



# **RFP VistA Gateway “Primer” for DataBridge**

**Version 1.1**

**Overview:** The DSS VistA Gateway is a utility that allows developers to integrate their host applications into the Department of Veterans' Affairs (VA) legacy database: VistA. The DSS VistAGateway is based on "COM". The DSS VistAGateway possesses many properties and methods that allow the DataBridge developers to retrieve data out of VistA and input data into the VistA. The DataBridge project allows a CIS vendor to manage the below mentioned VistA data elements in an on-demand, real time basis using the VistA Gateway COM based methods and/or properties to supplement HL7 data feeds.

- User Authentication (Security)
- Allergies
- Coding Status (Patient Care Encounter)
- Lab Orders
- Patient Selection
- Patient Registration
- Patient Record Flags
- Pharmacy Orders
- Problem List
- TIU (Progress Notes)
- VistA Imaging
- Radiology Orders
- Vitals Management
- Miscellaneous Supporting APIs

The intent of this manual is not to replace the existing VistA Gateway Programming API guide; this manual is meant to act as a primer to accompany the VistA Gateway Programming API guide for CIS DataBridge developers. Before reading this manual, please verify you have the following components in place: the VistA Gateway dll, access to the VistA Gateway Programming API guide, the VistA Gateway tester and source code, and connection to a VistA database supplied to you by DSS running in a Cache environment. If you do not have one of the above components in place, stop! Call Document Storage Systems at 561-227-0207 and ask for the DataBridge Project Manager. It's assumed your company has a DataBridge contract in place with Document Storage Systems.

The remaining part of this document will cross reference the APIs detailed within VistA Gateway Programming API guide. The purpose of this manual is to assist you (the DataBridge API developer) to understand (1) the relevant API methods, (2) the object properties, (3) the interdependencies of the APIs in question, and (4) the purpose and best practice implementation for the API(s) in question. This document is not intended as a replacement for the VistA Gateway Programming API guide.

### **VistA Gateway User Authentication Implementation**

The DataBridge applications must connect to VistA, maintain that persistent connection for further data retrieval/processing, and authenticate the user in question as a valid and legal VistA user. The VistA Gateway allows the CIS software to access VistA in the same manner as CPRS, BCMA, and other RPC broker based applications. Once an authenticated token is returned to the integrated

VistA Gateway system, the software can assess object properties to store within the offline database in question and use VistA "pointer" information for further VistA package processing later on the software session. There are several "touchpoints" within VistA that require VistA user review, validation and interaction. It's because of these rules based processes the VistA Gateway COM object is recommended to be implemented within the host application software. The outcome is a persistent connection to the VistA database so on-demand data can be retrieved and data prompts to induce the user into determining certain data elements to be sent back into VistA; abiding by inherit VistA module rules in a real time interactive session.

### **VistA Gateway Allergy Implementation**

After the CIS system connects to VistA through the VistA Gateway, a variety of method calls are available to the VistA Gateway developers to process and display clinical information to the user. Two such method calls give the VistA Gateway developer the ability to retrieve a list of patient allergies from VistA as well as retrieving the details about a given allergy.

### **VistA Gateway Coding Status Implementation**

The primary coding interface(s) is HL7 based. The appointments and checked out visits will be sent to the CIS through a SIU HL7 outbound from VistA into the CIS; this will allow the CIS to send the associated coding event back into VistA. There will be an in-place VistA based HL7 inbound listener that will accept and acknowledge the CIS coding event HL7 messages. Two HL7 VistA based "application" components are in place to direct coding messages from the CIS to specified 'IN' protocols to a processing routine (EN^DSIHIN) that will digest the coding messages. Once the routine has parsed the message into local arrays, the data enclosed in the messages will be stored in PCE using the DATA2PCE^PXAPI M based API. For more details on the HL7 DataBridge coding interface, please be referred to the DataBridge HL7 Specification; particularly the SIU and DFT sections.

