



VISTA DATA EXTRACTION FRAMEWORK (VDEF)

TECHNICAL MANUAL

VERSION 1.0

PATCH VDEF*1.0*3

DOCUMENT V. 1.0, DECEMBER 2004

REVISED, V. 1.1, January 2006

Department of Veterans Affairs (VA)
VHA OI Health Systems Design & Development (HSD&D)
Messaging & Interface Services (M&IS)

DOCUMENT REVISION HISTORY

Date	Version No.	Change	Author
December 2004	1.0	Initial publication release	Randy Stone
January 2006	1.1	Documents false VDEF alerts and restarting VDEF queue processor. Reflects changes made by patch VDEF*1.0*3.	Brian Lynch

TABLE OF CONTENTS

1. Introduction.....	1
2. Implementation and Maintenance	2
2.1. VDEF Initialization at System Startup	2
2.2. VDEF Request Queue Population	2
2.3. VDEF Request Fulfillment by the Request Processor.....	3
2.4. VDEF Monitoring and Other Site-Wide Parameters.....	5
2.4.1. Custodial Package Deactivation.....	5
2.4.2. Request Processor Scheduling.....	6
2.4.3. VDEF Request Queue Processor Monitor.....	6
2.5. VDEF Status	7
2.6. Deleting Request Queue Entries	7
3. Files	8
4. Routines	9
5. Exported Options.....	10
6. Archiving and Purging.....	11
7. Entry Points / Callable Routines / APIs.....	12
7.1. Entry Point \$\$QUEUE^VDEFQM.....	12
7.1.1. Calling Syntax	12
7.1.2. Input Parameters.....	12
7.1.3. Function Return Value	13
7.2. Entry Point SETDLMS^VDEFEL.....	13
7.2.1. Calling Syntax	13
7.2.2. Input Parameters.....	13
7.2.3. Output Value	13
7.3. Entry Point \$\$XCN200^VDEFEL().....	13
7.3.1. Calling Syntax	13
7.3.2. Prerequisite.....	14
7.3.3. Input Parameters.....	14
7.3.4. Function Return Value	14
7.4. Entry Point \$\$STS^VDEFEL().....	14
7.4.1. Calling Syntax	14
7.4.2. Input Parameter	14
7.4.3. Function Return Value	15
7.5. Entry Point POSTKID^VDEFVU().....	15
7.5.1. Calling syntax.....	15
7.5.2. Input Variables	15
7.5.3. Output.....	16
7.6. Entry Point ERR^VDEFREQ()	16
7.6.1. Calling syntax.....	16
7.6.2. Input Parameter	16
7.6.3. Output.....	16
8. External Interfaces	17

9. External Relations	18
9.1. Integration Agreements.....	18
9.2. Package Dependencies.....	19
10. Internal Relations	20
11. Package-Wide Variables	21
12. Software Products Security.....	22
12.1. Mail Groups and Alerts.....	22
12.2. Remote Systems.....	22
12.2.1. Acknowledgements	22
12.3. Contingency Planning.....	22
13. Appendix A: VDEF Shut Down Procedures	23
14. Appendix B: VDEF Initial Setup Procedures	24
15. Appendix C: VDEF Files.....	25
16. Appendix D: VDEF Alert Messages.....	26

1. INTRODUCTION

VISTA Data Extraction Framework (VDEF) is a **VISTA** package that uses hard-coded MUMPS (M) routines to create and deliver Health Level 7 (HL7) messages. The VDEF package supports queuing requests for messages, control of the timing of message creation, monitoring of the request queue, and recording of errors encountered during message creation. The hard-coded programs are M programs belonging to an application's namespace. Messages are delivered using the **VISTA** HL7 package.

2. IMPLEMENTATION AND MAINTENANCE

The VDEF package is installed as a regular Kernel Installation and Distribution System (KIDS) build. Once installed, Information Resource Management (IRM) staff will need to set up and activate the appropriate HL7 link parameters and add the VDEF STARTUP OPTION to the list of Scheduled Startup Options. (See Section 4.1 of the VDEF Installation and User Configuration Guide for details.)

Once the configuration process is complete and the decision has been made to turn on VDEF messaging, IRM staff will need to go into the VDEF Configuration and Status menu and select a number of menu options. (See Section 4 of the VDEF Installation and User Configuration Guide for details.)

Once these tasks are performed, VDEF messaging will be activated. VDEF messaging consists of the activities outlined in the sections below.

Note: Per VHA Directive 2004-038, this package should not be modified after installation.

