PHARMACY RE-ENGINEERING (PRE)

APPLICATION PROGRAM INTERFACE (API) MANUAL

Version 1.0
October 2004

(Revised February 2006)

Department of Veterans Affairs
VistA Health Systems Design and Development

Revision History

Each time this manual is updated, the Title Page lists the new revised date and this page describes the changes. Either update the existing manual with the Change Pages document, or replace it with the updated manual.

Note: The Change Pages document may include unedited pages needed for two-sided copying. Only edited pages display the patch number and revision date in the page footer.

Date	Revised Pages	Patch Number	Description
02/06	All pages	PSO*7*229 PSS*1*104 PSN*4*109	Encapsulation Cycle II follow-up patches. Added descriptions to each API detail documentation. Updated detail information for the APIs PSO525AP, PSO59, and PSO52API. Added the PSN5067 API, component FNAME to the PSSDI API, and component ZERO to the PSS50P7 API. Removed component FNAME of the PSS50 API. Updated the Active API table with these additions and deletion. Added PSNDATE INPUT parameter to the DIC^PSNDI and IX^PSNDI components. Added PSSSCRDT and PSSSCRUS input parameters to DIC^PSSDI component. Added PSSDATE and PSSUSAGE input parameters to MIX^PSSDI component. Added a note for the input value PSSVAL to NDC^PSS50. Added Service Code information and notes to components DATA^PSN50P68 and DATA^PSS50. Generated new Table of Contents and Index. L Woodson, (S. Brown, D. Blocker, R. Ruzbacki, M. Vo), H. Whitley

Date	Revised Pages	Patch Number	Description
11/05	All pages	PSO*7*213 PSS*1*101 PSJ*5*163	Encapsulation Cycle II. Updated the *Active APIs* table with the Routines PSJ59P5, PSO5241, PSO525AP, PSO5252, PSO5291, PSO52API, PSO53, PSO59, PSS55, and PSS59P7. Added these routines to their respective sections. Renumbered pages due to the insertion of these routines and their components. Updated the Table of Contents and Index (List of File Numbers and Names). L. Woodson, (S. Brown, D. Blocker), H. Whitley
08/05	5, 52-53, 55, 73	PSN*4*104 PSS*1*97	Added the following APIs to the *Active APIs* table and changed each Inactive flag to Active due to its being activated on FORUM: NDF^PSS50, ZERO^PSS50, ARWS^PSS50 and DRGIEN^PSS50P7.
	13-15, 19-24, 46-91		Added word “name” to LIST = the array name in the Input parameters.
	23		Added a new output parameter.
	24		Expanded the definition of input parameters.
	25, 94		Added periods to indicate input parameters can be an array.
	94		Added new component EN.
			Global change to VistA format.
			L. Woodson/V. Howell, (R. Ruzbacki, M. Vo), M. Mims

Date	Revised Pages	Patch Number	Description
12/04	All	PSS*1*91 PSN*4*94	Identified active APIs from the inactive APIs. Edited text for PSSPK for PSS50 APIs. Updated DATA^PSS50 to add field: #27 CMOP ID. Updated AP^PSS51P1 to add the fields: #2 FREQUENCY (IN MINUTES), #5 TYPE OF SHEDULE, #2.5 MAX DAYS FOR ORDERS, and #8 OUTPATIENT EXPANSION. Updated DATA^PSN50P68 to add the fields: #3 UNITS and #4 NATIONAL FORMULARY NAME. Updated DATA^PSN50625 to add the fields: #.01 NUMBER and #2 SPECIFIC TO GENDER. Updated DATA^PSN50626 to add the fields: #.01 NUMBER and #2 SPECIFIC TO GENDER. Updated ^PSSDI to add DO entry point. K.Stark/V. Howell, (N. Goyal, R. Ruzbacki, L. Teague, M. Vo), M. Mims
10/04		PSS*1*85 PSN*4*80	Original Release of Pharmacy Re-Engineering API Manual. K.Stark, (N. Goyal, R. Ruzbacki, L. Teague, M. Vo), M. Mims

(*This page left blank for two-sided copying.*)

Table of Contents

Introduction

To meet the current and future business needs of the Department of Veterans Affairs (VA) Pharmacy, and to support the overall architecture planned for Health*e*Vet VistA, the existing pharmacy software modules are being re-engineered through new development and the purchase of commercial off-the-shelf products (COTS).

Transition to the new Pharmacy Re-Engineered system eliminates the VistA Pharmacy files currently referenced. Therefore, all existing references (direct global reads/writes, VA FileMan reads/writes) to all pharmacy application files must be replaced with Application Programmer Interfaces (APIs).

This *Pharmacy Re-Engineering (PRE) Application Program Interface (API) Manual* is designed to document the APIs provided by the Pharmacy suite of applications consisting of the following:

- National Drug File (NDF)
- Bar Code Medication Administration (BCMA)
- Outpatient Pharmacy (OP)
- Inpatient Medications (IPM) – Unit Dose and IV
- Inpatient Pharmacy Automatic Replenishment/Ward Stock (AR/WS)
- Controlled Substances (CS)
- Drug Accountability (DA)
- Consolidated Mail Outpatient Pharmacy (CMOP)
- Pharmacy Data Management (PDM)
- Pharmacy Benefits Management (PBM)
- Pharmacy Prescription Practices (PPP)

The plan is to encapsulate in two cycles. Encapsulation Cycle 1 focuses on APIs for National Drug File (NDF) and Pharmacy Data Management (PDM), and only APIs for these two packages will be released as part of Cycle 1. The one active (supported) API for Outpatient Pharmacy that existed prior to encapsulation is also included.

A table of all currently active Pharmacy APIs can be found following the Interfaces section. Each application has a section that begins with a list of all the associated active (supported) and inactive APIs.

If you need the use of an API that has not yet been made active, please email the Pharmacy Re-Engineering team at VHA OI SDD PHARM REENG DEV.

Intended Audience

The intended audiences for this document are Provider Systems, Health Systems Design and Development (HSD&D) staff, National VistA support staff, and Software Quality Assurance (SQA) staff.

Document Conventions

Convention	Description
Field Names	Field names are written in the format: FIELD NAME (FILE NUMBER,FIELD NUMBER) and sometimes as IEN(FILE NUMBER).
Status	Active indicates the supported API has been made active with a Forum DBIA. Inactive indicates the supported API has not yet been made Active with a Forum DBIA.
☛	The pointing hand refers to Notes addressed to the reader.

Acronyms and Definitions

Acronyms

Acronym	Definition
API	Application Program Interface
AR/WS	Inpatient Pharmacy Automatic Replenishment/Ward Stock (namespace PSGW)
BCMA	Bar Code Medication Administration (namespace PSB)
CMOP	Consolidated Mail Outpatient Pharmacy (namespace PSX)
CS	Controlled Substances (namespace PSD)
DA	Drug Accountability/Inventory Interface (namespace PSA)
DBIA	Database Integration Agreement
DSM	Digital Standard MUMPS
EVS	Enterprise VistA Support
HSD&D	Health Systems Design and Development
IEN	Internal Entry Number
IPM	Inpatient Medications – Unit Dose and IV (namespace PSJ, PSG, and PSIV)
MUMPS	Massachusetts General Hospital Utility Multi-Programming System
NDF	National Drug File (namespace PSN)
OP	Outpatient Pharmacy (namespace PSO)
PBM	Pharmacy Benefits Management (namespace PSU)
PDM	Pharmacy Data Management (namespace PSS)
PPP	Pharmacy Prescription Practices (namespace PPP)
PRE	Pharmacy Re-Engineering
SQA	Software Quality Assurance
SRS	Software Requirements Specification
VA	Department of Veterans Affairs
VistA	Veterans Health Information Systems and Technology Architecture
VMS	Virtual Memory System

Definitions

Term	Definition
Encapsulation	Encapsulation provides a form of "logical data independence." We can change the implementation of a type without changing any of the programs using that type. Therefore, the application programs are protected from implementation changes in the lower layers of the system.
HealtheVet VistA	HealtheVet VistA is a collection of information systems, technologies and standards strategically designed to support patients, providers and administrators in the Department of Veterans Affairs' current and future health system.

Interfaces

User Interfaces

The APIs are used by all VistA applications that currently reference, via direct global reads/writes or VA FileMan reads/writes to, PDM and NDF application files. This enables other VistA applications uninterrupted access to pharmacy data during and after the transition to the Pharmacy Re-Engineered system.

Software Interfaces

The following software must be running to support the Pharmacy APIs.

- Kernel V. 8.0
- VA FileMan V. 22.0
- MailMan V. 8.0
- Pharmacy Data Management (PDM) V. 1.0
- National Drug File (NDF) V. 4.0
- Outpatient Pharmacy V. 7.0
- Inpatient Medications V. 5.0

Hardware Interfaces

The APIs run on standard hardware platforms used by the Department of Veterans Affairs Healthcare facilities. These systems consist of standard or upgraded Alpha AXP clusters and run VMS DSM, Cache NT, or Cache VMS.

(*This page left blank for two-sided copying.*)

Active APIs

The following table lists the APIs that are active as of the release of this manual. If you need the use of an API that has not yet been made active, please email the Pharmacy Re-Engineering team using the Outlook mail group VHA OI SDD PHARM REENG DEV.

Active APIs and their components	New DBIA #	Old DBIA #
PSJ59P5		
ALL	4819	1884
PSN50P41		
B	4531	2196
ZERO	4531	149 (149-A)
PSN50P6		
ROOT	4540	N/A
ZERO	4540	302 (302-B), 2016, 2394, 3111, 209, 149 (149-B)
PSN50P65		
C	4543	2393, 149 (149-C)
IEN	4543	2472, 2473
ROOT	4543	N/A
PSN5067		
ALL	4829	2221
PSN50P68		
DATA	4545	3735
PSNAPIS		
B	2531	N/A
CIRN	2531	N/A
CIRN2	2574	N/A
CLASS	2531	N/A
CLASS2	2574	N/A
CLIST	2574	N/A
CMOP	2574	N/A
CPRS	2531	N/A
DCLASS	2531	N/A
DCLCODE	2531	N/A
DDIEX	2574	N/A

Active APIs and their components	New DBIA #	Old DBIA #
PSNAPIS (continued)		
DFSU	2531	N/A
DRUG	2531	N/A
DSS	2531	N/A
FORMI	2574	N/A
FORMR	2574	N/A
FORMRX	2574	N/A
PROD0	2531	N/A
PSJST	2531	N/A
PSPT	2531	N/A
PROD2	2531	N/A
PSA	2531	N/A
PSJDF	2531	N/A
PSJING	2531	N/A
T	2531	N/A
TGTOG	2574	N/A
TGTOG2	2574	N/A
TTOG	2574	N/A
VAGN	2531	N/A
VAP	2531	N/A
PSNDI		
DIC	4554	149 (149-A), (149-C), 618
IX	4554	149 (149-C)
PSO5241		
PEN	4821	2906
PSO525AP		
SUS	4822	786, 4703
PSO5252		
EN	4823	782
PSO5291		
PSO	4824	4223
PSO52API		
PROF	4820	920

Active APIs and their components	New DBIA #	Old DBIA #
RX	4820	53, 90, 523, 591, 678, 780, 785, 824, 885, 1079, 1977, 2020, 2471, 2512, 2513, 2680, 2905, 3792, 4182
PSO53		
PSO	4825	1975, 2511
PSO59		
PSS	4827	1976, 2621
PSOORDER		
EN	1878	N/A
PSS50		
AND	4533	N/A
ARWS	4533	N/A
ASP	4533	N/A
CLOZ	4533	N/A
DATA	4533	25, 106, 221, 276, 302, 553, 477
LAB	4533	273
NDC	4533	N/A
NDF	4533	N/A
VAC	4533	N/A
ZERO	4533	N/A
PSS50P7		
DRGIEN	4662	N/A
ZERO	4662	N/A
PSS51P1		
AP	4546	N/A
ZERO	4546	N/A
PSS51P2		
ALL	4548	N/A
PSS52P6		
ZERO	4549	536, 568
PSS52P7		
ZERO	4550	437, 537
PSS55		
PSS431	4826	117, 552, 2497, 677, 3785, 4181, 533, 2228, 922, 567

Active APIs and their components	New DBIA #	Old DBIA #
PSS55 (continued)		
PSS432	4826	2497, 677, 2475, 2228, 567
PSS433	4826	117, 677, 2228
PSS435	4826	677, 2228, 922
PSS436	4826	2497, 677, 3785, 4181, 535, 2228, 922, 567
PSS59P7		
PSS	4828	2682
PSSDI		
DIC	4551	N/A
DO	4551	N/A
EN	4551	N/A
FNAME	4551	N/A
MIX	4551	N/A

National Drug file (NDF)

The National Drug File application, namespace PSN, provides the following Application Program Interfaces (APIs). If you need the use of an API that has not yet been made active, please email the Pharmacy Re-Engineering team using the Outlook mail group VHA OI SDD PHARM REENG DEV.

New DBIA #	Old DBIA #	Component	Description
2531	N/A	B^PSNAPIS	Returns the closed global root of the "B" cross reference on the VA GENERIC NAME from the VA GENERIC file (#50.6).
	N/A	CIRN^PSNAPIS	Returns specific data from VA GENERIC file (#50.6), NDC/UPN file (#50.67), and VA PRODUCT file (#50.68).
	N/A	CLASS^PSNAPIS	Validates the VA DRUG CLASSIFICATION CODE in the VA DRUG CLASS file (#50.605).
	N/A	CPRS^PSNAPIS	Returns the DOSAGE FORM from DOSAGE FORM file (#50.606), CLASS CODE from VA DRUG CLASS file (#50.605), STRENGTH from VA PRODUCT file (#50.68), and UNITS from DRUG UNITS file (#50.607) for the selected drug.
	N/A	DCLASS^PSNAPIS	Returns the PRIMARY VA DRUG CLASS from VA PRODUCT file (#50.68) and the CLASSIFICATION from VA DRUG CLASS file (#50.605) for the selected drug.
	N/A	DCLCODE^PSNAPIS	Returns the 5 character CLASS CODE from the VA DRUG CLASS file (#50.605) for the selected drug.
	N/A	DFSU^PSNAPIS	Returns the DOSE FORM from DOSAGE FORM file (#50.606), STRENGTH from VA PRODUCT file (#50.68), and UNITS from DRUG UNITS file (#50.607).
	N/A	DRUG^PSNAPIS	Validates if the drug name is either a valid NAME entry in the VA GENERIC file (#50.6) or a valid TRADE NAME entry in the NDC/UPN file (#50.67).
	N/A	DSS^PSNAPIS	Returns the five or seven digit DSS NUMBER from the VA PRODUCT file (#50.68) for the selected drug.
	N/A	PROD0^PSNAPIS	Returns the NAME, DOSAGE FORM, STRENGTH, UNITS, GCNSEQNO, and PREVIOUS GCNSEQNO fields from the VA PRODUCT file (#50.68).
	N/A	PROD2^PSNAPIS	Returns the VA PRINT NAME, VA PRODUCT IDENTIFIER,TRANSMIT TO CMOP fields from the VA PRODUCT file (#50.68) and the NAME from the VA DISPENSE UNIT file (#50.64).

New DBIA #	Old DBIA #	Component	Description
2531 cont.	N/A	PSA^PSNAPIS	Returns Drug names from the DRUG file (#50) for every drug which has the same product name as the name of the drug identified by the passed in NDC.
	N/A	PSJDF^PSNAPIS	Returns the DOSE FORM from DOSAGE FORM file (#50.606).
	N/A	PSJING^PSNAPIS	Returns the IEN and NAME in the DRUG INGREDIENTS file (#50.416), STRENGTH from the VA PRODUCT file (#50.68) and NAME from the DRUG UNITS file (#50.607) for every ingredient in the selected drug.
	N/A	PSJST^PSNAPIS	Returns the STRENGTH from VA PRODUCT file (#50.68).
	N/A	PSPT^PSNAPIS	Returns the IEN and NAME from PACKAGE SIZE file (#50.609) and the IEN and NAME from PACKAGE TYPE file (#50.608).
	N/A	T^PSNAPIS	Returns the closed global root of the "T" cross reference on the VA GENERIC NAME for the NDC/UPN file (#50.67).
	N/A	VAGN^PSNAPIS	Returns the VA GENERIC NAME from the VA GENERIC file (#50.6) for the selected drug.
	N/A	VAP^PSNAPIS	Returns the array VA PRODUCT NAME from the VA PRODUCT file (#50.68), DOSE FORM from DOSAGE FORM file (#50.606), and CLASS CLASSIFICATION from VA DRUG CLASS file (#50.605).
2574	N/A	CIRN2^PSNAPIS	Returns the list of the NDCs from the NDC/UPN file (#50.67) for a product.
	N/A	CLASS2^PSNAPIS	Returns the first CODE and CLASSIFICATION fields for the VA DRUG CLASS file (#50.605).
	N/A	CLIST^PSNAPIS	Returns a list of IEN and CODE from the VA DRUG CLASS file (#50.605).
	N/A	CMOP^PSNAPIS	Returns the 5 character CMOP CODE for the VA PRODUCT file (#50.68).
	N/A	DDIEX^PSNAPIS	Returns the indication from VA PRODUCT file (#50.68) if a drug is exempted from drug-drug interaction order checking.
	N/A	FORMI^PSNAPIS	Gets the NATIONAL FORMULARY INDICATOR for a VA PRODUCT file (#50.68).
	N/A	FORMR^PSNAPIS	Returns the indication if the NATIONAL FORMULARY RESTRICTION is existent for the VA PRODUCT file (#50.68).
	N/A	FORMRX^PSNAPIS	Returns the list NATIONAL FORMULARY RESTRICTION for the VA PRODUCT file (#50.68).
	N/A	TGTOG^PSNAPIS	Returns the IEN from the VA GENERIC file (#50.6) if the passed in NAME is either a valid NAME from the VA GENERIC file (#50.6) or a valid TRADE NAME from the NDC/UPN file (#50.67).

New DBIA #	Old DBIA #	Component	Description
2574 cont.	N/A	TGTOG2^PSNAPIS	Returns the IEN in VA GENERIC file (#50.6) and GENERIC NAME is the .01 field of that entry for all entries for which the passed in NAME is a partial or exact match to either the TRADE NAME or the GENERIC NAME.
	N/A	TTOG^PSNAPIS	Returns the IEN and GENERIC NAME for every entry in the VA GENERIC file (#50.6) which matches the TRADE NAME.
4531	2196	B^PSN50P41	Returns the file root of the "B" cross-reference on the DRUG INGREDIENTS file (#50.416).
	149 (149-A)	ZERO^PSN50P41	Checks for the existence of the entry in the DRUG INGREDIENTS file (#50.416). The NAME field (#.01) and the PRIMARY INGREDIENT field (#2) of the DRUG INGREDIENTS file (#50.416) will be returned.
4540	N/A	ROOT^PSN50P6	Returns the root of the zero node of the VA GENERIC file (#50.6).
	302 (302-B), 2016, 2394, 3111, 209, 149 (149-B)	ZERO^PSN50P6	Returns the zero node of the VA GENERIC file (#50.6) in the array defined by the calling application. The "B" cross-reference in the format of ^PS(50.6,"B",NAME(50.6,.01),IEN(50.6)) will be used.
4543	2393, 149 (149-C)	C^PSN50P65	Returns the CODE field (#.01) and CLASSIFICATION field (#1) of the VA DRUG CLASS file (#50.605). The "C" cross-reference in the format of ^PS(50.605,"C", CLASSIFICATION(50.605,1),IEN(50.605)) will be used.
	2472, 2473	IEN^PSN50P65	Returns the CODE field (#.01) and CLASSIFICATION field (#1) of the VA DRUG CLASS file (#50.605). The "B" cross-reference in the format of ^PS(50.605,"B", CODE(50.605,.01),IEN(50.605)) will be used.
	N/A	ROOT^PSN50P65	Returns the global root of the zero node of the "C" cross-reference of the VA DRUG CLASS file (#50.605).
4829	2221	ALL^PSN5067	Returns data from the NDC/UPN file (#50.67).
4545	3735	DATA^PSN50P68	Returns the NAME field (#.01), NDC LINK TO GCNSEQNO field (#13) and CS FEDERAL SCHEDULE field (#19) of the VA PRODUCT file (#50.68) in the array defined by the calling application.
4554	149 (149-A), (149-C), 618	DIC^PSNDI	This API will accept input values and return output values as defined by VA FileMan Lookup call ^DIC.
	149 (149-C)	IX^PSNDI	This API will accept input values and return output values as defined by VA FileMan call IX^DIC.

New DBIA #	Old DBIA #	Component	Description
Inactive		DATA^PSN50612	Returns data from the NATIONAL DRUG TRANSLATION file (#50.612) in the array defined by the calling application.
Inactive		DATA^PSN50625	Returns data from the WARNING LABEL-ENGLISH file (#50.625) in the array defined by the calling application.
Inactive		DATA^PSN50626	Returns data from the WARNING LABEL-SPANISH file (#50.626) in the array defined by the calling application.
Inactive		DATA^PSN50627	Returns data from the WARNING LABEL MAP file (#50.627) in the array defined by the calling application.
Inactive		APD^PSN50P41	Returns the NAME field (#.01) and PRIMARY INGREDIENTS field (#2) of the DRUG INGREDIENTS file (#50.416) in the array defined by the calling application. The “APD” cross-reference in the format of ^PS(50.416, “APD”,DRUG IDENTIFIER(50.4161,.01),IEN(50.416) from the PRIMARY INGREDIENTS(50.416,2)) will be used.
Inactive		APS^PSN50P41	Returns the DRUG IDENTIFIER field (#.01) of the DRUG IDENTIFIER multiple of the DRUG INGREDIENTS file (#50.416) in the array defined by the calling application. The “APS” cross-reference in the format of ^PS(50.416, “APS”,PRIMARY INGREDIENT(50.416,2),IEN(50.416)) will be used.
Inactive		ID^PSN50P41	Returns the DRUG IDENTIFIER field (#.01) of the DRUG IDENTIFIER multiple of the DRUG INGREDIENTS file (#50.416) in the array defined by the calling application.
Inactive		NAME^PSN50P41	Returns fields in the zero node in the DRUG INGREDIENTS file (#50.416) in the array defined by the calling application. The “P” cross-reference in the format of ^PS(50.416,“P”,NAME(50.416,.01),IEN(50.416)) will be used.
Inactive		ALL^PSN50P67	Returns all of the data from the DRUG UNITS file (#50.607) in the array defined by the calling application.
Inactive		FORM^PSN50P68	Returns the NATIONAL FORMULARY NAME field of the VA PRODUCT file (#50.68).
Inactive		ALL^PSN56	Returns all of the data from the DRUG INTERACTION file (#56) in the array defined by the calling application.

New DBIA #	Old DBIA #	Component	Description
Inactive		IEN^PSN56	Returns the IEN(s) of the DRUG INTERACTION file (#56) when passed the DRUG IDENTIFIER sub-field (#.01) of the DRUG IDENTIFIER multiple (#1) from the DRUG INGREDIENTS file (#50.416) for INGREDIENT 1 field (#1) and for INGREDIENT 2 field (#2) in the DRUG INTERACTION file (#56). The "APD" cross-reference in the format of ^PS(56,"APD",DRUG IDENTIFIER(50.4161,.01) of the INGREDIENT 1(56,1),DRUG IDENTIFIER(50.4161,.01) of the INGREDIENT 2(56,2),IEN of the DRUG INTERACTION file (#56)) will be used.
Inactive		DIE^PSNDI	This API will accept input values and return output values as defined by VA FileMan Edit Data call ^DIE.

PSNAPIS API – API for NDF files

Component: PSA

Returns the Drug names from the DRUG file (#50) for every drug which has the same product name as the name of the drug identified by the passed in NDC.

Status: Active

```
S X=$$PSA^PSNAPIS(ndc,.array)
Input:
      ndc
      array
Where:
   ndc = 12-digit National Drug Code (NDC)
   array = defined by the calling application
Output:
   array(IEN) = name
   $$PSA = number of entries in the array
Where:
   array(IEN) = IEN in that file for every drug that has the same product name
   as the name of the drug identified by the NDC
   name = name of the drug identified by the NDC
```

Component: PSJING

Returns the IEN and NAME in the DRUG INGREDIENTS file (#50.416), STRENGTH from the VA PRODUCT file (#50.68) and NAME from the DRUG UNITS file (#50.607) for every ingredient in the selected drug.

Status: Active

```
S X=$$PSJING^PSNAPIS(p1,p3,.array)
Input:
   pl
   p3
   array = defined by the calling application
```

```
Where:
   p1 = first piece of the "ND" node in the DRUG file (#50) for a selected
   drug
   p3 = third piece of the "ND" node in the DRUG file (#50) for a selected
   drug
   array = defined by the calling application
Output:
   array(IEN) = IEN^INGREDIENT
   $$PSJING = number of entries in the array
Where:
   IEN = IEN in the DRUG INGREDIENTS file (#50.416)
   INGREDIENT = name for every ingredient for the selected drug
```

Component: PSJDF

Returns the DOSE FORM from DOSAGE FORM file (#50.606).

Status: Active

```
S X=$$PSJDF^PSNAPIS(p1,p3)
Input:
   pl
   p3
Where:
   pl = first piece of the "ND" node in the DRUG file (#50) for a selected
   drug
   p3 = third piece of the "ND" node in the DRUG file (#50) for a selected
   drug
Output:
   $$PSJDF = IEN^DOSE
Where:
   IEN = IEN in the DOSAGE FORM file (#50.606)
   DOSE = name of the Dosage Form for a selected drug
```

Component: PSJST

Returns the STRENGTH from VA PRODUCT file (#50.68).

Status: Active

```
S X=$$PSJST^PSNAPIS(p1,p3)
Input:
   pl
   p3
Where:
   p1 = first piece of the "ND" node in the DRUG file (#50) for a selected
   drug
   p3 = third piece of the "ND" node in the DRUG file (#50) for a selected
   drug returns
Output:
   $$PSJST = IEN^STRENGTH
Where:
   IEN = 1
   STRENGTH = the strength of a selected drug
```

Component: CLASS

Validates the VA DRUG CLASSIFICATION CODE in the VA DRUG CLASS file (#50.605).

Status: Active

```
S X=$$CLASS^PSNAPIS(class)
Input:
   class
Where:
   class = 5-character VA Drug Classification Code
   (e.g. CN103)
Output:
      $$CLASS
Where $$CLASS is:
   1 = VA Drug Classification Code is valid
   0 = VA Drug Classification Code is not valid
```

Component: DRUG

Validates if the drug name is either a valid NAME entry in the VA GENERIC file (#50.6) or a valid TRADE NAME entry in the NDC/UPN file (#50.67).

Status: Active

```
S X=$$DRUG^PSNAPIS(drug name)
Input:
   drug name
Where:
   drug name = VA Generic Name from the VA GENERIC file
   (#50.6) or the Trade Name from the NDC/UPN file (#50.67)
Output:
   $$DRUG
Where $$DRUG is:
   1 = VA Generic Name or Trade Name is valid
   0 = VA Generic Name or Trade Name is not valid
```

Component: PROD0

Returns the NAME, DOSAGE FORM, STRENGTH, UNITS, GCNSEQNO, and PREVIOUS GCNSEQNO fields from the VA PRODUCT file (#50.68).

Status: Active

```
S X=$$PROD0^PSNAPIS(p1,p3)
Input:
   pl
   p3
Where:
   p1 = first piece of the "ND" node in the DRUG file (#50) for a selected
   drug
   p3 = third piece of the "ND" node in the DRUG file (#50) for a selected
   drug
Output:
   $$PROD0 = VA PRODUCT NAME^DOSAGE FORM^STRENGTH^UNITS
   ^^^PMIS^PREVIOUS PMIS
Where:
   VA PRODUCT NAME = name of VA Product in the VA PRODUCT file (#50.68) for a
   selected drug
   DOSAGE FORM = name of the Dosage Form for a selected drug
   STRENGTH = the strength of a selected drug
   UNITS = number of units
   PMIS = this contains the GCNSEQNO Code, which relates to mapping of the
   Patient Medication Information Sheet (PMIS) and Rx warning labels
   PREVIOUS PMIS = this contains the PREVIOUS GCNSEQNO Code, which relates to
   mapping of the PMIS and Rx warning labels
```

☛ Note: DBIA was not updated with PMIS project. The PMIS ENTRY field is no longer called USP GENERIC NAME in the VA PRODUCT file (#50.68). The source is now FDB and not USP.

Component: PROD2

Returns the VA PRINT NAME, VA PRODUCT IDENTIFIER, TRANSMIT TO CMOP fields from the VA PRODUCT file (#50.68) and the NAME from the VA DISPENSE UNIT file (#50.64).

Status: Active

```
S X=$$PROD2^PSNAPIS(p1,p3)
Input:
   pl
   p3
Where:
   p1 = first piece of the "ND" node in the DRUG file (#50) for a selected
   drug
   p3 = third piece of the "ND" node in the DRUG file (#50) for a selected
   drug
Output:
   $$PROD2 = VA PRINT NAME^VA PRODUCT IDENTIFIER^TRANSMIT TO CMOP^VA DISPENSE
   UNIT
Where:
   VA PRINT NAME = print name for a VA Product
   VA PRODUCT IDENTIFIER = 5-character CMOP ID
   TRANSMIT TO CMOP = Y or N
   VA DISPENSE UNIT = VA Dispense Unit for a VA Product
```

Component: DCLASS

Returns the PRIMARY VA DRUG CLASS from VA PRODUCT file (#50.68) and the CLASSIFICATION from VA DRUG CLASS file (#50.605) for the selected drug.

Status: Active

```
S X=$$DCLASS^PSNAPIS(p1,p3)
Input:
   pl
   p3
Where:
   p1 = first piece of the "ND" node in the DRUG file (#50)
   p3 = third piece of the "ND" node in the DRUG file (#50)
Output:
   $$DCLASS = PRIMARY VA DRUG CLASS^VA DRUG CLASSIFICATION
Where:
   PRIMARY VA DRUG CLASS = IEN for the VA DRUG CLASS file (#50.605)
   VA DRUG CLASSIFICATION = VA DRUG CLASSIFICATION description for a selected
   drug, (e.g. CYANIDE ANTIDOTES)
```

Component: DCLCODE

Returns the 5 character CLASS CODE from the VA DRUG CLASS file (#50.605) for the selected drug.

Status: Active

```
S X=$$DCLCODE^PSNAPIS(p1,p3)
Input:
   pl
   p3
Where:
   p1 = first piece of the "ND" node in the DRUG file (#50) for a selected
   drug
   pl = third piece of the "ND" node in the DRUG file (#50) for a selected
   drug
Output:
   $$DCLCODE = 5-character VA Drug Class Code for a selected drug, (e.g.
   AD200)
```

Component: VAGN

Returns the VA GENERIC NAME from the VA GENERIC file (#50.6) for the selected drug.

Status: Active

```
S X=$$VAGN^PSNAPIS(p1)
Input:
   pl
Where:
   p1 = first piece of the "ND" node in the DRUG file (#50) for a selected
   drug
Output:
   $$VAGN = VA GENERIC NAME from the VA GENERIC file (#50.6)
```

Component: DFSU

Returns the DOSE FORM from DOSAGE FORM file (#50.606), STRENGTH from VA PRODUCT file (#50.68), and UNITS from DRUG UNITS file (#50.607).

Status: Active

```
S X=$$DFSU^PSNAPIS(p1,p3)
Input:
   pl
   p3
Where:
   p1 = first piece of the "ND" node in the DRUG file (#50) for a selected
   drug
   p3 = third piece of the "ND" node in the DRUG file (#50) for a selected
   drug
Output:
   $$DFSU = DFIEN^DOSE^STIEN^STRENGTH^UNIEN^UNITS
Where:
   DFIEN = IEN of the Dosage Form
   DOSE = name of the Dosage Form
   STIEN = IEN of the strength
   STRENGTH = the strength
   UNIEN = IEN of the units
   UNITS = the unit
```

Component: VAP

Returns the array VA PRODUCT NAME from the VA PRODUCT file (#50.68), DOSE FORM from DOSAGE FORM file (#50.606), and CLASS CLASSIFICATION from VA DRUG CLASS file (#50.605).

Status: Active

```
S X=$$VAP^PSNAPIS(da,.array)
Input:
   da
   array
Where:
   da = IEN in the VA GENERIC file (#50.6)
   array = array defined by the calling application
   $$VAP = number of entries in the array
Output:
   array(ien) = IEN^VA PRODUCT NAME^DFIEN^DOSE^CLIEN^CLASS
Where:
   IEN = IEN from the VA PRODUCT file (#50.68)
   VA PRODUCT NAME = Name of the VA Product
   DFIEN = IEN of the Dosage Form
   DOSE = name of the Dosage Form
   CLIEN = IEN of the class from the VA DRUG CLASS file (#50.605)
   CLASS = the 5-character VA Drug Class Code for the VA Product associated
   with a selected VA Generic Drug
```

Component: PSPT

Returns the IEN and NAME from PACKAGE SIZE file (#50.609) and the IEN and NAME from PACKAGE TYPE file (#50.608).

Status: Active

```
S X=$$PSPT^PSNAPIS(p1,p3,array)
Input:
   pl
   p3
   array
Where:
   p1 = first piece of the "ND" node in the DRUG file (#50) for a selected
   drug
   p3 = third piece of the "ND" node in the DRUG file (#50) for a selected
   drug
   array = defined by the calling application
Output:
      (for each package size and type combination)
   array(PSIEN^PTIEN) = PSIEN^PSIZE^PTIEN^PTYPE
   $$PSPT = number of entries in the array
Where:
   PSIEN = IEN of the package size
   PSIZE = package size
   PTIEN = IEN of the package type
   PTYPE = package type
```

Component: DSS

Returns the five or seven digit DSS NUMBER from the VA PRODUCT file (#50.68) for the selected drug.

Status: Active

```
S X=$$DSS^PSNAPIS(p1,p3)
Input:
   pl
   P3
Where:
   p1 = first piece of the "ND" node in the DRUG file (#50) for a selected
   drug
   p3 = third piece of the "ND" node in the DRUG file (#50) for a selected
   drug
Output:
   $$DSS = 5 or 7-digit DSS identifier for a selected drug
```

Component: CPRS

Returns the DOSAGE FORM from DOSAGE FORM file (#50.606), CLASS CODE from VA DRUG CLASS file (#50.605), STRENGTH from VA PRODUCT file (#50.68), and UNITS from DRUG UNITS file (#50.607) for the selected drug.

Status: Active

```
S X=$$CPRS^PSNAPIS(p1,p3)
Input:
   pl
   p3
Where:
   p1 = first piece of the "ND" node in the DRUG file (#50) for a selected
   drug
   p3 = third piece of the "ND" node in the DRUG file (#50) for a selected
   drug
Output:
   $$CPRS = DOSAGE FORM^CLASS CODE^STRENGTH^UNITS
Where:
   DOSAGE FORM = name of Dosage Form
   CLASS CODE = 5-character VA Drug Class Code for a selected drug
   STRENGTH = the strength
   UNITS = the unit
```

Component: CIRN

Returns specific data from the VA GENERIC file (#50.6), NDC/UPN file (#50.67), and VA PRODUCT file (#50.68).

Status: Active

```
S X=$$CIRN^PSNAPIS(ndc,.array)
Input:
   ndc
   array
Where:
   ndc = 12-digit National Drug Code
   array = defined by the calling application
```

```
Output:
   $$CIRN = number of entries in the array
Where:
   array = NDC
   array(0) = IEN in the VA GENERIC file (#50.6)
   array(1) = VA GENERIC NAME
   array(2) = Dose Form IEN^^CLASS IEN
   array(3) = strength
   array(4) = units IEN
   array(5) = package size IEN
   array(6) = package type pointer
   array(7) = NDC^MANUFACTURER^TRADE NAME^VA PRODUCT NAME POINTER^ROUTE OF
   ADMINISTRATION
      If the NDC is not a valid entry, then array is returned as
      "000000000000", and seven elements of array returned as null
```

Component: B

Returns the closed global root of the "B" cross reference on the VA GENERIC NAME from the VA GENERIC file (#50.6).

Status: Active

```
S X=$$B^PSNAPIS returns the closed global root of the
"B" cross-reference field in the VA GENERIC file (#50.6)
```

Component: T

Returns the closed global root of the "T" cross reference on the VA GENERIC NAME from the NDC/UPN file (#50.67).

Status: Active

```
S X=$$T^PSNAPIS returns the closed global root of the
"T" cross-reference field in the NDC/UPN file (#50.67)
```

Component: TTOG

Returns the IEN and GENERIC NAME for every entry in the VA GENERIC file (#50.6) which matches the TRADE NAME.

Status: Active

```
S X=$$TTOG^PSNAPIS(TRADE,.LIST)
Input:
   TRADE
   LIST
Where:
   TRADE = a free text trade name
   LIST = array name defined by the calling application
Output:
   LIST(IEN) = IEN^GENERIC NAME
   $$TTOG = number of entries in the LIST
Where:
   IEN = IEN from the VA GENERIC file (#50.6)
   GENERIC NAME = NAME field (#.01) in the VA GENERIC file (#50.6) for every
   entry that has a defined Trade Name
```

Component: CLIST

Returns a list of IEN and CODE from the VA DRUG CLASS file (#50.605).

