

Scheduling Package Refactoring API

Release 1.00

Ray Group International

Sep 04, 2012

Abstract

As part of the Open Source EHR Refactoring project, we have continued working on refactoring the Scheduling module code. This document is to describe what we have done in term of code changes for all interested stakeholders.

Contents

1	Introduction	1
2	Code Walkthrough	3
3	How to use the code	31
4	Installation	32
5	Conclusions	32

1. Introduction

The overarching goal of any code changes we implement is to make the VistA code more modular and readable without altering functionality. *Problem List* was chosen as the first module to be refactored because it has a moderate number of dependencies to other packages, allowing our team to finalize a refactoring strategy and allocate ample time for certification. To date, we have completed a *Problem List* application programming interface (API) that is used both by the roll-and-scroll interface and CPRS.

With the lessons learned from *Problem List*, we are now applying these and our refactoring methodology to *Scheduling*. This module has a significant number of dependencies to other VistA packages, and thus requires an increased concentration to the interactions it has within the system and how we will develop an API. Our team has prepared a dependency document [*RGI_Open Source EHR Refactoring Services_Scheduling Dependency Notes.docx*] outlining these interactions and it is available on our project page on the OSEHRA website [<http://www.osehra.org/group/ehr-refactoring-services>]. We have

Latest version available at the OSEHRA Journal [<http://hdl.handle.net/10909/2>]

Distributed under [Creative Commons Attribution License](#)

specifically chosen this module in order to align with VA's active search into improving scheduling functionality within VistA; any refactoring completed by our project team should be helpful to future *Scheduling* design and implementation efforts..

Scheduling has a well-defined API for the core information that is being used by its dependent packages; details on this API may be found in the PIMS Technical Manual [<http://www.va.gov/vdl/documents/Clinical/Scheduling/pimstm.pdf>]. However, this API is read-only, meaning it provides the ability to view currently scheduled appointments but is not being used by the roll-and-scroll interface. Currently there appears to be no API for the core *Scheduling* functionality of making appointments, check-ins and check-outs, etc.

There is an additional API that is being used by the Primary Care Management module, *Managed Care*. This module does not deal directly with the core *Scheduling* functionality, and thus its interactions with the module can be considered unique and separate; it has a graphical user interface (GUI) and gets/sets its associated data through RPC tags within the *Scheduling* module.

Additionally, we found that there are a number of packages that currently use globals belonging to *Scheduling* directly. Details on these packages can be found in the dependency notes document. (Within the roll-and-scroll interface, routines using globals directly is considered the norm, not the exception.)

Since there is a well-defined API for reading *Scheduling* data, we have focused on the roll-and-scroll interface that is being used for making, changing, or editing appointments. Most of this core functionality resides under the Appointment Management option. We have applied the methodology used for *Problem List* and developed an API that is capable of exposing most of the functionality of the Appointment Management menu. Similar to the API created for *Problem List*, we have worked to minimize direct global access and the number of assumed variables within the *Scheduling* API.

2. Code Walkthrough

Several APIs were developed during the refactoring effort. They are described below.

2.1 Clinic management API

\$\$GETCLN^SDMAPI1() - Return Clinic details

This extrinsic function returns detailed information on a clinic.

Format

\$\$GETCLN^SDMAPI1(.RETURN,CLN)

Input Parameters

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

RETURN("CREDIT STOP CODE") – Clinic credit stop code

RETURN("DEFAULT APPOINTMENT TYPE") – Default appointment type

RETURN("DISPLAY INCREMENTS PER HOUR")

RETURN("HOUR CLINIC DISPLAY BEGINS")

RETURN("INACTIVATE DATE")

RETURN("LENGTH OF APP'T")

RETURN("MAX # DAYS FOR AUTO-REBOOK")

RETURN("MAX # DAYS FOR FUTURE BOOKING")

RETURN("NAME")

RETURN("NON-COUNT CLINIC?")