As an alternative to the HL7 coding event DataBridge interfaces, the VistA Gateway has encapsulated much of the visit/appointment retrieval events as well as coding update events through a variety of COM APIs. These APIs will allow the CIS to retrieve a patient based appointment/visit listing for a particular date range and then take coding action on a given visit.

### **VistA Gateway Lab Ordering Information Implementation**

The primary lab data retrieval interface(s) is HL7 based. The lab data will be sent to the CIS through an ORM HL7 outbound from VistA into the CIS. For more details on the HL7 DataBridge lab interface, please be referred to the DataBridge HL7 Specification; particularly the ORM section.

As an alternative to the HL7 lab event(s) DataBridge interfaces, the VistA Gateway has encapsulated much of the lab results retrieval events through a variety of COM APIs. After the CIS system connects to VistA through the VistA Gateway, a variety of method calls are available to the VistA Gateway developers to process and display clinical information to the user. Two such method calls give the VistA Gateway developer the ability to retrieve a list of patient lab orders

from VistA as well as retrieving the details about a given lab order. These APIs will allow the CIS to retrieve patient based lab results for a particular date range.

### **VistA Gateway Patient Selection Implementation**

After the CIS system connects to VistA through the VistA Gateway, a patient context change can occur through a variety of different means. There are "silent" APIs to assist in inherit patient context changes (CCOW, native SQL patient search dialog) as well as interactive APIs that allow the VistA Gateway developer to invoke a VistA Patient Search/Select interactive dialog box. The primary methodology to retrieve patient related data from VistA will be through native DataBridge ADT HL7 feeds. These HL7 feeds will seed the initial patient records within the CIS SQL database. There several existing VistA Gateway methods in place to search for specific VistA patients in real time as the session demands. The VistA Gateway can allow the CIS system to verify a CCOW message and retrieve necessary patient related information to seed a new record on the fly, if the existing CIS database does not have that patient on file via historical HL7 data.

### **VistA Gateway Patient Registration Implementation**

Please refer to the above stated "PAT\_SetNewPatient" and "PAT\_GetDemographics" API within the patient selection section of this document. After the above patient search APIs are invoked the following object property values will be available to the CIS to introduce into their offline patient table (all string values):

**PatientIFN:** A number that reflects the patient's internal file number.  
**PatientName:** 3-30 upper-case characters in length, Last,First Middle.  
**SensitivePatient:** 0=Not a sensitive patient, 1=Sensitive patient.  
**PatientGender:** 'M' for male, 'F' for female.  
**PatientDOB:** MM/DD/YYYY, could be imprecise i.e. 00/DD/YYYY  
**PatientAge:** TCalculation based on the above filed DOB i.e. 65.  
**PatientMaritalStatus:** See set of codes below.  
**PatientRace:** See set of codes below.  
**PatientReligion:** See set of codes below.  
**PatientSSN:** 9 (xxxxxxxxx) characters in length, or 10 (xxxxxxxxxP) in length for a pseudo-SSN.  
**PatientAddress1:** 3-35 characters in length.  
**PatientAddress2:** 3-30 characters in length.  
**PatientAddress3:** 3-30 characters in length.  
**PatientCity:** 3-15 characters in length.  
**PatientCounty:** 3-15 characters in length.  
**PatientState:** 2 upper-case characters in length.  
**PatientZip:** 5 characters in length.  
**PatientHomePhone:** 4-20 characters in length.  
**PatientWorkPhone:** 4-20 characters in length.  
**PatientEmploymentStatus:** See set of codes below.  
**PatientDeathDate:** MM/DD/YYYY, could be imprecise i.e. 00/DD/YYYY.

### **VistA Gateway Patient Record Flags Implementation**

After the CIS system connects to VistA through the VistA Gateway, a variety of method calls are available to the VistA Gateway developers to process and display clinical information to the user. Two such method calls give the VistA Gateway developer the ability to retrieve a list of patient record flags from VistA as well as retrieving the details about a given patient record flag. Patient record flags within VistA alert the user to special circumstances that involve a particular patient; such "flags" could be "behavioral" as one example.