Status: Active

```
S X=$$CLIST^PSNAPIS(DA,.LIST)
Input:
   DA
   LIST
Where:
   DA = IEN from the VA GENERIC file (#50.6)
   LIST = array name defined by the calling application
Output:
   LIST(IEN) = IEN^CLASS
   $$CLIST = number of entries in the LIST
Where:
   IEN = IEN from the VA DRUG CLASS file (#50.605)
   CLASS = 5-character VA Drug Class Code for every class associated with that
   entry in the VA GENERIC file (#50.6)
```

Component: TGTOG

Returns the IEN from the VA GENERIC file (#50.6) if the passed in NAME is either a valid NAME from the VA GENERIC file (#50.6) or a valid TRADE NAME from the NDC/UPN file (#50.67).

Status: Active

```
S X=$$TGTOG^PSNAPIS(NAME)
Input:
   NAME
Where:
   NAME = a free text entry VA Generic Name from the VA GENERIC file (#50.6)
   or Trade Name from the NDC/UPN file (#50.67)
Output:
   $$TGTOG = IEN or 0
Where:
   IEN = if the "B" cross-reference existed for the VA Generic Name, return
   the IEN from the VA GENERIC file (#50.6), otherwise return the IEN from the
   VA PRODUCT file (#50.68)
   0 = not a valid VA Generic Name or Valid Trade name
```

Component: TGTOG2

Returns the IEN in VA GENERIC file (#50.6) and GENERIC NAME is the .01 field of that entry for all entries for which the passed in NAME is a partial or exact match to either the TRADE NAME or the GENERIC NAME.

Status: Active

```
S X=$$TGTOG2^PSNAPIS(NAME,.LIST)
Input:
   NAME
   LIST
Where:
   NAME = a free text entry VA Generic Name from the VA GENERIC file (#50.6)
   or Trade Name from the NDC/UPN file (#50.67)
   LIST = array name defined by the calling application
```

```
Output:
   LIST(IEN) = IEN^GENERIC NAME
   $$TGTOG2 = number of entries in the LIST
Where:
   IEN = IEN in VA GENERIC file (#50.6)
   GENERIC NAME = the NAME field (#.01) in the VA GENERIC file (#50.6) of that
   entry for all entries where the NAME is a partial or an exact match to
   either the Trade Name or the VA Generic Name entered
```

Component: CIRN2

Returns the list of the NDCs from the NDC/UPN file (#50.67) for a product.

Status: Active

```
S X=$$CIRN2^PSNAPIS(p1,p3,.LIST)
Input:
   p1
   p3
   LIST
Where:
   p1 = the first piece of the "ND" node in the DRUG file (#50)
   p3 = the third piece of the "ND" node in the DRUG file (#50)
   LIST = array name defined by the calling application
Output:
   LIST(J) = NDC
   $$CIRN2 = number of entries in the LIST
Where:
   J = a simple index
   NDC = 12-character National Drug Code for all NDCs association with the VA
   Product specified by p1 and p3
```

Component: CLASS2

Returns the first CODE and CLASSIFICATION fields for the VA DRUG CLASS file (#50.605).

Status: Active

```
S X=$$CLASS2^PSNAPIS(IEN)
Input:
   IEN
Where:
   IEN = IEN in the VA DRUG CLASS file (#50.605)
Output:
   $$CLASS2 = CODE^CLASSIFICATION, (e.g. AD900^ANTIDOTES/DETERRENTS, OTHER)
Where:
   CODE = CODE field (#.01) of the VA DRUG CLASS file (#50.605)
   CLASSIFICATION = CLASSIFICATION field (#1) of the VA DRUG CLASS file
   (#50.605)
```

Component: FORMI

Gets the NATIONAL FORMULARY INDICATOR for a VA PRODUCT file (#50.68).

Status: Active

```
S X=$$FORMI^PSNAPIS(p1,p3)
Input:
   p1
   p3
Where:
   p1 = first piece of the "ND" node in the DRUG file (#50) for a selected
   entry
   p3 = third piece of the "ND" node in the DRUG file (#50) for a selected
   entry
Output:
   $$FORMI = 1 or 0
Where:
   1 = the item is on the National Formulary
   0 = the item is not on the National Formulary
```

Component: FORMR

Returns the indication if the NATIONAL FORMULARY RESTRICTION is existent for the VA PRODUCT file (#50.68).

Status: Active

```
S X=$$FORMR^PSNAPIS(p1,p3)
Input:
   p1
   p3
Where:
   p1 = first piece of the "ND" node in the DRUG file (#50) for a selected
   entry
   p3 = third piece of the "ND" node in the DRUG file (#50) for a selected
   entry
Output:
   $$FORMR = 1 or 0
Where:
   1 = there are National Formulary restrictions
   0 = there are no National formulary restrictions
```

Component: CMOP

Returns the 5 character CMOP CODE for the VA PRODUCT file (#50.68).

Status: Active

```
S X=$$CMOP^PSNAPIS(CODE)
Input:
   CODE
Where:
   CODE = 5-character CMOP ID
Output:
   $$CMOP
Where:
   $$CMOP = corresponding VA Product Name if the CODE is valid
   $$CMOP = null if the CODE is not valid
```

Component: FORMRX

Returns the list NATIONAL FORMULARY RESTRICTION for the VA PRODUCT file (#50.68).

Status: Active

```
S X=$$FORMRX^PSNAPIS(DA,K,.LIST)
Input:
   DA
   K
   LIST
Where:
   DA = first piece of the "ND" node in the DRUG file (#50) for a selected
   drug
   K = third piece of the "ND" node in the DRUG file (#50) for a selected drug
   LIST = an array defined by the calling application
Output:
   $$FORMRX = 1 or 0
Where:
   1 = there are National Formulary restrictions
   0 = there are no National formulary restrictions
   LIST = word processing field describing the restrictions, if restrictions
   exist for the VA Product
```

Component: DDIEX

Returns the indication from VA PRODUCT file (#50.68) if a drug is exempted from drug-drug interaction order checking.

Status: Active

```
S X=$$DDIEX^PSNAPIS(VAR1,VAR2)
Input:
   VAR1
   VAR2
Where:
   VAR1 = first piece of the "ND" node in the DRUG file (#50) for a selected
   drug
   VAR2 = third piece of the "ND" node in the DRUG file (#50) for a selected
   drug
Output:
   $$DDIEX = 1 or 0
Where:
   1 = indicates that a drug has been exempted from drug-drug interaction
   order checking
   0 = indicates that a drug is not exempted from drug-drug interaction order
   checking
```

PSN50P41 API – Drug Ingredients file (#50.416)

Component: B

Returns the file root of the “B” cross-reference on the DRUG INGREDIENTS file (#50.416).

Status: Active

```
S X=$$B^PSN50P41
Input:
   N/A
Output:
   $$B
Where:
   $$B = "^PS(50.416,"B")" the file root of the "B" cross-reference on the
   DRUG INGREDIENTS file (#50.416)
```

Component: ZERO

Checks for the existence of the entry in the DRUG INGREDIENTS file (#50.416). The NAME field (#.01) and the PRIMARY INGREDIENT field (#2) of the DRUG INGREDIENTS file (#50.416) will be returned.

Status: Active

```
ZERO^PSN50P41(PSNIEN,PSNFT,PSNFL,LIST)
Input:
   PSNIEN
   PSNFT
   PSNFL
   LIST
Where:
   PSNIEN = IEN of entry in DRUG INGREDIENTS file (#50.416) [optional]
   PSNFT = the NAME field (#.01) of the DRUG INGREDIENTS file (#50.416) (a
   value of "??" may be used) [optional]
   PSNFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   LIST = defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSNIEN,.01) = NAME (50.416,.01)
   ^TMP($J,LIST,PSNIEN,2) = PRIMARY INGREDIENT (50.416,2)^NAME (50.416,.01)
   (The PRIMARY INGREDIENT field (#2) stores the IEN of the DRUG INGREDIENTS
   file (#50.416) that point to other entry in this file)
   ^TMP($J,LIST,PSNIEN,3) = INACTIVATION DATE (50.416,3)^External format (ex:
   SEP 12,1999)
   ^TMP($J,LIST,"B",NAME,PSNIEN) = ""
```

Component: NAME

Returns fields in the zero node in the DRUG INGREDIENTS file (#50.416) in the array defined by the calling application. The “P” cross-reference in the format of ^PS(50.416,“P”,NAME(50.416,.01),IEN(50.416)) will be used.

Status: Inactive

```
NAME^PSN50P41(PSNFT,LIST)
Input:
   PSNFT
   LIST
Where:
   PSNFT = NAME field (#.01) of the DRUG INGREDIENTS file (#50.416)
   (a value of "??" may be used) [required]
   LIST = defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSNIEN,.01) = NAME (50.416,.01)
   ^TMP($J,LIST,PSNIEN,2) = PRIMARY INGREDIENT (50.416,2)^NAME (50.416,.01)
   (The PRIMARY INGREDIENT field (#2) stores the IEN of the DRUG INGREDIENTS
   file (#50.416) that point to other entry in this file)
   ^TMP($J,LIST,"P",NAME,PSNIEN) = ""
Where:
   PSNIEN is IEN of entry in the DRUG INGREDIENTS file (#50.416)
```

Component: ID

Returns the DRUG IDENTIFIER field (#.01) of the DRUG IDENTIFIER multiple of the DRUG INGREDIENTS file (#50.416) in the array defined by the calling application.

Status: Inactive

```
ID^PSN50P41(PSNIEN,PSNFT,LIST)
Input:
   PSNIEN
   PSNFT
   LIST
Where:
   PSNIEN = IEN of entry in DRUG INGREDIENTS file (#50.416) [optional]
   PSNFT = NAME field (#.01) of the DRUG INGREDIENTS file (#50.416) (a value
   of "??" may be used) [optional]
   LIST = defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSNIEN,.01) = NAME (50.416,.01)
   ^TMP($J,LIST,PSSIEN,"ID",0) = Total entries returned for this sub-file or -
   1^NO DATA FOUND
   ^TMP($J,LIST,PSNIEN,"ID",PSN(1),.01) = DRUG IDENTIFIER (50.4161,.01)
   ^TMP($J,LIST,"B",NAME,PSNIEN) = ""
Where:
   PSN(1) is the IEN of entry in the DRUG IDENTIFIER multiple
```

Component: APS

Returns the DRUG IDENTIFIER field (#.01) of the DRUG IDENTIFIER multiple of the DRUG INGREDIENTS file (#50.416) in the array defined by the calling application. The “APS” cross-reference in the format of ^PS(50.416, “APS”,PRIMARY INGREDIENT(50.416,2),IEN(50.416)) will be used.

Status: Inactive

```
APS^PSN50P41(PSNPI,LIST)
Input:
   PSNPI
   LIST
Where:
   PSNPI = PRIMARY INGREDIENT field (#2) of the DRUG INGREDIENTS file
   (#50.416) [required]
   LIST = defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSNIEN,.01) = NAME (50.416,.01)
   ^TMP($J,LIST,PSNIEN,"ID",0) = Total entries returned for this sub-file or -
   1^NO DATA FOUND
   ^TMP($J,LIST,PSNIEN,"ID",PSN(1),.01) = DRUG IDENTIFIER (50.4161,.01)
   ^TMP($J,LIST,"APS",NAME,PSNIEN) = ""
Where:
   PSNIEN is the IEN of entry in the DRUG INGREDIENTS file (#50.416)
   PSN(1) is the IEN of entry in the DRUG IDENTIFIER multiple
```

Component: APD

Returns the NAME field (#.01) and PRIMARY INGREDIENTS field (#2) of the DRUG INGREDIENTS file (#50.416) in the array defined by the calling application. The “APD” cross-reference in the format of ^PS(50.416, “APD”,DRUG IDENTIFIER(50.4161,.01), IEN(50.416) from the PRIMARY INGREDIENTS(50.416,2)) will be used.

Status: Inactive

```
APD^PSN50P41(PSNID,LIST)
Input:
   PSNID
   LIST
Where:
   PSNID = DRUG IDENTIFIER field (#.01) of the DRUG IDENTIFIER multiple of the
   DRUG INGREDIENTS file (#50.416) [optional]
   LIST = defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSNIEN,.01) = NAME(50.416,.01)
   ^TMP($J,LIST,PSNIEN,2) = PRIMARY INGREDIENT (50.416,2)^NAME (50.416,.01)
   (The PRIMARY INGREDIENT field (#2)
   stores the IEN of the DRUG INGREDIENTS file (#50.416) that point to other
   entry in this file)
   ^TMP($J,LIST,PSNIEN,"ID",0) = Total entries returned for this sub-file or -
   1^NO DATA FOUND
   ^TMP($J,LIST,PSNIEN,"ID",PSN(1),.01) = DRUG IDENTIFIER (50.4161,.01)
   ^TMP($J,LIST,"APD",NAME,PSNIEN) = ""
Where:
   PSNIEN is the IEN of entry in the DRUG INGREDIENTS file (#50.416)
   PSN(1) is the IEN of entry in the DRUG IDENTIFIER multiple
```

PSN50P6 API - VA Generic file (#50.6)

Component: ROOT

Returns the root of the zero node of the VA GENERIC file (#50.6).

Status: Active

```
S X=$$ROOT^PSN50P6
Input:
   N/A
Output:
   $$ROOT
Where:
   $$ROOT is "^PSNDF(50.6," the file root of the VA GENERIC file (#50.6)
```

Component: ZERO

Returns the zero node of the VA GENERIC file (#50.6) in the array defined by the calling application. The “B” cross-reference in the format of ^PS(50.6,“B” ,NAME(50.6,.01),IEN(50.6)) will be used.

Status: Active

```
ZERO^PSN50P6(PSNIEN,PSNFT,PSNFL,PSNX,LIST)
Input:
   PSNIEN
   PSNFT
   PSNFL
   PSNX
   LIST
Where:
   PSNIEN = IEN of entry in VA GENERIC file (#50.6) [optional]
   PSNFT = NAME field (#.01) of the VA GENERIC file (#50.6)
   (a value of "??" may be used) [optional]
   PSNFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   PSNX = 1 for exact match flag [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSNIEN,.01) = NAME (50.6,.01)
   ^TMP($J,LIST,PSNIEN,1) = INACTIVATION DATE (50.6,1)^External format (ex:
   SEP 12,1999)
   ^TMP($J,LIST,"B",NAME,PSNIEN) = ""
```

PSN50P65 API - VA Drug Class file (#50.605)

Component: ROOT

Returns the global root of the zero node of the “C” cross-reference of the VA DRUG CLASS file (#50.605).

Status: Active

```
S X=$$ROOT^PSN50P65(PSNC)
Input:
   PSNC = If "1" the global root of the "C" cross-reference is returned
   otherwise, the global root of the zero node is returned
Output:
   $$ROOT
Where:
   $$ROOT = "^PS(50.605,"C")" if PSNC is passed in as 1
   $$ROOT = "^PS(50.605," if PSNC is null
```

Component: C

Returns the CODE field (#.01) and CLASSIFICATION field (#1) of the VA DRUG CLASS file (#50.605). The “C” cross-reference in the format of ^PS(50.605,“C”,CLASSIFICATION(50.605,1),IEN(50.605)) will be used.

Status: Active

```
C^PSN50P65(PSNIEN,PSNFT,LIST)
Input:
   PSNIEN
   PSNFT
   LIST
Where:
   PSNIEN = IEN of entry in VA DRUG CLASS file (#50.605) [optional]
   PSNFT = the CLASSIFICATION field (#1) of the VA DRUG CLASS file (#50.605)
   (a value of "??" may be used) [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSNIEN,.01) = CODE (50.605,.01)
   ^TMP($J,LIST,PSNIEN,1) = CLASSIFICATION (50.605,1)
   ^TMP($J,LIST,PSNIEN,2) = PARENT CLASS (50.605,2)^CODE (50.605,.01)
   ^TMP($J,LIST,PSNIEN,3) = TYPE (50.605,3)
   ^TMP($J,LIST,"C",CODE,PSNIEN) = ""
```

☛ Note: The "C" cross-reference in the format of ^PS(50.605,"C",CLASSIFICATION (50.605,1),IEN(50.605)) will be used for the lookup.

Component: IEN

Returns the CODE field (#.01) and CLASSIFICATION field (#1) of the VA DRUG CLASS file (#50.605). The “B” cross-reference in the format of ^PS(50.605,“B”, CODE(50.605,.01),IEN(50.605)) will be used.

Status: Active

```
IEN^PSN50P65(PSNIEN,PSNFT,LIST)
Input:
   PSNIEN
   PSNFT
   LIST
Where:
   PSNIEN = IEN of entry in VA DRUG CLASS file (#50.605) [optional]
   PSNFT = CODE field (#.01) of VA DRUG CLASS file (#50.605) (a value of "??"
   may be used) [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSNIEN,.01) = CODE (50.605,.01)
   ^TMP($J,LIST,PSNIEN,1) = CLASSIFICATION (50.605,1)
   ^TMP($J,LIST,"B",CODE,PSNIEN) = ""
```

PSN50P67 API – Drug Units file (#50.607)

Component: ALL

Returns all of the data from the DRUG UNITS file (#50.607) in the array defined by the calling application.

Status: Inactive

```
ALL^PSN50P67(PSNIEN,PSNFT,PSNFL,LIST)
Input:
   PSNIEN
   PSNFT
   PSNFL
   LIST
Where:
   PSNIEN = IEN of entry in DRUG UNITS file (#50.607) [optional]
   PSNFT = NAME field (#.01) of the DRUG UNITS file (#50.607)
   (a value of "??" may be used) [optional]
   PSNFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSNIEN,.01) = NAME (50.607,.01)
   ^TMP($J,LIST,PSNIEN,1) = INACTIVATION DATE (50.607,1)^External format (ex:
   SEP 12,1999)
   ^TMP($J,LIST,"B",NAME,PSNIEN) = ""
```

PSN50612 API – National Drug Translation file (#50.612)

Component: DATA

Returns data from the NATIONAL DRUG TRANSLATION file (#50.612) in the array defined by the calling application.

Status: Inactive

```
DATA^PSN50612(PSNIEN,LIST)
Input:
   PSNIEN
   LIST
Where:
   PSNIEN = IEN of entry in NATIONAL DRUG TRANSLATION file (#50.612)
   [required]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSNIEN,10) = END (50.612,10)
   ^TMP($J,LIST,PSNIEN,11) = LAST UNMATCHED DRUG CLASSED(50.612,11)
```

PSN50625 API – Warning Label – English file (#50.625)

Component: DATA

Returns data from the WARNING LABEL-ENGLISH file (#50.625) in the array defined by the calling application.

Status: Inactive

```
DATA^PSN50625 (PSNIEN,LIST)
Input:
   PSNIEN
   LIST
Where:
   PSNIEN = IEN of entry in WARNING LABEL-ENGLISH file (#50.625) [required]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSNIEN,.01) = NUMBER (50.625,.01)
   ^TMP($J,LIST,PSNIEN,2) = SPECIFIC TO GENDER (50.625,2)^External format for
   the set of codes
   ^TMP($J,LIST,PSNIEN,"WLT",0) = Total entries returned for this sub-file or
   -1^NO DATA FOUND
   ^TMP($J,LIST,PSNIEN,"WLT",PSN(1),.01) = WARNING LABEL TEXT (50.6251,.01)
Where:
   PSN(1) is the IEN of entry in the WARNING LABEL TEXT multiple
```

PSN50626 API – Warning Label – Spanish file (#50.626)

Component: DATA

Returns data from the WARNING LABEL-SPANISH file (#50.626) in the array defined by the calling application.

Status: Inactive

```
DATA^PSN50626(PSNIEN,LIST)
Input:
   PSNIEN
   LIST
Where:
   PSNIEN = IEN of entry in WARNING LABEL-SPANISH file (#50.626) [required]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSNIEN,.01) = NUMBER (50.626,.01)
   ^TMP($J,LIST,PSNIEN,2) = SPECIFIC TO GENDER (50.626,2)^External format for
   the set of codes
   ^TMP($J,LIST,PSNIEN,"WLT",0) = Total entries returned for this sub-file or
   -1^NO DATA FOUND
   ^TMP($J,LIST,PSNIEN,"WLT",PSN(1),.01) = SPANISH WARNING LABEL TEXT
   (50.6261,.01)
Where:
   PSN(1) is the IEN of entry in the SPANISH WARNING LABEL TEXT multiple
```

PSN50627 API – Warning Label Map file (#50.627)

Component: DATA

Returns data from the WARNING LABEL MAP file (#50.627) in the array defined by the calling application.

Status: Inactive

```
DATA^PSN50627(PSNIEN,PSNFT,LIST)
Input:
   PSNIEN
   PSNFT
   LIST
Where:
   PSNIEN = IEN of entry in WARNING LABEL MAP file (#50.627) [optional]
   PSNFT = NAME field (#.01) of the WARNING LABEL MAP file (#50.627)
   (a value of "??" may be used) [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSNIEN,.01) = GCNSEQNO (50.627,.01)
   ^TMP($J,LIST,PSNIEN,1) = NUMBER (50.627,1)
   ^TMP($J,LIST,PSNIEN,.2) = PRIORITY (50.627,2)
   ^TMP($J,LIST,"B",GCNSEQNO,PSNIEN) = ""
```

PSN5067 API – NDC/UPN file (#50.67)

Component: ALL

Returns data from the NDC/UPN file (#50.67).

Status: Active

```
ALL^PSN5067(PSNIEN,PSNFT,PSNFL,LIST)
Input:
   PSNIEN
   PSNFT
   PSNFL
   LIST
Where:
   PSNIEN = IEN of entry in NDC/UPN file (#50.67) If a value is passed in for
   PSNIEN, then any value passed in for the PSNFT parameter will be ignored.
   [optional]
   PSNFT = TRADE NAME field (#4) of the NDC/UPN file (#50.67) (a value of "??"
   may be used). This value will be ignored if a value for PSNIEN is passed
   in. [optional]
   PSNFL = Inactivation date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSNIEN,.01) = SEQUENCE NUMBER (50.67,.01)
   ^TMP($J,LIST,PSNIEN,1) = NDC (50.67,1)
   ^TMP($J,LIST,PSNIEN,2) = UPN (50.67,2)
   ^TMP($J,LIST,PSNIEN,3) = MANUFACTURER (50.67,3)^NAME (55.95,.01)
   ^TMP($J,LIST,PSNIEN,4) = TRADE NAME (50.67,4)
   ^TMP($J,LIST,PSNIEN,5) = VA PRODUCT NAME (50.67,5)^NAME (50.68,.01)
   ^TMP($J,LIST,PSNIEN,7) = INACTIVATION DATE (50.67,7)^External format (ex:
   SEP 12,1999)
   ^TMP($J,LIST,PSNIEN,8) = PACKAGE SIZE (50.67,8)^NAME (50.609,.01)
   ^TMP($J,LIST,PSNIEN,9) = PACKAGE TYPE (50.67,9)^NAME (50.608,.01)
   ^TMP($J,LIST,PSNIEN,10) = OTX/RX INDICATOR (50.67,10)^ External format for
   the set of codes
   ^TMP($J,LIST,"B",NAME,PSNIEN) = ""
```

☛ Note: Please note the value NAME in the “B” cross reference returned is from the TRADE NAME field (#4). This is not a required field.

PSN50P68 API – VA Product file (#50.68)

Component: DATA

Returns the NAME field (#.01), NDC LINK TO GCNSEQNO field (#13) and CS FEDERAL SCHEDULE field (#19) of the VA PRODUCT file (#50.68) in the array defined by the calling application.

Status: Active

```
DATA^PSN50P68(PSNIEN,PSNFT,LIST)
Input:
   PSNIEN
   PSNFT
   LIST
Where:
   PSNIEN = IEN of entry in VA PRODUCT file (#50.68) [optional]
   PSNFT = NAME field (#.01) of the VA PRODUCT file (#50.68) (a value of "??"
   may be used) [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSNIEN,.01) = NAME (50.68,.01)
   ^TMP($J,LIST,PSNIEN,.05) = VA GENERIC NAME (50.68,.05)^NAME (50.6,.01)
   ^TMP($J,LIST,PSNIEN,3) = UNITS (50.68,3)^NAME (50.607,.01)
   ^TMP($J,LIST,PSNIEN,4) = NATIONAL FORMULARY NAME (50.68,4)
   ^TMP($J,LIST,PSNIEN,11) = GCNSEQNO (50.68,11)
   ^TMP($J,LIST,PSNIEN,12) = PREVIOUS GCNSEQNO (50.68,12)
   ^TMP($J,LIST,PSNIEN,13) = NDC LINK TO GCNSEQNO (50.68,13)
   ^TMP($J,LIST,PSNIEN,19) = CS FEDERAL SCHEDULE (50.68,19)^External format
   for the set of codes
   ^TMP($J,LIST,PSNIEN,2000) = SERVICE CODE (50.68,2000)
   ^TMP($J,LIST,"B",NAME,PSNIEN) = ""
   ☛Note: If there is no data in the SERVICE CODE field (#2000) of the VA
   PRODUCT file (#50.68), the value 600000 will be returned for the SERVICE
   CODE.
```

Component: FORM

Returns the NATIONAL FORMULARY NAME field of the VA PRODUCT file (#50.68).

Status: Inactive

```
S X=$$FORM^PSN50P68(PSNIEN)
Input:
   PSNIEN
Where:
   PSNIEN = IEN of entry in VA PRODUCT file (#50.68) [required]
Output:
   $$FORM
Where:
   $$FORM = the NATIONAL FORMULARY NAME field (#4) of the VA PRODUCT file
   (#50.68)
```

PSN56 API – Drug Interaction file (#56)

Component: ALL

Returns all of the data from the DRUG INTERACTION file (#56) in the array defined by the calling application.

Status: Inactive

```
ALL^PSN56(PSNIEN,PSNFT,LIST)
Input:
   PSNIEN
   PSNFT
   LIST
Where:
   PSNIEN = IEN of entry in DRUG INTERACTION file (#56) [optional]
   PSNFT = NAME field (#.01) of DRUG INTERACTION file (#56)
   (a value of "??" may be used) [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSNIEN,.01) = NAME (56,.01)
   ^TMP($J,LIST,PSNIEN,1) = INGREDIENT 1 (56,1)^NAME (50.416,.01)
   ^TMP($J,LIST,PSNIEN,2) = INGREDIENT 2 (56,2)^NAME (50.416,.01)
   ^TMP($J,LIST,PSNIEN,3) = SEVERITY (56,3)^External format for the set of
   codes
   ^TMP($J,LIST,PSNIEN,4) = NATIONALLY ENTERED (56,4)^External format for the
   set of codes
   ^TMP($J,LIST,PSNIEN,5) = TOTAL INDEXES (56,5)
   ^TMP($J,LIST,PSNIEN,6) = LOCALLY EDITED (56,6)
   ^TMP($J,LIST,PSNIEN,7) = INACTIVATION DATE (56,7)^External format (ex: SEP
   12,1999)
   ^TMP($J,LIST,"B",NAME,PSNIEN) = ""
```

Component: IEN

Returns the IEN(s) of the DRUG INTERACTION file (#56) when passed the DRUG IDENTIFIER sub-field (#.01) of the DRUG IDENTIFIER multiple (#1) from the DRUG INGREDIENTS file (#50.416) for INGREDIENT 1 field (#1) and for INGREDIENT 2 field (#2) in the DRUG INTERACTION file (#56). The "APD" cross-reference in the format of ^PS(56,"APD",DRUG IDENTIFIER(50.4161,.01) of the INGREDIENT 1(56,1),DRUG IDENTIFIER(50.4161,.01) of the INGREDIENT 2(56,2),IEN of the DRUG INTERACTION file (#56)) will be used.

Status: Inactive

```
IEN^PSN56(PSNING1,PSNING2,PSNFL,LIST)
Input:
   PSNING1
   PSNING2
   PSNFL
   LIST
Where:
   PSNING1 = DRUG IDENTIFIER (#50.4161,.01) associated with INGREDIENT 1
   [required]
   PSNING2 = DRUG IDENTIFIER (#50.4161,.01) associated with INGREDIENT 2
   [required]
   PSNFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   LIST = array name defined by calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSNIEN,.01) = NAME (56,.01)
   ^TMP($J,LIST,PSNIEN,1) = INGREDIENT 1 (56,1)^NAME (50.416,.01)
   ^TMP($J,LIST,PSNIEN,2) = INGREDIENT 2 (56,2)^NAME (50.416,.01)
   ^TMP($J,LIST,PSNIEN,3) = SEVERITY (56,3)^External format for the set of
   codes
   ^TMP($J,LIST,PSNIEN,7) = INACTIVATION DATE (56,7)^External format (ex: SEP
   12,1999)
   ^TMP($J,LIST,"APD",NAME,PSNIEN) = ""
Where:
   PSNIEN is IEN of entry in the DRUG INTERACTION file (#56)
```

PSNDI API – FileMan Calls

Component: DIC - Lookup

This API will accept input values and return output values as defined by VA FileMan Lookup call ^DIC.

Status: Active

```
DIC^PSNDI(PSNFILE,PSNPACK,.DIC,.X,DLAYGO,PSNDATE)
Input:
   PSNFILE
   PSNPACK
   PSNDATE
Where:
   PSNFILE = File number used for validation of access [required]
   PSNPACK = Name space of the calling application. Ex: PSJ for Inpatient
   Meds, PSO for Outpatient.  PSNPACK is used to check if write access is
   allowed [optional]
   See VA FileMan Programmer Manual for ^DIC call, for DIC, X, AND DLAYGO
   input definitions
   PSNDATE = Inactivation Date. If the file has an Inactivation Date, then any
   entry with an Inactivation Date on or before PSNDATE will not be returned.
Output:
   PSNDIY will return null if the values for PSNFILE and PSNPACK are valid (it
   will return -1 if conditions were not met)
Where:
   See VA FileMan Programmer Manual for ^DIC output definition
   DLAYGO should only be passed in if the calling application has this type of
   access through another Integration Agreement
```

Component: IX - Lookup/Add

This API will accept input values and return output values as defined by VA FileMan call IX^DIC.

Status: Active

```
IX^PSNDI(PSNFILE,PSNPACK,.DIC,D,.X,DLAYGO,PSNDATE)
Input:
   PSNFILE
   PSNPACK
   PSNDATE
Where:
   PSNFILE = File number used for validation of access [required]
   PSNPACK = Name space of the calling application. Ex: PSJ for Inpatient
   Meds, PSO for Outpatient.  PSNPACK is used to check if write access is
   allowed [optional]
   See VA FileMan Programmer Manual for IX^DIC call, for DIC, D, X, and DLAYGO
   input definitions
   PSNDATE = Inactivation Date. If the file has an Inactivation Date, then any
   entry with an Inactivation Date on or before PSNDATE will not be returned.
Output:
   PSNDIY will return null if the values for PSNFILE and PSNPACK are valid (it
   will return -1 if conditions were not met)
Where:
   See VA FileMan Programmer Manual for IX^DIC output definition
   DLAYGO should only be passed in if the calling application has this type of
   access through another Integration Agreement
```

Component: DIE - Edit Data

This API will accept input values and return output values as defined by VA FileMan Edit Data call ^DIE.

Status: Inactive

```
DIE^PSNDI(PSNFILE,PSNPACK,.DIE,.DA,DR,DIDEL)
Input:
   PSNFILE
   PSNPACK
Where:
   PSNFILE = File number used for validation of access [required]
   PSNPACK = Name space of the calling application. Ex: PSJ for Inpatient
   Meds, PSO for Outpatient.  PSNPACK is used to check if write access is
   allowed [optional]
   See VA FileMan Programmer Manual for ^DIE call, for DIE, DA, DR, and DIDEL
   input definitions
Output:
   PSNDIY will return null if the values for PSNFILE and PSNPACK are valid (it
   will return -1 if conditions were not met)
Where:
   See VA FileMan Programmer Manual for ^DIE output definition
```

(*This page left blank for two-sided copying.*)

Bar Code Medication Administration (BCMA)

At this time there are no supported APIs for the Bar Code Medication Administration application.

(This page left blank for two-sided copying.)

Outpatient Pharmacy (OP)

The Outpatient Pharmacy application, namespace PSO, provides the following Application Program Interfaces (APIs). If you need the use of an API that has not yet been made active, please email the Pharmacy Re-Engineering team using the Outlook mail group VHA OI SDD PHARM REENG DEV.

New DBIA #	Old DBIA #	Component	Description
1878	N/A	EN^PSOORDER	Open subscription for Outpatient Pharmacy prescription data.
4820	920	PROF^PSO52API	This component returns PRESCRIPTION file (#52) data elements by using the “P” & “A” cross reference (^PS(55,DFN,"P","A")) on the PHARMACY PATIENT file (#55).
	53, 90, 523, 591, 678, 780, 785, 824, 885, 1079, 1977,2020, 2471, 2512, 2513, 2680, 2905, 3792, 4182	RX^PSO52API	This API has defined one entry point by parameter passing, to return prescription data from the PRESCRIPTION file (#52).
4821	2906	PEN^PSO5241	Returns data elements for the PENDING OUTPATIENT ORDERS file (#52.41).
4822	786, 4703	SUS^PSO525AP	Returns data elements for the RX SUSPENSE file (#52.5).
4823	782	EN^PSO5252	Returns data elements for the CLOZAPINE PRESCRIPTION OVERRIDES file (#52.52).
4824	4223	PSO^PSO5291	Returns data elements for the TPB ELIGIBILITY file (#52.91).
4825	1975, 2511	PSO^PSO53	Returns data elements for the RX PATIENT STATUS file (#53).
4827	1976, 2621	PSS^PSO59	Returns data elements for the OUTPATIENT SITE file (#59).

PSO52API API – PRESCRIPTION file (#52)

Component: PROF

This component returns PRESCRIPTION file (#52) data elements by using the “P” & “A” cross reference (^PS(55,DFN,"P","A")) on the PHARMACY PATIENT file (#55).

Status: Active

```
PROF^PSO52API(DFN,LIST,SDATE,EDATE)

Input:
   DFN
   LIST
   SDATE
   EDATE
Where:
   DFN = IEN from the PATIENT file (#2) [required]
   LIST = Subscript name used in the ^TMP global [required]
   SDATE = Starting Expiration Date [optional]
   EDATE = Ending Expiration Date [optional]

   ☛ Note: If patient does not have any prescriptions, but has archived
   prescriptions, a ^TMP($J,LIST,DFN,”ARC”)=”PATIENT HAS ARCHIVED PRESCRIPTION”
   shall be defined. If SDATE is passed in, suspended RXs starting with this
   date shall be returned (^PS(55,DFN,”P”,”A”)). If EDATE is passed in, RXs up
   to and including this date shall be returned (^PS(55,DFN,”P”,”A”)).
Output:
   ^TMP($J,LIST,DFN,0)=Total entries returned by patient or -1^NO PRESCRIPTION
   DATA FOUND
   ^TMP($J,LIST,DFN,IEN,.01)=RX #(52,.01)^IEN
   ^TMP($J,LIST,DFN,IEN,1)=ISSUE DATE (52,1)^External Format (ex: Sep. 12,
   1999)
   ^TMP($J,LIST,DFN,IEN,2)=PATIENT (52,2)^NAME (2,.01)
   ^TMP($J,LIST,DFN,IEN,3)=PATIENT STATUS (52,3)^NAME (53,.01)
   ^TMP($J,LIST,DFN,IEN,4)=PROVIDER (52,4)^NAME (200,.01)
   ^TMP($J,LIST,DFN,IEN,5)=CLINIC (52,5)^NAME (44,.01)
   ^TMP($J,LIST,DFN,IEN,6)=DRUG (52,6)^GENERIC NAME (50,.01)
   ^TMP($J,LIST,DFN,IEN,6.5)=TRADE NAME (52,6.5)
   ^TMP($J,LIST,DFN,IEN,7)=QTY (52,7)
   ^TMP($J,LIST,DFN,IEN,8)=DAYS SUPPLY (52,8)
   ^TMP($J,LIST,DFN,IEN,9)=# OF REFILLS (52,9)
   ^TMP($J,LIST,DFN,IEN,10.3)=ORDER CONVERTED (52,10.3) ^External Format for
   the Set of Codes
   ^TMP($J,LIST,DFN,IEN,10.6)=COPIES (52,10.6)
   ^TMP($J,LIST,DFN,IEN,11)=MAIL/WINDOW (52,11)^External Format for the Set of
   Codes
   ^TMP($J,LIST,DFN,IEN,16)=ENTERED BY (52,16)^NAME (200,.01)
   ^TMP($J,LIST,DFN,IEN,17)=UNIT PRICE OF DRUG (52,17)
   ^TMP($J,LIST,DFN,IEN,100)=STATUS (52,100)^External Format for the Set of
   Codes
   ^TMP($J,LIST,DFN,”ARC”)=”PATIENT HAS ARCHIVED PRESCRIPTION”
   ^TMP($J,LIST,”B”,RX #,IEN)=””
```

Component: RX

This API has defined one entry point by parameter passing, to return prescription data from the PRESCRIPTION file (#52).