RETURN("OVERBOOKS/DAY MAXIMUM")

RETURN("PRIVILEGED USER",IFN) – List of users that have rights to make appointments.

RETURN("PROHIBIT ACCESS TO CLINIC?")

RETURN("REACTIVATE DATE")

RETURN("SCHEDULE ON HOLIDAYS?")

RETURN("STOP CODE NUMBER")

RETURN("TYPE") – Type of clinic

RETURN("VARIABLE APP'NTMENT LENGTH")

CLN (Required) Clinic IFN

Output

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

Error Codes Returned

None

\$\$LSTCLNS^SDMAPI1() - Return clinics filtered by name

This extrinsic function returns a list of clinics filtered by name.

Format

\$\$LSTCLNS^SDMAPI1(.RETURN,SEARCH,.START,NUMBER)

Input Parameters

.RETURN (Required) Array passed by reference that will receive the data.

RETURN(IND,"ID") – Clinic IFN

RETURN(IND,"NAME") – Clinic name

SEARCH (Optional) Partial match restriction. Default: All entries

.START (Required) The entry from which to begin the list.

NUMBER (Optional) Number of entries to return. Default: All entries

Output

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

Error Codes Returned

None

\$\$SLOTS^SDMAPI1() – Get available slots

This extrinsic function returns the available slots for a clinic.

Format

\$\$SLOTS^SDMAPI1(.RETURN,SC)

Input Parameters

.RETURN (Required) Array passed by reference that will receive the data

RETURN(IND,"DATE") – AvailAbility string

SC (Required) Clinic IFN

Output

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

Error Codes Returned

None

\$\$FRSTAVBL^SDMAPI1() - Find first available appointment date

This extrinsic function finds the first available appointment date for a clinic.

Format

\$\$FRSTAVBL^SDMAPI1(.RETURN,SC)

Input Parameters

.RETURN (Required) Passed by reference, set to the first available date.

SC (Required) Clinic IFN

Output

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

Error Codes Returned

None

\$\$GETAPTS^SDMAPI1() - Get patient appointments

This extrinsic function returns a list of patient appointments.

Format

\$\$GETAPTS^SDMAPI1(.RETURN,DFN,SD)

Input Parameters

.RETURN (Required) Array passed by reference that will receive the data

RETURN("APT",DT," APPOINTMENT DATE/TIME") – Date of appointment

RETURN("APT",DT," APPOINTMENT TYPE") – Type of appointment

RETURN("APT",DT," APPT. CANCELLED") – Cancellation flag

RETURN("APT",DT," CANCELLATION REASON") – Cancellation reason

RETURN("APT",DT," CANCELLATION REMARKS") – Cancellation remarks

RETURN("APT",DT,"CLINIC") - Clinic

RETURN("APT",DT," COLLATERAL VISIT") – Collateral visit

RETURN("APT",DT," CURRENT STATUS") – Current status of appointment

RETURN("APT",DT," DATA ENTRY CLERK") – User who made appointment

RETURN("APT",DT," PURPOSE OF VISIT") – Type of appointment

RETURN("APT",DT," STATUS") – Appointment status

DFN (Required) Patient IFN

SD (Optional) – if not set returns all patient appointments.

- If only SD is set returns that appointment if it exists.
- If SD(0) is set (0/1) returns patient appointments before/after specified date (SD).

Output

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

Error Codes Returned

None

\$\$HASPEND^SDMAPI1() – Check if patient has pending appointments

This extrinsic function verifies if a patient has pending appointments.

Format

\$\$HASPEND^SDMAPI1(.RETURN,DFN,DT)

Input Parameters

.RETURN (Required) Set to 1 if the patient has pending appointments, 0 otherwise

DFN (Required) Patient IFN

DT (Required) Start date

Output

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

Error Codes Returned

PATNFND Patient not found.

\$\$GETPEND^SDMAPI1() - Return patient pending appointments

This extrinsic function returns a list of patient pending appointments.