### **VistA Gateway Pharmacy Ordering Information Implementation**

The primary medication data retrieval interface(s) is HL7 based. The medication data will be sent to the CIS through an OBX HL7 outbound from VistA into the CIS. For more details on the HL7 DataBridge medication interface, please be referred to the DataBridge HL7 Specification; particularly the OBX section.

As an alternative to the HL7 pharmacy event(s) DataBridge interfaces, the VistA Gateway has encapsulated much of the medication results retrieval events through a variety of COM APIs. After the CIS system connects to VistA through the VistA Gateway, a variety of method calls are available to the VistA Gateway developers to process and display clinical information to the user. Two such method calls give the VistA Gateway developer the ability to retrieve a list of medication orders from VistA as well as retrieving the details about a given pharmacy order. These APIs will allow the CIS to retrieve patient based pharmacy ACTIVE orders and their given details.

### **VistA Gateway Problem List Implementation**

After the CIS system connects to VistA through the VistA Gateway, a variety of method calls are available to the VistA Gateway developers to process and display clinical information to the user. Two such method calls give the VistA Gateway developer the ability to retrieve a list of patient problems from VistA as well as retrieving the details about a given problem list entry.

### **VistA Gateway TIU Implementation**

TIU is a module within the VistA database that would commonly be associated to other EMRs "charting notes" or "progress notes" implementation. The TIU notes package within VistA is very much "rules" oriented that at times requires immediate user feedback to process information correctly back into VistA. This package requires the user to electronically sign the note in question and verify that the electronic signature is correct at the time of filing. To create a TIU note, various supporting VistA data elements associated with the TIU note must be identified by either the system or the user. The Patient, Title, Location, Visit/Appointment, Author/User, and Note Text are required data element for a TIU note. This means, the system must collect certain pieces of information before the main TIU note creation API is called via the VistA Gateway. Some of the ancillary information is collected within other areas of the application, such as user/author data, patient data, and patient specific appointment/visit data. Past implementation of TIU note creation by other VistA Gateway developers stored the TIU title, and clinic/ward information as a default within the vendor SQL database; this information was seeded upon initialization of the system.

COM object API implementation to create TIU notes is the primary technology that will be use by the DataBridge partners.

### **VistA Gateway VistA Imaging Implementation**

At times, the CIS will have the need to send CIS produced images back into VistA. The host VistA system that stores images is known as "VistA Imaging". The VistA Imaging system provides a finite number of APIs to the VistA Gateway for imaging processing. The CIS must associate a TIU note to each image it expects to send back to the VistA Imaging system. The outcome of this API set is an image association to a TIU note within the CPRS system; this association is graphically depicted with a picture icon to the left of each TIU note reference with the CPRS TIU note listing.

VistA Imaging is a package within the VistA database that allows images produced by a CIS to be stored as part of the patient record. The capable image types range from picture files of various formats to PDF images. Images in VistA are frequently associated with a TIU progress note to provide supporting documentation and, therefore, a more complete record.

To upload a TIU associated image to VistA Imaging, various supporting elements (parameters) must first be identified by either the system or the user. The patient, image file name and network location, image date, unique imaging identifier, and TIU note for association are all required for VistA Imaging to store the image and associate it with a TIU note.

The patient and TIU note are readily available from the creation of a TIU note described in the "TIU Implementation" section of this document. The file name and network location is simply a fully qualified server name and path to where the image file is located. This file an location must be accessible to VistA Imaging across the hospital network as VistA Imaging will be making a copy of the file to a more permanent storage location. The unique imaging identifier is described in detail in the VistA Gateway Programming API guide, but is simply a unique key for VistA Imaging to track the status of the image upload and provide feedback on that status.