Status: Active

```
RX^PSO52API(DFN,LIST,IEN,RX#,NODE,SDATE,EDATE)
```

```
Input:
   DFN
   LIST
   IEN
   RX#
   NODE
   SDATE
   EDATE
Where:
   DFN = IEN from the PATIENT file (#2) [required]
   LIST = Subscript name used in the ^TMP global [required]
   IEN = Internal prescription number [optional]
   RX# = RX # field (#.01) of the PRESCRIPTION file (#52) [optional]
   NODE = will be passed “null” or as a string of data containing some or all
   of the following 0,1,2,3,R,I,P,O,T,L,S,M,C,A,and/or ST [optional]
   SDATE = Start Date [optional]
   EDATE = End Date [optional]
```

☛ Note**:** This API has defined one entry point by parameter passing, to return data from available data nodes from the PRESCRIPTION file (#52). The variable “NODE” shall be passed null or as a string of data separated by commas. Every field from the PRESCRIPTION file (#52) shall not be returned, though the string must be defined to contain some or all of the following: 0, 2, 3, R, I, P, O, T, L, S, M, C, A, ST. Each section describes what is returned by each string variable. TRADE NAME field (#6.5) (piece 1 of the "TN" node), from the PRESCRIPTION file (#52) that is associated with DRUG NAME field (#6) will be returned as part of the 0 node data, if populated. The status and “B” cross-reference of **valid** prescriptions is always returned.

0 - returns the Zero node
2 - returns the Zero and Two nodes
3 - returns the Two and Three nodes
R - returns all refills for a prescription (If a specific refill is wanted, this variable must be passed in the following format: “R^^n” (indicating which refill is wanted))
I - returns Integrated Billing (IB) data ([^null, O or R^n] if 2nd piece is defined as null IB data for both original and all refills are returned. If 2nd piece is defined as ‘O’ only IB data for the original fill is returned. If 2nd piece is defined as ‘R’ refill data is returned and the 3rd piece is null, all refill IB data are returned, or if the 3rd piece has a number it shall return the IB node for that refill only.)
P - returns all partial fills for a prescription
O - returns the Orderable Item for a prescription
T - returns the TPB node
L - returns the Label multiple
S - returns the SAND node
M - returns the medication instructions (SIG)
C - returns CMOP data
A - returns the Activity Logs for a prescription
ST - returns the status of the prescription

☛Note: If NODE is passed in null, all possible nodes are returned.

☛Note: If an invalid parameter is passed into the NODE variable the following node shall be returned: ^TMP($J,LIST,DFN,IEN,"INVALID REQUEST",NODE)=Invalid Data Requested

Output: (**If node = 0 (zero)**)

```
   ^TMP($J,LIST,DFN,0)=Total entries returned by patient or -1^NO DATA FOUND
   ^TMP($J,LIST,DFN,IEN,.01)=RX #(52,.01)
   ^TMP($J,LIST,DFN,IEN,1)=ISSUE DATE (52,1)^External Format (ex: Sep. 12,
   1999)
   ^TMP($J,LIST,DFN,IEN,2)=PATIENT (52,2)^NAME (2,.01)
   ^TMP($J,LIST,DFN,IEN,3)=PATIENT STATUS (52,3)
   ^TMP($J,LIST,DFN,IEN,4)=PROVIDER (52,4)^NAME (200,.01)
   ^TMP($J,LIST,DFN,IEN,5)=CLINIC (52,5) ^NAME(44,.01)
   ^TMP($J,LIST,DFN,IEN,6)=DRUG (52,6)^GENERIC NAME(50,.01)
   ^TMP($J,LIST,DFN,IEN,6.5)=TRADE NAME (52,6.5)
   ^TMP($J,LIST,DFN,IEN,7)=QTY (52,7)
   ^TMP($J,LIST,DFN,IEN,8)=DAYS SUPPLY (52,8)
   ^TMP($J,LIST,DFN,IEN,9)=# OF REFILLS (52,9)
   ^TMP($J,LIST,DFN,IEN,10.3)=ORDER CONVERTED (52,10.3)^External Format for
   the Set of Codes
   ^TMP($J,LIST,DFN,IEN,10.6)=COPIES (52,10.6)
   ^TMP($J,LIST,DFN,IEN,11)=MAIL/WINDOW (52,11) ^External Format for the Set
   of Codes
   ^TMP($J,LIST,DFN,IEN,16)=ENTERED BY (52,16)^NAME (200,.01)
   ^TMP($J,LIST,DFN,IEN,17)=UNIT PRICE OF DRUG (52,17)
```

Output: (**If node = 2**)

```
   ^TMP($J,LIST,DFN,0)=Total entries returned by patient or -1^NO DATA FOUND
   ^TMP($J,LIST,DFN,IEN,.01)=RX #(52,.01)
   ^TMP($J,LIST,DFN,IEN,1)=ISSUE DATE (52,1)^External Format (ex: Sep. 12,
   1999)
   ^TMP($J,LIST,DFN,IEN,2)=PATIENT (52,2)^NAME (2,.01)
   ^TMP($J,LIST,DFN,IEN,3)=PATIENT STATUS (52,3)
   ^TMP($J,LIST,DFN,IEN,4)=PROVIDER (52,4)^NAME (200,.01)
   ^TMP($J,LIST,DFN,IEN,5)=CLINIC (52,5) ^NAME(44,.01)
   ^TMP($J,LIST,DFN,IEN,6)=DRUG (52,6)^GENERIC NAME(50,.01)
   ^TMP($J,LIST,DFN,IEN,6.5)=TRADE NAME (52,6.5)
   ^TMP($J,LIST,DFN,IEN,7)=QTY (52,7)
   ^TMP($J,LIST,DFN,IEN,8)=DAYS SUPPLY (52,8)
   ^TMP($J,LIST,DFN,IEN,9)=# OF REFILLS (52,9)
   ^TMP($J,LIST,DFN,IEN,10.3)=ORDER CONVERTED (52,10.3)^External Format for
   the Set of Codes
   ^TMP($J,LIST,DFN,IEN,10.6)=COPIES (52,10.6)
   ^TMP($J,LIST,DFN,IEN,11)=MAIL/WINDOW (52,11) ^External Format for the Set
   of Codes
   ^TMP($J,LIST,DFN,IEN,16)=ENTERED BY (52,16)^NAME (200,.01)
   ^TMP($J,LIST,DFN,IEN,17)=UNIT PRICE OF DRUG (52,17)
   ^TMP($J,LIST,DFN,IEN,20)=DIVISION (52,20)^NAME(59,.01)
   ^TMP($J,LIST,DFN,IEN,21)=LOGIN DATE (52,21)^External Format (ex: Sep. 12,
   1999)
   ^TMP($J,LIST,DFN,IEN,22)=FILL DATE (52,22) ^External Format (ex: Sep. 12,
   1999)
   ^TMP($J,LIST,DFN,IEN,23)=PHARMACIST (52,23)^NAME (200,.01)
   ^TMP($J,LIST,DFN,IEN 24)=LOT # (52,24)
   ^TMP($J,LIST,DFN,IEN 25)=DISPENSED DATE (52,25)^External Format (ex: Sep.
   12, 1999)
   ^TMP($J,LIST,DFN,IEN,26)=EXPIRATION DATE (52,26)^External Format (ex: Sep.
   12, 1999)
   ^TMP($J,LIST,DFN,IEN,27)=NDC (52,27)
   ^TMP($J,LIST,DFN,IEN,28)=MANUFACTURER (52,28)
   ^TMP($J,LIST,DFN,IEN,29)=DRUG EXPIRATION DATE (52,29)^External Format (ex:
   Sep. 12, 1999)
   ^TMP($J,LIST,DFN,IEN,30)=GENERIC PROVIDER (52,30)
   ^TMP($J,LIST,DFN,IEN,31)=RELEASE DATE/TIME (52,31)^External Format (ex:
   Sep. 12, 1999)
   ^TMP($J,LIST,DFN,IEN 32.1)=RETURNED TO STOCK (52,32.1)^External Format (ex:
   Sep. 12, 1999)
   ^TMP($J,LIST,DFN,IEN,32.2)=REPRINT (52,32.2)^External Format for the Set of
   Codes
   ^TMP($J,LIST,DFN,IEN,32.3)=BINGO WAIT TIME (52,32.3)
   ^TMP($J,LIST,DFN,IEN,100)=STATUS (52,100) ^External Format for the Set of
   Codes
   ^TMP($J,LIST,DFN,IEN,104)=VERIFYING PHARMACIST (52,104)^NAME (200,.01)
   ^TMP($J,LIST,”B”,RX#,IEN)=””
```

Output: (**If node = 3**)

```
   ^TMP($J,LIST,DFN,0)=Total entries returned by patient or -1^NO DATA FOUND
   ^TMP($J,LIST,DFN,IEN,12)=REMARKS (52,12)
   ^TMP($J,LIST,DFN,IEN,20)=DIVISION (52,20)^NAME (59,.01)
   ^TMP($J,LIST,DFN,IEN,21)=LOGIN DATE (52,21)^External Format (ex: Sep. 12,
   1999)
   ^TMP($J,LIST,DFN,IEN,22)=FILL DATE (52,22)^External Format (ex: Sep. 12,
   1999)
   ^TMP($J,LIST,DFN,IEN,23)=PHARMACIST (52,23)^NAME (200,.01)
   ^TMP($J,LIST,DFN,IEN 24)=LOT # (52,24)
   ^TMP($J,LIST,DFN,IEN 25)=DISPENSED DATE (52,25)^External Format (ex: Sep.
   12, 1999)
   ^TMP($J,LIST,DFN,IEN,26)=EXPIRATION DATE (52,26)^External Format (ex: Sep.
   12, 1999)
   ^TMP($J,LIST,DFN,IEN,26.1)=CANCEL DATE (52,26.1)^External Format (ex: Sep.
   12, 1999)
   ^TMP($J,LIST,DFN,IEN,27)=NDC (52,27)
   ^TMP($J,LIST,DFN,IEN,28)=MANUFACTURER (52,28)
   ^TMP($J,LIST,DFN,IEN,29)=DRUG EXPIRATION DATE (52,29)^External Format (ex:
   Sep. 12, 1999)
   ^TMP($J,LIST,DFN,IEN,30)=GENERIC PROVIDER (52,30)
   ^TMP($J,LIST,DFN,IEN,31)=RELEASE DATE/TIME (52,31)^External Format (ex:
   Sep. 12, 1999)
   ^TMP($J,LIST,DFN,IEN 32.1)=RETURNED TO STOCK (52,32.1)^External Format (ex:
   Sep. 12, 1999)
   ^TMP($J,LIST,DFN,IEN,32.2)=REPRINT (52,32.2) ^External Format for the Set
   of Codes
   ^TMP($J,LIST,DFN,IEN,32.3)=BINGO WAIT TIME (52,32.3)
   ^TMP($J,LIST,DFN,IEN,34.1)=DRUG ALLERGY INDICATION (52,34.1) ^External
   Format for the Set of Codes
   ^TMP($J,LIST,DFN,IEN,100)=STATUS (52,100) ^External Format for the Set of
   Codes
   ^TMP($J,LIST,DFN,IEN,101)=LAST DISPENSED DATE (52,101)^External Format (ex:
   Sep. 12, 1999)
   ^TMP($J,LIST,DFN,IEN,102)=NEXT POSSIBLE FILL (52,102)^External Format (ex:
   Sep. 12, 1999)
   ^TMP($J,LIST,DFN,IEN,102.1)=PRIOR FILL DATE (52,102.1)^External Format (ex:
   Sep. 12, 1999)
   ^TMP($J,LIST,DFN,IEN,102.2)=PENDING NEXT POSSIBLE FILLDATE
   (52,102.2)^External Format (ex: Sep. 12, 1999)
   ^TMP($J,LIST,DFN,IEN,104)=VERIFYING PHARMACIST (52,104)^NAME (200,.01)
   ^TMP($J,LIST,DFN,IEN,109)=COSIGNING PHYSICIAN (52,109)^NAME (200,.01)
   ^TMP($J,LIST,DFN,IEN,112)=ORIGINAL QTY (52,112)
   ^TMP($J,LIST,”B”,RX#,IEN)=””
```

Output: (**If node = R(r)[^^n(o)]**)

```
   ^TMP($J,LIST,DFN,0)=Total entries returned by patient or -1^NO DATA FOUND
   ^TMP($J,LIST,DFN,IEN,100)=STATUS (52,100) ^External Format for the Set of
   Codes
   ^TMP($J,LIST,DFN,IEN,”RF”,0)=Total entries returned by patient or -1^NO
   DATA FOUND
   ^TMP($J,LIST,DFN,IEN,”RF”,n,.01)=REFILL DATE (52.1,.01)^External Format
   (ex: Sep. 12, 1999)
   ^TMP($J,LIST,DFN,IEN,”RF”,n,1)=QTY (52.1,1)
   ^TMP($J,LIST,DFN,IEN,”RF”,n,1.1)=DAYS SUPPLY (52.1,1.1)
   ^TMP($J,LIST,DFN,IEN,”RF”,n,1.2)=CURRENT UNIT PRICE OF DRUG (52.1,1.2)
   ^TMP($J,LIST,DFN,IEN,”RF”,n,2)=MAIL/WINDOW (52.1,2) ^External Format for
   the Set of Codes
   ^TMP($J,LIST,DFN,IEN,”RF”,n,3)=REMARKS (52.1,3)
   ^TMP($J,LIST,DFN,IEN,”RF”,n,4)=PHARMACIST NAME (52.1,4)^NAME (200,.01)
   ^TMP($J,LIST,DFN,IEN,”RF”,n,5)=LOT # (52.1,5)
   ^TMP($J,LIST,DFN,IEN,”RF”,n,6)=CLERK CODE (52.1,6)^NAME (200,.01)
   ^TMP($J,LIST,DFN,IEN,”RF”,n,7)=LOGIN DATE (52.1,7)^External Format (ex:
   Sep. 12, 1999)
   ^TMP($J,LIST,DFN,IEN,”RF”,n,8)=DIVISION (52.1,8) ^NAME (59,.01)
   ^TMP($J,LIST,DFN,IEN,”RF”,n,10.1)=DISPENSE DATE (52.1,10.1)^External Format
   (ex: Sep. 12, 1999)
   ^TMP($J,LIST,DFN,IEN,”RF”,n,12)=MANUFACTURER (52.1,12)
   ^TMP($J,LIST,DFN,IEN,”RF”,n,13)=DRUG EXPIRATION DATE (52.1,13)^External
   Format (ex: Sep. 12, 1999)
   ^TMP($J,LIST,DFN,IEN,”RF”,n,14)=RETURNED TO STOCK (52.1,14)External Format
   (ex: Sep. 12, 1999)
   ^TMP($J,LIST,DFN,IEN,”RF”,n,15)=PROVIDER (52.1,15)^NAME (200,.01)
   ^TMP($J,LIST,DFN,IEN,”RF”,n,17)=RELEASED DATE/TIME (52.1,17)^External
   Format (ex: Sep. 12, 1999)
   ^TMP($J,LIST,”B”,RX#,IEN)=””
Where:
   n = the IEN of entry in the Refill multiple
```

Output: (**If node = I(r)[^null,O or R^n])**

```
   ^TMP($J,LIST,DFN,0)=Total entries returned by patient or -1^NO DATA FOUND
   ^TMP($J,LIST,DFN,IEN,100)=STATUS (52,100)^External Format for the Set of
   Codes
   ^TMP($J,LIST,”B”,RX#,IEN)=””
```

[ORIGINAL FILLS]

```
   ^TMP($J,LIST,DFN,IEN,105)=COPAY TRANSACTION TYPE (52,105)
   ^TMP($J,LIST,DFN,IEN,106)=IB NUMBER (52,106)^REFERENCE NUMBER(350,.01)
   ^TMP($J,LIST,DFN,IEN,106.5)=COPAY TYPE AUDIT (52,106.5)
   ^TMP($J,LIST,DFN,IEN,106.6)=COPAY EXCEEDING CAP (52,106.6)^TRANSACTION
   NUMBER(354.71,.01)
```

[REFILLS]

```
   ^TMP($J,LIST,DFN,IEN,”IB”,0)=Total entries returned by patient or -1^NO
   DATA FOUND
   ^TMP($J,LIST,DFN,IEN,”IB”,n,9)= IB NUMBER (52.1,9) ^REFERENCE
   NUMBER(350,.01)
   ^TMP($J,LIST,DFN,IEN,”IB”,n,9.1)=COPAY EXCEEDING CAP (52.1,9.1)^TRANSACTION
   NUMBER(354.71,.01)
Where:
   n = the IEN of entry in the Refill multiple for IB
```

Output: (**If node = P**)

```
   ^TMP($J,LIST,DFN,0)=Total entries returned by patient or -1^NO DATA FOUND
   ^TMP($J,LIST,DFN,IEN,100)=STATUS (52,100) ^External Format for the Set of
   Codes
   ^TMP($J,LIST,DFN,IEN,”P”,0)= Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,DFN,IEN,”P”,n,.01)=PARTIAL DATE (52.2,.01)^External Format
   (ex: Sep. 12, 1999)
   ^TMP($J,LIST,DFN,IEN,”P”,n,.02)=MAIL/WINDOW (52.2,.02) ^External Format for
   the Set of Codes
   ^TMP($J,LIST,DFN,IEN,”P”,n,.03)=REMARKS (52.2,.03)
   ^TMP($J,LIST,DFN,IEN,”P”,n,.04)=QTY (52.2,.04)
   ^TMP($J,LIST,DFN,IEN,”P”,n,.041)=DAYS SUPPLY (52.2,.041)
   ^TMP($J,LIST,DFN,IEN,”P”,n,.042)=CURRENT UNIT PRICE OF DRUG (52.2,.042)
   ^TMP($J,LIST,DFN,IEN,”P”,n,.05)=PHARMACIST NAME (52.2,.05)^NAME (200,.01)
   ^TMP($J,LIST,DFN,IEN,”P”,n,.06)=LOT # (52.2,.06)
   ^TMP($J,LIST,DFN,IEN,”P”,n,.08)=LOGIN DATE (52.2,.08)^External Format (ex:
   Sep. 12, 1999)
   ^TMP($J,LIST,DFN,IEN,”P”,n,.09)=DIVISION (52.2,.09) ^NAME (59,.01)
   ^TMP($J,LIST,DFN,IEN,”P”,n,1)=NDC (52.2,1)
   ^TMP($J,LIST,DFN,IEN,”P”,n,5)=RETURNED TO STOCK (52.2,5)^External Format
   (ex: Sep. 12, 1999)
   ^TMP($J,LIST,DFN,IEN,”P”,n,6)=PROVIDER (52.2,6)^NAME (200,.01)
   ^TMP($J,LIST,DFN,IEN,”P”,n,8)=RELEASE DATE/TIME (52.2,8)^External Format
   (ex: Sep. 12, 1999)
   ^TMP($J,LIST,”B”,RX#,IEN)=””
Where:
   n = the IEN of entry in the Partial multiple
```

Output: (**If node = O**)

```
   ^TMP($J,LIST,DFN,0)=Total entries returned by patient or -1^NO DATA FOUND
   ^TMP($J,LIST,DFN,IEN,100)=STATUS (52,100) ^External Format for the Set of
   Codes
   ^TMP($J,LIST,DFN,IEN,”OI”)=IEN(52,39.2)^NAME(50.7,.01)^DOSAGE
   FORM(50.7,.02)^NAME (50.606,.01)
   ^TMP($J,LIST,”B”,RX#,IEN)=””
```

Output: (**If node = T**)

```
   ^TMP($J,LIST,DFN,0)=Total entries returned by patient or -1^NO DATA FOUND"
   ^TMP($J,LIST,DFN,IEN,100)=STATUS (52,100) ^External Format for the Set of
   Codes
   ^TMP($J,LIST,DFN,IEN,201)=TPB RX (52,201) ^External Format for the Set of
   Codes
   ^TMP($J,LIST,”B”,RX#,IEN)=””
```

Output: (**If node = L**)

```
   ^TMP($J,LIST,DFN,0)=Total entries returned by patient or -1^NO DATA FOUND
   ^TMP($J,LIST,DFN,IEN,100)=STATUS (52,100) ^External Format for the Set of
   Codes
   ^TMP($J,LIST,DFN,IEN,”L”,0)=Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,DFN,IEN,”L”,n,.01)=LABEL DATE/TIME (52.032,.01)^External
   Format (ex: Sep. 12, 1999)
   ^TMP($J,LIST,DFN,IEN,”L”,n,1)=RX REFERENCE (52.032,1)
   ^TMP($J,LIST,DFN,IEN,”L”,n,2)=LABEL COMMENT (52.032,1)
   ^TMP($J,LIST,DFN,IEN,”L”,n,3)=PRINTED BY (52.032,3)^NAME (200,.01)
   ^TMP($J,LIST,DFN,IEN,”L”,n,4)=WARNING LABEL TYPE (52.032,4) ^External
   Format for the Set of Codes
   ^TMP($J,LIST,”B”,RX#,IEN)=””
Where:
   n = the IEN of entry in the Label multiple
```

Output: (**If node = S**)

```
   ^TMP($J,LIST,DFN,0)=Total entries returned by patient or -1^NO DATA FOUND
   ^TMP($J,LIST,DFN,IEN,100)=STATUS (52,100) ^External Format for the Set of
   Codes
   ^TMP($J,LIST,DFN,IEN,301)=CLOZAPINE DOSAGE (MG/DAY) (52,301)
   ^TMP($J,LIST,DFN,IEN,302)=WBC RESULTS (52,302)
   ^TMP($J,LIST,DFN,IEN,303)=DATE OF WBC TEST (52,303)^External Format (ex:
   Sep. 12, 1999)
   ^TMP($J,LIST,”B”,RX#,IEN)=””
```

Output: (**If node = M**)

```
   ^TMP($J,LIST,DFN,0)=Total entries returned by patient or -1^NO DATA FOUND
   ^TMP($J,LIST,DFN,IEN,100)=STATUS (52,100) ^External Format for the Set of
   Codes
   ^TMP($J,LIST,DFN,IEN,”M”,0)=Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,DFN,IEN,”M”,n,0)=MEDICATION INSTRUCTIONS(52.04,.01)
   ^TMP($J,LIST,”B”,RX#,IEN)=””
Where:
   n = the IEN of entry in the Medication Instruction (SIG) multiple
```

Output: (**If node = C**)

```
   ^TMP($J,LIST,DFN,0)=Total entries returned by patient or -1^NO DATA FOUND
   ^TMP($J,LIST,DFN,IEN,100)=STATUS (52,100) ^External Format for the Set of
   Codes
   ^TMP($J,LIST,DFN,IEN,”C”,0)=Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,DFN,IEN,”C”,n,2)=RX INDICATOR (52.01,2)
   ^TMP($J,LIST,DFN,IEN,”C”,n,3)=STATUS (52.01,3) ^External Format for the Set
   of Codes
   ^TMP($J,LIST,DFN,IEN,”C”,n,4)=NDC (52.10,4)
   ^TMP($J,LIST,”B”,RX#,IEN)=””
Where:
   n = the IEN of entry in the CMOP Event multiple
```

Output: (**If node = A**)

```
   ^TMP($J,LIST,DFN,0)=Total entries returned by patient or -1^NO DATA FOUND
   ^TMP($J,LIST,DFN,IEN,100)=STATUS (52,100) ^External Format for the Set of
   Codes
   ^TMP($J,LIST,DFN,IEN,”A”,0)=Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,DFN,IEN,”A”,n,.01)=ACTIVITY LOG (52.03,.01)^External Format
   (ex: Sep. 12, 1999)
   ^TMP($J,LIST,DFN,IEN,”A”,n,.02)=REASON (52.03,.02) ^External Format for the
   Set of Codes
   ^TMP($J,LIST,DFN,IEN,”A”,n,.03)=INITIATOR OF ACTIVITY (52.03,.03)^NAME
   (200,.01)
   ^TMP($J,LIST,DFN,IEN,”A”,n,.04)=RX REFERENCE (52.03,.04)
   ^TMP($J,LIST,DFN,IEN,”A”,n,.05)=COMMENTS (52.30,.05)
   ^TMP($J,LIST,”B”,RX#,IEN)=””
Where:
   n = the IEN of entry in the Activity Log multiple
```

Output: (**If node = ST**)

```
   ^TMP($J,LIST,DFN,0)=Total entries returned by patient or -1^NO DATA FOUND
   ^TMP($J,LIST,DFN,IEN,100)=STATUS (52,100) ^External Format for the Set of
   Codes
   ^TMP($J,LIST,”B”,RX#,IEN)=””
```

PSO5241 API – PENDING OUTPATIENT ORDERS file (#52.41)

Component: PEN

Returns data elements for the PENDING OUTPATIENT ORDERS file (#52.41).

Status: Active

```
PEN^PSO5241(DFN,LIST,IEN,PLACER NUMBER)

Input:
   DFN
   LIST
   IEN
   PLACER NUMBER
Where:
   DFN = IEN from the PATIENT file (#2) [required]
   LIST = Subscript name used in the ^TMP global [required]
   IEN = Internal record number [optional]
   PLACER NUMBER = Pointer to ORDERS file (#100) [optional]
```

☛Note: If IEN is passed in, PLACER NUMBER is ignored. If PLACER NUMBER is passed in without IEN, lookup is done on ^PS(52.41,“B” cross-reference. If no IEN and no PLACER NUMBER is passed, ^PS(52.41,"P" xref is used to return all pending orders (ORDER TYPE = NW:NEW ORDER;HD:HOLD;RNW:RENEW;RF:REFILL REQUEST)

```
Output:
   ^TMP($J,LIST,DFN,0)=Total entries returned by patient or -1^NO DATA FOUND
   ^TMP($J,LIST,DFN,IEN,.01)=PLACER NUMBER(52.41,.01)
   ^TMP($J,LIST,DFN,IEN,2)=ORDER TYPE (52.41,2) ^External Format for the Set
   of Codes
   ^TMP($J,LIST,DFN,IEN,8)=IEN(52.41,8)^NAME(50.7,.01)^DOSAGE
   FORM(50.7,.02)^NAME (50.606,.01) – Orderable Item
   ^TMP($J,LIST,DFN,IEN,11)=DRUG (52.41,11) ^GENERIC NAME(50,.01)
   ^TMP($J,LIST,DFN,”B”,PLACER NUMBER,IEN)=””
```

PSO525AP API – RX SUSPENSE file (#52.5)

Component: SUS

Returns data elements for the RX SUSPENSE file (#52.5).

Status: Active

```
SUS^PSO525AP(LIST,DFN,IEN,RX#,SDATE,EDATE)

Input:
   LIST
   DFN
   IEN
   RX#
   SDATE
   EDATE
Where:
   LIST = Subscript name used in the ^TMP global [required]
   DFN = IEN from the PATIENT file (#2) [optional]
   IEN = Internal record number [optional]
   RX# = Pointer to PRESCRIPTION file (#52) [optional]
   SDATE = Starting Suspense Date [optional]
   EDATE = Ending Suspense Date [optional]
```

☛ Note: If IEN is passed in, RX # is ignored. If RX # is passed in without IEN, lookup is done on ^PS(52.5,“B” cross-reference. If no IEN and no RX # is passed, “AF” cross-reference shall be used to return suspended RXs. If SDATE is passed in, suspended RXs starting with this date shall be returned. If EDATE is passed in, suspended RXs up to and including this date shall be returned.

```
Output:
   ^TMP($J,LIST,DFN,0)=Total entries returned by patient or -1^NO DATA FOUND
   ^TMP($J,LIST,DFN,IEN,.01)=RX # (52.5,01)
   ^TMP($J,LIST,DFN,IEN,.02)=Suspense Date (52.5,.02)^External Format (ex:
   Sep. 12, 1999)
   ^TMP($J,LIST,DFN,IEN,.05)=Partial (52.5,.05)
   ^TMP($J,LIST,DFN,IEN,2)=Printed (52.5,2)^External Format for the Set of
   Codes
   ^TMP($J,LIST,DFN,IEN,3)=CMOP INDICATOR (52.5,3)^External Format for the Set
   of Codes
   ^TMP($J,LIST,DFN,IEN,9)=Fill (52.5,9)
   ^TMP($J,LIST,”B”,RX#,IEN)=””
```

PSO5252 API – CLOZAPINE PRESCRIPTION OVERRIDES file (#52.52)

Component: EN

Returns data elements for the CLOZAPINE PRESCRIPTION OVERRIDES file (#52.52).

Status: Active

```
EN^PSO5252(LIST,IEN,RX#,SDATE,EDATE)

Input:
   LIST
   IEN
   RX#
   SDATE
   EDATE
Where:
   LIST = Subscript name used in the ^TMP global [required]
   IEN = Internal record number [optional]
   RX# = Pointer to PRESCRIPTION file (#52) [optional]
   SDATE = Starting Date [optional]
   EDATE = Ending Date [optional]
```

☛ Note: If RX # is passed in without IEN, lookup is done on ^PS(52.52,“A”. If no IEN and no RX # is passed, "B" cross-reference shall be used to return Override RXs.

```
Output:
   ^TMP($J,LIST,0)=Total entries returned by patient or -1^NO DATA FOUND
   ^TMP($J,LIST,IEN,.01)= DATE TIME (52.52,.01)
   ^TMP($J,LIST,IEN,1)=PRESCRIPTION NUMBER (52.52,1)^RX # (52,.01)
   ^TMP($J,LIST,IEN,2)=USER ENTERING (52.52,2)^NAME (200,.01)
   ^TMP($J,LIST,IEN,3)=APPROVING TEAM MEMBER (52.52,3)^NAME (200,.01)
   ^TMP($J,LIST,IEN,4)=REASON FOR LOCKOUT (52.52,4) ^External Format for the
   Set of Codes
   ^TMP($J,LIST,IEN,5)=COMMENTS (52.52,5)
   ^TMP($J,LIST,”B”,DATE TIME,IEN)=””
```

PSO5291 API – TPB ELIGIBILITY file (#52.91)

Component: PSO

Returns data elements for the TPB ELIGIBILITY file (#52.91).

Status: Active

```
PSO^PSO5291(PSOIEN,PSOTXT,LIST)

Input:
   PSOIEN
   PSOTXT
   LIST
Where:
   PSOIEN = IEN [optional]
   PSOTXT = Free text entry [optional]
   LIST = Subscript name used in the ^TMP global [required]
```

☛ Note**:** Either the IEN or free text entry must be present.

```
Output:
   ^TMP($J,LIST,PSOIEN,0)=TOTAL entries returning or -1^NO DATA FOUND
   ^TMP($J,LIST,PSOIEN,.01)=PATIENT (IEN) (52.91,.01)^NAME (2,.01)
   ^TMP($J,LIST,PSOIEN,1)=DATE PHARMACY BENEFIT BEGAN(52.91,1)^External Format
   (ex: Sep. 12, 1999)
   ^TMP($J,LIST,PSOIEN,2)=INACTIVATION OF BENEFIT DATE(52.91,2)^External
   Format (ex: Sep. 12, 1999)
   ^TMP($J,LIST,PSOIEN,3)=INACTIVATION REASON CODE (52.91,3)^External Format
   for the Set of Codes
   ^TMP($J,LIST,PSOIEN,4)=DESIRED APPOINTMENT DATE(52.91,4)^External Format
   (ex: Sep. 12, 1999)
   ^TMP($J,LIST,PSOIEN,5)=WAIT TYPE (52.91,5)^External Format for the Set of
   Codes
   ^TMP($J,LIST,PSOIEN,6)=STATION NUMBER(52.91,6)
   ^TMP($J,LIST,PSOIEN,7)=INSTITUTION (52.91,7)^NAME (4,.01)
   ^TMP($J,LIST,PSOIEN,8)=EXCLUSION REASON (52.91,8)^External Format for the
   Set of Codes
   ^TMP($J,LIST,PSOIEN,9)=PRIMARY CARE SCHEDULE APT DATE(52.91,9)^External
   Format (ex: Sep. 12, 1999)
   ^TMP($J,LIST,PSOIEN,10)=RX #(52.91,10)
   ^TMP($J,LIST,PSOIEN,11)=DATE LETTER PRINTED(52.91,11)^External Format (ex:
   Sep. 12, 1999)
   ^TMP($J,LIST,”B”,PATIENT,PSOIEN)=””
```

PSO53 API – RX PATIENT STATUS file (#53)

Component: PSO

Returns data elements for the RX PATIENT STATUS file (#53).

Status: Active

```
PSO^PSO53(PSOIEN,PSOTXT,LIST)
```

```
Input:
   PSOIEN
   PSOTXT
   LIST
Where:
   PSOIEN = IEN [optional]
   PSOTXT = Free text entry [optional]
   LIST = Subscript name used in the ^TMP global [required]
```

☛ Note**:** `Either the IEN or free text entry must be present.`

```
Output:
   ^TMP($J,LIST,PSOIEN,0)=TOTAL entries returning or -1^NO DATA FOUND
   ^TMP($J,LIST,PSOIEN,.001)=NUMBER(53,.001)
   ^TMP($J,LIST,PSOIEN,.01)=NAME(53,.01)
   ^TMP($J,LIST,PSOIEN,2)=ABBR(53,2)
   ^TMP($J,LIST,PSOIEN,3)=DAYS SUPPLY(53,3)
   ^TMP($J,LIST,PSOIEN,4)=REFILLS(53,4)
   ^TMP($J,LIST,PSOIEN,5)=RENEWABLE (53,5)^External Format for the Set of
   Codes
   ^TMP($J,LIST,PSOIEN,6)=SC/A&A/OTHER/INPATIENT/NVA (53,6)^External Format
   for the Set of Codes
   ^TMP($J,LIST,PSOIEN,15)=EXEMPT FROM COPAYMENT (53,15)^External Format for
   the Set of Codes
   ^TMP($J,LIST,PSOIEN,16)=EXEMPT FROM CHAMPUS BILLING (53,16)^External Format
   for the Set of Codes
   ^TMP($J,LIST,”B”,NAME,PSOIEN)=””
```

PSO59 API – OUTPATIENT SITE file (#59)

Component: PSS

Returns data elements for the OUTPATIENT SITE file (#59).

Status: Active

```
PSS^PSO59(PSOIEN,PSOTXT,LIST)
```

```
Input:
   PSOIEN
   PSOTXT
   LIST
Where:
   PSOIEN = IEN [optional]
   PSOTXT = Free text entry [optional]
   LIST = Subscript name used in the ^TMP global [required]
```

☛ Note**:** `Either the IEN or free text entry must be present.`

```
Output:
   ^TMP($J,LIST,PSOIEN,0)=Total entries returned by patient or -1^NO DATA
   FOUND
   ^TMP($J,LIST,PSOIEN,.01)=NAME(59,.01)
   ^TMP($J,LIST,PSOIEN,.02)=MAILING FRANK STREET ADDRESS(59,.02)
   ^TMP($J,LIST,PSOIEN,.05)=MAILING FRANK ZIP+4 CODE(59,.05)
   ^TMP($J,LIST,PSOIEN,.06)=SITE NUMBER(59,.06)
   ^TMP($J,LIST,PSOIEN,.07)=MAILING FRANK CITY(59,.07)
   ^TMP($J,LIST,PSOIEN,.08)=MAILING FRANK STATE (59,.08)^NAME (5,.01)
   ^TMP($J,LIST,PSOIEN,1)=SITE DEA NUMBER (59,1))
   ^TMP($J,LIST,PSOIEN,2)=SITE (NATIONAL NAME)(59,2)^NAME (736,.01)
   ^TMP($J,LIST,PSOIEN,100)=RELATED INSTITUTION (59,100)^NAME (4,.01)
   ^TMP($J,LIST,PSOIEN,1003)=IB SERVICE/SECTION (59,1003)^NAME (49,.01)
   ^TMP($J,LIST,”B”,NAME,PSOIEN)=””
```

PSOORDER API – Prescription file (#52)

Component: EN

Open subscription for Outpatient Pharmacy prescription data.