Format

\$\$GETPEND^SDMAPI1(.RETURN,DFN,DT)

Input Parameters

.RETURN (Required) Array passed by reference that will receive the data

RETURN(APPTDT," APPOINTMENT TYPE") – Type of appointment

RETURN(APPTDT," CLINIC") - Clinic

RETURN(APPTDT," COLLATERAL VISIT") – Collateral visit

RETURN(APPTDT," CONSULT LINK")

RETURN(APPTDT," EKG DATE/TIME")

RETURN(APPTDT," LAB DATE/TIME")

Latest version available at the [OSEHRA Journal](http://hdl.handle.net/10909/2) [<http://hdl.handle.net/10909/2>]

Distributed under [Creative Commons Attribution License](#)

RETURN(APPTDT," LENGTH OF APP'T") – Length of appointment

RETURN(APPTDT," X-RAY DATE/TIME")

DFN (Required) Patient IFN

DT (Required) Start date

Output

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

Error Codes Returned

None

2.2 Appointment management API

\$MAKE^SDMAPI2() - Make appointment

This extrinsic function creates a new appointment.

Format

\$MAKE^SDMAPI2(.RETURN,DFN,SC,SD,TYPE,STYP,LEN,SRT,OTHR,LVL)

Input Parameters

.RETURN (Required) Set to 1 if the update succeeded, 0 otherwise

RETURN(0) – Error code^Text^Level (1 for error, 2 for warning, 3 for warning)

DFN (Required) Patient IFN

SC (Required) Clinic IFN

SD (Required) Appointment date

TYPE (Required) Purpose of visit

STYP (Required) Sub-category associated with this appointment

LEN (Required) Appointment length

SRT (Required) Scheduling request type

OTHR (Optional) Any other tests ordered in association with the appointment

LVL (Optional) Forces appointment creation if it is set to 1, otherwise will throw an error if the patient has an active appointment on same time, or on same day, or a canceled appointment on same time, or there are no open slots.

Output

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

Error Codes Returned

PATDIED PATIENT HAS DIED.

Latest version available at the [OSEHRA Journal](http://hdl.handle.net/10909/2) [<http://hdl.handle.net/10909/2>]

Distributed under [Creative Commons Attribution License](#)

APTPAB	Appointment date is prior to the patient's date of birth.
APTPCLA	Appointment date is prior to the clinic's availability date.
APTCLUV	There is no available slot for this date/time.
APTEXCD	EXCEEDS MAXIMUM DAYS FOR FUTURE APPOINTMENT!!
APTSHOL	HOLIDAY??
APTPAHA	PATIENT ALREADY HAS APPOINTMENT ... THEN.
APTPHSD	PATIENT ALREADY HAS APPOINTMENT ON THE SAME DAY ...
APTPPCP	THIS TIME WAS PREVIOUSLY CANCELLED BY THE PATIENT ...
APTOVBK	OVERBOOK!
APTOVOS	THAT TIME IS NOT WITHIN SCHEDULED PERIOD!
APTOAPD	ONLY ... OVERBOOK PER DAY!!
APTCBCP	CAN'T BOOK WITHIN A CANCELLED TIME PERIOD
APTNOST	NO OPEN SLOTS THEN

\$MAKEUS^SDMAPI2() – Make unscheduled appointment

This extrinsic function creates a new unscheduled appointment.

Format

\$MAKEUS^SDMAPI2(.RETURN,DFN,SC,SD,TYP,STYP)

Input Parameters

.RETURN	(Required) Set to 1 if the update succeeded, 0 otherwise
	RETURN(0) – Error code^Text^Level (1 for error, 2 for warning, 3 for warning)
DFN	(Required) Patient IFN
SC	(Required) Clinic IFN
SD	(Required) Appointment date
TYP	(Required) Purpose of visit
STYP	(Required) Sub-category associated with this appointment

Output

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

Error Codes Returned

PATDIED	PATIENT HAS DIED.
---------	-------------------

APTCINV Clinic is scheduled to be inactivated on ...