Another aspect of implementing the use of VistA Imaging is in getting back the status from VistA as to the success of the image upload. The VA closely regulates the use of VistA Imaging at all hospital sites and will require the CIS to go through a certification process in order to be approved to use the VistA Imaging package. One of the criteria the CIS will be judged on is in providing feedback as to the successful upload of an image. The upload of images to VistA imaging is not instantaneous. It is a queued batch process where the time of completion depends on the current workload of the VistA Imaging processor. The VistA Gateway provides additional API's to monitor the success of any upload request made to VistA Imaging. That status is tracked via the unique imaging identifier that was generated as part of the image upload request.

There is a DICOM gateway available within the VistA Imaging system that CIS provider can subscribe to. This DICOM interface is outside of the scope of the VistA Gateway implementation. Those CIS vendor who wish to send DICOM images directly to VistA Imaging should contact the DSS DataBridge Product Line Manager for further specifications.

### **VistA Gateway Radiology Ordering Information Implementation**

The primary radiology data retrieval interface(s) is HL7 based. The radiology data will be sent to the CIS through an ORM HL7 outbound from VistA into the CIS. For more details on the HL7 DataBridge radiology interface, please be referred to the DataBridge HL7 Specification; particularly the ORM section.

As an alternative to the HL7 radiology event(s) DataBridge interfaces, the VistA Gateway has encapsulated much of the radiology results retrieval events through a variety of COM APIs. After the CIS system connects to VistA through the VistA Gateway, a variety of method calls are available to the VistA Gateway developers to process and display clinical information to the user. Two such method calls give the VistA Gateway developer the ability to retrieve a list of radiology orders from VistA as well as retrieving the details about a given radiology order. These APIs will allow the CIS to retrieve patient based radiology ACTIVE orders and their given details.

### **VistA Gateway Vitals Implementation**

The primary vitals data retrieval interface(s) is HL7 based. The radiology data will be sent to the CIS through an OBX or OBR HL7 outbound from VistA into the CIS. For more details on the HL7 DataBridge radiology interface, please be referred to the DataBridge HL7 Specification; particularly the OBX or OBR section.

As an alternative to the HL7 vitals intake event(s) DataBridge interfaces, the VistA Gateway has encapsulated much of the vitals results retrieval events through a single COM API. After the CIS system connects to VistA through the VistA Gateway, a method call is available to the VistA Gateway developers to process and display clinical information to the user. One such method call gives the VistA Gateway developer the ability to retrieve a list of the most current vitals that are on file for the given VistA patient in question.

### **VistA Gateway Miscellaneous Supporting API Usage**

There are miscellaneous "helper" APIs to assist in the VistA Gateway developer create the necessary API parameter values. The "MISC\_ConvertDisplayDateToVistADate" is an extremely valuable API; this API converts a window formatted date/time (mm/dd/yyyy hh:mm:ss) to a VistA readable format. The "USR\_GetUserInformation" and "PAT\_GetPrimaryProvider" APIs allow the caller to retrieve VistA user details and the primary provider that is known to be assigned to the patient in question. The "MISC\_GetTableEntryData" API is another extremely useful method. This API allow a VistA Gateway developer to look up any field value within any VistA file (table) if the entry number is known i.e. patient race for patient "441" is ASIAN. At times, the CIS will need to "dump" legacy VistA data into it's data repository to initialize it's database tables to be in sync with VistA. There are a variety of PT\_\* API methods that allow the CIS vendor to acquire VistA table information to make their CIS database "VistA relevant".

At times, the CIS system date/time will not be in sync with the legacy system: VistA. The "MISC\_DSICDateConvert" API will allow the CIS caller to aquire the VistA system NOW date/time.

**Revision History**

<b>Version</b>	<b>Date</b>	<b>Author</b>	<b>Change Description</b>
1.0.1	03/26/2008	J Flejter	Initial Definition
1.0.2	03/28/2008	J Flejter	Integrated V Hornback VistA Imaging Information
1.0.3	03/28/2008	J Flejter	Identify Remote Procedure Calls within Appendix D
1.1.1	06/11/2008	J Flejter	Miscellaneous Date/Time NOW API Introduced
1.1.2	03/23/2009	E OBrien	Updated for RFP