Status: Active

```
EN^PSOORDER(DFN,RX#)
Input:
   DFN
   RX#
Where:
   DFN = IEN from the PATIENT file (#2); variable is optional
   If the patient’s IEN is not sent; then a null value must be
   passed in its place
   RX# = IEN of a prescription. The IEN is obtained from either the
   PRESCRIPTION PROFILE MULTIPLE file (#52) in the
   PHARMACY PATIENT file (#55), or from the PRESCRIPTION file
   (#52). A separate DBIA may be required to obtain the RX#.

   The ^TMP("PSOR",$J) is killed each time the entry point is called. It will
   be the responsibility of each developer to kill the ^TMP("PSOR",$J) global
   and the DFN and Rx# variables when finished
```

☛ Note**:** Requires version 7.0 of Outpatient Pharmacy software to be installed.

```
Output:
   ^TMP("PSOR",$J,RXN,0) = ID^FD^LSFD^ST^RX#^QTY^DS^RF^RFM^DRCT^
   RXCT^EXDT^RELDT^RTSDT^WPC^PAT^LDT
Where:
   RXN = Internal Rx#
   ID = Issue date of Rx
   FD = Fill date of Rx
   LSFD = Last fill date of Rx
   ST = Status of Rx; variable has format A;B where:
        A = code
        B = external printable form, (e.g., A for ACTIVE, DC for
        discontinued)
        If the status of a prescription = Hold:
Output:
   ^TMP("PSOR",$J,RXN,"HOLD",0) = HDRS^HDCOM^HDDT
```

```
Where:
   HDRS = Hold Reason
   HDCOM = Hold Comments
   HDDT = Hold Date
   If the status of a prescription = Suspended or has had an entry in the RX
   SUSPENSE file (#52.5)
Output:
   ^TMP("PSOR",$J,RXN,"SUS",0) = PRT^CMIND
Where:
   PRT = Printed Status: Printed, or Not Printed
   CMIND = CMOP Indicator; where:
           Q = Queued for Transmission
           X = Transmission Completed
           L = Loading for Transmission
           P = Printed Locally
```

☛ Note**:** Data only appears in this field if sites have the CMOP package installed and implemented.

```
RX# = External Rx number
QTY = Quantity Dispensed
DS = Day Supply
RF = Number of refills
RFM = Number of refills remaining
DRCT = Drug Cost of Original Fill
RXCT = Rx Cost of Original Fill
EXDT = Expiration/Cancel date of Rx
RELDT = Release Date/Time
RTSDT = Returned to Stock Date
WPC = Was Patient Counseled (Yes/No)
PAT = Patient; variable format A;B where:
      A = internal pointer and
      B = external printable form
LDT = This date is the date the original Rx was added to the prescription file
(#52). Login Date.
Output:
   ^TMP("PSOR",$J,RXN,1) = PR^CLK^VRP^CLN^RXP^MW^DIV^OERR#^FP^NDC^TPBRX
Where:
   PR = Provider, variable format A;B where:
        A = internal pointer and
        B = external printable form
   CLK = Entered By (clerk code); variable format A;B where:
         A = internal pointer and
         B = external printable form
   VRP = Verifying Pharmacist; variable format A;B where:
         A = internal pointer and
         B = external printable form
   CLN = Clinic; variable format A;B where:
         A = internal pointer
         B = external printable form. Data comes from the HOSPITAL
             LOCATION file (#44)
   RXP = Rx Patient Status; variable format A;B where:
         A = internal pointer and
         B = external printable form
   M/W = Mail/Window Routing; variable format A;B where:
         A = code
         B = external printable form
   DIV = Pointer to the OUTPATIENT SITE file (#59)
   OERR# = Order number; points to the ORDER file (#100)

   FP = Finishing Person. This variable has the format A;B
   where A is the internal pointer and B is external printable form.
   NDC = National Drug Code.
   TPBRX = This variable indicates that the Rx has been created as part of the
   Transitional Pharmacy Benefit project.
CMOP Data: If applicable:
Output:
   ^TMP("PSOR",$J,RXN,"CMOP",n,0) = TRANS #^SEQ#^FILL#^CMSTA^CMDCDT^NDC
Where:
   TRANS # = Transaction number; points to the CMOP TRANSMISSION file (#550.2)
   SEQ # = Sequence number; represents order number sent to the CMOP
   host facility
   Fill # = Fill #; where:
            0 = original,
            1-11 = refills
   CMSTA = CMOP Status; variable format A;B where:
           A = code
           B = external printable form
   CMDCDT = CMOP cancel date
   NDC = National Drug File Code - free text
Output:
   ^TMP("PSOR",$J,RXN,"CMOP",1,1,0) = CMDC REASON
Where:
   CMDC REASON = CMOP cancel reason
```

☛ Note**:** CMDCDT and CMDC REASON fields only populated if CMSTA = 3.

```
Drug Data:
Output:
   ^TMP("PSOR",$J,RXN,"DRUG",0) = DR^VA PRINT NAME^DRUGID^VA DRUG CLASS
Where:
   DR = Drug in Rx; variable format A;B where:
        A = internal pointer and
        B = is external printable form
   VA Print Name = VA Print name found in the VA PRODUCT file (#50.68)
   DRUGID = CMOP ID (VA PRODUCT IDENTIFIER) found in the VA PRODUCT file
   (#50.68)
   VA DRUG CLASS = VA DRUG CLASSIFICATION
Output:
   ^TMP("PSOR",$J,RXN,"DRUGOI",0) = ORDERABLE ITEM
Where:
   ORDERABLE ITEM = Pharmacy Orderable Item tied to the drug in the Rx;
   variable format A;B where:
      A = internal pointer and
      B = external printable form concatenated with dose form,
      (i.e., ASPIRIN TAB)
Copay: If applicable
Output:
   ^TMP("PSOR",$J,RXN,"IB",0) = COPAY TRANSACTION TYPE^IB NUMBER
Where:
   COPAY TRANSACTION TYPE = 1 or 2
   IB NUMBER = pointer to the INTEGRATED BILLING ACTION file (#350)
Refills:
Output:
   ^TMP("PSOR",$J,RXN,"REF",n,0) =
   RFD^PR^CLK^QTY^DS^DRCT^RXCT^RELDT^RTSDT^M/W^DIV^LDT^NDC
Where:
   RXN = Internal Rx #
   RFD = Refill Date
   PR = Provider; variable format A;B where:
        A = internal pointer and
        B = external printable form
   CLK = Refill entry by; variable format A;B where:
         A = internal pointer and
         B = external printable form
   QTY = Quantity Dispensed
   DS = Day Supply
   DRCT = Drug Cost of Refill
   RXCT = Rx Cost of Refill
   RELDT = Release Date/Time
   RTSDT = Returned to Stock Date
   M/W = Mail/Window Routing; variable format A;B where:
         A = code
         B = external printable form
   DIV = pointer to the OUTPATIENT SITE file (#59)
   LDT = This date indicates the date the refill was requested.  This is not
   the date the refill will be dispensed (Login Date)
   NDC = National Drug Code
Partial Fills:
Output:
   ^TMP("PSOR",$J,RXN,"RPAR",n,0) =
   PRD^PR^CLK^QTY^DS^DRCT^RXCT^RELDT^RTSDT^M/W^DIV^LDT^NDC
Where:
   RXN = Internal Rx #
   PRD = Partial Date
   PR = Provider; Variable format A;B where:
        A = internal pointer and
        B = external printable form
   CLK = Refill entry by; variable format A;B where:
         A = internal pointer and
         B = external printable form
   QTY = Quantity Dispensed
   DS = Day Supply
   DRCT = Drug Cost of partial fill
   RXCT = Rx Cost of partial fill
   RELDT = Release Date/Time
   RTSDT = Returned to Stock Date
   M/W = Mail/Window Routing; variable format A;B where:
         A = code
         B = external printable form
   DIV = pointer to the OUTPATIENT SITE file (#59)
   LDT = This date indicates the date the partial fill was created(Login Date)
   NDC = National Drug Code
Activity Log:
Output:
   ^TMP("PSOR",$J,RXN,"ACT",n,0) = D/T^REA^NEW PERSON^RX #^COMMENTS
Where:
   D/T = Date/Time entry made
   REA = Reason entry was made
   NEW PERSON = Entry created by
   NEW PERSON = Entry created by; variable format A;B where:
                A = internal pointer and
                B = external printable form
   RX # = which fill the activity occurred on (original, refill, or partial)
   COMMENTS = Comments about the activity that occurred
Medication Instructions:
Output:
   ^TMP("PSOR",$J,RXN,"SIG",n,0) = Condensed Medication Instructions (SIG)
   ^TMP("PSOR",$J,RXN,"SIG1",n,0) = Expanded Medication Instructions (SIG)
   Dispensing Instructions:
Output:
   ^TMP("PSOR",$J,RXN,"MI",N,0) =
   DOSAGE^DOSE^UNITS^NOUN^DURATION^CONJUNCTION^MRT^SCH^VERB
   Where:
   DOSAGE = This is the strength of the medication dispensed.  This variable
   can be a numeric value or free-text
   DOSE = This numeric value represents the total number of pills to make a
   total dosage.  This value is only returned when the dosage is numeric
   UNITS = This data element is the unit of measure the medication is
   dispense.  This variable has the format A;B where:
      A is the internal pointer
      B is external printable form
   NOUN = This data element indicates the form the medication was dispensed,
   i.e., tablet
   DURATION = This indicates how long this dosage should be taken
   CONJUNCTION = This data element is used for complex dosing instructions
   MRT = This data element indicates how the medication is ingested
   (medication route). This variable has the format A;B where:
      A is the internal pointer and
      B is external printable form
   SCH = This data indicates when the medication is taken (schedule)
   VERB = This data element indicates what action is taken to ingest the
   medication
   Patient Instructions:
Output:
   ^TMP("PSOR",$J,RXN,"PI",n,0) = EXPANDED PATIENT INSTRUCTIONS
```

(*This page left blank for two-sided copying.*)

Inpatient Medications (IPM) - Unit Dose and IV

The Inpatient Medications application, namespace PSJ, provides the following Application Program Interfaces (APIs). If you need the use of an API that has not yet been made active, please email the Pharmacy Re-Engineering team using the Outlook mail group VHA OI SDD PHARM REENG DEV.

New DBIA #	Old DBIA #	Component	Description
4819	1884	ALL^PSJ59P5	Returns the NAME field (#.01), DIVISION field (#.02) and INACTIVATION DATE field (#19) from the IV ROOM file (#59.5)

PSJ59P5 API – IV ROOM file (#59.5)

Component: ALL

Returns the NAME field (#.01), DIVISION field (#.02) and INACTIVATION DATE field (#19) from the IV ROOM file (#59.5)

Status: Active

```
ALL^PSJ59P5(PSJIEN,PSJTXT,LIST)

Input:
   PSJIEN
   PSJTXT
   LIST
Where:
   PSJIEN = IEN [optional]
   PSJTXT = Free text entry [optional]
   LIST = Subscript name used in the ^TMP global [required]
Output:
   ^TMP($J,LIST,0)=Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSJIEN,.01)=NAME (59.5,.01)
   ^TMP($J,LIST,PSJIEN,.02)= DIVISION (59.5,.02 - P)^NAME(40.8,.01)
   ^TMP($J,LIST, PSJIEN,19)= INACTIVATION DATE (59.5,19)^External Format (ex:
   Sep. 12, 1999)
   ^TMP($J,LIST,”B”,NAME,PSJIEN)=””
```

(*This page left blank for two-sided copying.*)

Inpatient Pharmacy Automatic Replenishment/Ward Stock (AR/WS)

At this time there are no supported APIs for the Inpatient Pharmacy Automatic Replenishment/ Ward Stock application.

(*This page left blank for two-sided copying.*)

Controlled Substances (CS)

At this time there are no supported APIs for the Controlled Substances application.

(*This page left blank for two-sided copying.*)

Drug Accountability/Inventory Interface (DA)

At this time there are no supported APIs for the Drug Accountability application.

(*This page left blank for two-sided copying.*)

Consolidated Mail Outpatient Pharmacy (CMOP)

At this time there are no supported APIs for the Consolidated Mail Outpatient Pharmacy application.

(*This page left blank for two-sided copying.*)

Pharmacy Data Management (PDM)

The Pharmacy Data Management application, namespace PSS, provides the following Application Program Interfaces (APIs). If you need the use of an API that has not yet been made active, please email the Pharmacy Re-Engineering team using the Outlook mail group VHA OI SDD PHARM REENG DEV.

New DBIA #	Old DBIA #	Component	Description
4533	N/A	AND^PSS50	Returns the IEN(s) and GENERIC NAME field (#.01) in the DRUG file (#50) in the array defined by the calling application. The "AND" cross-reference in the format of ^PSDRUG("AND",NATIONAL DRUG FILE ENTRY(50,20),IEN(50)) will be used.
	N/A	ARWS^PSS50	Returns fields utilized by the Automatic Replenishment/Ward Stock extract in the Pharmacy Benefits Management (PBM) application, in the DRUG file (#50) in the array defined by the calling application.
	N/A	ASP^PSS50	Returns the IEN(s) and GENERIC NAME field (#.01) in the DRUG file (#50) in the array defined by the calling application. The "ASP" cross-reference in the format of ^PSDRUG("ASP",PHARMACY ORDERALBE ITEM(50,2.1),IEN(50)) will be used.
	N/A	CLOZ^PSS50	Returns the Clozapine lab test monitor drug fields of the DRUG file (#50) in the array defined by the calling application.
	25, 106, 221, 276, 302, 553, 477	DATA^PSS50	Returns most fields in the DRUG file (#50) in the array defined by the calling application.
	273	LAB^PSS50	Returns the lab test monitor drug fields of the DRUG file (#50) in the array defined by the calling application.
	N/A	NDC^PSS50	Returns the IEN(s) and GENERIC NAME field (#.01) in the array defined by the calling application. The "NDC" cross-reference in the format of ^PSDRUG("NDC",NDC(50,31),IEN(50)) will be used.
	N/A	NDF^PSS50	Returns the National Drug File (NDF) drug fields of the DRUG file (#50) in the array defined by the calling application.

New DBIA #	Old DBIA #	Component	Description
4533 cont.	N/A	VAC^PSS50	Returns the IEN(s) and GENERIC NAME field (#.01) in the array defined by the calling application. The “VAC” cross-reference in the format of ^PSDRUG(“VAC”,NATIONAL DRUG CLASS(50,25),IEN(50)) will be used.
	N/A	ZERO^PSS50	Returns the zero node in the DRUG file (#50) in the array defined by the calling application.
4546	N/A	AP^PSS51P1	Returns the IEN(s) and NAME field (#.01) of the ADMINISTRATION SCHEDULE file (#51.1) in the array defined by the calling application when passed in the PACKAGE PREFIX field (#4) of the ADMINISTRATION SCHEDULE file (#51.1). The “AP” cross-reference in format of ^PS(51.1,“AP”_PACKAGE PREFIX(4),NAME(.01),IEN(51.1)) will be used.
	N/A	ZERO^PSS51P1	Returns the zero node and the OTHER LANGUAGE EXPASION field (#8.1) of the ADMINISTRATION SCHEDULE file (#51.1) in the array defined by the calling application.
4548	N/A	ALL^PSS51P2	Returns all fields in the MEDICATION ROUTES file (#51.2) in the array defined by the calling application.
4549	536,568	ZERO^PSS52P6	Returns the zero node and the INACTIVATION DATE field (#12) of the IV ADDITIVES file (#52.6) in the array defined by the calling application.
4550	437, 537	ZERO^PSS52P7	Returns the zero node, the INACTIVATION DATE field (#12) and the ELECTROLYTES multiple of the IV SOLUTIONS file (#52.7) in the array defined by the calling application.
4551	N/A	DIC^PSSDI	Accepts input values and return output values as defined by VA FileMan Lookup call ^DIC.
	N/A	DO^PSSDI	This API will accept input values and return output values as defined by VA FileMan Lookup call DO^DIC1.
	N/A	EN^PSSDI	This API will accept input values and return output values as defined by VA FileMan Data Retrieval call EN^DIQ1.
	N/A	FNAME^PSSDI	Returns the field name of the specified Pharmacy file for the field number and file number passed in.
	N/A	MIX^PSSDI	Accepts input values and return output values as defined by VA FileMan Lookup/Add call MIX^DIC1.

New DBIA #	Old DBIA #	Component	Description
4662	N/A	DRGIEN^PSS50P7	Returns the IEN(s) of the DRUG file (#50) in the array defined by the calling application when passing in the IEN of the PHARMACY ORDER ITEM file (#50.7). The “A50” cross-reference in format of ^PS(50.7,“A50”,IEN(50.7),IEN(50)) will be used.
	N/A	ZERO^PSS50P7	Returns the zero node of the PHARMACY ORDERABLE ITEM file (#50.7) in the array defined by the calling application. A check for the existence of the zero node will be performed.
4826	117, 552, 2497 677, 3785, 4181, 533, 2228, 922, 567	PSS431^PSS55	Returns a list of Unit Dose medications from data in the UNIT DOSE sub-file (#55.06) of the PHARMACY PATIENT file (#55) and dispensed drugs from the DISPENSED DRUGS sub-file (#55.07).
	2497, 677, 2475, 2228, 567	PSS432^PSS55	Returns a listing of active orders by scanning the PHARMACY PATIENT file (#55) utilizing the “AUS” cross-reference.
	117, 677, 2228	PSS433^PSS55	Returns the Unit Dose data from the “2” node of the UNIT DOSE sub-file (#55.06) of the PHARMACY PATIENT file (#55) for the DFN received.
	677, 2228, 922	PSS435^PSS55	Returns the Start Date and Time and/or a list of active Hyperal IV Orders utilizing the “AIT” cross-reference on the STOP DATE/TIME field (#.02) for the DFN received.
	2497, 677, 3785, 4181, 535, 2228, 922, 567	PSS436^PSS55	Returns a list of the active IV Additive Information, found on the AD nodes of the PHARMACY PATIENT file (#55) for the DFN received.
4828	2682	PSS^PSS59P7	Returns the OUTPATIENT VERSION field (#49.99) and ADMISSION CANCEL OF RXS field (#40.1) from the PHARMACY SYSTEM file (#59.7) for the IEN or free text entry received.
Inactive	N/A	ALL^PSS32P3	Returns the TYPE field (#.01) of the APSP INTERVENTION TYPE file (#9009032.3).
Inactive	N/A	ALL^PSS32P5	Returns the RECOMMENDATION field (#.01) of the APSP INTERVENTION RECOMMENDATION file (#9009032.5).
Inactive	N/A	A526^PSS50	Returns the IEN(s) of the IV ADDITIVE file (#52.6) when passed in the IEN of the DRUG file (#50) in the array defined by the calling application.

New DBIA #	Old DBIA #	Component	Description
Inactive	N/A	A527^PSS50	Returns the IEN(s) of the IV SOLUTION file (#52.7) when passed in the IEN of the DRUG file (#50) in the array defined by the calling application.
Inactive	N/A	AB^PSS50	Returns the IEN(s) and GENERIC NAME field (#.01) in the DRUG file (#50) in the array defined by the calling application. External formats will be included if applicable. The “AB” cross-reference in the format of ^PSDRUG(“AB”,ITEM NUMBER(50,.0441,.01),IEN(50),IEN(50.0441)) will be used.
Inactive	N/A	ADDOLDNM^PSS50	Adds a new entry to the OLD NAME multiple of the DRUG file (#50).
Inactive	N/A	AIU^PSS50	Returns the GENERIC NAME field (#.01) and IEN(s) of the DRUG file (#50) in the array defined by the calling application. The “AIU” cross-reference in the format of PSDRUG(“AIU”,NAME(50,.01),IEN(50)) will be used.
Inactive	N/A	AOC^PSS50	Returns the IEN(s) and GENERIC NAME field (#.01) in the DRUG file (#50) in the array defined by the calling application. The “AOC” cross-reference in the format of ^PSDRUG(“AOC”,IEN(50.7),VA CLASSIFICATION (50.2),IEN(50)) will be used.
Inactive	N/A	AP^PSS50	Returns the IEN(s) and GENERIC NAME field (#.01) in the DRUG file (#50) in the array defined by the calling application. The “AP” cross-reference in the format of ^PSDRUG (“AP”,PRIMARY DRUG (50,64),IEN(50)) will be used.
Inactive	N/A	AQ^PSS50	Indicates if the IEN passed into the DRUG file (#50) existed in the “AQ” cross-reference. The “AQ” cross-reference in the format of ^PSDRUG(“AQ”,IEN(50)) will be used.
Inactive	N/A	AQ1^PSS50	Returns the GENERIC NAME field (#.01) and IEN(s) of the DRUG file (#50) in the array defined by the calling application. The “AQ1” cross-reference in the format of ^PSDRUG(“AQ1”, CMOP ID(27),IEN(50)) will be used.
Inactive	N/A	ATC^PSS50	Returns the ATC drug fields of the DRUG file (#50) in the array defined by the calling application.

New DBIA #	Old DBIA #	Component	Description
Inactive	N/A	AVSN^PSS50	Returns the GENERIC NAME field (#.01), IEN(s) and SYNONYM multiple of the DRUG file (#50) in the array defined by the calling application. The “AVSN” cross-reference in the format of ^PSDRUG(“AVSN”,VSN(50.1.,400),IEN(50.1), IEN(50)) will be used.
Inactive	N/A	B^PSS50	Returns the IEN(s) of the DRUG file (#50) in the array defined by the calling application. The “B” cross-reference in the format of ^PSDRUG(“B”,NAME(50,.01),IEN(50)) will be used.
Inactive	N/A	C^PSS50	Returns the SYNONYM multiple of the DRUG file (#50) in the array defined by the calling application. The “C” cross-reference in the format of ^PSDRUG(“C”,SYNONYM(50.1.,01),IEN(50.1)) will be used.
Inactive	N/A	CMOP^PSS50	Returns the CMOP drug fields of the DRUG file (#50) in the array defined by the calling application.
Inactive	N/A	CSYN^PSS50	Returns the SYNONYM subfield (#.01) of the SYNONYM multiple of the DRUG file (#50).
Inactive	N/A	DOSE^PSS50	Returns the dosage related fields of the DRUG file (#50) in the array defined by the calling application.
Inactive	N/A	DRG^PSS50	Returns the drug fields of the DRUG file (#50) in the array defined by the calling application.
Inactive	N/A	DSPUNT^PSS50	Returns the DISPENSE UNITS PER ORDER UNIT field (#403) of the SYNONYM multiple of the DRUG file (#50) in the array defined by the calling application.
Inactive	N/A	EDTIFCAP^PSS50	Allows editing of the ITEM NUMBER sub-field (#.01) of the IF CAP ITEM NUMBER multiple of the DRUG file (#50). The lookup call is provided in AB^PSS50 API.
Inactive	N/A	FRMALT^PSS50	Looks up the FORMULARY ALTERNATIVE multiple from the DRUG file (#50) in the array defined by the calling application.
Inactive	N/A	IEN^PSS50	Returns the IEN(s) of the DRUG file (#50) in the array defined by the calling application.
Inactive	N/A	INV^PSS50	Returns the inventory drug fields of the DRUG file (#50) in the array defined by the calling application.

New DBIA #	Old DBIA #	Component	Description
Inactive	N/A	IU^PSS50	Returns the GENERIC NAME field (#.01) and IEN(s) of the DRUG file (#50) in the array defined by the calling application. The "IU" cross-reference in the format of ^PSDRUG("IU",APPLICATION PACKAGES'USE(50,63),IEN(50)) will be used.
Inactive	N/A	LABEL^PSS50	Returns the fields from the DRUG file (#50) when passed the IEN of the DRUG file (#50) in the array defined by the calling application. External formatting will be included if applicable. The last piece of the output contains the field name of the returned field.
Inactive	N/A	LIST^PSS50	Returns all GENERIC NAME field (#.01) and PHARMACY ORDERABLE ITEM field (#2.1) of the DRUG file (#50) in the array defined by the calling application.
Inactive	N/A	LOOKUP^PSS50	Looks up a drug(s) from the DRUG file (#50) in the array defined by the calling application.
Inactive	N/A	MRTN^PSS50	Returns the monitor routine drug fields of the DRUG file (#50) in the array defined by the calling application.
Inactive	N/A	MSG^PSS50	Returns the IEN(s) of the DRUG file (#50) if there is data in the QUANTITY DISPENSE MESSAGE field (#215) in the array defined by the calling application.
Inactive	N/A	NOCMOP^PSS50	Returns the IEN(s) in the DRUG file (#50) in the array defined by the calling application.
Inactive	N/A	OLDNM^PSS50	Returns the OLD NAME multiple of the DRUG file (#50) in the array defined by the calling application.
Inactive	N/A	SKAIU^PSS50	Sets and kills the "AIU" cross-references of the DRUG file (#50) in the array defined by the calling application.
Inactive	N/A	SKAQ^PSS50	Sets and kills the "AQ" cross-reference of the DRUG file (#50) in the array defined by the calling application.
Inactive	N/A	SKAQ1^PSS50	Sets and kills the "AQ1" cross-reference of the DRUG file (#50) in the array defined by the calling application.
Inactive	N/A	SKB^PSS50	Will set and kill the "B" cross-reference of the DRUG file (#50) in the array defined by the calling application.

New DBIA #	Old DBIA #	Component	Description
Inactive	N/A	SKIU^PSS50	Sets and kills the “AU” cross-references of the DRUG file (#50) in the array defined by the calling application.
Inactive	N/A	SORT^PSS50	Returns the NAME field (#.10) of the DRUG file (#50) in the array defined by the calling application. Original sort template can be found in PSOUPAT.
Inactive	N/A	WS^PSS50	Returns the ward stock drug fields of the DRUG file (#50) in the array defined by the calling application.
Inactive	N/A	ALL^PSS50P4	Returns all fields in the DRUG ELECTROLYTES file (#50.4) in the array defined by the calling application.
Inactive	N/A	ADD^PSS50P66	The first API, ALL^PSS50P66, does a look-up from the DOSAGE FORM file (#50.606). The second API, ADD^PSS50P66 will add the MEDICATION ROUTES FOR DOSAGE FORM sub-field (#.01) of the MED ROUTE FOR DOSAGE FORM multiple from the DOSAGE FORM file (#50.606).
Inactive	N/A	ALL^PSS50P66	Returns most of the fields from the DOSAGE FORM file (#50.606) in the array defined by the calling application.
Inactive	N/A	IEN^PSS50P7	Returns the IEN(s) of the PHARMACY ORDERABLE ITEM file (#50.7) in the array defined by the calling application when passing in the NAME field (#.01) of the PHARMACY ORDERABLE ITEM file (#50.7). The “B” cross-reference in the format of ^PS(50.7,“B”,NAME(50.7,.01),IEN(50.7)) will be used.
Inactive	N/A	INSTR^PSS50P7	Returns the PATIENT INSTRUCTIONS field (#7) and OTHER LANGUAGE INSTRUCTIONS field (#7.1) of the PHARMACY ORDERABLE ITEM file (#50.7) in the array defined by the calling application.

New DBIA #	Old DBIA #	Component	Description
Inactive	N/A	LOOKUP^PSS50P7	Returns the NAME field (#.01) of the PHARMACY ORDERABLE ITEM file (#50.7) along with the NAME field (#.01) of the DOSAGE FORM file (#50.606). Screening for the active entries from the DRUG file (#50) for specified usage will be applied before including the data in the return array. The ACTIVE DATE field (#.04) of the PHARMACY ORDERABLE ITEM file (#50.7) should be included in the screening.
Inactive	N/A	NAME^PSS50P7	Returns the NAME field (#.01) of the PHARMACY ORDERABLE ITEM file (#50.7).
Inactive	N/A	SYNONYM^PSS50P7	Returns the SYNONYM sub-field (#.01) of the SYNONYM multiple of the PHARMACY ORDERABLE ITEM file (#50.7) in the array defined by the calling application.
Inactive	N/A	A^PSS51	Returns the MEDICATION INSTRUCTION EXPANSION fields of the MEDICATION INSTRUCTION file (#51) in the array defined by the calling application.
Inactive	N/A	ALL^PSS51	Returns most of the fields from the MEDICATION INSTRUCTION file (#51) in the array defined by the calling application.
Inactive	N/A	CHK^PSS51	Checks if the value is a valid entry and the INTENDED USE field (#30) from the MEDICATION INSTRUCTION file (#51) is for Outpatient use (Less than 2).
Inactive	N/A	LOOKUP^PSS51	Returns the NAME field (#.01) and EXPANSION field (#1) from the MEDICATION INSTRUCTION file (#51) in the array defined by the calling application.
Inactive	N/A	WARD^PSS51	Returns most of the fields from the MEDICATION INSTRUCTION file (#51) in the array defined by the calling application.
Inactive	N/A	ADM^PSS51P1	Validates the STANDARD ADMINISTRATION TIMES field (#1) of the ADMINISTRATION SCHEDULE file (#51.1).
Inactive	N/A	ALL^PSS51P1	Returns all data of the ADMINISTRATION SCHEDULE file (#51.1) in the array defined by the calling application.
Inactive	N/A	HOSP^PSS51P1	Returns the fields from HOSPITAL LOCATION sub-field of the ADMINISTRATION SCHEDULE file (#51.1) in the array defined by the calling application.

New DBIA #	Old DBIA #	Component	Description
Inactive	N/A	IEN^PSS51P1	Returns the IEN(s) of the ADMINISTRATION SCHEDULE file (#51.1) in the array defined by the calling application.
Inactive	N/A	IX^PSS51P1	Returns the zero node of the ADMINISTRATION SCHEDULE file (#51.1) in the array defined by the calling application when passed in the NAME field (#.01) and PACKAGE PREFIX field (#4) of the ADMINISTRATION SCHEDULE file (#51.1). The “AP” cross-reference in format of ^PS(51.1,“AP”_PACKAGE PREFIX(4),NAME(.01),IEN(51.1)) will be used.
Inactive	N/A	WARD^PSS51P1	Returns the WARD sub-field of the ADMINISTRATION SCHEDULE file (#51.1) in the array defined by the calling application.
Inactive	N/A	ACP^PSS51P15	Returns the IEN(s) and NAME field (#.01) of the ADMINISTRATION SHIFT file (#51.15) in the array defined by the calling application when passed in the PACKAGE field (#4) and the ABBREVIATION field (#1) of the ADMINISTRATION SHIFT file (#51.15). The “ACP” cross-reference in format of ^PS(51.15,“ACP”_PACKAGE(4),ABBREVIATION(1),IEN(51.15)) will be used.
Inactive	N/A	ALL^PSS51P15	Returns all fields in the ADMINISTRATION SHIFT file (#51.15) in the array defined by the calling application.
Inactive	N/A	IEN^PSS51P2	Returns the IEN(s) of the MEDICATION ROUTES file (#51.2) when passed the ABBREVIATION field (#1) of the MEDICATION ROUTES file (#51.2) in the array defined by the calling application. The “C” cross-reference in format of ^PS(51.2,“C”,ABBREVIATION(52.1,1),IEN(51.2)) will be used.
Inactive	N/A	NAME^PSS51P2	Returns the IEN of the MEDICATION ROUTES file (#51.2) in the array defined by the calling application. The “B” cross-reference in format of ^PS(51.2,“B”,NAME(52.1,.01),IEN(51.2)) will be used.
Inactive	N/A	ALL^PSS51P5	Returns all fields in the ORDER UNIT file (#51.5) in the array defined by the calling application.
Inactive	N/A	EXPAN^PSS51P5	Returns all fields in the ORDER UNIT file (#51.5) in the array defined by the calling application.

New DBIA #	Old DBIA #	Component	Description
Inactive	N/A	DRGIEN^PSS52P6	Returns the IEN(s) from the IV ADDITIVES file (#52.6) in the array defined by the calling application with the DRUG file (#50) IEN. The “AC” cross-reference in format of ^PS(52.6,“AC”,IEN(50),IEN(52.6)) will be used.
Inactive	N/A	DRGINFO^PSS52P6	Returns the fields in the DRUG INFORMATION multiple of the IV ADDITIVES file (#52.6) in the array defined by the calling application.
Inactive	N/A	ELYTES^PSS52P6	Returns the fields in the ELECTROLYTES multiple of the IV ADDITIVES file (#52.6) in the array defined by the calling application.
Inactive	N/A	LOOKUP^PSS52P6	Returns the PRINT NAME, QUICK CODE and SYNONYM of the IV ADDITIVES file (#52.6) to the calling application to display to the user. The screening considers only active entries in the IV ADDITIVES file (#52.6) and the PHARMACY ORDERABLE ITEM field (#15) of the IV ADDITIVES file (#52.6) that are not null. Those entries with the INACTIVE DATE field (#100) in the DRUG file (#50) will be excluded.
Inactive	N/A	POI^PSS52P6	Returns the IEN(s) from the IV ADDITIVES file (#52.6) in the array defined by the calling application with the given PHARMACY ORDERABLE ITEM file (#50.7). The “AOI” cross-reference in format of ^PS(52.6,“AOI”,IEN(50.7),IEN(52.6)) will be used.
Inactive	N/A	QCODE^PSS52P6	Returns the fields in the QUICK CODE multiple of the IV ADDITIVES file (#52.6) in the array defined by the calling application.
Inactive	N/A	SYNONYM^PSS52P6	Returns the fields in the SYNONYM multiple of the IV ADDITIVES file (#52.6) in the array defined by the calling application.
Inactive	N/A	ACTSOL^PSS52P7	Returns the IEN(s), PRINT NAME, and VOLUME from the IV SOLUTIONS file (#52.7) for active entries. A count of all active entries shall be included.
Inactive	N/A	DRGIEN^PSS52P7	Returns the IEN(s) from the IV SOLUTIONS file (#52.7) in the array defined by the calling application with the DRUG file (#50) IEN. The “AC” cross-reference in format of ^PS(52.7,“AC”,IEN(50),IEN(52.7)) will be used.
Inactive	N/A	INACTDT^PSS52P7	Returns the INACTIVATION DATE (52.7,8) (ex: 3030915 or null).

New DBIA #	Old DBIA #	Component	Description
Inactive	N/A	LOOKUP^PSS52P7	Returns PRINT NAME, PRINT NAME(2), and VOLUME from the IV SOLUTIONS file (#52.7).
Inactive	N/A	POI^PSS52P7	Returns the IEN(s) from the IV SOLUTIONS file (#52.7) in the array defined by the calling application with the given PHARMACY ITEM ORDERABLE file (#50.7) IEN. The “AOI” cross-reference in format of ^PS(52.7,“AOI”,IEN(50.7),IEN(52.7)) will be used.
Inactive	437	POICHK^PSS52P7	Returns the PHARMACY ORDERABLE ITEM file (#50.7) IEN if the value in the PHARMACY ORDERABLE ITEM field (#9) of the IV SOLUTIONS file (#52.7) existed.
Inactive	N/A	ALL^PSS54	Returns the NAME field (#.01) and the TEXT multiple of the RX CONSULT file (#54) in the array defined by the calling application.
Inactive	N/A	LOOKUP^PSS54	Returns the NAME field (#.01) from the RX CONSULT file (#54).
Inactive	N/A	DIE^PSSDI	Accepts input values and return output values as defined by VA FileMan Edit Data call ^DIE.
Inactive	N/A	EN1^PSSDI	Accepts input values and return output values as defined by VA FileMan Print Data call EN1^DIP.
Inactive	N/A	FILE^PSSDI	Accepts input values and return output values as defined by VA FileMan Add call FILE^DIC1.
Inactive	N/A	IX^PSSDI	Accepts input values and return output values as defined by VA FileMan Lookup/Add call IX^DIC.
Inactive	N/A	HLP^PSSFILES	Returns help text for “?” input value on the specified PDM file in the array defined by the calling application.

PSS50 API - DRUG file (#50)

Component: DATA

Returns most fields in the DRUG file (#50) in the array defined by the calling application.