\$\$CANCEL^SDMAPI2() - Cancel appointment

This extrinsic function cancels an existing appointment.

Format

\$\$CANCEL^SDMAPI2(.RETURN,DFN,SC,SD,TYP,RSN,RMK)

Input Parameters

.RETURN	(Required) Set to 1 if the update succeeded, 0 otherwise
	RETURN(0) – Error code^Text^Level (1 for error, 2 for warning, 3 for warning)
DFN	(Required) Patient IFN
SC	(Required) Clinic IFN
SD	(Required) Appointment date
TYP	(Required) Status (no-show, cancelled by clinic, etc.)
RSN	(Required) Cancellation reason (pointer to file 409.2)
RMK	(Optional) Remarks

Output

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

Error Codes Returned

APTCAND	Appointment already cancelled
APTCCHO	Appointment has a check out date and cannot be cancelled.
APTCRGT	Appt. ... NOT CANCELLED. Access to this clinic is restricted to only privileged users!
APTCNPE	You cannot cancel this appointment.

\$\$NOSHOW^SDMAPI2() - Mark appointment as no-show

This extrinsic function marks an existing appointment as no-show.

Format

\$\$NOSHOW^SDMAPI2(.RETURN,DFN,SC,SD,LVL)

Input Parameters

.RETURN	(Required) Set to 1 if the update succeeded, 0 otherwise
DFN	(Required) Patient IFN
SC	(Required) Clinic IFN
SD	(Required) Appointment date

LVL (Optional) Forces appointment creation if it is set to 1, otherwise will throw an error if any of the checks failed

Output

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

Error Codes Returned

None

\$\$CHECKIN^SDMAPI2() - Check in appointment

This extrinsic function checks in an existing appointment.

Format

\$\$CHECKIN^SDMAPI2(.RETURN,DFN,SD,SC)

Input Parameters

.RETURN (Required) Set to 1 if the check in succeeded, 0 otherwise

RETURN(0) – Error code^Text^Level (1 for error, 2 for warning, 3 for warning)

DFN (Required) Patient IFN

SD (Required) Appointment date

SC (Required) Clinic IFN

Output

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

Error Codes Returned

APTCIPE You cannot check in this appointment.

APTCITS It is too soon to check in this appointment.

\$\$CHECKO^SDMAPI4() - Check out appointment

This extrinsic function checks out an existing appointment.

Format

\$\$CHECKO^SDMAPI2(.RETURN,DFN,SD,SC)

Input Parameters

.RETURN (Required) Set to 1 if the check in succeeded, 0 otherwise

RETURN(0) – Error code^Text^Level (1 for error, 2 for warning, 3 for warning)

DFN (Required) Patient IFN

SD (Required) Appointment date

SC (Required) Clinic IFN

Output

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

Error Codes Returned

APTCOTS It is too soon to check out this appointment.

APTCOCN You cannot check out this appointment.

APTCOAC Appointment already checked out

\$\$DELCO^SDMAPI4() - Delete check out

This extrinsic function deletes an existing appointment check out.

Format

\$\$DELCO^SDMAPI4(.RETURN,DFN,SD)

Input Parameters

.RETURN (Required) Set to 1 if the check in succeeded, 0 otherwise

RETURN(0) – Error code^Text^Level (1 for error, 2 for warning, 3 for warning)

DFN (Required) Patient IFN

SD (Required) Appointment date

Output

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

Error Codes Returned

APTDCOD The appointment must have a check out date/time to delete.

2.3 Vocabularies API

\$\$GETAPPT^SDMAPI1() - Get Appointment Type details

This extrinsic function returns detailed information about an appointment type.

Format

\$\$GETAPPT^SDMAPI1(.RETURN,TYPE)

Input Parameters

.RETURN (Required) Array passed by reference that will receive the data.