2.1. VDEF Initialization at System Startup

VDEF STARTUP OPTION is a queueable option that will be added to the list of Scheduled Startup Options during VDEF installation. Refer to the VDEF Installation Guide for details on how to set it up in TaskMan.

When *VISTA* is brought up, TaskMan will run the VDEF STARTUP OPTION scheduled option. When the option runs, it submits the following TaskMan processes for subsequent execution:

VDEF Checked Out Monitor

VDEF Request Processor for <name of Request queue> (one per Request Queue)

VDEF Request Processor Monitor

IRM staff can prevent this option from starting any of the VDEF TaskMan processes except VDEF Monitor by changing the value of the SUSPENDED FLAG for the Request Queues that should not be running (see above). VDEF Monitor cannot be prevented from running at *VISTA* startup time short of taking VDEF STARTUP OPTION off the list of queueable options and then deleting the task that is scheduled to run VDEF Monitor with TaskMan. This should only occur if VDEF is being completely uninstalled at the site.

2.2. VDEF Request Queue Population

When certain events occur within the *VISTA* application, the *VISTA* software issues a call to the VDEF Application Programmer Interface (API). (Refer to the API specification for \$\$QUEUE^VDEFQM below.)

Each VDEF Request contains the VDEF Event Type that contains the HL7 Message Type and Event Type, and two Name-Value Pairs, which tell VDEF what data to extract. The recognized VDEF Event Types are stored in the VDEF EVENT DESCRIPTION file (#577).

It should be noted that custodial package-specific KIDS builds deploy individual VDEF Event Type entries in the VDEF EVENT DESCRIPTION file (#577). For example, a KIDS build will deploy the Event Types that cover files like the GMRV VITAL MEASUREMENT. If a particular VDEF Event Type is not installed at a given site, then the VDEF API will reject any VDEF Requests for this Event Type.

When \$\$QUEUE^VDEFQM is called, the VDEF API retrieves the default Requestor from the VDEF REQUESTOR file (#579.1). Once the Requestor is established, the VDEF API checks the following:

Whether this Requestor is active, and

Whether the custodial package for this VDEF Event Type has VDEF messaging turned on.

If these conditions are satisfied, the VDEF API creates an entry in the Request Queue associated with the given requestor.

WARNING	IRM staff can disable the creation of VDEF Requests for a given interface by deactivating a Requestor (see above). Be advised that preventing the generation of VDEF messages for one or more of the custodial packages for which it is enabled will result in a loss of synchronization between VISTA and the remote systems that receive these custodial package messages. Once synchronization is lost for a custodial package, there is no easy way of getting the remote system(s) back in synch with VISTA . Only turn VDEF messaging off for a custodial package if the custodial package is also disabled for all remote systems receiving this custodial package's data. Deactivation at the requestor level should only occur when that requestor system (e.g., HDR) is being completely disabled.
----------------	--

2.3. VDEF Request Fulfillment by the Request Processor

Upon system startup, through the VDEF STARTUP OPTION, TaskMan initiates one Request Processor job (EN^VDEFREQ) for every Request Queue. Once started, each Request Processor verifies that its assigned Request Queue has not been suspended since the Processor last ran. If it has, then the Request Processor does not start.

The Request Processor then checks if the current date/time falls inside the timeframe defined for this RUN TIME subfile (#579.32) when it should not run. If it does fall within this timeframe, the request Processor waits to run at the end of that timeframe and starts processing the Request queue.

After processing any queued up requests, the Request Processor waits for the number of seconds defined in the WAKEUP PERIOD to elapse. Then it processes new all entries in the Request Queue that are assigned to it that have accumulated while it was in the wait loop. It repeats this cycle continuously.

IRM staff can use the SUSPEND/RUN REQUEST QUEUE menu option to stop the Request Processor for an individual Request Queue. Request Processor tasks check internal TaskMan flags to see if they have been asked to stop after completing every VDEF Request. IRM staff can also use the REQUEST QUEUE PARAMETERS options on the VDEF Configuration and Status menu to change the WAKEUP PERIOD value for the Queue, or define timeframes when individual Request Processors will not run (see the description of the VDEF Request Processor Schedule menu option below).

If the Request Processor for a Request Queue is suspended, VDEF requests continue to accumulate in it until it is turned on again. The main reason for suspending a Request Processor is likely to be system performance deterioration due to VDEF activities running during regular business hours.