Status: Active

```
DATA^PSS50(PSSIEN,PSSFT,PSSFL,PSSPK,PSSRTOI,LIST)
Input:
   PSSIEN
   PSSFT
   PSSFL
   PSSPK
   PSSRTOI
   LIST
Where:
   PSSIEN = IEN from the DRUG file (#50) [optional]
   PSSFT = GENERIC NAME field (#.01) of the DRUG file (#50) (a value of "??"
   may be used) [optional]
   PSSFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   PSSPK = Returns only those entries containing at least one of the codes in
   the APPLICATION PACKAGES’ USE field (#63) of the DRUG file (#50) (ex: PSSPK
   = "IU" will return all entries for either IV, Unit Dose, or both IV and
   Unit Dose) [optional]
   PSSRTOI = If a "1" is passed in, then only those entries matched to a
   Pharmacy Orderable Item will be returned [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = GENERIC NAME (50,.01)
   ^TMP($J,LIST,PSSIEN,2) = VA CLASSIFICATION (50,2)
   ^TMP($J,LIST,PSSIEN,2.1) = PHARMACY ORDERABLE ITEM (50,2.1)
   ^NAME(50.7,.01)^IEN of the Dosage Form file (#50.606)^NAME (50.606,.01)
   ^TMP($J,LIST,PSSIEN,3) = DEA SPECIAL HDLG (50,3)
   ^TMP($J,LIST,PSSIEN,4) = MAXIMUM DOSE PER DAY (50,4)
   ^TMP($J,LIST,PSSIEN,5) = STANDARD SIG (50,5)
   ^TMP($J,LIST,PSSIEN,6) = FSN (50,6)
   ^TMP($J,LIST,PSSIEN,8) = WARNING LABEL (50,8)
   ^TMP($J,LIST,PSSIEN,12) = ORDER UNIT (50,12)^ABBREVIATION
   (51.5,.01)^EXPANSION (51.5,.02)
   ^TMP($J,LIST,PSSIEN,13) = PRICE PER ORDER UNIT (50,13)
   ^TMP($J,LIST,PSSIEN,14.5) = DISPENSE UNIT (50,14.5)
   ^TMP($J,LIST,PSSIEN,15) = DISPENSE UNITS PER ORDER UNIT (50,15)
   ^TMP($J,LIST,PSSIEN,16) = PRICE PER DISPENSE UNIT (50,16)
   ^TMP($J,LIST,PSSIEN,20) = NATIONAL DRUG FILE ENTRY (50,20)^NAME (50.6,.01)
   ^TMP($J,LIST,PSSIEN,21) = VA PRODUCT NAME (50,21)
   ^TMP($J,LIST,PSSIEN,22) = PSNDF VA PRODUCT NAME ENTRY
   (50,22)^NAME(50.68,.01)
   ^TMP($J,LIST,PSSIEN,25) = NATIONAL DRUG CLASS (50,25)^CODE
   (50.605,.01)^CLASSIFICATION (50.605,1)
   ^TMP($J,LIST,PSSIEN,27) = CMOP ID (50,27)
   ^TMP($J,LIST,PSSIEN,31) = NDC (50,31)
   ^TMP($J,LIST,PSSIEN,40) = ACTION PROFILE MESSAGE (50,40)
   ^TMP($J,LIST,PSSIEN,51) = LOCAL NON-FORMULARY (50,51)^External format for
   the set of codes
   ^TMP($J,LIST,PSSIEN,52) = VISN NON-FORMULARY (50,52)^External format for
   the set of codes
   ^TMP($J,LIST,PSSIEN,63) = APPLICATION PACKAGES’ USE (50,63)
   ^TMP($J,LIST,PSSIEN,64) = PRIMARY DRUG (50,64)^NAME (50.3,.01)
   ^TMP($J,LIST,PSSIEN,100) = INACTIVE DATE (50,100)^External format (ex: SEP
   12,1999)
   ^TMP($J,LIST,PSSIEN,101) = MESSAGE (50,101)
   ^TMP($J,LIST,PSSIEN,102) = RESTRICTION (50,102)
   ^TMP($J,LIST,PSSIEN,301) = AR/WS AMIS CATEGORY (50,301)^External format for
   the set of codes
   ^TMP($J,LIST,PSSIEN,302) = AR/WS AMIS CONVERSION NUMBER (50,302)
   ^TMP($J,LIST,PSSIEN,400) = SERVICE CODE (50,400)
   ^TMP($J,LIST,PSSIEN,"FRM",0) = Total entries returned for this sub-file or
   -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,"FRM",PSS(1),2) = FORMULARY ALTERNATIVE
   (50.065,.01)^GENERIC NAME (50,.01)
   ^TMP($J,LIST,PSSIEN,"OLD",0) = Total entries returned for this sub-file or
   -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,"OLD",PSS(1),.01) = OLD NAMES (50.01,.01)
   ^TMP($J,LIST,PSSIEN,"OLD",PSS(1),.02) = DATE CHANGED (50.01,.02)^External
   format (ex: SEP 12,1999)
   ^TMP($J,LIST,PSSIEN,"SYN",0) = Total entries returned for this sub-file or
   -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,"SYN",PSS(1),.01) = SYNONYM (50.1,.01)
   ^TMP($J,LIST,PSSIEN,"SYN",PSS(1),1) = INTENDED USE (50.1,1)^External format
   for the set of codes
   ^TMP($J,LIST,PSSIEN,"SYN",PSS(1),2) = NDC CODE (50.1,2)
   ^TMP($J,LIST,PSSIEN,"SYN",PSS(1),403) = DISPENSE UNITS PER ORDER UNIT
   (50.1,403)
   ^TMP($J,LIST,"B",GENERIC NAME,PSSIEN) = ""
Where:
```

☛ Note: PSS(1) is the IEN of the multiple it referenced

☛ Note: For the SERVICE CODE field (#400) entry returned, it will be retrieved in one of three ways:

1) If the DRUG entry from the DRUG file (#50) is matched to the VA PRODUCT file (#50.68), and there is data in the SERVICE CODE field (#2000) of the VA PRODUCT file (#50.68) of that match, the SERVICE CODE field will be retrieved from the SERVICE CODE field (#2000) of the VA PRODUCT file (#50.68).
2) If no SERVICE CODE data is found in the SERVICE CODE field (#2000) of the VA PRODUCT file (#50.68), and there is data in the SERVICE CODE field (#400) of the DRUG file (#50), the SERVICE CODE field will be retrieved from the SERVICE CODE field (#400) of the DRUG file (#50).
3) If no SERVICE CODE data is found in the SERVICE CODE field (#2000) of the VA PRODUCT file (#50.68), and in the SERVICE CODE field (#400) of the DRUG file (#50), the value 600000 will be returned as the SERVICE CODE.

Component: INV

Returns the inventory drug fields of the DRUG file (#50) in the array defined by the calling application.

Status: Inactive

```
INV^PSS50(PSSIEN,PSSFT,PSSFL,PSSPK,PSSRTOI,LIST)
   PSSIEN
   PSSFT
   PSSFL
   PSSPK
   PSSRTOI
   LIST
Where:
   PSSIEN = IEN from the DRUG file (#50) [optional]
   PSSFT = GENERIC NAME field (#.01) of the DRUG file (#50) (a value of "??"
   may be used) [optional]
   PSSFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   PSSPK = Returns only those entries containing at least one of the codes in
   the APPLICATION PACKAGES’ USE field (#63) of the DRUG file (#50) (ex: PSSPK
   = "IU" will return all entries for either IV, Unit Dose, or both IV and
   Unit Dose) [optional]
   PSSRTOI = If a "1" is passed in, then only those entries matched to a
   Pharmacy Orderable Item will be returned [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = GENERIC NAME (50,.01)
   ^TMP($J,LIST,PSSIEN,11) = REORDER LEVEL (50,11)
   ^TMP($J,LIST,PSSIEN,12) = ORDER UNIT (50,12)^ABBREVIATION
   (51.5,.01)^EXPANSION (51.5,.02)
   ^TMP($J,LIST,PSSIEN,13) = PRICE PER ORDER UNIT (50,13)
   ^TMP($J,LIST,PSSIEN,14) = NORMAL AMOUNT TO ORDER (50,14)
   ^TMP($J,LIST,PSSIEN,14.5) = DISPENSE UNIT (50,14.5)
   ^TMP($J,LIST,PSSIEN,15) = DISPENSE UNITS PER ORDER UNIT (50,15)
   ^TMP($J,LIST,PSSIEN,16) = PRICE PER DISPENSE UNIT (50,16)
   ^TMP($J,LIST,PSSIEN,17) = SOURCE OF SUPPLY (50,17)
   ^TMP($J,LIST,PSSIEN,17.1) = EXPIRATION DATE (50,17.1)
   ^TMP($J,LIST,PSSIEN,50) = CURRENT INVENTORY (50,50)
   ^TMP($J,LIST,PSSIEN,"IFC",0) = Total entries returned for this sub-file or
   -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,"IFC",PSS(1),.01) = ITEM NUMBER (50.0441,.01)
   ^TMP($J,LIST,PSSIEN,"SYN",0) = Total entries returned for this sub-file or
   -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,"SYN",PSS(1),.01) = SYNONYM (50.1,.01)
   ^TMP($J,LIST,PSSIEN,"SYN",PSS(1),1) = INTENDED USE (50.1,1)^External format
   for the set of codes
   ^TMP($J,LIST,PSSIEN,"SYN",PSS(1),2) = NDC CODE (50.1,2)
   ^TMP($J,LIST,PSSIEN,"SYN",PSS(1),400) = VSN (50.1,400)
   ^TMP($J,LIST,PSSIEN,"SYN",PSS(1),401) = ORDER UNIT (50.1,401)^ABBREVIATION
   (51.5,.01)^EXPANSION (51.5,.02)
   ^TMP($J,LIST,PSSIEN,"SYN",PSS(1),402) = PRICE PER ORDER UNIT (50.1,402)
   ^TMP($J,LIST,PSSIEN,"SYN",PSS(1),403) = DISPENSE UNITS PER ORDER UNIT
   (50.1,403)
   ^TMP($J,LIST,PSSIEN,"SYN",PSS(1),404) = PRICE PER DISPENSE UNIT (50.1,404)
   ^TMP($J,LIST,PSSIEN,"SYN",PSS(1),405) = VENDOR (50.1,405)
   ^TMP($J,LIST,"B",GENERIC NAME,PSSIEN) = ""
Where:
   PSS(1) is the IEN of the multiple it referenced
```

Component: CMOP

Returns the CMOP drug fields of the DRUG file (#50) in the array defined by the calling application.

Status: Inactive

```
CMOP^PSS50(PSSIEN,PSSFT,PSSFL,PSSPK,PSSRTOI,LIST)
Input:
   PSSIEN
   PSSFT
   PSSFL
   PSSPK
   PSSRTOI
   LIST
Where:
   PSSIEN = IEN from the DRUG file (#50) [optional]
   PSSFT = GENERIC NAME field (#.01) of the DRUG file (#50) (a value of "??"
   may be used) [optional]
   PSSFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   PSSPK = Returns only those entries containing at least one of the codes in
   the APPLICATION PACKAGES’ USE field (#63) of the DRUG file (#50) (ex: PSSPK
   = "IU" will return all entries for either IV, Unit Dose, or both IV and
   Unit Dose) [optional]
   PSSRTOI = If a "1" is passed in, then only those entries matched to a
   Pharmacy Orderable Item will be returned [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = GENERIC NAME (50,.01)
   ^TMP($J,LIST,PSSIEN,28) = OP EXTERNAL DISPENSE (50,28)^External format for
   the set of codes
   ^TMP($J,LIST,PSSIEN,213) = CMOP DISPENSE (50,213)^External format for the
   set of codes
   ^TMP($J,LIST,PSSIEN,215) = QUANTITY DISPENSE MESSAGE (50,215)
   ^TMP($J,LIST,PSSIEN,"AL",0) = Total entries returned for this sub-file or -
   1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,"AL",PSS(1),.01) = ACTIVITY LOG (50.0214,.01)^External
   format (ex: SEP 12,1999)
   ^TMP($J,LIST,PSSIEN,"AL",PSS(1),1) = REASON (50.0214,1)^External format for
   the set of codes
   ^TMP($J,LIST,PSSIEN,"AL",PSS(1),2) = INITIATOR OF ACTIVITY (50.0214,2)^
   NAME(200,.01)
   ^TMP($J,LIST,PSSIEN,"AL",PSS(1),3) = FIELD EDITED (50.0214,3)
   ^TMP($J,LIST,PSSIEN,"AL",PSS(1),4) = NEW VALUE (50.0214,4)
   ^TMP($J,LIST,PSSIEN,"AL",PSS(1),5) = NDF UPDATE (50.0214,5)
   ^TMP($J,LIST,"B",GENERIC NAME,PSSIEN) = ""
Where:
   PSS(1) is the IEN of entry in the ACTIVITY LOG multiple
```

Component: DRG

Returns the drug fields of the DRUG file (#50) in the array defined by the calling application.

Status: Inactive

```
DRG^PSS50(PSSIEN,PSSFT,PSSFL,PSSPK,PSSRTOI,LIST)
Input:
   PSSIEN
   PSSFT
   PSSFL
   PSSPK
   PSSRTOI
   LIST
Where:
   PSSIEN = IEN from the DRUG file (#50) [optional]
   PSSFT = GENERIC NAME field (#.01) of the DRUG file (#50) (a value of "??"
   may be used) [optional]
   PSSFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   PSSPK = Returns only those entries containing at least one of the codes in
   the APPLICATION PACKAGES’ USE field (#63) of the DRUG file (#50) (ex: PSSPK
   = "IU" will return all entries for either IV, Unit Dose, or both IV and
   Unit Dose) [optional]
   PSSRTOI – If "1" returns only those entries matched to a Pharmacy Orderable
   Item [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = GENERIC NAME (50,.01)
   ^TMP($J,LIST,PSSIEN,62.01) = DAY (nD) or DOSE (nL) LIMIT (50,62.01)
   ^TMP($J,LIST,PSSIEN,62.02) = UNIT DOSE MED ROUTE (50,62.02)^NAME (51.2,.01)
   ^TMP($J,LIST,PSSIEN,62.03) = UNIT DOSE SCHEDULE TYPE (50,62.03)^External
   format for the set of codes
   ^TMP($J,LIST,PSSIEN,62.04) = UNIT DOSE SCHEDULE (50,62.04)
   ^TMP($J,LIST,PSSIEN,62.05) = CORRESPONDING OUTPATIENT DRUG
   (50,62.05)^GENERIC NAME (50,.01)
   ^TMP($J,LIST,PSSIEN,905) = CORRESPONDING INPATIENT DRUG (50,905)^GENERIC
   NAME (50,.01)
   ^TMP($J,LIST,"B",GENERIC NAME,PSSIEN) = ""
```

Component: ATC

Returns the ATC drug fields of the DRUG file (#50) in the array defined by the calling application.

Status: Inactive

```
ATC^PSS50(PSSIEN,PSSFT,PSSFL,PSSPK,LIST)
Input:
   PSSIEN
   PSSFT
   PSSFL
   PSSPK
   LIST
Where:
   PSSIEN = IEN from the DRUG file (#50) [optional]
   PSSFT = GENERIC NAME field (#.01) of the DRUG file (#50) (a value of "??"
   may be used) [optional]
   PSSFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   PSSPK = Returns only those entries containing at least one of the codes in
   the APPLICATION PACKAGES’ USE field (#63) of the DRUG file (#50) (ex: PSSPK
   = "IU" will return all entries for either IV, Unit Dose, or both IV and
   Unit Dose) [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = GENERIC NAME (50,.01)
   ^TMP($J,LIST,PSSIEN,212.2) = ATC MNEMONIC (50,212.2)
   ^TMP($J,LIST,PSSIEN,"ATC",0) = Total entries returned for this sub-file or
   -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,"ATC",PSS(1),.01) = WARD GROUP FOR ATC CANISTER
   (50.0212,.01)^NAME (57.5,.01)
   ^TMP($J,LIST,PSSIEN,"ATC",PSS(1),1) = ATC CANISTER (50.0212,1)
   ^TMP($J,LIST,"C",GENERIC NAME,PSSIEN) = ""
Where:
   PSS(1) is the IEN of entry in the ATC CANISTER multiple
```

Component: LAB

Returns the lab test monitor drug fields of the DRUG file (#50) in the array defined by the calling application.

Status: Active

```
LAB^PSS50(PSSIEN,PSSFT,PSSFL,PSSPK,PSSRTOI,LIST)
Input:
   PSSIEN
   PSSFT
   PSSFL
   PSSPK
   PSSRTOI
   LIST
Where:
   PSSIEN = IEN from the DRUG file (#50) [optional]
   PSSFT = GENERIC NAME field (#.01) of the DRUG file (#50) (a value of "??"
   may be used) [optional]
   PSSFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   PSSPK = Returns only those entries containing at least one of the codes in
   the APPLICATION PACKAGES’ USE field (#63) of the DRUG file (#50) (ex: PSSPK
   = "IU" will return all entries for either IV, Unit Dose, or both IV and
   Unit Dose) [optional]
   PSSRTOI – If "1" returns only those entries matched to a Pharmacy Orderable
   Item [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = GENERIC NAME (50,.01)
   ^TMP($J,LIST,PSSIEN,17.2) = LAB TEST MONITOR (50,17.2)^NAME (60,.01)
   ^TMP($J,LIST,PSSIEN,17.3) = MONITOR MAX DAYS (50,17.3)
   ^TMP($J,LIST,PSSIEN,17.4) = SPECIMEN TYPE (50,17.4)^NAME (61,.01)
   ^TMP($J,LIST,PSSIEN,17.5) = MONITOR ROUTINE (50,17.5)
   ^TMP($J,LIST,PSSIEN,17.6) = LAB MONITOR MARK (50,17.6)^External format for
   the set of codes
   ^TMP($J,LIST,"B",GENERIC NAME,PSSIEN) = ""
```

Component: CLOZ

Returns the Clozapine lab test monitor drug fields of the DRUG file (#50) in the array defined by the calling application.

Status: Active

```
CLOZ^PSS50(PSSIEN,PSSFT,PSSFL,PSSPK,PSSRTOI,LIST)
Input:
   PSSIEN
   PSSFT
   PSSFL
   PSSPK
   PSSRTOI
   LIST
Where:
   PSSIEN = IEN from the DRUG file (#50) [optional]
   PSSFT = GENERIC NAME field (#.01) of the DRUG file (#50) (a value of "??"
   may be used) [optional]
   PSSFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   PSSPK = Returns only those entries containing at least one of the codes in
   the APPLICATION PACKAGES' USE field (#63) of the DRUG file (#50) (ex: PSSPK
   = "IU" will return all entries for either IV, Unit Dose, or both IV and
   Unit Dose) [optional]
   PSSRTOI – If "1" returns only those entries matched to a Pharmacy Orderable
   Item [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = GENERIC NAME (50,.01)
   ^TMP($J,LIST,PSSIEN,"CLOZ",0) = Total entries returned for this sub-file or
   -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,"CLOZ",PSS(1),.01) = LAB TEST MONITOR
   (50.02,.01)^NAME(60,.01)
   ^TMP($J,LIST,PSSIEN,"CLOZ",PSS(1),1) = MONITOR MAX DAYS (50.02,1)
   ^TMP($J,LIST,PSSIEN,"CLOZ",PSS(1),2) = SPECIMEN TYPE (50.02,2)^NAME
   (61,.01)
   ^TMP($J,LIST,PSSIEN,"CLOZ",PSS(1),3) = TYPE OF TEST (50.02,3)^External
   format for the set of codes
   ^TMP($J,LIST,"B",GENERIC NAME,PSSIEN) = ""
Where:
   PSS(1) is the IEN of entry in the CLOZAPINE LAB TEST multiple
```

Component: DOSE

Returns the dosage related fields of the DRUG file (#50) in the array defined by the calling application.

Status: Inactive

```
DOSE^PSS50(PSSIEN,PSSFT,PSSFL,PSSPK,PSSRTOI,LIST)
Input:
   PSSIEN
   PSSFT
   PSSFL
   PSSPK
   PSSRTOI
   LIST
Where:
   PSSIEN = IEN from the DRUG file (#50) [optional]
   PSSFT = GENERIC NAME field (#.01) of the DRUG file (#50) (a value of "??"
   may be used) [optional]
   PSSFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   PSSPK = Returns only those entries containing at least one of the codes in
   the APPLICATION PACKAGES’ USE field (#63) of the DRUG file (#50) (ex: PSSPK
   = "IU" will return all entries for either IV, Unit Dose, or both IV and
   Unit Dose) [optional]
   PSSRTOI – If "1" returns only those entries matched to a Pharmacy Orderable
   Item [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = GENERIC NAME (50,.01)
   ^TMP($J,LIST,PSSIEN,901) = STRENGTH (50,901)
   ^TMP($J,LIST,PSSIEN,902) = UNIT (50,902)^NAME (50.607,.01)
   ^TMP($J,LIST,PSSIEN,"LOC",0) = Total entries returned for this sub-file or
   -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,"LOC",PSS(1),.01) = LOCAL POSSIBLE DOSAGE (50.0904,.01)
   ^TMP($J,LIST,PSSIEN,"LOC",PSS(1),1) = PACKAGE (50.0904,1)^External format
   for the set of codes
   ^TMP($J,LIST,PSSIEN,"LOC",PSS(1),2) = BCMA UNITS PER DOSE (50.0904,2)
   ^TMP($J,LIST,PSSIEN,"LOC",PSS(1),3) = OTHER LANGUAGE DOSAGE NAME
   (50.0904,3)
   ^TMP($J,LIST,PSSIEN,"POS",0) = Total entries returned for this sub-file or
   -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,"POS",PSS(1),.01) = DISPENSE UNITS PER DOSE
   (50.0903,.01)
   ^TMP($J,LIST,PSSIEN,"POS",PSS(1),1) = DOSE (50.0903,1)
   ^TMP($J,LIST,PSSIEN,"POS",PSS(1),2) = PACKAGE (50.0903,2)^External format
   ^TMP($J,LIST,PSSIEN,"POS",PSS(1),3) = BCMA UNITS PER DOSE(50.0903,3)
   ^TMP($J,LIST,"B",GENERIC NAME,PSSIEN) = ""
Where:
   PSS(1) is the IEN of the multiple it referenced
```

Component: NDF

Returns the National Drug File (NDF) drug fields of the DRUG file (#50) in the array defined by the calling application.

Status: Active

```
NDF^PSS50(PSSIEN,PSSFT,PSSFL,PSSPK,PSSRTOI,LIST)
Input:
   PSSIEN
   PSSFT
   PSSFL
   PSSPK
   PSSRTOI
   LIST
Where:
   PSSIEN = IEN from the DRUG file (#50) [optional]
   PSSFT = GENERIC NAME field (#.01) of the DRUG file (#50) (a value of "??"
   may be used) [optional]
   PSSFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   PSSPK = Returns only those entries containing at least one of the codes in
   the APPLICATION PACKAGES’ USE field (#63) of the DRUG file (#50) (ex: PSSPK
   = "IU" will return all entries for either IV, Unit Dose, or both IV and
   Unit Dose) [optional]
   PSSRTOI – If "1" returns only those entries matched to a Pharmacy Orderable
   Item [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = GENERIC NAME (50,.01)
   ^TMP($J,LIST,PSSIEN,20) = NATIONAL DRUG FILE ENTRY (50,20)^NAME (50.6,.01)
   ^TMP($J,LIST,PSSIEN,21) = VA PRODUCT NAME (50,21)
   ^TMP($J,LIST,PSSIEN,22) = PSNDF VA PRODUCT NAME ENTRY
   (50,22)^NAME(50.68,.01)
   ^TMP($J,LIST,PSSIEN,23) = PACKAGE SIZE (50,23)^NAME (50.609,.01)
   ^TMP($J,LIST,PSSIEN,24) = PACKAGE TYPE (50,24)^NAME (50.608,.01)
   ^TMP($J,LIST,PSSIEN,25) = NATIONAL DRUG CLASS (50,25)^CODE
   (50.605,.01)^CLASSIFICATION (50.605,1)
   ^TMP($J,LIST,PSSIEN,27) = CMOP ID (50,27)
   ^TMP($J,LIST,PSSIEN,29) = NATIONAL FORMULARY INDICATOR (50,29)^External
   format for the set of codes
   ^TMP($J,LIST,"B",GENERIC NAME,PSSIEN) = ""
```

Component: WS

Returns the ward stock drug fields of the DRUG file (#50) in the array defined by the calling application.

Status: Inactive

```
WS^PSS50(PSSIEN,PSSFT,PSSFL,PSSPK,PSSRTOI,LIST)
Input:
   PSSIEN
   PSSFT
   PSSFL
   PSSPK
   PSSRTOI
   LIST
Where:
   PSSIEN = IEN from the DRUG file (#50) [optional]
   PSSFT = GENERIC NAME field (#.01) of the DRUG file (#50) (a value of "??"
   may be used) [optional]
   PSSFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   PSSPK = Returns only those entries containing at least one of the codes in
   the APPLICATION PACKAGES’ USE field (#63) of the DRUG file (#50) (ex: PSSPK
   = "IU" will return all entries for either IV, Unit Dose, or both IV and
   Unit Dose) [optional]
   PSSRTOI = If a "1" is passed in, then only those entries matched to a
   Pharmacy Orderable Item will be returned [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = GENERIC NAME (50,.01)
   ^TMP($J,LIST,PSSIEN,300) = INPATIENT PHARMACY LOCATION (50,300)
   ^TMP($J,LIST,PSSIEN,301) = AR/WS AMIS CATEGORY (50,301)^External format for
   the set of codes
   ^TMP($J,LIST,PSSIEN,302) = AR/WS AMIS CONVERSION NUMBER (50,302)
   ^TMP($J,LIST,"B",GENERIC NAME,PSSIEN) = ""
```

Component: ARWS

Returns fields utilized by the Automatic Replenishment/Ward Stock extract in the Pharmacy Benefits Management (PBM) application, in the DRUG file (#50) in the array defined by the calling application.

Status: Active

```
ARWS^PSS50(PSSIEN,PSSFT,LIST)
Input:
   PSSIEN
   PSSFT
   LIST
Where:
   PSSIEN = IEN of entry in DRUG file (#50) [optional]
   PSSFT = GENERIC NAME field (#.01) of the DRUG file (#50) (a value of "??"
   may be used) [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = GENERIC NAME (50,.01)
   ^TMP($J,LIST,PSSIEN,2) = VA CLASSIFICATION (50,2)
   ^TMP($J,LIST,PSSIEN,3) = DEA SPECIAL HDLG (50,3)
   ^TMP($J,LIST,PSSIEN,12) = ORDER UNIT (50,12)^ABBREVIATION
   (51.5,.01)^EXPANSION (51.5,.02)
   ^TMP($J,LIST,PSSIEN,13) = PRICE PER ORDER UNIT (50,13)
   ^TMP($J,LIST,PSSIEN,14.5) = DISPENSE UNIT (50,14.5)
   ^TMP($J,LIST,PSSIEN,15) = DISPENSE UNITS PER ORDER UNIT (50,15)
   ^TMP($J,LIST,PSSIEN,16) = PRICE PER DISPENSE UNIT (50,16)
   ^TMP($J,LIST,PSSIEN,20) = NATIONAL DRUG FILE ENTRY (50,20)^NAME (50.6,.01)
   ^TMP($J,LIST,PSSIEN,21) = VA PRODUCT NAME (50,21)
   ^TMP($J,LIST,PSSIEN,22) = PSNDF VA PRODUCT NAME ENTRY
   (50,22)^NAME(50.68,.01)
   ^TMP($J,LIST,PSSIEN,23) = PACKAGE SIZE (50,23)^NAME (50.609,.01)
   ^TMP($J,LIST,PSSIEN,25) = NATIONAL DRUG CLASS (50,25)^CODE
   (50.605,.01)^CLASSIFICATION (50.605,1)
   ^TMP($J,LIST,PSSIEN,31) = NDC (50,31)
   ^TMP($J,LIST,PSSIEN,51) = LOCAL NON-FORMULARY (50,51)^External format for
   the set of codes
   ^TMP($J,LIST,PSSIEN,52) = VISN NON-FORMULARY (50,52)^External format for
   the set of codes
   ^TMP($J,LIST,PSSIEN,301) = AR/WS AMIS CATEGORY (50,301)^External format for
   the set of codes
   ^TMP($J,LIST,PSSIEN,302) = AR/WS AMIS CONVERSION NUMBER (50,302)
   ^TMP($J,LIST,"B",GENERIC NAME,PSSIEN) = ""
```

Component: CSYN

Returns the SYNONYM subfield (#.01) of the SYNONYM multiple of the DRUG file (#50).

Status: Inactive

```
CSYN^PSS50(PSSIEN,PSSVAL,LIST)
Input:
   PSSIEN
   PSSVAL
   LIST
Where:
   PSSIEN = IEN of entry in DRUG file (#50) [required]
   PSSVAL = SYNONYM subfield (#.01) of the SYNONYM multiple (#9) of the DRUG
   file (#50) [required]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = GENERIC NAME (50,.01)
   ^TMP($J,LIST,PSSIEN,"SYN",0) = Total entries returned for this sub-file or
   -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,PSS(1),.01) = SYNONYM (50.1,.01)
   ^TMP($J,LIST,PSSIEN,"SYN",PSS(1),403) = DISPENSE UNITS PER ORDER
   UNIT(50.1,403)
   ^TMP($J,LIST,"C",SYNONYM,PSSIEN) = ""
Where:
   PSS(1) is the IEN of the SYNONYM multiple (#9)
```

Component: MRTN

Returns the monitor routine drug fields of the DRUG file (#50) in the array defined by the calling application.

Status: Inactive

```
MRTN^PSS50(PSSIEN,PSSFT,PSSFL,PSSPK,PSSRTOI,LIST)
Input:
   PSSIEN
   PSSFT
   PSSFL
   PSSPK
   PSSRTOI
   LIST
Where:
   PSSIEN = IEN from the DRUG file (#50) [optional]
   PSSFT = GENERIC NAME field (#.01)of the DRUG file (#50) (a value of "??"
   may be used) [optional]
   PSSFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   PSSPK = Returns only those entries containing at least one of the codes in
   the APPLICATION PACKAGES’ USE field (#63) of the DRUG file (#50) (ex: PSSPK
   = "IU" will return all entries for either IV, Unit Dose, or both IV and
   Unit Dose) [optional]
   PSSRTOI = If a "1" is passed in, then only those entries matched to a
   Pharmacy Orderable Item will be returned [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = GENERIC NAME (50,.01)
   ^TMP($J,LIST,PSSIEN,17.2) = LAB TEST MONITOR (50,17.2)^NAME (60,.01)
   ^TMP($J,LIST,PSSIEN,17.5) = MONITOR ROUTINE (50,17.5)
   ^TMP($J,LIST,PSSIEN,31) = NDC (50,31)
   ^TMP($J,LIST,"B",GENERIC NAME,PSSIEN) = ""
```

Component: DSPUNT

Returns the DISPENSE UNITS PER ORDER UNIT field (#403) of the SYNONYM multiple of the DRUG file (#50) in the array defined by the calling application.

Status: Inactive

```
DSPUNT^PSS50(PSSIEN,PSSIEN2,LIST)
Input:
   PSSIEN
   PSSIEN2
   LIST
Where:
   PSSIEN = IEN of entry in DRUG file (#50) [required]
   PSSIEN2 = IEN of entry in SYNONYM multiple (#9) of the DRUG file (#50)
   [required]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = GENERIC NAME (50,.01)
   ^TMP($J,LIST,PSSIEN,"SYN",0) = Total entries returned for this sub-file or
   -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,PSS(1),.01) = SYNONYM (50.1,.01)
   ^TMP($J,LIST,PSSIEN,"SYN",PSS(1),403) = DISPENSE UNITS PER ORDER
   UNIT(50.1,403)
   ^TMP($J,LIST,"C",SYNONYM,PSSIEN) = ""
Where:
   PSS(1) is the IEN of the SYNONYM multiple (#9)
```

Component: ZERO

Returns the zero node in the DRUG file (#50) in the array defined by the calling application.

Status: Active

```
ZERO^PSS50(PSSIEN,PSSFT,PSSFL,PSSPK,PSSRTOI,LIST)
Input:
   PSSIEN
   PSSFT
   PSSFL
   PSSPK
   PSSRTOI
   LIST
Where:
   PSSIEN = IEN from the DRUG file (#50) [optional]
   PSSFT = GENERIC NAME field (#.01) of the DRUG file (#50) (a value of "??"
   may be used) [optional]
   PSSFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   PSSPK = Returns only those entries containing at least one of the codes in
   the APPLICATION PACKAGES’ USE field (#63) of the DRUG file (#50) (ex: PSSPK
   = "IU" will return all entries for either IV, Unit Dose, or both IV and
   Unit Dose) [optional]
   PSSRTOI = If passed in a 1, return only those entries matched to a Pharmacy
   Orderable Item [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = GENERIC NAME (50,.01)
   ^TMP($J,LIST,PSSIEN,2) = VA CLASSIFICATION (50,2)
   ^TMP($J,LIST,PSSIEN,3) = DEA SPECIAL HDLG (50,3)
   ^TMP($J,LIST,PSSIEN,4) = MAXIMUM DOSE PER DAY (50,4)
   ^TMP($J,LIST,PSSIEN,5) = STANDARD SIG (50,5)
   ^TMP($J,LIST,PSSIEN,6) = FSN (50,6)
   ^TMP($J,LIST,PSSIEN,8) = WARNING LABEL (50,8)
   ^TMP($J,LIST,PSSIEN,51) = LOCAL NON-FORMULARY (50,51)^External format for
   the set of codes
   ^TMP($J,LIST,PSSIEN,52) = VISN NON-FORMULARY (50,52)^External format for
   the set of codes
   ^TMP($J,LIST,PSSIEN,101) = MESSAGE (50,101)
   ^TMP($J,LIST,"B",GENERIC NAME,PSSIEN) = ""
```

Component: NOCMOP

Returns the IEN(s) in the DRUG file (#50) in the array defined by the calling application.

Status: Inactive

```
S X=$$NOCMOP^PSS50(PSSIEN,PSSFL)
Input:
   PSSIEN
   PSSFL
Where:
   PSSIEN = IEN from the DRUG file (#50) [optional]
   PSSFL = Consider true for one of the screening conditions:
   If PSSFL = 1 and the CMOP DISPENSE field (#213) is defined with a value of
   0 (zero)
   If PSSFL = 0 or not passed in and the CMOP DISPENSE field (#213) is not
   defined [optional]
Output:
   $$NOCMOP
Where $$NOCMOP is:
   0 (zero) if conditions are not met
   1 (one) condition is met
```

Component: MSG

Returns the IEN(s) of the DRUG file (#50) if there is data in the QUANTITY DISPENSE MESSAGE field (#215) in the array defined by the calling application.

Status: Inactive

```
MSG^PSS50(LIST)
Input:
   LIST
Where:
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = GENERIC NAME (50,.01)
   ^TMP($J,LIST,PSSIEN,100) = INACTIVE DATE (50,100)^External format (ex: SEP
   12,1999)
   ^TMP($J,LIST,"B",GENERIC NAME,PSSIEN) = ""
Where:
   PSSIEN is IEN of entry in the DRUG file (#50)
```

Component: IEN

Returns the IEN(s) of the DRUG file (#50) in the array defined by the calling application.

Status: Inactive

```
IEN^PSS50(LIST)
Input:
   LIST
Where:
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = GENERIC NAME (50,.01)
   ^TMP($J,LIST,"AIU",GENERIC NAME,PSSIEN) = ""
Where:
   PSSIEN is IEN of entry in the DRUG file (#50)
```

Component: B

Returns the IEN(s) of the DRUG file (#50) in the array defined by the calling application. The “B” cross-reference in the format of ^PSDRUG(“B”,NAME(50,.01),IEN(50)) will be used.

Status: Inactive

```
B^PSS50(PSSFT,PSSFL,PSSPK,PSSRTOI,LIST)
Input:
   PSSFT
   PSSFL
   PSSPK
   PSSRTOI
   LIST
Where:
   PSSFT = GENERIC NAME field (#.01) of the DRUG file (#50) (a value of "??"
   may be used) [optional]
   PSSFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   PSSPK = Returns only those entries containing at least one of the codes in
   the APPLICATION PACKAGES’ USE field (#63) of the DRUG file (#50) (ex: PSSPK
   = "IU" will return all entries for either IV, Unit Dose, or both IV and
   Unit Dose) [optional]
   PSSRTOI = If passed in a 1, return only those entries matched to a Pharmacy
   Orderable Item [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = GENERIC NAME (50,.01)
   ^TMP($J,LIST,"B",GENERIC NAME,PSSIEN) = ""
Where:
   PSSIEN is IEN of entry in the DRUG file (#50)
```

Component: SKB

Will set and kill the “B” cross-reference of the DRUG file (#50) in the array defined by the calling application.

Status: Inactive

```
S X=$$SKB^PSS50(PSSIEN,PSSFL)
Input:
   PSSIEN
   PSSFL
Where:
   PSSIEN = IEN of entry in DRUG file (#50) [required]
   PSSFL = Either "S" (set) or "K" (kill) action [required]
Output:
   $$SKB
Where $$SKB is:
   0 (zero) action unsuccessful
   1 (one) action is accomplished
```

Component: VAC

Returns the IEN(s) and GENERIC NAME field (#.01) in the array defined by the calling application. The “VAC” cross-reference in the format of ^PSDRUG(“VAC”,NATIONAL DRUG CLASS(50,25),IEN(50)) will be used.

Status: Active

```
VAC^PSS50(PSSVAL,PSSFL,PSSPK,LIST)
Input:
   PSSVAL = the NATIONAL DRUG CLASS field (#25) of the DRUG file (#50)
   [required]
   PSSFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   PSSPK = Returns only those entries containing at least one of the codes in
   the APPLICATION PACKAGES’ USE field (#63) of the DRUG file (#50) (ex: PSSPK
   = "IU" will return all entries for either IV, Unit Dose, or both IV and
   Unit Dose) [optional]
   LIST = the array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = GENERIC NAME (50,.01)
   ^TMP($J,LIST,"VAC",GENERIC NAME,PSSIEN) = ""
```

☛ Note: The "VAC" cross-reference in the format of ^PSDRUG("VAC",NATIONAL DRUG CLASS (50,25),IEN(50) will be used for the lookup.

```
Where:
   PSSIEN is IEN of entry in the DRUG file (#50)
```

Component: NDC

Returns the IEN(s) and GENERIC NAME field (#.01) in the array defined by the calling application. The “NDC” cross-reference in the format of ^PSDRUG(“NDC”,NDC(50,31),IEN(50)) will be used.

Status: Active

```
NDC^PSS50(PSSVAL,PSSFL,PSSPK,LIST)
Input:
   PSSVAL
   PSSFL
   PSSPK
   LIST
Where:
   PSSVAL = NDC field (#31) of the DRUG file (#50) (ex: "053905099101"
   (without dashes) as being used in the "NDC" cross-reference) [required]
```

☛ Note: Use quotes around the PSSVAL value, or use quotes around the value if setting a variable to this value to be used as the parameter, to prevent leading zeros from being truncated.

```
   PSSFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   PSSPK = Returns only those entries containing at least one of the codes in
   the APPLICATION PACKAGES’ USE field (#63) of the DRUG file (#50) (ex: PSSPK
   = "IU" will return all entries for either IV, Unit Dose, or both IV and
   Unit Dose) [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = GENERIC NAME (50,.01)
   ^TMP($J,LIST,PSSIEN,"NDC",GENERIC NAME,PSSIEN) = ""
Where:
   PSSIEN is IEN of entry in the DRUG file (#50)
```

☛ Note: The "NDC" cross-reference in the format of ^PSDRUG("NDC",NDC (50,31),IEN(50)) will be used for the lookup.

Component: ASP

Returns the IEN(s) and GENERIC NAME field (#.01) in the DRUG file (#50) in the array defined by the calling application. The “ASP” cross-reference in the format of ^PSDRUG(“ASP”,PHARMACY ORDERALBE ITEM(50,2.1),IEN(50)) will be used.