RETURN("NAME") – Appointment type name

RETURN("NUMBER") – Appointment type IFN

RETURN("SYNONIM") - Appointment type synonym

TYPE (Required) Appointment type IEN (File 409.1)

Output

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

Latest version available at the [OSEHRA Journal \[http://hdl.handle.net/10909/2\]](http://hdl.handle.net/10909/2)

Distributed under [Creative Commons Attribution License](#)

Error Codes Returned

None

\$\$GETELIG^SDMAPI1() - Return Eligibility Code details

This extrinsic function returns detailed information about an eligibility code.

Format

\$\$GETELIG^SDMAPI1(.RETURN,ELIG)

Input Parameters

.RETURN (Required) Array passed by reference that will receive the data

RETURN("ABBREVIATION") – Eligibility Code abbreviation

RETURN("MAS ELIGIBILITY CODE")

RETURN("NAME") - Eligibility Code name

RETURN("PRINT NAME") - Eligibility Code print name

RETURN("TYPE") - Eligibility Code type

RETURN("VA CODE NUMBER") – VA Code Number

RETURN("INACTIVE") – Eligibility Code status

ELIG (Required) Eligibility code IEN

Output

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

Error Codes Returned

None

\$\$LSTAPPT^SDMAPI1() - List appointment types filtered by name

This extrinsic function returns a list of appointment types. The list can be filtered by name.

Format

\$\$LSTAPPTS^SDMAPI1(.RETURN,SEARCH,START,NUMBER)

Input Parameters

.RETURN (Required) Array passed by reference that will receive the data

RETURN(0) - # of entries found^maximum requested^any more?^flags

(consistent with LIST^DIC header node)

RETURN(IND,"ID") – Appointment type IFN

RETURN(IND,"NAME") – Appointment type name

SEARCH (Optional) Partial match restriction. Default: All entries

Latest version available at the [OSEHRA Journal](http://hdl.handle.net/10909/2) [<http://hdl.handle.net/10909/2>]

Distributed under [Creative Commons Attribution License](#)

START (Optional) The entry from which to begin the list. Default: ""

NUMBER (Optional) Number of entries to return. Default: All entries

Output

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

Error Codes Returned

None

\$\$LSTCRSNS^SDMAPI1() - List cancellation reasons filtered by name

This extrinsic function returns a list of cancellation reasons. The list can be filtered by name.

Format

\$\$LSTCRSNS^SDMAPI1(.RETURN,SEARCH,START,NUMBER)

Input Parameters

.RETURN (Required) Array passed by reference that will receive the data

RETURN(0) - # of entries found^maximum requested^any more?^flags
(consistent with LIST^DIC header node)

RETURN(IND,"ID") – Cancellation reason IEN

RETURN(IND,"NAME") – Cancellation reason name

SEARCH (Optional) Partial match restriction. Default: All entries

START (Optional) The entry from which to begin the list. Default: ""

NUMBER (Optional) Number of entries to return. Default: All entries

Output

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

Error Codes Returned

None

2.4 Electronic Wait List API

\$\$LIST^SDWLAPI1() - List EWL entries for patient

This extrinsic function returns the list of a patient EWL entries.

Format

\$\$LIST^SDWLAPI1(.RETURN,DFN,STATUS,BEGIN,END)

Input Parameters

.RETURN (Required) Array passed by reference that will receive the data

RETURN - # of entries found

RETURN(ID,"DISPTYPE")=Disposition type

RETURN(ID,"IEN")=EWL entry IEN

RETURN(10,"INSTITUTION")=Institution

RETURN(10,"ORIGDT")=Originating date

RETURN(10,"PRIORITY")=Priority

RETURN(10,"STATUS")=Status

RETURN(10,"WAITFOR")=Pointer to Team/Position/Specialty/Clinic depending on
Wait List Type

RETURN(10,"WLTYPE")=Wait List Type

DFN (Required) Patient IFN

STATUS (Optional) EWL entry status (O=Open, C=Closed, ""=Both). Default ""

BEGIN (Optional) Start date

END (Optional) End date

Output

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

Error Codes Returned

INVPARAM Invalid parameter value

PATNFND Patient not found.

