PROBNAR^IBDFBK3   Retrieves provider narrative for a problem using PROBNARR^GMPLAPI4. Previously this information was directly obtained from Problem List globals ^AUPNPROB and ^AUTNPOV.

PROBDIA^IBDFBK3   Retrieves ICD9 code id for a problem using DIAG^GMPLAPI4 . Previously this information was directly obtained from^AUPNPROB.

IBDFN11 - ENCOUNTER FORM - (entry points for reprint of dynamic data)

REPRINT^IBDFN11   Retrieves the ICD9 code for a problem using DIAG^GMPLAPI4. Previously this information was directly obtained from Problem List global ^AUPNPROB and ICD Diagnosis global ^ICD9.

Clinical Case Registries

RORHL17 - HL7 PROBLEM LIST: OBR,OBX

EN1^RORHL17   Retrieves the list of patient problems using GETPLIST^GMPLAPI4. Previously the list was retrieved by direct Problem List global access.

LOAD^RORHL17   Gets problem details using DETAIL^GMPLAPI2. Previously the list was retrieved by direct Problem List global access.

Clinical Reminders

PXRMISE - Index size estimating routines.

NEPROB^PXRMISE   Retrieves the number of entries in Problem List using PRBCNT^GMPLAPI4. Previously direct Problem List global access is used to retrieve the same information.

PXRMPROB - Code for Problem List.

OUTPUT^PXRMPROB   Retrieves provider narrative using PROBNARR^GMPLAPI4. Previously direct Problem List global access is used to retrieve the same information.

Kernel

XQOR3 - Process Menus, Protocol Menus

MENU^XQOR3   Added initialization for a new XQORM subscript ("R") from new item 54 from File 101.

XQORM1 - Display selections & prompt

DISP^XQORM1   Added functionality to hide the menu if XQORM("R") is defined.

Order Entry/Results Reporting

ORCPROB - Problem List interface

REMOVE^ORCPROB   Calls DELETE^GMPLUTL2 to remove a problem. Previously it called REMOVE^GMPLUTL2 which included replicated code from roll&scroll for the same purpose.

VERIFY^ORCPROB   Retrieves Verify Required parameter using PARAM^GMPLUTL2 and calls VERIFY^GMPL1 to verify a problem. Previously a direct access to Problem List global ^AUPNPROB is used to check if verification is successful and actual location of the verify parameter in global is used to retrieve the parameter. Now both information is returned as parameters from the Problem List routines.

ORQQPL1 - PROBLEM LIST FOR CPRS GUI

EDLOAD^ORQQPL1   Retrieves problem details using DETAIL^GMPLAPI2 which replaced the previously used entry pointGETFLDS^GMPLEDT3. New entry point is located in the API routines and used by additional entry points.

EDSAVE^ORQQPL1   Saves problem details using UPDATE^GMPLAPI2 which replaced the previously used entry point EN^GMPLSAVE. New entry point is located in the API routines and accepts formal parameters as inputs instead of assumed variables.

ADDSAVE^ORQQPL1   Add a new problem using NEW^GMPLAPI2 which replaced the previously used entry point NEW^GMPSAVE. The new entry point is located in the API routines and accepts formal parameters as inputs instead of assumed variables.

INITUSER^ORQQPL1	Retrieves sites settings using GET^GMPLSITE. These settings previously were retrieved by direct access to Problem List global ^GMPL(125.99). Also retrieves user settings using USERVIEW^GMPLEXT and SERVICE^GMPLEXT. These retriavals were moved to GMPLEXT from VIEW^GMPLX1 and SERVICE^GMPLX1 in an attempt to group access to other packages.
DUP^ORQQPL1	Retrieves active duplicate problems using DUPL^GMPLAPI2 and ACTIVE^GMPLAPI2. DUPL^GMPLAPI2 replaces previously used entry point DUPL^GMPLX. It is in the api routines and used from an additional locations in Problem List code which previously replicated code to retrieve duplicates. ACTIVE^GMPLAPI2 replaced direct access to Problem List global ^AUPNPROB.