Status: Active

```
ASP^PSS50(PSSVAL,PSSFL,PSSPK,LIST)
Input:
   PSSVAL
   PSSFL
   PSSPK
   LIST
Where:
   PSSVAL = PHARMACY ORDERABLE ITEM field (#2.1) of the DRUG file (#50)
   [required]
   PSSFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   PSSPK = Returns only those entries containing at least one of the codes in
   the APPLICATION PACKAGES’ USE field (#63) of the DRUG file (#50) (ex: PSSPK
   = "IU" will return all entries for either IV, Unit Dose, or both IV and
   Unit Dose) [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = GENERIC NAME (50,.01)
   ^TMP($J,LIST,"ASP",GENERIC NAME,PSSIEN) = ""
Where:
   PSSIEN is IEN of entry in the DRUG file (#50)
```

☛ Note: The "ASP" cross-reference in the format of ^PSDRUG("ASP",PHARMACY ORDERABLE ITEM (50,2.1),IEN(50)) will be used for the lookup.

Component: AND

Returns the IEN(s) and GENERIC NAME field (#.01) in the DRUG file (#50) in the array defined by the calling application. The “AND” cross-reference in the format of ^PSDRUG(“AND”,NATIONAL DRUG FILE ENTRY(50,20),IEN(50)) will be used.

Status: Active

```
AND^PSS50(PSSVAL,PSSFL,PSSPK,LIST)
Input:
   PSSVAL
   PSSFL
   PSSPK
   LIST
Where:
   PSSVAL = NATIONAL DRUG FILE ENTRY field (#20) of the DRUG file (#50)
   [required]
   PSSFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   PSSPK = Returns only those entries containing at least one of the codes in
   the APPLICATION PACKAGES’ USE field (#63) of the DRUG file (#50) (ex: PSSPK
   = "IU" will return all entries for either IV, Unit Dose, or both IV and
   Unit Dose) [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = GENERIC NAME (50,.01)
   ^TMP($J,LIST,"AND",GENERIC NAME,PSSIEN) = ""
Where:
   PSSIEN is IEN of entry in the DRUG file (#50)
```

☛ Note: The "AND" cross-reference in the format of ^PSDRUG("AND",NATIONAL DRUG FILE ENTRY (50,20),IEN(50)) will be used for the lookup.

Component: AB

Returns the IEN(s) and GENERIC NAME field (#.01) in the DRUG file (#50) in the array defined by the calling application. External formats will be included if applicable. The “AB” cross-reference in the format of ^PSDRUG(“AB”,ITEM NUMBER(50,.0441,.01),IEN(50),IEN(50.0441)) will be used.

Status: Inactive

```
AB^PSS50(PSSVAL,PSSFL,PSSPK,LIST)
Input:
   PSSVAL
   PSSFL
   PSSPK
   LIST
Where:
   PSSVAL = ITEM NUMBER sub-field (#.01) of the IFCAP ITEM NUMBER multiple in
   the DRUG file (#50) [required]
   PSSFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   PSSPK = Returns only those entries containing at least one of the codes in
   the APPLICATION PACKAGES’ USE field (#63) of the DRUG file (#50) (ex: PSSPK
   = "IU" will return all entries for either IV, Unit Dose, or both IV and
   Unit Dose) [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = GENERIC NAME (50,.01)
   ^TMP($J,LIST,PSSIEN,"IFC",0) = Total entries returned for this sub-file or
   -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,"IFC",PSS(1),.01) = ITEM NUMBER (50.0441,.01)
   ^TMP($J,LIST,"AB",GENERIC NAME,PSSIEN) = ""
Where:
   PSSIEN is IEN of entry in the DRUG file (#50)
   PSS(1) is the IEN of entry in the IFCAP ITEM NUMBER multiple
```

Component: AP

Returns the IEN(s) and GENERIC NAME field (#.01) in the DRUG file (#50) in the array defined by the calling application. The “AP” cross-reference in the format of ^PSDRUG (“AP”,PRIMARY DRUG (50,64),IEN(50)) will be used.

Status: Inactive

```
AP^PSS50(PSSVAL,PSSFL,PSSPK,LIST)
Input:
   PSSVAL
   PSSFL
   PSSPK
   LIST
Where:
   PSSVAL = PRIMARY DRUG field (#64) of the DRUG file (#50) [required]
   PSSFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   PSSPK = Returns only those entries containing at least one of the codes in
   the APPLICATION PACKAGES’ USE field (#63) of the DRUG file (#50) (ex: PSSPK
   = "IU" will return all entries for either IV, Unit Dose, or both IV and
   Unit Dose) [optional]
   LIST = array name defined by the calling application [required]
```

```
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = GENERIC NAME (50,.01)
   ^TMP($J,LIST,"AP",GENERIC NAME,PSSIEN) = ""
Where:
   PSSIEN is IEN of entry in the DRUG file (#50)
```

Component: AOC

Returns the IEN(s) and GENERIC NAME field (#.01) in the DRUG file (#50) in the array defined by the calling application. The “AOC” cross-reference in the format of ^PSDRUG(“AOC”,IEN(50.7),VA CLASSIFICATION (50.2),IEN(50)) will be used.

Status: Inactive

```
AOC^PSS50(PSSVAL,PSSFL,PSSPK,LIST)
Input:
   PSSVAL
   PSSFL
   PSSPK
   LIST
Where:
   PSSVAL = VA CLASSIFICATION field (#2) of the DRUG file (#50) [required]
   PSSFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   PSSPK = Returns only those entries containing at least one of the codes in
   the APPLICATION PACKAGES’ USE field (#63) of the DRUG file (#50) (ex: PSSPK
   = "IU" will return all entries for either IV, Unit Dose, or both IV and
   Unit Dose) [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = GENERIC NAME (50,.01)
   ^TMP($J,LIST,"AOC",GENERIC NAME,PSSIEN) = ""
Where:
   PSSIEN is IEN of entry in the DRUG file (#50)
```

Component: C

Returns the SYNONYM multiple of the DRUG file (#50) in the array defined by the calling application. The “C” cross-reference in the format of ^PSDRUG(“C”,SYNONYM(50.1.,01),IEN(50.1)) will be used.

Status: Inactive

```
C^PSS50(PSSVAL,PSSFL,PSSPK,LIST)
Input:
   PSSVAL
   PSSFL
   PSSPK
   LIST
Where:
   PSSVAL = SYNONYM sub-field (#.01) of the SYNONYM multiple (#9) of the DRUG
   file (#50) [required]
   PSSFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   PSSPK = Returns only those entries containing at least one of the codes in
   the APPLICATION PACKAGES’ USE field (#63) of the DRUG file (#50) (ex: PSSPK
   = "IU" will return all entries for either IV, Unit Dose, or both IV and
   Unit Dose) [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,PSS(1),.01) = SYNONYM (50.1,.01)
   ^TMP($J,LIST,PSSIEN,PSS(1),2) = NDC CODE (50.1,2)
   ^TMP($J,LIST,PSSIEN,PSS(1),3) = INTENDED USE (50.1,3)^External format for
   the set of codes
   ^TMP($J,LIST,PSSIEN,PSS(1),400) = VSN (50.1,400)
   ^TMP($J,LIST,PSSIEN,PSS(1),401) = ORDER UNIT (50.1,401)^ABBREVIATION
   (51.5,.01)
   ^TMP($J,LIST,PSSIEN,PSS(1),402) = PRICE PER ORDER UNIT (50.1,402)
   ^TMP($J,LIST,PSSIEN,PSS(1),403) = DISPENSE UNITS PER ORDER UNIT (50.1,403)
   ^TMP($J,LIST,PSSIEN,PSS(1),404) = PRICE PER DISPENSE UNIT (50.1,404)
   ^TMP($J,LIST,PSSIEN,PSS(1),405) = VENDOR (50.1,405)
   ^TMP($J,LIST,"C",GENERIC NAME,PSSIEN) = ""
Where:
   PSSIEN is IEN of entry in the DRUG file (#50)
   PSS(1) is the IEN of entry in the SYNONYM multiple (#9)
```

Component: AVSN

Returns the GENERIC NAME field (#.01), IEN(s) and SYNONYM multiple of the DRUG file (#50) in the array defined by the calling application. The “AVSN” cross-reference in the format of ^PSDRUG(“AVSN”,VSN(50.1.,400),IEN(50.1),IEN(50)) will be used.

Status: Inactive

```
AVSN^PSS50(PSSVAL,PSSFL,PSSPK,LIST)
Input:
   PSSVAL
   PSSFL
   PSSPK
   LIST
Where:
   PSSVAL = VSN sub-field (#400) of the SYNONYM multiple (#9) of the DRUG file
   (#50) [required]
   PSSFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   PSSPK = Returns only those entries containing at least one of the codes in
   the APPLICATION PACKAGES’ USE field (#63) of the DRUG file (#50) (ex: PSSPK
   = "IU" will return all entries for either IV, Unit Dose, or both IV and
   Unit Dose) [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = GENERIC NAME (50,.01)
   ^TMP($J,LIST,PSSIEN,"SYN",0) = Total entries returned for this sub-file or
   -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,"SYN",PSS(1),.01) = SYNONYM (50.1,.01)
   ^TMP($J,LIST,PSSIEN,"SYN",PSS(1),2) = NDC CODE (50.1,2)
   ^TMP($J,LIST,PSSIEN,"SYN",PSS(1),3) = INTENDED USE (50.1,3)^External format
   for the set of codes
   ^TMP($J,LIST,PSSIEN,"SYN",PSS(1),400) = VSN (50.1,400)
   ^TMP($J,LIST,PSSIEN,"SYN",PSS(1),,401) = ORDER UNIT
   (50.1,401)^ABBREVIATION(51.5,.01)
   ^TMP($J,LIST,PSSIEN,"SYN",PSS(1),402) = PRICE PER ORDER UNIT (50.1,402)
   ^TMP($J,LIST,PSSIEN,"SYN",PSS(1),403) = DISPENSE UNITS PER ORDER UNIT
   (50.1,403)
   ^TMP($J,LIST,PSSIEN,"SYN",PSS(1),404) = PRICE PER DISPENSE UNIT (50.1,404)
   ^TMP($J,LIST,PSSIEN,"SYN",PSS(1),405) = VENDOR (50.1,405)
   ^TMP($J,LIST,"AVSN",GENERIC NAME,PSSIEN) = ""
Where:
   PSSIEN is IEN of entry in the DRUG file (#50)
   PSS(1) is the IEN of entry in the SYNONYM multiple (#9)
```

Component: AQ

Indicates if the IEN passed into the DRUG file (#50) existed in the “AQ” cross-reference. The “AQ” cross-reference in the format of ^PSDRUG(“AQ”,IEN(50)) will be used.

Status: Inactive

```
S X=$$AQ^PSS50(PSSIEN)
Input:
   PSSIEN
Where:
   PSSIEN = IEN of the DRUG in the DRUG file (#50) [required]
Output:
   $$AQ
Where $$AQ is:
   0 (zero) = IEN does not exist in the cross-reference
   1 (one) = IEN exists in the cross-reference
```

Component: SKAQ

Sets and kills the “AQ” cross-reference of the DRUG file (#50) in the array defined by the calling application.

Status: Inactive

```
S X=$$SKAQ^PSS50(PSSIEN,PSSFL)
Input:
   PSSIEN
   PSSFL
Where:
   PSSIEN = IEN of entry in DRUG file (#50) [required]
   PSSFL = Either "S" (set) or "K" (kill) action [required]
Output:
   $$SKAQ
Where $$SKAQ is:
   0 (zero) = action unsuccessful
   1 (one) = action accomplished
```

Component: SKAQ1

Sets and kills the “AQ1” cross-reference of the DRUG file (#50) in the array defined by the calling application.

Status: Inactive

```
S X=$$SKAQ1^PSS50(PSSIEN)
Input:
   PSSIEN
Where:
   PSSIEN = IEN of entry in DRUG file (#50) [required]
Output:
   $$SKAQ1
```

```
Where $$SKAQ1 is:
   0 (zero) = action unsuccessful
   1 (one) = action accomplished
```

Component: AQ1

Returns the GENERIC NAME field (#.01) and IEN(s) of the DRUG file (#50) in the array defined by the calling application. The “AQ1” cross-reference in the format of ^PSDRUG(“AQ1”, CMOP ID(27),IEN(50)) will be used.

Status: Inactive

```
AQ1^PSS50(PSSVAL,PSSFL,PSSPK,LIST)
Input:
   PSSVAL
   PSSFL
   PSSPK
   LIST
Where:
   PSSVAL = CMOP ID field (#27) of the DRUG file (#50) [required]
   PSSFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   PSSPK = Returns only those entries containing at least one of the codes in
   the APPLICATION PACKAGES’ USE field (#63) of the DRUG file (#50) (ex: PSSPK
   = "IU" will return all entries for either IV, Unit Dose, or both IV and
   Unit Dose) [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = GENERIC NAME (50,.01)
   ^TMP($J,LIST,"AQ1",GENERIC NAME,PSSIEN) = ""
Where:
   PSSIEN is IEN of entry in the DRUG file (#50)
```

Component: OLDNM

Returns the OLD NAME multiple of the DRUG file (#50) in the array defined by the calling application.

Status: Inactive

```
OLDNM^PSS50(PSSIEN,PSSVAL,PSSFL,PSSPK,LIST)
Input:
   PSSIEN
   PSSVAL
   PSSFL
   PSSPK
   LIST
Where:
   PSSIEN = IEN of entry in DRUG file (#50) [optional]
   PSSVAL = GENERIC NAME field (#.01) of the DRUG file (#50) (a value of "??"
   may be used) [optional]
   PSSFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   PSSPK = Returns only those entries containing at least one of the codes in
   the APPLICATION PACKAGES’ USE field (#63) of the DRUG file (#50) (ex: PSSPK
   = "IU" will return all entries for either IV, Unit Dose, or both IV and
   Unit Dose) [optional]
   LIST = array name defined by the calling application [required]
```

```
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = GENERIC NAME (50,.01)
   ^TMP($J,LIST,PSSIEN,"OLD",0) = Total entries returned for this sub-file or
   -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,"OLD",PSS(1),.01) = OLD NAMES (50.01,.01)
   ^TMP($J,LIST,PSSIEN,"OLD",PSS(1),.02) = DATE CHANGED (50.01,.02)^External
   format (ex: SEP 12,1999)
   ^TMP($J,LIST,"B",GENERIC NAME,PSSIEN) = ""
Where:
   PSS(1) is the IEN of entry in the OLD NAMES multiple
```

Component: ADDOLDNM

Adds a new entry to the OLD NAME multiple of the DRUG file (#50).

Status: Inactive

```
S X=$$ADDOLDNM^PSS50(PSSIEN,PSSIEN2,PSSONM,PSSDT)
Input:
   PSSIEN
   PSSIEN2
   PSSONM
   PSSDT

Where:
   PSSIEN = IEN of entry in DRUG file (#50) [optional]
   PSSIEN2 = IEN of entry in OLD NAMES multiple of the DRUG file (#50)
   [required]
   PSSONM = OLD NAMES field (#.01) of the OLD NAMES multiple of the DRUG file
   (#50) [optional]
   PSSDT = DATE CHANGED field (#.02) of the OLD NAMES multiple of the DRUG
   file (#50) [optional]
Output:
   $$ADDOLDNM
Where $$ADDOLDNM is:
   0 (zero) = entry was unsuccessful
   1 (one) = entry was added
```

Component: A526

Returns the IEN(s) of the IV ADDITIVE file (#52.6) when passed in the IEN of the DRUG file (#50) in the array defined by the calling application.

Status: Inactive

```
A526^PSS50(PSSIEN,LIST)
Input:
   PSSIEN
   LIST
Where:
   PSSIEN = IEN of the entry in DRUG file (#50)[required]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN2,.01) = PRINT NAME (52.6,.01)
   ^TMP($J,LIST,"A526",PRINT NAME,PSSIEN2) = ""
Where:
   PSSIEN2 is IEN of entry in IV ADDITIVES file (#52.6)
```

Component: A527

Returns the IEN(s) of the IV SOLUTION file (#52.7) when passed in the IEN of the DRUG file (#50) in the array defined by the calling application.

Status: Inactive

```
A527^PSS50(PSSIEN,LIST)
Input:
   PSSIEN
   LIST
Where:
   PSSIEN = IEN of entry in DRUG file (#50) [required]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN2,.01) = PRINT NAME (52.7,.01)
   ^TMP($J,LIST,"A527",PRINT NAME,PSSIEN2) = ""
Where:
   PSSIEN2 is IEN of entry in IV SOLUTIONS file (#52.7)
```

Component: AIU

Returns the GENERIC NAME field (#.01) and IEN(s) of the DRUG file (#50) in the array defined by the calling application. The “AIU” cross-reference in the format of PSDRUG(“AIU”,NAME(50,.01),IEN(50)) will be used.

Status: Inactive

```
AIU^PSS50(PSSFT,PSSPK,PSSFL,LIST)
Input:
   PSSFT
   PSSPK
   PSSFL
   LIST
Where:
   PSSFT = GENERIC NAME field (#.01) of the DRUG file (#50) [required]
   PSSPK = Returns only those entries containing at least one of the codes in
   the APPLICATION PACKAGES’ USE field (#63) of the DRUG file (#50) (ex: PSSPK
   = "IU" will return all entries for either IV, Unit Dose, or both IV and
   Unit Dose) [optional]
   PSSFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = GENERIC NAME (50,.01)
Where:
   PSSIEN is IEN of entry in the DRUG file (#50)
```

Component: IU

Returns the GENERIC NAME field (#.01) and IEN(s) of the DRUG file (#50) in the array defined by the calling application. The “IU” cross-reference in the format of ^PSDRUG(“IU”,APPLICATION PACKAGES’USE(50,63),IEN(50)) will be used.

Status: Inactive

```
IU^PSS50(PSSFL,LIST)
Input:
   PSSFL
   LIST
Where:
   PSSFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = GENERIC NAME (50,.01)
Where:
   PSSIEN is IEN of entry in the DRUG file (#50)
```

Component: SKAIU

Sets and kills the “AIU” cross-references of the DRUG file (#50) in the array defined by the calling application.

Status: Inactive

```
S X=$$SKAIU^PSS50(PSSIEN,PSSFL)
Input:
   PSSIEN
   PSSFL
Where:
   PSSIEN = IEN of entry in DRUG file (#50) [required]
   PSSFL = Either "S" (set) or "K" (kill) action [required]
```

☛ Note: The APPLICATION PACKAGES’ USE field (#63) of the DRUG file (#50) must be updated prior to calling this API.

```
Output:
   $$SKAIU
Where $$SKAIU is:
   0 (zero) = action unsuccessful
   1 (one) = action accomplished
```

Component: SKIU

Sets and kills the “AU” cross-references of the DRUG file (#50) in the array defined by the calling application.

Status: Inactive

```
S X=$$SKIU^PSS50(PSSIEN)
Input:
   PSSIEN
Where:
   PSSIEN = IEN of entry in DRUG file (#50) [required]
```

☛ Note: The APPLICATION PACKAGES’ USE field (#63) of the DRUG file (#50) must be updated prior to calling this API.

```
Output:
   $SKIU
Where $SKIU is:
   0 (zero) = action unsuccessful
   1 (one) = action accomplished
```

Component: LOOKUP

Looks up a drug(s) from the DRUG file (#50) in the array defined by the calling application.

Status: Inactive

```
LOOKUP^PSS50(PSSFT,PSSFL,PSSPK,PSSIFCAP,PSSCMOP,PSSRTOI,PSSCRFL,LIST)
Input:
   PSSFT
   PSSFL
   PSSPK
   PSSIFCAP
   PSSCMOP
   PSSRTOI
   PSSCRFL
   LIST
Where:
   PSSFT = A look up value that could be the GENERIC NAME field (#.01), IEN,
   VA PRODUCT NAME (#21), NATIONAL DRUG CLASS (#25), or SYNONYM (#.01)
   multiple of the DRUG file (#50) (a value of"??" may be used) [optional]
   PSSFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   PSSPK = Returns only those entries containing at least one of the codes in
   the APPLICATION PACKAGES’ USE field (#63) of the DRUG file (#50) (ex: PSSPK
   = "IU" will return all entries for either IV, Unit Dose, or both IV and
   Unit Dose) [optional]
   PSSIFCAP = If this flag is passed in with a 1, the API returns those drug
   entries that do not have the IFCAP ITEM NUMBER (#441) multiple existed in
   the DRUG file (#50) [optional]
   PSSCMOP = If this flag is passed in with a 1, the API returns those drug
   entries that do not have data in the CMOP ID field (#27) of the DRUG file
   (#50) [optional]
   PSSRTOI = If a "1" is passed in, then only those entries matched to a
   Pharmacy Orderable Item will be returned [optional]
   PSSD = Multiple index lookup is performed if passed in
   (ex: "B", "C", "VAPN", "VAC"). Otherwise only the "B" cross-reference is
   used [optional]
   LIST = array name defined by the calling application [required]
```

☛ Note: If PSSFT = "??", the "B" cross-reference will be used and PSSD is ignored. PSSFT will allow the leading back tick to indicate an IEN lookup is desired. If there is a back tick, value passed into PSSD is ignored.

```
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = GENERIC NAME (50,.01)
   ^TMP($J,LIST,PSSIEN,25) = NATIONAL DRUG CLASS (50,25)^CODE
   (50.605,.01)^CLASSIFICATION (50.605,1)
   ^TMP($J,LIST,PSSIEN,100) = INACTIVE DATE (50,100)^External format
   (ex: SEP 12,1999)
   ^TMP($J,LIST,PSSIEN,101) = MESSAGE (50,101)
   ^TMP($J,LIST,PSSDX,GENERIC NAME,PSSIEN) = ""
Where:
   PSSIEN is IEN of entry in the DRUG file (#50)
   PSSDX is value passed in the PSSD. If multiple values are passed in, or the
   back tick IEN lookup is used, "B" will be used
```

Component: FRMALT

Looks up the FORMULARY ALTERNATIVE multiple from the DRUG file (#50) in the array defined by the calling application.

Status: Inactive

```
FRMALT^PSS50(PSSIEN,PSSFT,PSSFL,PSSPK,LIST)
Input:
   PSSIEN
   PSSFT
   PSSFL
   PSSPK
   LIST
Where:
   PSSIEN = IEN of entry in DRUG file (#50) [required]
   PSSFT = GENERIC NAME field (#.01) of the DRUG file (#50) (a value of "??"
   may be used)  [required]
   PSSFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   PSSPK = Returns only those entries containing at least one of the codes in
   the APPLICATION PACKAGES’ USE field (#63) of the DRUG file (#50) (ex: PSSPK
   = "IU" will return all entries for either IV, Unit Dose, or both IV and
   Unit Dose) [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = GENERIC NAME (50,.01)
   ^TMP($J,LIST,PSSIEN,25) = NATIONAL DRUG CLASS (50,25)
   ^TMP($J,LIST,PSSIEN,100) = INACTIVE DATE (50,100)^External format (ex: SEP
   12,1999)
   ^TMP($J,LIST,PSSIEN,101) = MESSAGE (50,101)
   ^TMP($J,LIST,PSSIEN,"FRM",0) = Total entries returned for this subfile or -
   1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,"FRM",PSS(1),2) = FORMULARY ALTERNATIVE
   (50.065,2)^GENERIC NAME (50,.01)
   ^TMP($J,LIST,"B",GENERIC NAME,PSSIEN) = ""
Where:
   PSS(1) is the IEN of the FORMULARY ALTERNATIVE multiple
```

Component: LABEL

Returns the fields from the DRUG file (#50) when passed the IEN of the DRUG file (#50) in the array defined by the calling application. External formatting will be included if applicable. The last piece of the output contains the field name of the returned field.

Status: Inactive

```
LABEL^PSS50(PSSIEN,LIST)
Input:
   PSSIEN
   LIST
Where:
   PSSIEN = IEN of entry in the DRUG file (#50) [required]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = GENERIC NAME (50,.01)
   ^TMP($J,LIST,PSSIEN,25) = NATIONAL DRUG CLASS (50,25)^CODE
   (50.605,.01)^CLASSIFICATION (50.605,1)
   ^TMP($J,LIST,PSSIEN,51) = LOCAL NON FORMULARY (50,51)^External format for
   the set of codes
   ^TMP($J,LIST,PSSIEN,100) = INACTIVE DATE (50,100)^External format (ex: SEP
   12,1999)
   ^TMP($J,LIST,PSSIEN,101) = MESSAGE (50,101)
   ^TMP($J,LIST,PSSIEN,102) = RESTRICTION (50,102)
   ^TMP($J,LIST,"B",GENERIC NAME,PSSIEN) = ""
```

Component: LIST

Returns all GENERIC NAME field (#.01) and PHARMACY ORDERABLE ITEM field (#2.1) of the DRUG file (#50) in the array defined by the calling application.

Status: Inactive

```
LIST^PSS50(PSSFT,PSSFL,PSSD,PSSPK,LIST)
Input:
   PSSFT
   PSSFL
   PSSD
   PSSPK
   LIST
Where:
   PSSFT = GENERIC NAME field (#.01) of the DRUG file (#50)
   (a value of "??" may be used) [optional]
   PSSFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   PSSD = the index used in the lookup, (ex: "B","C" cross-reference)
   PSSPK = Returns only those entries containing at least one of the codes in
   the APPLICATION PACKAGES’ USE field (#63) of the DRUG file (#50) (ex: PSSPK
   = "IU" will return all entries for either IV, Unit Dose, or both IV and
   Unit Dose) [optional]
   LIST = array name defined by the call application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = GENERIC NAME (50,.01)
   ^TMP($J,LIST,PSSIEN,100) = INACTIVE DATE (50,100)^External format (ex: SEP
   12,1999)
   ^TMP($J,LIST,PSSIEN,2.1) = PHARMACY ORDERABLE ITEM
   (50,2.1)^NAME(50.7,.01)^IEN of the Dosage Form file (#50.606)
   ^TMP($J,LIST,"B",GENERIC NAME,PSSIEN) = ""
```

Component: SORT

Returns the NAME field (#.10) of the DRUG file (#50) in the array defined by the calling application. Original sort template can be found in PSOUPAT.

Status: Inactive

```
SORT^PSS50(PSSIEN,LIST)
Input:
   PSSIEN
   LIST
Where:
   PSSIEN = IEN of entry in the DRUG file (#50) [required]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,NAME,PSSIEN) = ""
Where:
   NAME is the value in the GENERIC NAME field (#.01) of the DRUG file (#50)
```

Component: EDTIFCAP

Allows editing of the ITEM NUMBER sub-field (#.01) of the IF CAP ITEM NUMBER multiple of the DRUG file (#50). The lookup call is provided in AB^PSS50 API.

Status: Inactive

```
S X=$$EDTIFCAP^PSS50(PSSIEN,PSSVAL)
Input:
   PSSIEN
   PSSVAL
Where:
   PSSIEN = IEN of entry in the DRUG file (#50) [required]
   PSSVAL = the ITEM NUMBER subfield (#.01) of the IFCAP ITEM NUMBER multiple
   in the DRUG file (#50) [required]
Output:
   $$EDTIFCAP
Where $$EDTIFCAP is:
   0 (zero) = action unsuccessful
   1 (one) = action accomplished
```

PSS50P4 API – Drug Electrolytes file (#50.4)

Component: ALL

Returns all fields in the DRUG ELECTROLYTES file (#50.4) in the array defined by the calling application.

Status: Inactive

```
ALL^PSS50P4(PSSIEN,PSSFT,LIST)
Input:
   PSSIEN
   PSSFT
   LIST
```

```
Where:
   PSSIEN = the IEN of entry in DRUG ELECTROLYTES file (#50.4) [optional]
   PSSFT = NAME field (#.01) of the DRUG ELECTROLYTES file (#50.4) (a value of
   "?" may be used) [optional]
   LIST = array name defined by the call application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = NAME (50.4,.01)
   ^TMP($J,LIST,PSSIEN,1) = CONCENTRATION UNITS (50.4,1)^External format for
   the set of codes
   ^TMP($J,LIST,"B",NAME,PSSIEN) = ""
```

PSS50P66 API – Dosage Form file (#50.606)

Component: ALL

Returns most of the fields from the DOSAGE FORM file (#50.606) in the array defined by the calling application.

Status: Inactive

```
ALL^PSS50P66(PSSIEN,PSSFT,PSSFL,LIST)
Input:
   PSSIEN
   PSSFT
   PSSFL
   LIST
Where:
   PSSIEN = IEN of entry in DOSAGE FORM file (#50.606) [optional]
   PSSFT = NAME field (#.01) of the DOSAGE FORM file (#50.606)
   (a value of "??" may be used) [optional]
   PSSFL = Inactive date. A null value will return all entries. Entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date. [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = NAME (50.606,.01)
   ^TMP($J,LIST,PSSIEN,3) = VERB (50.606,3)
   ^TMP($J,LIST,PSSIEN,5) = PREPOSITION (50.606,5)
   ^TMP($J,LIST,PSSIEN,7) = INACTIVATION DATE (50.606,7)^External format (ex:
   SEP 12,1999)
   ^TMP($J,LIST,PSSIEN,"MRDSFRM",0) = Total entries returned for this sub-file
   or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,"MRDSFRM",PSS(1),.01) = MED ROUTE FOR DOSAGE FORM
   (50.6061,.01)^NAME (51.2,.01)
   ^TMP($J,LIST,PSSIEN,"NOUN",0) = Total entries returned for this sub-file or
   -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,"NOUN",PSS(1),.01) = NOUN (50.6066,.01)
   ^TMP($J,LIST,"B",NAME,PSSIEN) = ""
Where:
   PSS(1) is the IEN of the multiple it referenced
```

Component: ADD

The first API, ALL^PSS50P66, does a look-up from the DOSAGE FORM file (#50.606). The second API, ADD^PSS50P66, will add the MEDICATION ROUTES FOR DOSAGE FORM sub-field (#.01) of the MED ROUTE FOR DOSAGE FORM multiple from the DOSAGE FORM file (#50.606).

Status: Inactive

```
S X=$$ADD^PSS50P66(PSSIEN,PSSMR)
Input:
   PSSIEN
   PSSMR
Where:
   PSSIEN = IEN of entry in DOSAGE FORM file (#50.606) [required]
   PSSMR = IEN of entry in the MEDICATION ROUTE file (#51.2) (it is to be
   added to the MEDICATION ROUTE FOR DOSAGE FORM multiple (#50.6061) if is not
   a duplicate) [required]
Output:
   $$ADD
Where $$ADD:
   1 (zero) = entry was added
   0 (one) = entry was not added
```

PSS50P7 API – Pharmacy Orderable Item file (#50.7)

Component: ZERO

Returns the zero node of the PHARMACY ORDERABLE ITEM file (#50.7) in the array defined by the calling application. A check for the existence of the zero node will be performed.

Status: Active

```
ZERO^PSS50P7(PSSIEN,PSSFT,PSSFL,LIST)
Input:
   PSSIEN
   PSSFT
   PSSFL
   LIST
Where:
   PSSIEN = IEN of entry in the PHARMACY ORDERABLE ITEM file (#50.7)
   [optional]
   PSSFT = NAME field (#.01) of the PHARMACY ORDERABLE ITEM file (#50.7) (a
   value of "??" may be used) [optional]
   PSSFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = NAME (50.7,.01)
   ^TMP($J,LIST,PSSIEN,.02) = DOSAGE FORM(50.7,.02)^NAME (50.606,.01)
   ^TMP($J,LIST,PSSIEN,.03) = IV FLAG (50.7,.03)^External format for the set
   of codes (ex: 1 if flagged for IV)
   ^TMP($J,LIST,PSSIEN,.04) = INACTIVE DATE (50.7,.04)^External format (ex:
   SEP 12,1999)
   ^TMP($J,LIST,PSSIEN,.05) = DAY (nD) or DOSE (nL) LIMIT (50.7,.05)
   ^TMP($J,LIST,PSSIEN,.06) = MED ROUTE (50.7,.06)^NAME (51.2,.01)
   ^TMP($J,LIST,PSSIEN,.07) = SCHEDULE TYPE (50.7,.07)^External format for the
   set of codes
   ^TMP($J,LIST,PSSIEN,.08) = SCHEDULE (50.7,.08)
   ^TMP($J,LIST,PSSIEN,.09) = SUPPLY (50.7,.09)^External format for the set of
   codes
   ^TMP($J,LIST,PSSIEN,5) = FORMULARY STATUS (50.7,5)^External format for the
   set of codes
   ^TMP($J,LIST,PSSIEN,8) = NON-VA MED (50.7,8)^External format for the set of
   codes (ex: 1 flagged the med as Non-VA)
   ^TMP($J,LIST,"B",NAME,PSSIEN) = ""
```

Component: SYNONYM

Returns the SYNONYM sub-field (#.01) of the SYNONYM multiple of the PHARMACY ORDERABLE ITEM file (#50.7) in the array defined by the calling application.

Status: Inactive

```
SYNONYM^PSS50P7(PSSIEN,PSSFT,PSSFL,LIST)
Input:
   PSSIEN
   PSSFT
   PSSFL
   LIST
Where:
   PSSIEN = IEN of entry in the PHARMACY ORDERABLE ITEM file (#50.7)
   [optional]
   PSSFT = NAME field in the PHARMACY ORDERABLE ITEM file (#50.7) (a value of
   "??" may be used) [optional]
   PSSFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = NAME (50.7,.01)
   ^TMP($J,LIST,PSSIEN,.02) = DOSAGE FORM (50.7,.02)^NAME (50.606,.01)
   ^TMP($J,LIST,PSSIEN,"SYN",0) = Total entries returned for this sub-file or
   -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,"SYN",PSS(1),.01) = SYNONYM (50.72,.01)
   ^TMP($J,LIST,"B",NAME,PSSIEN) = ""
Where:
   PSS(1) is the IEN of entry the SYNONYM multiple
```

Component: INSTR

Returns the PATIENT INSTRUCTIONS field (#7) and OTHER LANGUAGE INSTRUCTIONS field (#7.1) of the PHARMACY ORDERABLE ITEM file (#50.7) in the array defined by the calling application.

Status: Inactive

```
INSTR^PSS50P7(PSSIEN,PSSFT,PSSFL,LIST)
Input:
   PSSIEN
   PSSFT
   PSSFL
   LIST
Where:
   PSSIEN = IEN of entry in PHARMACY ORDERABLE ITEM file (#50.7) [optional]
   PSSFT = NAME field (#.01) in PHARMACY ORDERABLE ITEM file (#50.7) (a value
   of "??" may be used) [optional]
   PSSFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = NAME (50.7,.01)
   ^TMP($J,LIST,PSSIEN,.02) = DOSAGE FORM (50.7,.02)^NAME (50.606,.01)
   ^TMP($J,LIST,PSSIEN,7) = PATIENT INSTRUCTIONS (50.7,7)
   ^TMP($J,LIST,PSSIEN,7.1) = OTHER LANGUAGE INSTRUCTIONS (50.7,7.1)
   ^TMP($J,LIST,"B",NAME,PSSIEN) = ""
```

Component: IEN

Returns the IEN(s) of the PHARMACY ORDERABLE ITEM file (#50.7) in the array defined by the calling application when passing in the NAME field (#.01) of the PHARMACY ORDERABLE ITEM file (#50.7). The “B” cross-reference in the format of ^PS(50.7,“B”,NAME(50.7,.01),IEN(50.7)) will be used.

Status: Inactive

```
IEN^PSS50P7(PSSFT,PSSFL,LIST)
Input:
   PSSFT
   PSSFL
   LIST
Where:
   PSSFT = NAME field (#.01) in the PHARMACY ORDERABLE ITEM file (#50.7) (a
   value of "??" may be used) [required]
   PSSFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN) = NAME (50.7,.01)
   ^TMP($J,LIST,PSSIEN,.02) = DOSAGE FORM (50.7,.02)^NAME (50.606,.01)
   ^TMP($J,LIST,"B",NAME,PSSIEN) = ""
Where:
   PSSIEN is IEN of entry in the PHARMACY ORDERABLE ITEM file (#50.7)
```

Component: DRGIEN

Returns the IEN(s) of the DRUG file (#50) in the array defined by the calling application when passing in the IEN of the PHARMACY ORDER ITEM file (#50.7). The “A50” cross-reference in format of ^PS(50.7,“A50”,IEN(50.7),IEN(50)) will be used.

Status: Active

```
DRGIEN^PSS50P7(PSSIEN,PSSFL,LIST)
Input:
   PSSIEN
   PSSFL
   LIST
Where:
   PSSIEN = IEN of entry in PHARMACY ORDERABLE ITEM file (#50.7) [required]
   PSSFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   LIST = array name defined by the calling application [required]
```

```
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSS50) = ""
Where:
   PSS50 is the IEN of the DRUG file (#50)
```

Component: NAME

Returns the NAME field (#.01) of the PHARMACY ORDERABLE ITEM file (#50.7).

Status: Inactive

```
S X=$$NAME^PSS50P7(PSSIEN)
Input:
   PSSIEN
Where:
   IEN = entry in PHARMACY ORDERABLE ITEM file (#50.7) [required]
Output:
   $$NAME
Where:
   $$NAME = NAME field (#.01) of the PHARMACY ORDERABLE ITEM file (#50.7)
```

Component: LOOKUP

Returns the NAME field (#.01) of the PHARMACY ORDERABLE ITEM file (#50.7) along with the NAME field (#.01) of the DOSAGE FORM file (#50.606). Screening for the active entries from the DRUG file (#50) for specified usage will be applied before including the data in the return array. The ACTIVE DATE field (#.04) of the PHARMACY ORDERABLE ITEM file (#50.7) should be included in the screening.