\$\$DETAIL^SDWLAPI1() - Wait List entry detailed information

This extrinsic function returns detailed information for a EWL entry

Format

\$\$DETAIL^SDWLAPI1(.RETURN,IEN)

Input Parameters

.RETURN (Required) Array passed by reference that will receive the data

RETURN("APPTCLERK")=Who made the appointment

RETURN("APPTCLIN")=Appointment clinic

RETURN("APPTCREDSC")=Appointment credit stop code

RETURN("APPTDATE")=Date appointment made

RETURN("APPTINST")=Appointment institution

RETURN("APPTSC")=Appointment stop code

Latest version available at the [OSEHRA Journal](http://hdl.handle.net/10909/2) [<http://hdl.handle.net/10909/2>]

Distributed under [Creative Commons Attribution License](#)

RETURN("APPTSCHED")=Scheduled date of appointment
 RETURN("APPTSN")=Appointment station number
 RETURN("APPTSTATUS")=Appointment status
 RETURN("CHDCLINP")=Changed clinic parent pointer
 RETURN("CMNTS")=Comment
 RETURN("DISPBY")=Dispositioned by
 RETURN("DISPDT")=Date dispositioned
 RETURN("DISPTYPE")=Disposition
 RETURN("DNRCMT")=Do not remove comment
 RETURN("DNRDT")=Do not remove date
 RETURN("DNRRSN")=Do not remove reason
 RETURN("DNRUSR")=User entering no remove
 RETURN("DSRDDT")=Desired date of appointment
 RETURN("ENTEREDBY")=Originating user
 RETURN("INSTITUTION")=Institution
 RETURN("ORIGDT")=Originating date
 RETURN("PATIENT")=Patient
 RETURN("PRIORITY")=Priority
 RETURN("PROVIDER")=Provider
 RETURN("REOPENCMT")=Reopen comment
 RETURN("REOPENRSN")=Reopen reason
 RETURN("REQBY")=Request by
 RETURN("SCPRIORITY")=Service connected priority
 RETURN("STATUS")=Status
 RETURN("WAITFOR")= Pointer to Team/Position/Specialty/Clinic depending on
 Wait List Type
 RETURN("WLTYPE")= Wait List Type

Output

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

Error Codes Returned

None

\$\$NEW^SDWLAPI1Q – Add new EWL entry

This extrinsic function creates a new EWL entry

Format

`$$NEW^SDWLAPI1(.RETURN,.SDWLD)`

Input Parameters

<code>.RETURN</code>	(Required) Will return the new IEN if the call succeeded, 0 otherwise
<code>.SDWLD</code>	(Required) Array containing the data to be saved
	<code>SDWLD("WLTYPE")=Wait List Type</code>
	<code>SDWLD("PATIENT")=Patient</code>
	<code>SDWLD("INSTITUTION")=Institution</code>
	<code>SDWLD("WAITFOR")= Pointer to Team/Position/Specialty/Clinic depending on Wait List Type</code>
	<code>SDWLD("PRIORITY")=Priority</code>
	<code>SDWLD("REQBY")=Request by</code>
	<code>SDWLD("PROVIDER")=Provider</code>
	<code>SDWLD("SCPRCNT")=Service connected percentage</code>
	<code>SDWLD("SCPRIORITY")=Service connected priority</code>
	<code>SDWLD("DSRDDT")=Desired date of appointment</code>
	<code>SDWLD("CMNTS")=Comments</code>
	<code>SDWLD("ENRSTAT")=EWL enrollee status</code>
	<code>SDWLD("ENRDU")=EWL enrollee date used</code>
	<code>SDWLD("ENRDF")=EWL enrollee database file</code>
	<code>SDWLD("TICKLER")=Scheduling reminder flag</code>
	<code>SDWLD("CHDCLINP")=Changed clinic parent pointer</code>

Output

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

Error Codes Returned

None

\$\$UPDATE^SDWLAPI1() – Update EWL entry

This extrinsic function returns a list of cancellation reasons. The list can be filtered by name.