WARNING	There is no loss of functionality associated with suspending a Request Processor for a certain period of time and then turning it back on. The only impact on the system is that it may cause the receiving remote systems to fall further behind VISTA while the processor is down and while it is catching up. Although the remote systems will catch up, suspending a queue could, depending on the nature of the data involved, result in a patient safety issue.
----------------	--

When a Request Processor is running, it examines its assigned Request Queue for Queued up Requests. If any are found, the Request Processor takes the first Queued Up request and changes the status to Checked Out. The Request Processor then calls the VDEF message-building M routine for the Internal Entry Number (IEN) specified in the Name/Value Pair that was sent the trigger application's VDEF API call. The IEN is the IEN of the primary global that the data associated with the trigger event will be found.

Once each HL7 message is built, the Request Processor retrieves the **VISTA** HL7 protocol for this VDEF Event Type, identifies its associated subscriber protocols, calls **VISTA** HL7, and passes the body of the message to it for delivery to the designated remote systems. If the **VISTA** HL7 package encounters a problem and returns an error (e.g., if the Dynamic Addressing array passed to VDEF was invalid), then the Request Processor marks the Request as "Errored Out" and stores the error text returned by **VISTA** HL7 in the Request's Error field.

Once the HL7 message for a given request has been generated, the Request Processor changes the status of the Request from Checked Out to Processed and moves on to the next Queued up request.

Because maintenance requests are eventually routed to their destination remote systems via the **VISTA** HL7 package, IRM staff may need to monitor and possibly adjust VDEF-specific **VISTA** HL7 filers and links as per their standard **VISTA** HL7 monitoring procedures.

IRM staff can modify the following parameters for any of the VDEF Request Queues using menu option VDEF REQUEST QUEUE PARAMETERS:

ARCHIVAL PARAMETER: Minimum amount of time (in days, hours, minutes, and seconds) that a processed VDEF Request remains in this Request Queue before it is purged by the VDEF Monitor (see below).

REQUEST QUEUE WAKEUP PERIOD: Period of time (in days, hours, minutes, and seconds) that TaskMan uses to calculate how long to wait before it resumes processing the Request queue

VDEF Request Processors use two more configuration settings that are not Request Queue-specific. These settings can be viewed and changed via option VDEF Site-Wide Parameters on the VDEF Configuration and Status menu and are as follows:

VDEF SYSTEM: <system's domain name>//

VDEF MONITOR DELAY: 5M//

2.4. VDEF Monitoring and Other Site-Wide Parameters

The VDEF Monitor process MONITOR^VDEFCONT is submitted by the VDEF STARTUP OPTION at *VISTA* startup (see above). It is not associated with any specific queue, so there is only one VDEF Monitor process per *VISTA* system. The VDEF Monitor resubmits itself every so many seconds based on the value of the VDEF MONITOR DELAY parameter that can be changed in the SITE-WIDE PARAMETERS option.

When the VDEF Monitor runs, it checks all Request Queues for Checked Out entries. If it finds any, it gets the Check Out date & time and the current system date and time for each Checked Out entry and compares the time difference against the. If it is greater than the allowed time, the VDEF Monitor sends an alert to the VDEF NATIONAL ALERTS group specified in the VDEF ALERTS mail group and requeues the request.

The VDEF Monitor also looks at all processed entries in each Request Queue, starting with the oldest one. The entries that have been in the *Processed* state longer than the value of the ARCHIVAL PARAMETER for that Request Queue are deleted.

The other site-wide parameters that IRM staff can modify are as follows:

2.4.1. Custodial Package Deactivation

When a custodial package-specific VDEF KIDS build is installed at a site, a custodial package record is added to the VDEF CUSTODIAL PACKAGE file (#579.6) **only if it is a new entry** and a VDEF Event record is created in the VDEF EVENT DESCRIPTION file (#577). They are both created with status defaulted to INACTIVE.

IRM staff can activate and deactivate VDEF messaging for individual custodial packages using menu option VDEF CUSTODIAL PACKAGE. When prompted for the custodial package to (de)activate, IRM staff can either enter “??” to list all custodial packages that currently have VDEF messaging installed on site or enter an individual custodial package to (de)activate VDEF messaging. When trying to deactivate (but not when trying to activate) VDEF messaging for a custodial package, a warning is generated because *VISTA* and the receiving system(s) become out of sync and there is no easy way to “re-sync” them.

The only custodial package that IRM staff cannot deactivate VDEF messaging for is Registration. If IRM staff select and attempt to deactivate it, an error message will be displayed.

2.4.2. Request Processor Scheduling

A Request Queue Processor runs at the interval specified in the REQUEST QUEUE WAKEUP PERIOD and processes the “Queued up” Requests in the Request Queue for which it is responsible. During peak operating hours when the system is under heavy load, IRM staff may decide to suspend the VDEF queue processing temporarily. IRM staff can prevent a Request Processor from running by manually suspending it using the VDEF SUSPEND/RUN REQUEST QUEUE option. Although useful as an ad hoc tool, it is a manual process and IRM staff will need to remember to un-suspend the Processor when the load becomes more manageable.