Status: Inactive

```
LOOKUP^PSS50P7(PSSFT,PSSD,PSSS,LIST)
Input:
   PSSFT
   PSSD
   PSSS
   LIST
Where:
   PSSFT = NAME field (#.01) of PHARMACY ORDERABLE ITEM file (#50.7) (a value
   of "??" may be used) [required]
   PSSD = Index being traversed [optional]
   PSSS = Screening information as defined in DIC("S") [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = NAME (50.7,.01)
   ^TMP($J,LIST,PSSIEN,.02) = DOSAGE FORM (50.7,.02)^NAME (50.606,.01)
   ^TMP($J,LIST,PSSIEN,.04) = INACTIVE DATE (50.7,.04)^External format (ex:
   SEP 12,1999)
   ^TMP($J,LIST,PSSDX,NAME,PSSIEN) = ""
Where:
   PSSIEN is the IEN of the PHARMACY ORDERABLE ITEM file (#50.7)
   PSSDX is value passed in the PSSD (if multiple values are passed in, "B"
   will be used)
```

PSS51 API – Medication Instruction file (#51)

Component: ALL

Returns most of the fields from the MEDICATION INSTRUCTION file (#51) in the array defined by the calling application.

Status: Inactive

```
ALL^PSS51(PSSIEN,PSSFT,LIST)
Input:
   PSSIEN
   PSSFT
   LIST
Where:
   PSSIEN = IEN of entry in MEDICATION INSTRUCTION file (#51) [optional]
   PSSFT = NAME field (#.01) of MEDICATION INSTRUCTION file (#51) (a value of
   "??" may be used) [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = NAME (51,.01)
   ^TMP($J,LIST,PSSIEN,.5) = SYNONYM (51,.5)
   ^TMP($J,LIST,PSSIEN,1) = EXPANSION (51,1)
   ^TMP($J,LIST,PSSIEN,1.1) = OTHER LANGUAGE EXPANSION (51,1.1)
   ^TMP($J,LIST,PSSIEN,2) = MED ROUTE (51,2)^NAME (51.2,.01)
   ^TMP($J,LIST,PSSIEN,3) = SCHEDULE (51,3)
   ^TMP($J,LIST,PSSIEN,4) = INSTRUCTIONS (51,4)
   ^TMP($J,LIST,PSSIEN,5) = ADDITIONAL INSTRUCTION (51,5)
   ^TMP($J,LIST,PSSIEN,9) = PLURAL (51,9)
   ^TMP($J,LIST,PSSIEN,10) = DEFAULT ADMIN TIMES (51,10)
   ^TMP($J,LIST,PSSIEN,30) = INTENDED USE (51,30)^External format for the set
   of codes
   ^TMP($J,LIST,PSSIEN,31) = FREQUENCY (IN MINUTES) (51,31)
   ^TMP($J,LIST,"B",NAME,PSSIEN) = ""
```

Component: WARD

Returns most of the fields from the MEDICATION INSTRUCTION file (#51) in the array defined by the calling application.

Status: Inactive

```
WARD^PSS51(PSSIEN,PSSFT,LIST)
Input:
   PSSIEN
   PSSFT
   LIST
Where:
   PSSIEN = IEN of entry in MEDICATION INSTRUCTION file (#51) [optional]
   PSSFT = NAME field (#.01) of MEDICATION INSTRUCTION file (#51) (a value of
   "??" may be used) [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = NAME (51,.01)
   ^TMP($J,LIST,"WARD",0) = Total entries returned for this sub-file or -1^NO
   DATA FOUND
   ^TMP($J,LIST,PSSIEN,"WARD",PSS(1),.01) = WARD (51.01,.01)^NAME (42,.01)
   ^TMP($J,LIST,PSSIEN,"WARD",PSS(1),.02) = DEFAULT ADMIN TIMES (51.01,.02)
   ^TMP($J,LIST,"B",NAME,PSSIEN) = ""
```

```
Where:
   PSS(1) is the IEN of entry in the WARD multiple
```

Component: LOOKUP

Returns the NAME field (#.01) and EXPANSION field (#1) from the MEDICATION INSTRUCTION file (#51) in the array defined by the calling application.

Status: Inactive

```
LOOKUP^PSS51(PSSIEN,PSSFT,LIST)
Input:
   PSSIEN
   PSSFT
   LIST
Where:
   PSSIEN = IEN of entry in MEDICATION INSTRUCTION file (#51) [optional]
   PSSFT = NAME field (#.01) of MEDICATION INSTRUCTION file (#51) (a value of
   "??" may be used) [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = NAME (51,.01)
   ^TMP($J,LIST,PSSIEN,1) = EXPANSION (51,1)
   ^TMP($J,LIST,"B",NAME,PSSIEN) = ""
```

Component: CHK

Checks if the value is a valid entry and the INTENDED USE field (#30) from the MEDICATION INSTRUCTION file (#51) is for Outpatient use (Less than 2).

Status: Inactive

```
CHK^PSS51(PSSFT,LIST)
Input:
   PSSFT
   LIST
Where:
   PSSFT = NAME (#.01) field of MEDICATION INSTRUCTION file (#51) (a value of
   "??" may be used) [required]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = NAME (51,.01)
   ^TMP($J,LIST,"B",NAME,PSSIEN) = ""
Where:
   PSSIEN is IEN of entry in the MEDICATION INSTRUCTION file (#51)
```

Component: A

Returns the MEDICATION INSTRUCTION EXPANSION fields of the MEDICATION INSTRUCTION file (#51) in the array defined by the calling application.

Status: Inactive

```
A^PSS51(PSSFT,LIST)
Input:
   PSSFT
   LIST
Where:
   PSSFT = NAME field (#.01) of the MEDICATION INSTRUCTION file (#51)
   [required]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = NAME (51,.01)
   ^TMP($J,LIST,PSSIEN,1) = EXPANSION (51,1)
   ^TMP($J,LIST,PSSIEN,1.1) = OTHER LANGUAGE EXPANSION (51,1.1)
   ^TMP($J,LIST,PSSIEN,9) = PLURAL (51,9)
   ^TMP($J,LIST,"A",NAME,PSSIEN) = ""
Where:
   PSSIEN is IEN of entry in the MEDICATION INSTRUCTION file (#51)
```

PSS51P1 API – Administration Schedule file (#51.1)

Component: ZERO

Returns the zero node and the OTHER LANGUAGE EXPASION field (#8.1) of the ADMINISTRATION SCHEDULE file (#51.1) in the array defined by the calling application.

Status: Active

```
ZERO^PSS51P1(PSSIEN,PSSFT,PSSPP,PSSTSCH,LIST)
Input:
   PSSIEN
   PSSFT
   PSSPP
   PSSTSCH
   LIST
Where:
   PSSIEN = IEN of entry in ADMINISTRATION SCHEDULE file (#51.1) [optional]
   PSSFT = NAME field (#.01) of ADMINISTRATION SCHEDULE file (#51.1) (a value
   of "??" may be used) [optional]
   PSSPP = the PACKAGE PREFIX field (#4) of the ADMINISTRATION SCHEDULE file
   (#51.1) (screening for the Package Prefix if this field is passed in (ex:
   PSJ, LR…)) [optional]
   PSSTSCH = TYPE OF SCHEDULE field (#5) of ADMINISTRATION SCHEDULE file
   (#51.1) (screening for One-time "O" if PSSTSCH passed in) [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = NAME (51.1,.01)
   ^TMP($J,LIST,PSSIEN,1) = STANDARD ADMINISTRATION TIMES (51.1,1)
   ^TMP($J,LIST,PSSIEN,2) = FREQUENCY (IN MINUTES) (51.1,2)
   ^TMP($J,LIST,PSSIEN,2.5) = MAX DAYS FOR ORDERS (51.1,2.5)
   ^TMP($J,LIST,PSSIEN,4) = PACKAGE PREFIX (51.1,4)
   ^TMP($J,LIST,PSSIEN,5) = TYPE OF SCHEDULE (51.1,5)^External format for the
   set of code
   ^TMP($J,LIST,PSSIEN,6) = STANDARD SHIFTS (51.1,6)
   ^TMP($J,LIST,PSSIEN,8) = OUTPATIENT EXPANSION (51.1,8)
   ^TMP($J,LIST,PSSIEN,8.1) = OTHER LANGUAGE EXPANSION (51.1,8.1)
   ^TMP($J,LIST,"B",NAME,PSSIEN) = ""
```

Component: WARD

Returns the WARD sub-field of the ADMINISTRATION SCHEDULE file (#51.1) in the array defined by the calling application.

Status: Inactive

```
WARD^PSS51P1(PSSIEN,PSSFT,PSSIEN2,LIST)
Input:
   PSSIEN
   PSSFT
   PSSIEN2
   LIST
Where:
   PSSIEN = IEN of entry in ADMINISTRATION SCHEDULE file (#51.1) [optional]
   PSSFT = NAME field (#.01) of ADMINISTRATION SCHEDULE file (#51.1) (a value
   of "??" may be used) [optional]
   PSSIEN2 = IEN of entry in WARD sub-file (#51.11) [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = NAME (51.1,.01)
   ^TMP($J,LIST,"WARD",0) = Total entries returned for this sub-file or -1^NO
   DATA FOUND (If PSSIEN2 is passed in and there are no ward entries, -1^NO
   DATA FOUND FOR PSSIEN2 concatenated with the value of PSSIEN2 is returned)
   ^TMP($J,LIST,PSSIEN,"WARD",PSS(1),.01) = WARD (51.11,.01)^NAME (42,.01)
   ^TMP($J,LIST,PSSIEN,"WARD",PSS(1),1) = WARD ADMINISTRATION TIMES (51.11,1)
   ^TMP($J,LIST,"B",NAME,PSSIEN) = ""
```

☛ Notes:

If PSSIEN = "", PSSFT = "??" and PSSIEN2 is passed in, then all entries in the Administration Schedule file will be returned and only Ward data matching the value passed in PSSIEN2 will be included.

If PSSIEN = "", PSSFT = "??" and PSSIEN2 = "", then all entries in the Administration Schedule file and existing Ward data will be returned.

If PSSIEN is passed in, PSSFT = "" and PSSIEN2 = "", the entry matching the value passed in PSSIEN will be returned along with the associated Ward data.

If PSSIEN, PSSFT & PSSIEN2 are passed in, the value in PSSFT will be ignored, the entry matching the value passed in PSSIEN will be returned (only Ward data matching the value passed in PSSIEN2 will be included).

```
Where:
   PSS(1) is the IEN of entry in the WARD multiple
```

Component: HOSP

Returns the fields from HOSPITAL LOCATION sub-field of the ADMINISTRATION SCHEDULE file (#51.1) in the array defined by the calling application.

Status: Inactive

```
HOSP^PSS51P1(PSSIEN,PSSFT,LIST)
Input:
   PSSIEN
   PSSFT
   LIST
Where:
   PSSIEN = IEN of entry in ADMINISTRATION SCHEDULE file (#51.1) [optional]
   PSSFT = NAME field (#.01) of ADMINISTRATION SCHEDULE file (#51.1) (a value
   of "??" may be used) [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = NAME (51.1,.01)
   ^TMP($J,LIST,"HOSP",0) = Total entries returned for this sub-file or -1^NO
   DATA FOUND
   ^TMP($J,LIST,PSSIEN,"HOSP",PSS(1),.01) = HOSPITAL LOCATION (51.17,.01)^NAME
   (44,.01)
   ^TMP($J,LIST,PSSIEN,"HOSP",PSS(1),1) = ADMINISTRATION TIMES (51.17,1)
   ^TMP($J,LIST,PSSIEN,"HOSP",PSS(1),2) = SHIFTS (51.17,2)
   ^TMP($J,LIST,"B",NAME,PSSIEN) = ""
Where:
   PSS(1) is the IEN of entry the HOSPITAL LOCATION multiple
```

Component: IEN

Returns the IEN(s) of the ADMINISTRATION SCHEDULE file (#51.1) in the array defined by the calling application.

Status: Inactive

```
IEN^PSS51P1(PSSFT,LIST)
Input:
   PSSFT
   LIST
Where:
   PSSFT = NAME field (#.01) of ADMINISTRATION SCHEDULE file (#51.1) (a value
   of "??" may be used) [required]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = NAME(51.1,.01)
   ^TMP($J,LIST,PSSIEN,1) = STANDARD ADMINISTRATION TIMES (51.1,1)
   ^TMP($J,LIST,"B",NAME,PSSIEN) = ""
Where:
   PSSIEN is IEN of entry in the ADMINISTRATION SCHEDULE file (#51.1)
```

Component: AP

Returns the IEN(s) and NAME field (#.01) of the ADMINISTRATION SCHEDULE file (#51.1) in the array defined by the calling application when passed in the PACKAGE PREFIX field (#4) of the ADMINISTRATION SCHEDULE file (#51.1). The “AP” cross-reference in format of ^PS(51.1,“AP”_PACKAGE PREFIX(4),NAME(.01),IEN(51.1)) will be used.

Status: Active

```
AP^PSS51P1(PSSPP,PSSFT,PSSWDIEN,PSSSTPY,LIST)
Input:
   PSSPP
   PSSFT
   PSSWDIEN
   PSSSTPY
   LIST
Where:
   PSSPP = PACKAGE PREFIX field (#4) of the ADMINISTRATION SCHEDULE file
   (#51.1) [required]
   PSSFT = NAME field (#.01) of the ADMINISTRATION SCHEDULE file (#51.1) (a
   value of "??" may be used) [optional]
   PSSWDIEN = IEN of entry of WARD multiple in ADMINISTRATION SCHEDULE file
   (#51.1) [optional]
   PSSSTPY = TYPE OF SCHEDULE field (#5) OF ADMINISTRATION SCHEDULE file
   (#51.1) [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = NAME (51.1,.01)
   ^TMP($J,LIST,PSSIEN,1) = STANDARD ADMINISTRATION TIMES (51.1,1)
   ^TMP($J,LIST,PSSIEN,2) = FREQUENCY (IN MINUTES) (51.1,2)
   ^TMP($J,LIST,PSSIEN,2.5) = MAX DAYS FOR ORDERS (51.1,2.5)
   ^TMP($J,LIST,PSSIEN,4) = PACKAGE PREFIX (51.1,4)
   ^TMP($J,LIST,PSSIEN,5) = TYPE OF SCHEDULE (51.1,5)^External format for the
   set of code
   ^TMP($J,LIST,PSSIEN,8) = OUTPATIENT EXPANSION (51.1,8)
   ^TMP($J,LIST,PSSIEN,"WRD",0) = Total entries returned for this sub-file or
   -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,"WRD",PSSWDIEN,.01) = WARD (51.11,.01)^NAME (42,.01)
   ^TMP($J,LIST,PSSIEN,"WRD",PSSWDIEN,1) = WARD ADMINISTRATION TIMES (51.11,1)
   ^TMP($J,LIST,"AP"_PACKAGE PREFIX,NAME,PSSIEN) = ""
```

☛ Notes:

If PSSSTYP is passed in the API will screen on this value.
If PSSPP is passed in, PSSFT = "" or PSSFT = "??", PSSWDIEN = "" and PSSTYP = "", then all schedules associated with the PSSPP and Ward will be returned.
If PSSPP is passed in, PSSFT = "" or PSSFT = "??", PSSWDIEN is passed in and PSSTYP = "", then all schedules associated with the PSSPP will be returned and only Ward data matching the PSSWDIEN will be included.
If PSSPP = "", PSSFT = "QID", PSSWDIEN = "" and PSSTYP = "", then QID data and associated Wards data will be returned.
If PSSPP = "", PSSFT = "QID", PSSWDIEN is passed in and PSSTYP = "", then QID data will be returned and only Ward data matching PSSWDIEN will be included.

Component: IX

Returns the zero node of the ADMINISTRATION SCHEDULE file (#51.1) in the array defined by the calling application when passed in the NAME field (#.01) and PACKAGE PREFIX field (#4) of the ADMINISTRATION SCHEDULE file (#51.1). The “AP” cross-reference in format of ^PS(51.1,“AP”_PACKAGE PREFIX(4),NAME(.01),IEN(51.1)) will be used.

Status: Inactive

```
IX^PSS51P1(PSSFT,PSSPP,LIST)
Input:
   PSSFT
   PSSPP
   LIST
Where:
   PSSFT = NAME field (#.01) of ADMINISTRATION SCHEDULE file (#51.1) (a value
   of "??" may be used) [required]
   PSSPP = PACKAGE PREFIX field (#4) of the ADMINISTRATION SCHEDULE file
   (#51.1) [required]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = NAME (51.1,.01)
   ^TMP($J,LIST,PSSIEN,1) = STANDARD ADMINISTRATION TIMES (51.1,1)
   ^TMP($J,LIST,PSSIEN,2) = FREQUENCY (IN MINUTES) (51.1,2)
   ^TMP($J,LIST,PSSIEN,2.5) = MAX DAYS FOR ORDERS (51.1,2.5)
   ^TMP($J,LIST,PSSIEN,4) = PACKAGE PREFIX (51.1,4)
   ^TMP($J,LIST,PSSIEN,5) = TYPE OF SCHEDULE (51.1,5)^External format for the
   set of code
   ^TMP($J,LIST,PSSIEN,6) = STANDARD SHIFTS (51.1,6)
   ^TMP($J,LIST,PSSIEN,8) = OUTPATIENT EXPANSION (51.1,8)
   ^TMP($J,LIST,PSSIEN,8.1) = OTHER LANGUAGE EXPANSION (51.1,8.1)
   ^TMP($J,LIST,"AP"_PACKAGE PREFIX,NAME,PSSIEN) = ""
Where:
   PSSIEN is IEN of entry in the ADMINISTRATION SCHEDULE file (#51.1)
```

Component: ADM

Validates the STANDARD ADMINISTRATION TIMES field (#1) of the ADMINISTRATION SCHEDULE file (#51.1).

Status: Inactive

```
S X=$$ADM^PSS51P1(PSSADM)
Input:
   PSSADM
Where:
   PSSADM = STANDARD ADMINISTRATION TIMES field (#1) of the ADMINISTRATION
   SCHEDULE file (#51.1) [required]
Output:
   $$ADM
Where:
   $$ADM = either "^" (for invalid) or the value of the STANDARD
   ADMINISTRATION TIMES field (#1) of the ADMINISTRATION SCHEDULE file (#51.1)
```

Component: ALL

Returns all data of the ADMINISTRATION SCHEDULE file (#51.1) in the array defined by the calling application.

Status: Inactive

```
ALL^PSS51P1(PSSIEN,PSSFT,LIST)
Input:
   PSSIEN
   PSSFT
   LIST
Where:
   PSSIEN = IEN of entry in ADMINISTRATION SCHEDULE file (#51.1) [optional]
   PSSFT = NAME field (#.01) of ADMINISTRATION SCHEDULE file (#51.1) (a value
   of "??" may be used) [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = NAME (51.1,.01)
   ^TMP($J,LIST,PSSIEN,1) = STANDARD ADMINISTRATION TIMES (51.1,1)
   ^TMP($J,LIST,PSSIEN,2.5) = MAX DAYS FOR ORDERS (51.1,2.5)
   ^TMP($J,LIST,PSSIEN,2) = FREQUENCY (IN MINUTES) (51.1,2)
   ^TMP($J,LIST,PSSIEN,4) = PACKAGE PREFIX (51.1,4)
   ^TMP($J,LIST,PSSIEN,5) = TYPE OF SCHEDULE (51.1,5)^External format for the
   set of code
   ^TMP($J,LIST,PSSIEN,6) = STANDARD SHIFTS (51.1,6)
   ^TMP($J,LIST,PSSIEN,8) = OUTPATIENT EXPANSION (51.1,8)
   ^TMP($J,LIST,PSSIEN,8.1) = OTHER LANGUAGE EXPANSION (51.1,8.1)
   ^TMP($J,LIST,PSSIEN,"HOSP",0) = Total entries returned for this sub-file or
   -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,"HOSP",PSS(1),.01) = HOSPITAL LOCATION (51.17,.01)^NAME
   (44,.01)
   ^TMP($J,LIST,PSSIEN,"HOSP",PSS(1),1) = ADMINISTRATION TIMES (51.17,1)
   ^TMP($J,LIST,PSSIEN,"HOSP",PSS(1),2) = SHIFTS (51.17,2)
   ^TMP($J,LIST,PSSIEN,"WRD",0) = Total entries returned for this sub-file or
   -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,"WRD",PSS(1),.01) = WARD (51.11,.01)^NAME (42,.01)
   ^TMP($J,LIST,PSSIEN,"WRD",PSS(1),1) = WARD ADMINISTRATION TIMES (51.11,1)
   ^TMP($J,LIST,"B",NAME,PSSIEN) = ""
Where:
   PSS(1) is the IEN of the multiple it referenced
```

PSS51P15 API – Administration Shift file (#51.15)

Component: ALL

Returns all fields in the ADMINISTRATION SHIFT file (#51.15) in the array defined by the calling application.

Status: Inactive

```
ALL^PSS51P15(PSSIEN,PSSFT,LIST)
Input:
   PSSIEN
   PSSFT
   LIST
Where:
   PSSIEN = IEN of entry in ADMINISTRATION SHIFT file (#51.15) [optional]
   PSSFT = NAME field (#.01) of the ADMINISTRATION SHIFT file (#51.15) (a
   value of "??" may be used) [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = NAME (51.15,.01)
   ^TMP($J,LIST,PSSIEN,1) = ABBREVIATION (51.15,1)
   ^TMP($J,LIST,PSSIEN,2) = STANDARD START/STOP TIMES (51.15,2)
   ^TMP($J,LIST,PSSIEN,4) = PACKAGE (51.15,4)
   ^TMP($J,LIST,PSSIEN,"HOSP",0) = Total entries returned for this sub-file or
   -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,"HOSP",PSS(1),.01) = HOSPITAL LOCATION
   (51.153,.01)^NAME (44,.01)
   ^TMP($J,LIST,PSSIEN,"HOSP",PSS(1),1) = START/STOP TIMES (51.153,1)
   ^TMP($J,LIST,"B",NAME,PSSIEN) = ""
```

Component: ACP

Returns the IEN(s) and NAME field (#.01) of the ADMINISTRATION SHIFT file (#51.15) in the array defined by the calling application when passed in the PACKAGE field (#4) and the ABBREVIATION field (#1) of the ADMINISTRATION SHIFT file (#51.15). The “ACP” cross-reference in format of ^PS(51.15,“ACP”_PACKAGE(4),ABBREVIATION(1),IEN(51.15)) will be used.

Status: Inactive

```
ACP^PSS51P15(PSSPK,PSSABBR,LIST)
Input:
   PSSPK
   PSSABBR
   LIST
Where:
   PSSPK = PACKAGE field (#4) of the ADMINISTRATION SHIFT file (#51.15)
   [required]
   PSSABBR = ABBREVIATION field (#1) of the ADMINISTRATION SHIFT file (#51.15)
   [required]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = NAME (51.15,.01)
   ^TMP($J,LIST,PSSIEN,1) = ABBREVIATION (51.15,1)
   ^TMP($J,LIST,PSSIEN,4) = PACKAGE (51.15,4)
   ^TMP($J,LIST,"ACP",PACKAGE,NAME,PSSIEN) = ""
```

```
Where:
   PSSIEN is IEN of entry in the ADMINISTRATION SHIFT file (#51.15)
```

PSS51P2 API – Medication Routes file (#51.2)

Component: ALL

Returns all fields in the MEDICATION ROUTES file (#51.2) in the array defined by the calling application.

Status: Active

```
ALL^PSS51P2(PSSIEN,PSSFT,PSSFL,PSSPK,LIST)
Input:
   PSSIEN
   PSSFT
   PSSFL
   PSSPK
   LIST
Where:
   PSSIEN = IEN of entry in MEDICATION ROUTES file (#51.2) [optional]
   PSSFT = NAME field (#.01) of the MEDICATION ROUTES file (#51.2) (a value of
   "??" may be used) [optional]
   PSSFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   PSSPK = PACKAGE USE field (#3) of the MEDICATION ROUTES file (#51.2)
   [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = NAME (51.2,.01)
   ^TMP($J,LIST,PSSIEN,1) = ABBREVIATION (51.2,1)
   ^TMP($J,LIST,PSSIEN,3) = PACKAGE USE (51.2,3)^External format for the set
   of codes
   ^TMP($J,LIST,PSSIEN,4) = OUTPATIENT EXPANSION (51.2,4)
   ^TMP($J,LIST,PSSIEN,4.1) = OTHER LANGUAGE EXPANSION (51.2,4.1)
   ^TMP($J,LIST,PSSIEN,5) = INACTIVATION DATE (51.2,5)^External format (ex:
   SEP 12,1999)
   ^TMP($J,LIST,PSSIEN,6) = IV FLAG (51.2,6)^External format for the set of
   codes (ex: 1 if flagged for IV)
   ^TMP($J,LIST,"B",NAME,PSSIEN) = ""
```

Component: NAME

Returns the IEN of the MEDICATION ROUTES file (#51.2) in the array defined by the calling application. The “B” cross-reference in format of ^PS(51.2,“B”, NAME(52.1,.01), IEN(51.2)) will be used.

Status: Inactive

```
NAME^PSS51P2(PSSFT,PSSPK,LIST)
Input:
   PSSFT
   PSSPK
   LIST
Where:
   PSSFT = NAME field (#.01) of MEDICATION ROUTES file (#51.2) (a value of
   "??" may be used) [required]
   PSSPK = PACKAGE USE field (#3) of the MEDICATION ROUTES file (#51.2) (if
   passed in, only entries containing this value will be returned; otherwise,
   all entries will be returned) [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = NAME (51.2,.01)
   ^TMP($J,LIST,PSSIEN,1) = ABBREVIATION (51.2,1)
   ^TMP($J,LIST,PSSIEN,5) = INACTIVATION DATE (51.2,5)^External format (ex:
   SEP 12,1999)
   ^TMP($J,LIST,"B",NAME,PSSIEN) = ""
Where:
   PSSIEN is IEN of entry in the MEDICATION ROUTES file (#51.2)
```

Component: IEN

Returns the IEN(s) of the MEDICATION ROUTES file (#51.2) when passed the ABBREVIATION field (#1) of the MEDICATION ROUTES file (#51.2) in the array defined by the calling application. The “C” cross-reference in format of ^PS(51.2,“C”,ABBREVIATION(52.1,1),IEN(51.2)) will be used.

Status: Inactive

```
IEN^PSS51P2(PSSABBR,LIST)
Input:
   PSSABBR
   LIST
Where:
   PSSABBR = ABBREVIATION field (#1) of the MEDICATION ROUTES file (#51.2)
   [required]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = NAME (51.2,.01)
   ^TMP($J,LIST,PSSIEN,1) = ABBREVIATION (51.2,1)
   ^TMP($J,LIST,PSSIEN,5) = INACTIVATION DATE (51.2,5)^External format (ex:
   SEP 12,1999)
   ^TMP($J,LIST,"B",NAME,PSSIEN) = ""
Where:
   PSSIEN is IEN of entry in the MEDICATION ROUTES file (#51.2)
```

PSS51P5 API – Order Unit file (#51.5)

Component: ALL

Returns all fields in the ORDER UNIT file (#51.5) in the array defined by the calling application.

Status: Inactive

```
ALL^PSS51P5(PSSIEN,PSSFT,PSSCRFL,LIST)
Input:
   PSSIEN
   PSSFT
   PSSCRFL
   LIST
Where:
   PSSIEN = IEN of entry in ORDER UNIT file (#51.5) [optional]
   PSSFT = ABBREVIATION field (#.01) of the ORDER UNIT file (#51.5) (a value
   of "??" may be used) [optional]
   PSSCRFL = Multiple index lookup is performed if passed in a 1 (otherwise
   only the "B" cross-reference is used) [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = ABBREVIATION (51.5,.01)
   ^TMP($J,LIST,PSSIEN,.02) = EXPANSION (51.5,.02)
   ^TMP($J,LIST,"B",ABBREVIATION,PSSIEN) = ""
```

Component: EXPAN

Returns all fields in the ORDER UNIT file (#51.5) in the array defined by the calling application.

Status: Inactive

```
EXPAN^PSS51P5(PSSEXPAN,LIST)
Input:
   PSSEXPAN
   LIST
Where:
   PSSEXPAN = EXPANSION field (#.02) of the ORDER UNIT file (#51.5) [required]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = ABBREVIATION (51.5,.01)
   ^TMP($J,LIST,PSSIEN,.02) = EXPANSION (51.5,.02)
   ^TMP($J,LIST,"C",ABBREVIATION,PSSIEN) = ""
Where:
   PSSIEN is IEN of entry in the ORDER UNIT file (#51.5)
```

PSS52P6 API – IV Additives file (#52.6)

Component: ZERO

Returns the zero node and the INACTIVATION DATE field (#12) of the IV ADDITIVES file (#52.6) in the array defined by the calling application.

Status: Active

```
ZERO^PSS52P6(PSSIEN,PSSFT,PSSFL,LIST)
Input:
   PSSIEN
   PSSFT
   PSSFL
   LIST
Where:
   PSSIEN = IEN of entry in IV ADDITIVES file (#52.6) [optional]
   PSSFT = PRINT NAME field (#.01) of IV ADDITIVES file (#52.6) (a value of
   "??" may be used) [optional]
   PSSFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = PRINT NAME (52.6,.01)
   ^TMP($J,LIST,PSSIEN,1) = GENERIC DRUG (52.6,1)^GENERIC NAME (50,.01)
   ^TMP($J,LIST,PSSIEN,2) = DRUG UNIT (52.6,2)^Drug Unit external format (ex:
   MG, ML…)
   ^TMP($J,LIST,PSSIEN,3) = NUMBER OF DAYS FOR IV ORDER (52.6,3)
   ^TMP($J,LIST,PSSIEN,4) = USUAL IV SCHEDULE (52.6,4)
   ^TMP($J,LIST,PSSIEN,5) = ADMINISTRATION TIMES (52.6,5)
   ^TMP($J,LIST,PSSIEN,7) = AVERAGE DRUG COST PER UNIT (52.6,7)
   ^TMP($J,LIST,PSSIEN,12) = INACTIVATION DATE (52.6,12)^External format (ex:
   SEP 12,1999)
   ^TMP($J,LIST,PSSIEN,13) = CONCENTRATION (52.6,13)
   TMP($J,LIST,PSSIEN,14) = MESSAGE (52.6,14)
   ^TMP($J,LIST,PSSIEN,15) = PHARMACY ORDERABLE ITEM (52.6,15)^NAME (50.7,.01)
   ^TMP($J,LIST,PSSIEN,17) = USED IN IV FLUID ORDER ENTRY (52.6,17)^External
   format (ex: "YES" for 1 otherwise it is null)
   ^TMP($J,LIST,"B",PRINT NAME,PSSIEN) = ""
```

Component: QCODE

Returns the fields in the QUICK CODE multiple of the IV ADDITIVES file (#52.6) in the array defined by the calling application.

Status: Inactive

```
QCODE^PSS52P6(PSSIEN,PSSFT,PSSFL,LIST)
Input:
   PSSIEN
   PSSFT
   PSSFL
   LIST
Where:
   PSSIEN = IEN of entry in IV ADDITIVES file (#52.6) [optional]
   PSSFT = PRINT NAME field (#.01) of IV ADDITIVES file (#52.6) (a value of
   "??" may be used) [optional]
   PSSFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = PRINT NAME (52.6,.01)
   ^TMP($J,LIST,PSSIEN,"QCODE",0) = Total entries returned for this sub-file
   or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,"QCODE",PSS(1),.01) = QUICK CODE (52.61,.01)
   ^TMP($J,LIST,PSSIEN,"QCODE",PSS(1),1) = STRENGTH (52.61,1)
   ^TMP($J,LIST,PSSIEN,"QCODE",PSS(1),2) = USUAL INFUSION RATE (52.61,2)
   ^TMP($J,LIST,PSSIEN,"QCODE",PSS(1),3) = OTHER PRINT INFO (52.61,3)
   ^TMP($J,LIST,PSSIEN,"QCODE",PSS(1),4) = USUAL IV SCHEDULE (52.61,4)
   ^TMP($J,LIST,PSSIEN,"QCODE",PSS(1),5) = ADMINISTRATION TIMES (52.61,5)
   ^TMP($J,LIST,PSSIEN,"QCODE",PSS(1),6) = USUAL IV SOLUTION (52.61,6)^PRINT
   NAME (52.7,.01)_" "_VOLUME (52.7,2)
   ^TMP($J,LIST,PSSIEN,"QCODE",PSS(1),7) = MED ROUTE (52.61,7)^NAME (51.2,.01)
   ^TMP($J,LIST,"B",PRINT NAME,PSSIEN) = ""
Where:
   PSS(1) is the IEN of entry in the QUICK CODE multiple
```

Component: ELYTES

Returns the fields in the ELECTROLYTES multiple of the IV ADDITIVES file (#52.6) in the array defined by the calling application.

Status: Inactive

```
ELYTES^PSS52P6(PSSIEN,PSSFT,PSSFL,LIST)
Input:
   PSSIEN
   PSSFT
   PSSFL
   LIST
Where:
   PSSIEN = IEN of entry in IV ADDITIVES file (#52.6) [optional]
   PSSFT = PRINT NAME field (#.01) of IV ADDITIVES file (#52.6) (a value of
   "??" may be used) [optional]
   PSSFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = PRINT NAME (52.6,.01)
   ^TMP($J,LIST,PSSIEN,"ELYTES",0) = Total entries returned for this sub-file
   or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,"ELYTES",PSS(1),.01) = ELECTROLYTE (52.62,.01)^NAME
   (50.4,.01) (external format)
   ^TMP($J,LIST,PSSIEN,"ELYTES",PSS(1),1) = CONCENTRATION (52.62,1)
   ^TMP($J,LIST,"B",PRINT NAME,PSSIEN) = ""
Where:
   PSS(1) is the IEN of entry in the ELECTROLYTE multiple
```

Component: SYNONYM

Returns the fields in the SYNONYM multiple of the IV ADDITIVES file (#52.6) in the array defined by the calling application.

Status: Inactive

```
SYNONYM^PSS52P6(PSSIEN,PSSFT,PSSFL,LIST)
Input:
   PSSIEN
   PSSFT
   PSSFL
   LIST
Where:
   PSSIEN = IEN of entry in IV ADDITIVES file (#52.6) [optional]
   PSSFT = PRINT NAME field (#.01) of IV ADDITIVES file (#52.6) (a value of
   "??" may be used) [optional]
   PSSFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = PRINT NAME (52.6,.01)
   ^TMP($J,LIST,PSSIEN,"SYN",0) = Total entries returned for this sub-file or
   -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,"SYN",PSS(1),.01) = SYNONYM (52.63,.01)
   ^TMP($J,LIST,"B",PRINT NAME,PSSIEN) = ""
Where:
   PSS(1) is the IEN of entry in the SYNONYM multiple
```

Component: DRGINFO

Returns the fields in the DRUG INFORMATION multiple of the IV ADDITIVES file (#52.6) in the array defined by the calling application.

Status: Inactive

```
DRGINFO^PSS52P6(PSSIEN,PSSFT,PSSFL,LIST)
Input:
   PSSIEN
   PSSFT
   PSSFL
   LIST
Where:
   PSSIEN = IEN of entry in IV ADDITIVES file (#52.6) [optional]
   PSSFT = PRINT NAME field (#.01) of IV ADDITIVES file (#52.6) (a value of
   "??" may be used) [optional]
   PSSFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = PRINT NAME (52.6,.01)
   ^TMP($J,LIST,PSSIEN,"DRGINF",0) = Total entries returned for this sub-file
   or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,"DRGINF",PSS(1),.01) = DRUG INFORMATION (52.64,.01)
   ^TMP($J,LIST,"B",PRINT NAME,PSSIEN) = ""
Where:
   PSS(1) is the IEN of entry in the DRUG INFORMATION multiple
```

Component: DRGIEN

Returns the IEN(s) from the IV ADDITIVES file (#52.6) in the array defined by the calling application with the DRUG file (#50) IEN. The “AC” cross-reference in format of ^PS(52.6,“AC”,IEN(50),IEN(52.6)) will be used.

Status: Inactive

```
DRGIEN^PSS52P6(PSS50,PSSFL,LIST)
Input:
   PSS50
   PSSFL
   LIST
Where:
   PSS50 = IEN of entry in the DRUG file (#50) [required]
   PSSFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = PRINT NAME (52.6,.01)
   ^TMP($J,LIST,"AC",PSS50,PSSIEN) = ""
Where:
   PSSIEN is IEN of entry in the IV ADDITIVES file (#52.6)
```

Component: LOOKUP

Returns the PRINT NAME, QUICK CODE and SYNONYM of the IV ADDITIVES file (#52.6) to the calling application to display to the user. The screening considers only active entries in the IV ADDITIVES file (#52.6) and the PHARMACY ORDERABLE ITEM field (#15) of the IV ADDITIVES file (#52.6) that are not null. Those entries with the INACTIVE DATE field (#100) in the DRUG file (#50) will be excluded.