Format

\$\$UPDATE^SDWLAPI1(.RETURN,SDWLIEN,.SDWLD)

Input Parameters

.RETURN	(Required) Set to 1 if the the update succeeded, 0 otherwise
SDWLIEN	(Required) IEN of the EWL entry to be updated
.SDWLD	(Required) Array containing the data to be saved
	SDWLD("WLTYPE")=Wait List Type
	SDWLD("INSTITUTION")=Institution
	SDWLD("WAITFOR")= Pointer to Team/Position/Specialty/Clinic depending on Wait List Type
	SDWLD("PRIORITY")=Priority
	SDWLD("REQBY")=Request by
	SDWLD("PROVIDER")=Provider
	SDWLD("SCPRCNT")=Service connected percentage
	SDWLD("SCPRIORITY")=Service connected priority
	SDWLD("DSRDDT")=Desired date of appointment
	SDWLD("CMNTS")=Comments
	SDWLD("ENRSTAT")=EWL enrollee status
	SDWLD("REJECTED")=Rejection flag
	SDWLD("TICKLER")=Scheduling reminder flag
	SDWLD("INTRATF")=Intra-transfer flag
	SDWLD("MULTITEAM")=Mutli team flag

Output

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

Error Codes Returned

None

\$\$DISP^SDWLAPI1() – Disposition entry

This extrinsic function update the disposition data for an EWL entry.

Format

\$\$DISP^SDWLAPI1(.RETURN,SDWLDFN,SDWLIEN,SDWLDISP,.SDWLAPPT)

Input Parameters

.RETURN (Required) Set to 1 if the update succeeded, 0 otherwise

SDWLDFN (Required) Patient IFN

SDWLIEN (Required) EWL entry IEN

SDWLDISP (Required) Disposition type

.SDWLAPPT (Optional) Array containing appointment data (saved if disposition type is “SA”)

Output

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

Error Codes Returned

None

\$\$DELETE^SDWLAPI1() – Delete EWL entry

This extrinsic function deletes an existing EWL entry

Format

\$\$DELETE^SDWLAPI1(.RETURN,SDWLIEN)

Input Parameters

.RETURN (Required) Set to 1 if the delete succeeded, 0 otherwise

SDWLIEN (Required) EWL entry IEN

Output

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

Error Codes Returned

None

\$\$HASENTRY^SDWLAPI1() – Patient is on EW List?

This extrinsic function checks if patient has entries on the Electronic Wait List

Format

\$\$HASENTRY^SDWLAPI1(.RETURN,DFN)

Input Parameters

.RETURN (Required) Set to 1 if the patient has EWL entries on file, 0 otherwise

DFN (Required) Patient IFN.

Output

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

Error Codes Returned

None

3. How to use the code

API

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

Assumed Variables

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

Format and conventions of the calls

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

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

ErrorId^Message

Example

INVPARAM^Invalid parameter value – SDWLDFN

4. Installation

Along with this paper a KID Host File is provided. The steps required to install the distribution are outlined below:

1. From the *Systems Manager Menu* select *Programmer Options...*
2. Select *Kernel Installation & Distribution System*
3. Select *Installation*
4. Select *Load a Distribution*
5. Enter the host file path, for example C:\SD_5.3_260003.KID and load the distribution.
6. Use the *Install Package(s)* option and select SD*5.3*260003
7. When prompted *Want KIDS to Rebuild Menu Trees Upon Completion of Install?* Respond NO
8. When prompted *Want KIDS to INHIBIT LOGONs during the install?* Respond NO
9. When prompted *Want to DISABLE Scheduled Options, Menu Options, and Protocols?* Respond NO

5. Conclusions

This paper presented a set of APIs developed as part of the Open Source HER refactoring effort. They are meant to be used by both scroll & roll interface and RPC tags and to be accessible by other packages and applications. Also they should remove direct global access, uncouple the business logic from the user interface elements and minimize code duplication.