ORQQPL2 - RPCs FOR CPRS GUI IMPLEMENTATION

HIST^ORQQPL2	Retrieves Problem List audit history using GETHIST^GMPLHIST, AUDET^GMPLHIST, and DT^GMPLDIST. GETHIST^GMPLHIST replaced direct access to Problem List global ^GMPL(125.8). AUDET^GMPLHIST and DT^GMPLDIST replaced the previously used entry point DT^GMPLHIST to separate wording components of the output from the data retrieval and to use formal parameters instead of assumed variables.
DELETE^ORQQPL2	Removes a problem using the API entry DELETE^GMPLAPI2. Previously the code here was essentially a replication of the code used in Problem List roll&scroll for the same purpose.
REPLACE^ORQQPL2	Replaces a removed problem using the API entry UNDELETE^GMPLAPI4. Previously the code here was essentially a replication of the code used in Problem List roll&scroll for the same purpose.
VERIFY^ORQQPL2	Verifies a problem using VERIFY^GMPLAPI2. Previously the code here was essentially a replication of the codeused in Problem List roll&scroll for the same purpose.
INACT^ORQQPL2	Inactivates a problem using INACTV^GMPLAPI2. Previously the code here was essentially a replication of the code used in Problem List roll&scroll for the same purpose.
GETCOMM^ORQQPL2	Retrieves comments for a problem using NOTES^GMPLAPI3. Previously the code here was essentially a replication of the code used in Problem List roll& scroll for the same purpose.

ORQQPL3 - Problem List RPCs

LIST^ORQQPL3	Retrieves patient problems using LIST^GMPLAPI4 and GET^GMPLSITE. Previously custom code that was essentially repeated elsewhere and direct Problem List global access were used to retrieve the same list.
DELLIST^ORQQPL3	Retrieves removed patient problems using LIST^GMPLAPI4. Previously

custom code that was essentially repeated elsewhere and direct Problem List global access were used to retrieve the same list.

CAT^ORQQPL3	Tranfers contents of user's category list to a temporary global using GETLIST^GMPLAPI1. Previously the transfer was done by direct Problem List global access whose logic is essentially repeated elsewhere in Problem List code.
GETUSLST^ORQQPL3	Retrieves user's category list or default clinic category list using GETULST^GMPLAPI6 and GETCLST^GMPLAPI6. Previously retriaval is done by direct Problem List global access whose logic is essentially repeated elsewhere in Problem List code.
PROB^ORQQPL3	Retrieves problems in a category group using GETCAT^GMPLAPI1. Previously the problems were retrieved by direct Problem List global access whose logic is essentially repeated elsewhere in Proble List code.

PCE Patient Care Encounter

PXCAPL - Validates data from the PCE Device Interface into a call to update Problem List

PROBLEM^PXCAPL	Calls VALID^GMPLAPI4 to find if Problem is in file and calls PATIENT^GMPLAPI4 to find if a certain problem is associated with a certain patient. Previously direct Problem List global access is used to obtain the same information.

PXCAPOV - Validates data from the PCE Device Interface into PCE's PXK format for POV

DIAG^PXCAPOV	Calls VALID^GMPLAPI4 to find if Problem is in file and calls PATIENT^GMPLAPI4 to find if a certain problem is associated with a certain patient. Previously direct Problem List global access is used to obtain the same information.

QUASAR

ACKQUTL6 - Utilities routine

PLIST^ACKQUTL6	Retrieves the problem and its status (active or inactive) with a specific ICD9 code for a patient by calling GETPLIST^GMPLAPI4 and DETAIL^GMPLAPI2 respectively. Previously this information was obtained directly from Problem List global ^AUPNPROB.

4. How to use the code

API

The APIs presented in this paper cover most of the functionality found in the Problem List package. They follow a consistent calling convention that allows for returning rich error messages from the called routines, making them suitable as an interface to be used by external applications.