Status: Inactive

```
LOOKUP^PSS52P6(PSS50P7,PSSFL,LIST)
Input:
   PSS50P7
   PSSFL
   LIST
Where:
   PSS50P7 = A null value or the IEN of entry in the PHARMACY ORDERABLE ITEM
   file (#50.7) [optional]
   PSSFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = PRINT NAME (52.6,.01)
   ^TMP($J,LIST,PSSIEN,14) = MESSAGE (52.6,14)
   ^TMP($J,LIST,PSSIEN,"QCODE",0) = Total entries returned for this sub-file
   or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,"QCODE",PSS(1),.01) = QUICK CODE (52.61,.01)
   ^TMP($J,LIST,PSSIEN,"SYN",0) = Total entries returned for this sub-file or
   -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,"SYN",PSS(1),.01) = SYNONYM (52.63,.01)
   ^TMP($J,LIST,"B",PRINT NAME,PSSIEN) = ""
Where:
   PSSIEN is IEN of entry in the IV ADDITIVES file (#52.6)
   PSS(1) is the IEN of the multiple it referenced
```

☛ Note: If an additive does not tie to an entry in the DRUG file (#50) or PHARMACY ORDERABLE ITEM file (#50.7), and PSSFL is passed in, this additive will not be included in the returned ^TMP.

Component: POI

Returns the IEN(s) from the IV ADDITIVES file (#52.6) in the array defined by the calling application with the given PHARMACY ORDERABLE ITEM file (#50.7). The “AOI” cross-reference in format of ^PS(52.6,“AOI”,IEN(50.7),IEN(52.6)) will be used.

Status: Inactive

```
POI^PSS52P6(PSSOI,PSSFL,LIST)
Input:
   PSSOI
   PSSFL
   LIST
Where:
   PSSPOI = IEN of entry in the PHARMACY ORDERABLE ITEM file (#50.7)
   [required]
   PSSFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = PRINT NAME (52.6,.01)
   ^TMP($J,LIST,PSSIEN,2) = VOLUME (52.7,2)
   ^TMP($J,LIST,"AOI",PSSPOI,PSSIEN) = ""
Where:
   PSSIEN is IEN of entry in the IV ADDITIVES file (#52.6)
```

PSS52P7 API – IV Solutions file (#52.7)

Component: ZERO

Returns the zero node, the INACTIVATION DATE field (#12) and the ELECTROLYTES multiple of the IV SOLUTIONS file (#52.7) in the array defined by the calling application.

Status: Active

```
ZERO^PSS52P7(PSSIEN,PSSFT,PSSFL,LIST)
Input:
   PSSIEN
   PSSFT
   PSSFL
   LIST
Where:
   PSSIEN = IEN of entry in IV SOLUTIONS file (#52.7) [optional]
   PSSFT = PRINT NAME field (#.01) of IV SOLUTIONS file (#52.7) (a value of
   "??" may be used) [optional]
   PSSFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = PRINT NAME (52.7,.01)
   ^TMP($J,LIST,PSSIEN,.02) = PRINT NAME {2} (52.7,.02)
   ^TMP($J,LIST,PSSIEN,1) = GENERIC DRUG (52.7,1)^GENERIC NAME (50,.01)
   ^TMP($J,LIST,PSSIEN,2) = VOLUME (52.7,2)
   ^TMP($J,LIST,PSSIEN,7) = AVERAGE DRUG COST (52.7,7)
   ^TMP($J,LIST,PSSIEN,8) = INACTIVATION DATE (52.7,8)^External format (ex:
   SEP 12, 1999)
   ^TMP($J,LIST,PSSIEN,9) = PHARMACY ORDERABLE ITEM (52.7,9)^NAME (50.7,.01)
   ^TMP($J,LIST,PSSIEN,17) = USED IN IV FLUID ORDER ENTRY (52.7,17)^External
   format (ex: "YES" for 1 otherwise it is null)
   ^TMP($J,LIST,PSSIEN,"ELYTES",0) = Total entries returned for this sub-file
   or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,"ELYTES",PSS(1),.01) = ELECTROLYTES (52.702,.01)^NAME
   (50.4,.01) (external format)
   ^TMP($J,LIST,PSSIEN,"ELYTES",PSS(1),1) = CONCENTRATION (52.702,1)
   ^TMP($J,LIST,"B",PRINT NAME,PSSIEN) = ""
Where:
   PSS(1) is the IEN of entry in the ELECTROLYTES multiple
```

Component: DRGIEN

Returns the IEN(s) from the IV SOLUTIONS file (#52.7) in the array defined by the calling application with the DRUG file (#50) IEN. The “AC” cross-reference in format of ^PS(52.7,“AC”,IEN(50),IEN(52.7)) will be used.

Status: Inactive

```
DRGIEN^PSS52P7(PSS50,PSSFL,LIST)
Input:
   PSS50
   PSSFL
   LIST
Where:
   PSS50 = IEN of entry in the DRUG file (#50) [required]
   PSSFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = PRINT NAME (52.7,.01)
   ^TMP($J,LIST,PSSIEN,2) = VOLUME (52.7,2)
   ^TMP($J,LIST,"AC",PRINT NAME,PSSIEN) = ""
Where:
   PSSIEN is IEN of entry in the IV SOLUTIONS file (#52.7)
```

Component: INACTDT

Returns the INACTIVATION DATE (52.7,8) (ex: 3030915 or null).

Status: Inactive

```
S X=$$INACTDT^PSS52P7(PSSIEN)
Input:
   PSSIEN
Where:
   IEN = IEN of entry in the IV SOLUTIONS file (#52.7) [required]
Output:
   $$INACTDT
Where:
   X = internal value of the INACTIVATION DATE (52.7,8) (ex: 3030915 or null)
```

Component: LOOKUP

Returns PRINT NAME, PRINT NAME(2), and VOLUME from the IV SOLUTIONS file (#52.7).

Status: Inactive

```
LOOKUP^PSS52P7(PSSFT,LIST)
Input:
   PSSFT
   LIST
Where:
   PSSFT = PRINT NAME field (#.01) of IV SLUTIONS file (#52.7) (a value of
   "??" may be used) [required]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = PRINT NAME (52.7,.01)
   ^TMP($J,LIST,PSSIEN,.02) = PRINT NAME {2} (52.7,.02)
   ^TMP($J,LIST,PSSIEN,2) = VOLUME (52.7,2)
   ^TMP($J,LIST,"B",PRINT NAME,PSSIEN) = ""
Where:
   PSSIEN is IEN of entry in the IV SOLUTIONS file (#52.7)
```

Component: POICHK

Returns the PHARMACY ORDERABLE ITEM file (#50.7) IEN if the value in the PHARMACY ORDERABLE ITEM field (#9) of the IV SOLUTIONS file (#52.7) existed.

Status: Inactive

```
S X=$$POICHK^PSS52P7(PSSIEN)
Input:
   PSSIEN
Where:
   PSSIEN = IEN of the IV SOLUTIONS file (#52.7) [required]
Output:
   $$POICHK
Where:
   X = IEN of entry in the PHARMACY ORDERABLE ITEM file (#50.7) or 0 (zero)
   will be returned if it doesn’t exist
```

Component: POI

Returns the IEN(s) from the IV SOLUTIONS file (#52.7) in the array defined by the calling application with the given PHARMACY ITEM ORDERABLE file (#50.7) IEN. The “AOI” cross-reference in format of ^PS(52.7,“AOI”,IEN(50.7),IEN(52.7)) will be used.

Status: Inactive

```
POI^PSS52P7(PSSOI,PSSFL,LIST)
Input:
   PSSOI
   PSSFL
   LIST
Where:
   PSSOI = IEN of entry in the PHARMACY ORDERABLE ITEM file (#50.7) [required]
   PSSFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = PRINT NAME (52.7,.01)
   ^TMP($J,LIST,PSSIEN,2) = VOLUME (52.7,2)
   ^TMP($J,LIST,"AOI",PRINT NAME,PSSIEN) = ""
Where:
   PSSIEN is IEN of entry in the IV SOLUTIONS file (#52.7)
```

Component: ACTSOL

Returns the IEN(s), PRINT NAME, and VOLUME from the IV SOLUTIONS file (#52.7) for active entries. A count of all active entries shall be included.

Status: Inactive

```
ACTSOL^PSS52P7(PSSFL,LIST)
Input:
   PSSFL
   LIST
Where:
   PSSFL = Inactive date: A null value will return all entries (entry of a
   FileMan format date (ex: 3030917) will return active entries after this
   date) [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = PRINT NAME (52.7,.01)
   ^TMP($J,LIST,PSSIEN,2) = VOLUME (52.7,2)
   ^TMP($J,LIST,"B",PRINT NAME,PSSIEN) = ""
Where:
   PSSIEN is IEN of entry in the IV SOLUTIONS file (#52.7)
```

PSS54 API – RX Consult file (#54)

Component: ALL

Returns the NAME field (#.01) and the TEXT multiple of the RX CONSULT file (#54) in the array defined by the calling application.

Status: Inactive

```
ALL^PSS54(PSSIEN,PSSFT,LIST)
Input:
   PSSIEN
   PSSFT
   LIST
Where:
   PSSIEN = IEN of entry in RX CONSULT file (#54) [optional]
   PSSFT = NAME field(#.01) of RX CONSULT file (#54) (a value of "??" may be
   used) [optional]
   LIST = array name defined by the calling application [required]

Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = NAME (54,.01)
   ^TMP($J,LIST,PSSIEN,"TXT",0) = Total entries returned for this sub-file or
   -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,"TXT",PSS(1),.01) = TEXT (54.1,.01)
   ^TMP($J,LIST,"B",NAME,PSSIEN) = ""

Where:
   PSS(1) is the IEN of entry in the TEXT multiple
```

Component: LOOKUP

Returns the NAME field (#.01) from the RX CONSULT file (#54).

Status: Inactive

```
LOOKUP^PSS54(PSSSRCH,LIST)
Input:
   PSSSRCH
   LIST
Where:
   PSSSRCH = Lookup free text value entered by the user [required]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = NAME (54,.01)
   ^TMP($J,LIST,"B",NAME,PSSIEN) = ""
Where:
   PSSIEN is IEN of entry in the RX CONSULT file (#54)
```

PSS55 API – PHARMACY PATIENT file (#55)

For the Rx Profile information on a patient, please see PSO52API under the Outpatient Pharmacy section of this manual using the link provided below. The prescription profile fields include: RX #, PATIENT, PATIENT STATUS, PROVIDER, CLINIC, DRUG, QTY, DAYS SUPPLY, # OF REFILLS, MAIL/WINDOW, ISSUE DATE, ENTERED BY, UNIT PRICE OF DRUG, TRADE NAME, COPIES, and ORDER CONVERTED.

PSO52API - Component PROF

Component: PSS431

Returns a list of Unit Dose medications from data in the UNIT DOSE sub-file (#55.06) of the PHARMACY PATIENT file (#55) and dispensed drugs from the DISPENSED DRUGS sub-file (#55.07).

Status: Active

```
PSS431^PSS55(DFN,PO,PSDATE,PEDATE,LIST)
```

```
Input:
   DFN
   PO
   PSDATE
   PEDATE
   LIST
Where:
   DFN = IEN from the PATIENT file (#2) [required]
   PO = Order number [optional]
   PSDATE = Start Date [optional]
   PEDATE = End Date [optional]
   LIST = Subscript name used in the ^TMP global [required]
Output:
   ^TMP($J,LIST,0)=Total entries returned by patient or -1^NO DATA FOUND
   ^TMP($J,LIST ,PO,.01)=ORDER NUMBER(55.06,.01)
   ^TMP($J,LIST,PO,.5)=PATIENT NAME (DFN) (55.06,.5 - P)^NAME (2,.01)
   ^TMP($J,LIST,PO,1)=PROVIDER (pointer value) (55.06,1 - P)^NAME (200,.01)
   ^TMP($J,LIST,PO,3)=MED ROUTE (pointer value) (55.06,3 - P) ^NAME (51.2,.01)
   ^TMP($J,LIST,PO,4)=TYPE (55.06,4 - S)^External Format for the Set of Codes
   ^TMP($J,LIST,PO,5)=SELF MED (55.06,5 - S)^External Format for the Set of
   Codes
   ^TMP($J,LIST,PO,6)=HOSPITAL SUPPLIED SELF MED (55.06,6 - S)^External Format
   for the Set of Codes
   ^TMP($J,LIST,PO,7)=SCHEDULE TYPE (55.06,7 - S)^External Format for the Set
   of Codes
   ^TMP($J,LIST,PO,11)=DAY LIMIT(55.06,11)
   ^TMP($J,LIST,PO,12)=DOSE LIMIT(55.06,12)
   ^TMP($J,LIST,PO,27)=ORDER DATE(55.06,27 – D/T)^External Format (ex: Sep.
   12, 1999)
   ^TMP($J,LIST,PO,27.1)=LOG-IN DATE(55.06,27.1 – D/T)^External Format (ex:
   Sep. 12, 1999)
   ^TMP($J,LIST,PO,28)=STATUS (55.06,28 - S)^External Format for the Set of
   Codes
   ^TMP($J,LIST,PO,”DDRUG”,0)= Total entries returned or -1 ^NO DATA FOUND
   ^TMP($J,LIST,PO, ,”DDRUG”,.01)=DISPENSE DRUG (55.07,.01 - P)^GENERIC NAME
   (50,.01)
   ^TMP($J,LIST,PO,”DDRUG”,.02)=UNITS PER DOSE(55.07,.02)
   ^TMP($J,LIST,PO, ,”DDRUG”,.03)=INACTIVE DATE(55.07,.03 – D/T)^External
   Format (ex: Sep. 12, 1999)
   ^TMP($J,LIST,PO,”DDRUG”,.04)=TOTALS UNITS DISPENSED(55.07,.04)
   ^TMP($J,LIST,PO,”DDRUG”,.05)=UNITS CALLED FOR(55.07,.05)
   ^TMP($J,LIST,PO,”DDRUG”,.06)=UNITS ACTUALLY DISPENSED(55.07,.06)
   ^TMP($J,LIST,PO,”DDRUG”,.07)=TOTAL RETURNS(55.07,.07)
   ^TMP($J,LIST,PO,”DDRUG”,.08)=RETURNS(55.07,.08)
   ^TMP($J,LIST,PO,”DDRUG”,.09)=PRE-EXCHANGE UNITS(55.07,.09)
   ^TMP($J,LIST,PO,”DDRUG”,.1)=TOTALS EXTRA UNITS DISPENSED(55.07,.1)
   ^TMP($J,LIST,PO,”DDRUG”,.11)=EXTRA UNITS DISPENSED(55.07,.11)
   ^TMP($J,LIST,PO,”DDRUG”,.12)=TOTAL PRE-EXCHANGE UNITS(55.07,.12)
   ^TMP($J,LIST,”B”,ORDER NUMBER)=””
Where:
   DDRUG = IEN of entry in the DISPENSE DRUG multiple
```

Component: PSS432

Returns a listing of active orders by scanning the PHARMACY PATIENT file (#55) utilizing the “AUS” cross-reference.

Status: Active

```
PSS432^PSS55(DFN,PO,LIST)
```

```
Input:
   DFN
   PO
   LIST
Where:
   DFN = IEN from the PATIENT file (#2) [required]
   PO = Order number [optional]
   LIST = Subscript name used in the ^TMP global [required]
Output:
   ^TMP($J,LIST,0)=Total entries returned by patient or -1^NO DATA FOUND
   ^TMP($J,LIST,PO,.01)=ORDER NUMBER(55.06,.01)
   ^TMP($J,LIST,PO,.5)=PATIENT NAME (DFN) (55.06,.5 - P)^NAME (2,.01)
   ^TMP($J,LIST,PO,1)=PROVIDER (55.06,1 - P) ^NAME (200,.01)
   ^TMP($J,LIST,PO,3)=MED ROUTE (55.06,3 - P) ^NAME (51.2,.01)
   ^TMP($J,LIST,PO,4)=TYPE (55.06,4 - S)^External Format for the Set of Codes
   ^TMP($J,LIST,PO,5)=SELF MED (55.06,5 - S)^External Format for the Set of
   Codes
   ^TMP($J,LIST,PO,6)=HOSPITAL SUPPLIED SELF MED (55.06,6 - S)^External Format
   for the Set of Codes
   ^TMP($J,LIST,PO,7)=SCHEDULE TYPE  (55.06,7 - S)^External Format for the Set
   of Codes
   ^TMP($J,LIST,PO,11)=DAY LIMIT(55.06,11)
   ^TMP($J,LIST,PO,12)=DOSE LIMIT(55.06,12)
   ^TMP($J,LIST,PO,27)=ORDER DATE(55.06,27 – D/T)^External Format (ex: Sep.
   12, 1999)
   ^TMP($J,LIST,PO,27.1)=LOG-IN DATE(55.06,27.1 – D/T)^External Format (ex:
   Sep. 12, 1999)
   ^TMP($J,LIST,PO,28)=STATUS (55.06,28 - S)^External Format for the Set of
   Codes
   ^TMP($J,LIST,”B”,ORDER NUMBER)=””
```

Component: PSS433

Returns the Unit Dose data from the “2” node of the UNIT DOSE sub-file (#55.06) of the PHARMACY PATIENT file (#55) for the DFN received.

Status: Active

```
PSS433^PSS55(DFN,LIST)
```

```
Input:
   DFN
   LIST
Where:
   DFN = IEN from the PATIENT file (#2) [required]
   LIST = Subscript name used in the ^TMP global [required]
Output:
   ^TMP($J,LIST,0)=Total entries returned by patient or -1^NO DATA FOUND
   ^TMP($J,LIST,PO,.5)=PATIENT NAME (DFN) (55.06,.5 - P)^NAME (2,.01)
   ^TMP($J,LIST,PO,9)=ORIGINAL WARD (55.06,9 - P) ^NAME (42,.01)
   ^TMP($J,LIST,PO,25)=PREVIOUS STOP DATE/TIME(55.06,25 – D/T)^External Format
   (ex: Sep. 12, 1999)
   ^TMP($J,LIST,PO,26)=SCHEDULE(55.06,26)
   ^TMP($J,LIST,PO,34)=STOP DATE/TIME(55.06,34 – D/T)^External Format (ex:
   Sep. 12, 1999)
   ^TMP($J,LIST,PO,41)=ADMIN TIMES(55.06,41)
   ^TMP($J,LIST,PO,42)=FREQUENCY (in minutes)(55.06,42)
   ^TMP($J,LIST,PO,70)=ORIGINAL START DATE/TIME(55.06,70 – D/T)^External
   Format (ex: Sep. 12, 1999)
   ^TMP($J,LIST,”B”,ORDER NUMBER)=””
```

Component: PSS435

Returns the Start Date and Time and/or a list of active Hyperal IV Orders utilizing the “AIT” cross-reference on the STOP DATE/TIME field (#.02) for the DFN received.

Status: Active

```
PSS435^PSS55(DFN,PO,LIST)
```

```
Input:
   DFN
   PO
   LIST
Where:
   DFN = IEN from the PATIENT file (#2) [required]
   PO = Order number [optional]
   LIST = Subscript name used in the ^TMP global [required]
Output:
   ^TMP($J,LIST,0)=Total entries returned by patient or -1^NO DATA FOUND
   ^TMP($J,LIST,PO,.01)=ORDER NUMBER(55.01,.01)
   ^TMP($J,LIST,PO,.02)=START DATE/TIME(55.01,.02 – D/T)^External Format (ex:
   Sep. 12, 1999)
   ^TMP($J,LIST,PO,.03)=STOP DATE/TIME(55.01,.03 – D/T)^External Format (ex:
   Sep. 12, 1999)
   ^TMP($J,LIST,PO,.04)=TYPE (55.01,.04 - S)^External Format for the Set of
   Codes
   ^TMP($J,LIST,PO,.06)=PROVIDER (55.01,.06 - P)^NAME (200,.01)
   ^TMP($J,LIST,PO,.08)=INFUSION RATE(55.01,.08)
   ^TMP($J,LIST,PO,.09)=SCHEDULE(55.01,.09)
   ^TMP($J,LIST,PO,.12)=ADMINISTRATION TIMES(55.01,.12)
   ^TMP($J,LIST,PO,.17)=SCHEDULE INTERVAL(55.01,.17)
   ^TMP($J,LIST,PO,.24)=CUMULATIVE DOSES(55.01,.24)
   ^TMP($J,LIST,PO,100)=STATUS (55.01,100 - S)^External Format for the Set of
   Codes
   ^TMP($J,LIST,PO,104)=WARD(55.01,104)
   ^TMP($J,LIST,PO,106)=CHEMOTHERAPY TYPE (55.01,106 - S)^External Format for
   the Set of Codes
   ^TMP($J,LIST,PO,108)=INTERMITTENT SYRINGE(55.01,108)
   ^TMP($J,LIST,PO,110)=ORDERS FILE ENTRY(55.01,110)
   ^TMP($J,LIST,PO,112)=ATZERO (55.01,112 - S)^External Format for the Set of
   Codes
   ^TMP($J,LIST,PO,120)=OERR HOLD FLAG  (55.01,120 - S)^External Format for
   the Set of Codes
   ^TMP($J,LIST,PO,121)=AUTO DC (55.01,121 - S)^External Format for the Set of
   Codes
   ^TMP($J,LIST,”B”,ORDER NUMBER)=””
```

Component: PSS436

Returns a list of the active IV Additive Information, found on the AD nodes of the PHARMACY PATIENT file (#55) for the DFN received.

Status: Active

```
PSS436^PSS55(DFN,ORDER,LIST)

Input:
   DFN
   ORDER
   LIST
Where:
   DFN = IEN from the PATIENT file (#2) [required]
   ORDER = Order number [required]
   LIST = Subscript name used in the ^TMP global [required]
Output:
   ^TMP($J,LIST,0)=Total entries returned by patient or -1^NO DATA FOUND
   ^TMP($J,LIST,ORDER,.01)=ORDER NUMBER(55.01,.01)
   ^TMP($J,LIST,ORDER,.02)=START DATE/TIME(55.01,.02 – D/T)^External Format
   (ex: Sep. 12, 1999)
   ^TMP($J,LIST,ORDER,.03)=STOP DATE/TIME(55.01,.03- D/T)^External Format (ex:
   Sep. 12, 1999)
   ^TMP($J,LIST,ORDER,.04)=TYPE  (55.01,.04 - S)^External Format for the Set
   of Codes
   ^TMP($J,LIST,ORDER,.06)=PROVIDER(55.01,.06 - P)^NAME (200,.01)
   ^TMP($J,LIST,ORDER,.08)=INFUSION RATE(55.01,.08)
   ^TMP($J,LIST,ORDER,.09)=SCHEDULE(55.01,.09)
   ^TMP($J,LIST,ORDER,.12)=ADMINISTRATION TIMES(55.01,.12)
   ^TMP($J,LIST,ORDER,.17)=SCHEDULE INTERVAL(55.01,.17)
   ^TMP($J,LIST,ORDER,.24)=CUMULATIVE DOSES(55.01,.24)
   ^TMP($J,LIST,ORDER,100)=STATUS (55.01,100 - S)^External Format for the Set
   of Codes
   ^TMP($J,LIST,ORDER,104)=WARD(55.01,104)
   ^TMP($J,LIST,ORDER,106)=CHEMOTHERAPY TYPE (55.01,106 - S)^External Format
   for the Set of Codes
   ^TMP($J,LIST,ORDER,108)=INTERMITTENT SYRINGE (55.01,108 - S)^External
   Format for the Set of Codes
   ^TMP($J,LIST,ORDER,110)=ORDERS FILE ENTRY(55.01,110)
   ^TMP($J,LIST,ORDER,112)=ATZERO (55.01,112 - S)^External Format for the Set
   of Codes
   ^TMP($J,LIST,ORDER,120)=OERR HOLD FLAG (55.01,120 - S)^External Format for
   the Set of Codes
   ^TMP($J,LIST,ORDER,121)=AUTO DC (55.01,121 - S)^External Format for the Set
   of Codes
   ^TMP($J,LIST,ORDER,147)=BCMA EXPIRED FLAG (55.01,147 - S)^External Format
   for the Set of Codes
   ^TMP($J,LIST,IEN,”ADD”,0)=Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,ORDER,”ADD”,n1,.01)=ADDITIVE (55.02,.01 - P) ^PRINT NAME
   (52.6,.01)
   ^TMP($J,LIST,ORDER,”ADD”,n1,.02)=STRENGTH(55.02,.02)
   ^TMP($J,LIST,ORDER,”ADD”,n1,.03)=BOTTLE(55.02,.03)
   ^TMP($J,LIST,IEN,”SOL”,0)=Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,ORDER,”SOL”,n2,.01)=SOLUTION (55.11,.01 - P)^PRINT NAME
   (52.7,.01)
   ^TMP($J,LIST,ORDER,”SOL”,n2,1)=VOLUME(55.11, 1)
   ^TMP($J,LIST,”B”,ORDER NUMBER)=””
Where:
   n1 = IEN of entry in the ADDITIVE multiple
   n2 = IEN of entry in the SOLUTION multiple
```

PSS59P7 API – PHARMACY SYSTEM file (#59.7)

Component: PSS

Returns the OUTPATIENT VERSION field (#49.99) and ADMISSION CANCEL OF RXS field (#40.1) from the PHARMACY SYSTEM file (#59.7) for the IEN or free text entry received.

Status: Active

```
PSS^PSS59P7(PSSIEN,PSSTXT,LIST)

Input:
   PSSIEN
   PSSTXT
   LIST
Where:
   PSSIEN = IEN of entry in PHARMACY SYSTEM file (#59.7) [optional]
   PSSTXT = Free text entry [optional]
   LIST = Subscript name used in the ^TMP global [required]
```

☛ Note**:** `Either the IEN or free text entry must be present.`

```
Output:
   ^TMP($J,LIST,PSSIEN,40.1)=ADMISSION CANCEL OF RXS (59.7,40.1 - S)^External
   Format for the Set of Codes
   ^TMP($J,LIST,PSSIEN,49.99)=OUTPATIENT VERSION(59.7,49.99)
   ^TMP($J,LIST,”B”,SITE NAME,PSSIEN)=””
```

PSS32P3 API – APSP Intervention Type file (#9009032.3)

Component: ALL

Returns the TYPE field (#.01) of the APSP INTERVENTION TYPE file (#9009032.3).

Status: Inactive

```
ALL^PSS32P3(PSSIEN,PSSFT,LIST)
Input:
   PSSIEN
   PSSFT
   LIST
Where:
   PSSIEN = IEN of entry in APSP INTERVENTION TYPE file (#9009032.3)
   [optional]
   PSSFT = TYPE field (#.01) of APSP INTERVENTION TYPE file (#9009032.3) (a
   value of "??" may be used) [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = TYPE (9009032.3,.01)
   ^TMP($J,LIST,"B",TYPE,PSSIEN) = ""
```

PSS32P5 API – APSP Intervention Recommendation file (#9009032.5)

Component: ALL

Returns the RECOMMENDATION field (#.01) of the APSP INTERVENTION RECOMMENDATION file (#9009032.5).

Status: Inactive

```
ALL^PSS32P5(PSSIEN,PSSFT,LIST)
Input:
   PSSIEN
   PSSFT
   LIST
Where:
   PSSIEN = IEN of entry in APSP INTERVENTION RECOMMENDATION file (#9009032.5)
   [optional]
   PSSFT = RECOMMENDATION field (#.01) of APSP INTERVENTION RECOMMENDATION
   file (#9009032.5) (a value of "??" may be used) [optional]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,PSSIEN,.01) = RECOMMENDATION (9009032.5,.01)
   ^TMP($J,LIST,"B",RECOMMENDATION,PSSIEN) = ""
```

PSSFILES API – Help Text

Component: HLP

Returns help text for “?” input value on the specified PDM file in the array defined by the calling application.

Status: Inactive

```
HLP^PSSFILES(PSSFILE,LIST)
Input:
   PSSFILE
   LIST
Where:
   PSSFILE = PDM file number (ex: 50 for DRUG file, 52.6 for IV ADDITIVES
   file) [required]
   LIST = array name defined by the calling application [required]
Output:
   ^TMP($J,LIST,0) = Total entries returned or -1^NO DATA FOUND
   ^TMP($J,LIST,X) = HELP TEXT
Where:
   X is the index number
```

PSSDI API – FileMan Calls

Component: DIC - Lookup Call ^DIC

Accepts input values and return output values as defined by VA FileMan Lookup call ^DIC.

Status: Active

```
DIC^PSSDI(PSSFILE,PSSAPP,.DIC,.X,DLAYGO,PSSSCRDT,PSSSCRUS)
Input:
   PSSFILE
   PSSAPP
   PSSSCRDT
   PSSSCRUS
Where:
   PSSFILE = File number used for validation of access [required]
   PSSAPP = Name space of the calling application (ex: PSJ for Inpatient Meds,
   PSO for Outpatient; PSSAPP is used to check if write access is allowed)
   [optional]
   See VA FileMan Programmer Manual for ^DIC call, for DIC, X, and DLAYGO
   input definitions
   PSSSCRDT = Inactivation Date. If the file has an Inactivation Date, then
   any entry with an Inactivation Date on or before PSNDATE will not be
   returned.
   PSSSCRUS = APPLICATION PACKAGES' USE. This parameter only applies when the
   file is the DRUG file (#50). PSSSCRUS should be passed in the form of a
   String. If any of the characters in the PSSSCRUS String is contained in the
   APPLICATION PACKAGES' USE field (#63), then the entry will be returned.
Output:
   PSSDIY will return null if the values for PSSFILE and PSSAPP are valid (it
   will return -1 if conditions were not met)
Where:
   See VA FileMan Programmer Manual for ^DIC output definition
   DLAYGO should only be passed in if the calling application has this type of
   access through another Integration Agreement
```

Component: DO - Lookup Call DO^DIC1

This API will accept input values and return output values as defined by VA FileMan Lookup call DO^DIC1.

Status: Active

```
DO^PSSDI(PSSFILE,PSSAPP,.DIC)
Input:
   PSSFILE
   PSSAPP
Where:
   PSSFILE = File number used for validation of access [required]
   PSSAPP = Name space of the calling application (ex: PSJ for Inpatient Meds,
   PSO for Outpatient; PSSAPP is used to check if write access is allowed)
   [optional]
   See VA FileMan Programmer Manual for DO^DIC1 call , for DIC input
   definition
Output:
   PSSDIY will return null if the values for PSSFILE and PSSAPP are valid (it
   will return -1 if conditions were not met)
Where:
   See VA FileMan Programmer Manual for DO^DIC1 output definition
```

Component: IX - Lookup/Add call IX^DIC

Accepts input values and return output values as defined by VA FileMan Lookup/Add call IX^DIC.

Status: Inactive

```
IX^PSSDI(PSSFILE,PSSAPP,.DIC,D,.X,DLAYGO)
Input:
   PSSFILE
   PSSAPP
Where:
   PSSFILE = File number used for validation of access [required]
   PSSAPP = Name space of the calling application (ex: PSJ for Inpatient Meds,
   PSO for Outpatient; PSSAPP is used to check if write access is allowed)
   [optional]
   See VA FileMan Programmer Manual for IX^DIC call, for DIC, D, X, and DLAYGO
   input definitions
Output:
   PSSDIY will return null if the values for PSSFILE and PSSAPP are valid (it
   will return -1 if conditions were not met)
Where:
   See VA FileMan Programmer Manual for IX^DIC output definition
   DLAYGO should only be passed in if the calling application has this type of
   access through another Integration Agreement
```

Component: MIX

Accepts input values and return output values as defined by VA FileMan Lookup/Add call MIX^DIC1.

Status: Active

```
MIX^PSSDI(PSSFILE,PSSAPP,.DIC,D,.X,DLAYGO,PSSDATE,PSSUSAGE)
Input:
   PSSFILE
   PSSAPP
   PSSDATE
   PSSUSAGE
Where:
   PSSFILE = File number used for validation of access [required]
   PSSAPP = Name space of the calling application (ex: PSJ for Inpatient Meds,
   PSO for Outpatient; PSSAPP is used to check if write access is allowed)
   [optional]
   See VA FileMan Programmer Manual for MIX^DIC1 call, for DIC, D, X, and
   DLAYGO input definitions
   PSSDATE = Inactivation Date. If the file has an Inactivation Date, then any
   entry with an Inactivation Date on or before PSNDATE will not be returned.
   PSSUSAGE = APPLICATION PACKAGES' USE. This parameter only applies when the
   file is the DRUG file (#50). PSSUSAGE should be passed in the form of a
   String. If any of the characters in the PSSUSAGE String is contained in the
   APPLICATION PACKAGES' USE field (#63), then the entry will be returned.
Output:
   PSSDIY will return null if the values for PSSFILE and PSSAPP are valid (it
   will return -1 if conditions were not met)
Where:
   See VA FileMan Programmer Manual for MIX^DIC1 output definition
   DLAYGO should only be passed in if the calling application has this type of
   access through another Integration Agreement
```

Component: FILE

Accepts input values and return output values as defined by VA FileMan Add call FILE^DIC1.

Status: Inactive

```
FILE^PSSDI(PSSFILE,PSSAPP,.DIC,DA,X,DINUM,DLAYGO)
Input:
   PSSFILE
   PSSAPP
Where:
   PSSFILE = File number used for validation of access [required]
   PSSAPP = Name space of the calling application (ex: PSJ for Inpatient Meds,
   PSO for Outpatient; PSSAPP is used to check if write access is allowed)
   [optional]
   See VA FileMan Programmer Manual for FILE^DICN call, for DIC, DA, X, DINUM,
   and DLAYGO input definitions
Output:
   PSSDIY will return null if the values for PSSFILE and PSSAPP are valid (it
   will return -1 if conditions were not met)
Where:
   See VA FileMan Programmer Manual for FILE^DICN output definition
   DLAYGO should only be passed in if the calling application has this type of
   access through another Integration Agreement
```

Component: DIE

Accepts input values and return output values as defined by VA FileMan Edit Data call ^DIE.

Status: Inactive

```
DIE^PSSDI(PSSFILE,PSSAPP,DIE,DA,DR,DIDEL)
Input:
   PSSFILE
   PSSAPP
Where:
   PSSFILE = File number used for validation of access [required]
   PSSAPP = Name space of the calling application (ex: PSJ for Inpatient Meds,
   PSO for Outpatient; PSSAPP is used to check if write access is allowed)
   [optional]
   See VA FileMan Programmer Manual for ^DIE call, for DIE, DA, DR, and DIDEL
   input definitions
Output:
   PSSDIY will return null if the values for PSSFILE and PSSAPP are valid (it
   will return -1 if conditions were not met)
Where:
   See VA FileMan Programmer Manual for ^DIE output definition
```

Component: EN1

Accepts input values and return output values as defined by VA FileMan Print Data call EN1^DIP.

Status: Inactive

```
EN1^PSSDI(PSSFILE,PSSAPP,DIC,.L,FLDS,BY,.FR,.TO,DHD)
Input:
   PSSFILE
   PSSAPP
Where:
   PSSFILE = File number used for validation of access [required]
   PSSAPP = Name space of the calling application (ex: PSJ for Inpatient Meds,
   PSO for Outpatient; PSSAPP is used to check if write access is allowed)
   [optional]
   See VA FileMan Programmer Manual for EN1^DIP call, for DIC, L, FLDS, BY,
   FR, TO and DHD input definitions
Output:
   PSSDIY will return null if the values for PSSFILE and PSSAPP are valid (it
   will return -1 if conditions were not met)
Where:
   See VA FileMan Programmer Manual for EN1^DIP output definition
```

Component: EN

This API will accept input values and return output values as defined by VA FileMan Data Retrieval call EN^DIQ1.

Status: Active

```
EN^PSSDI(PSSFILE,PSSAPP,DIC,.DR,.DA,.DIQ)
Input:
   PSSFILE
   PSSAPP
Where:
   PSSFILE = File number used for validation of access [required]
   PSSAPP = Name space of the calling application (ex: PSJ for Inpatient Meds,
   PSO for Outpatient; PSSAPP is used to check if write access is allowed)
   [optional]
   See VA FileMan Programmer Manual for EN^DIQ1 call, for DIC, DR, DA, DIQ
   input definitions
Output:
   PSSDIY will return null if the values for PSSFILE and PSSAPP are valid (it
   will return -1 if conditions were not met)
Where:
   See VA FileMan Programmer Manual for EN^DIQ1 output definition
```

Component: FNAME

Returns the field name of the specified Pharmacy file for the field number and file number passed in.

Status: Active

```
S X=$$FNAME^PSSDI(PSSFNO,PSSFILE)
Input:
   PSSFNO
   PSSFILE
Where:
   PSSFNO = Field number [required]
   PSSFILE = File number or sub-file number [required]. This file must be a
   file that is owned by a Pharmacy application.
Output:
   $$FNAME
Where:
   $$FNAME = the field name. If an invalid field number or invalid file number
   is passed in, or the file number passed in is not a file owned by a
   pharmacy application, null will be returned.
```

(This page left blank for two-sided copying.)

Pharmacy Benefits Management (PBM)

At this time there are no supported APIs for the Pharmacy Benefits Management application.

(*This page left blank for two-sided copying.*)

Pharmacy Prescription Practices (PPP)

At this time there are no supported APIs for the Pharmacy Prescription Practices application.

(*This page left blank for two-sided copying.*)

List of File Numbers and Names