Assumed Variables

The only assumed variables used in this API are Kernel variables documented in section 2.3.1.3.2 of the SAC. In this version the following assumed variables were used: DUZ,DT,U

Format and conventions of the calls

The conventions used in this API are very similar to those enforced by the RPC Broker. Every tag has at least one parameter, passed by reference that will hold the result of the call. Every function returns a Boolean to signal if the processing was successful or not.

If the return value is 0, the first parameter (RETURN) will be structured as an array containing the errors encountered, numbered from 0 to the error count. The errors have the following form:

> ErrorId^Message

Example

> INVPARAM^Invalid parameter value – GMPIFN

5. Installation

Along with this paper a number of KID Host Files are provided. Each one of them can be used to install the changes done to a specific package. The KID files are as follows:

ACKQ_3.0_260002.KID	QUASAR package patch
GMPL_2.0_260002_RC6.KID	Problem List package patch
IBD_3.0_260002.KID	Automated Info Collection Sys package patch
OR_3.0_260002.KID	Order Entry Results Reporting package patch
PX_1.0_260002.KID	PCE Patient Care Encounter package patch
PXRM_2.0_260002.KID	Clinical Reminders package patch
ROR_1.5_260002.KID	Clinical Case Registries package patch
XU_8.0_260002.KID	Kernel package patch

All the above KID packages depend on GMPL_2.0_260002_RC6.KID so this one should be installed first. Along with the files presented above you can find a multi-build KID package named GMPL_2.0_260002_RC6_MB.KID. This file can be used to install all patches in one go.

The steps required to install the multi-build distribution are outlined below (similar steps could be followed to install the specific package patches one by one if you wish so):

1. From the *Systems Manager Menu* select *Programmer Options…*
2. Select *Kernel Installation & Distribution System*
3. Select *Installation*
4. Select *Load a Distribution*
5. Enter the host file path, for example C:\ GMPL_2.0_260002_RC6_MB.KID and load the distribution.
6. Use the *Install Package(s)* option and select GMPL*2.0*260002*RC6
7. When prompted *Want KIDS to Rebuild Menu Trees Upon Completion of Install?* Respond NO
8. When prompted *Want KIDS to INHIBIT LOGONs during the install?* Respond NO
9. When prompted *Want to DISABLE Scheduled Options, Menu Options, and Protocols?* Respond NO

6. Testing

As part of this project we also developed unit tests for the API entry points and functional tests for Problem List functionality in VistA Roll & Scroll interface. Both types of tests are submitted and became part of the OSEHRA repository https://github.com/OSEHRA/VistA. They are under Packages/Problem List/Testing directory. Documentation for how to set up and run these tests can be found in https://github.com/OSEHRA/VistA/tree/master/Documentation.   Please look for MUnit and RAS test

options.

CPRS functional test plans were developed and executed. The test plans for CPRS testing of the Problem List adhere to the approved guidelines as described in the NIST Test Procedure for §170.302 (c) Maintain up-to-date problem list found at http://healthcare.nist.gov/docs/170.302.c_problemlist_v1.1.pdf. The test plans can be found at http://www.osehra.org/document/cprs-test-cases. These CPRS tests were automated using Sikuli an image recognition testing tool. These tests can be found at http://review.code.osehra.org/#/c/197/. Note that the Sikuli tests are not yet portable – they can not be executed on different platforms without modifications to the image files. As such, the code found at the location above should be used as reference only.

7. GLOSSARY

IEN   Internal Entry Number. The number used to identify an entry within a file. Every record has a unique internal entry number. Often abbreviated as IEN.

8. Conclusions

This paper presented a set of APIs developed as part of the Open Source EHR Refactoring effort. They are meant to be used by both scroll & roll interface and RPC tags and to be accessible by other packages and applications.   They remove direct global access, uncouple the business logic from the user interface elements, minimize code duplication, and remove direct M read/write commands and hard coded text messages.