An alternative to manual suspension and un-suspension of Request Processors is the VDEF REQUEST PROCESSOR SCHEDULE menu option. When IRM staff access it and identify the Request Queue whose behavior needs to be modified, IRM staff can define any number of scheduling rules for the Request Processor that handles that Request Queue.

For every rule, IRM staff must enter the following information:

ENTRY: A unique alphanumeric identifier for this scheduling rule.

DAY OF THE WEEK: The day of the week to which this scheduling rule applies to (e.g., Sunday or Wednesday).

FORCED STATUS: Running or Suspended

FROM TIME: The time (in military time format) that the Request Processor for this Request Queue will be in the FORCED STATUS.

TO TIME: The time (in military time format) when the Request Processor for this Request Queue will resume normal execution.

Once this information is entered, the Request Processor checks all the rules that apply to the current day of the week. It will not run if the current time falls within any of the “time windows” defined for it.

2.4.3. VDEF Request Queue Processor Monitor

The VDEF Request Queue Processor Monitor MONITOR^VDEFCONT is submitted by the VDEF STARTUP OPTION at VISTA startup. It is not associated with any specific queue, so there will be only one of these processes. The VDEF Request Queue Process Monitor checks several things in the VDEF system and then goes into a 10-minute wait loop. It then repeats the checks. It does this continuously.

The monitor first checks to see if any Request Queue Processors are in the suspended state or have stopped running. It does this for each Request Queue that is defined in VDEF. If either of the above checks is true, the monitor will generate an alert. If a Request Queue Processor has stopped, it will be restarted by the monitor by creating a new TaskMan task for EN^VDEFREQ.

Then it checks for any requests that are hung in the Checked Out status and requeues the requests to be processed again. It also does the same check for requests that are in the Errored Out status.

Then it goes into a 10 minute wait loop.

2.5. VDEF Status

This option can be used to view the current status of the core VDEF components, including:

Logical links used by VDEF: VDEFVIE n

VDEF Requestors' activation status: MAINTENANCE.

VDEF Request Processors' suspension status and the number of Queued Up, Checked Out, and Errored Out entries in their associated request Queues.

2.6. Deleting Request Queue Entries

To delete an unnecessary VDEF request, use VA FileMan's Enter/Edit Option to delete the subentry in the REQUEST QUEUE ENTRIES multiple of the VDEF REQUEST QUEUE file (#579.3). Delete a subentry only when you are sure that the request was erroneous or that it has been otherwise processed. You can also manually change the status of a request by editing the REQUEST ENTRY STATUS field of the subentry. Check the ERROR TEXT field (#.17) of the REQUEST QUEUE ENTRIES multiple for information that might explain a problem with a VDEF request.

Section 6 of the VDEF Installation and User Configuration Guide describes in detail when to delete a record and how to do it.

You should not delete a VDEF record unless directed to do so by the VDEF Custodial developer.

3. FILES

The following FileMan files belong to this package:

File Number	File Name
577	VDEF EVENT DESCRIPTION
577.4	VDEF EVENT SUBTYPES
579.1	VDEF REQUESTOR
579.2	VDEF DESTINATION
579.3	VDEF REQUEST QUEUE
579.5	VDEF PARAMETERS
579.6	VDEF CUSTODIAL PACKAGE

4. ROUTINES

The following M routines belong to this package:

VDEFCONT
VDEFEL
VDEFKIDS
VDEFMON
VDEFMNU
VDEFQM
VDEFREQ
VDEFREQ1
VDEFUTIL
VDEFVU

5. EXPORTED OPTIONS

The following Options are delivered with this package:

Menu Option	Option Name	Parent Option	Comments
VDEF STARTUP OPTION	VDEF STARTUP OPTION	No parent: queueable option.	This Scheduled Option is run at system startup. It initiates the individual Request Queue processor jobs.
VDEF CONFIGURATION MENU	VDEF CONFIGURATION AND STATUS	No parent: assign at IRM's discretion.	Main VDEF menu
VDEF Activate/Inac Requestor	ACTIVATE/INACTIVATE REQUESTOR	VDEF Configuration and Status	Completely turns off VDEF messaging for a VDEF Requestor.
VDEF Site-Wide Parameters	SITE-WIDE PARAMETERS	VDEF Configuration and Status	Configures VDEF parameters for the whole <i>VISTA</i> site.
VDEF Request Queue Parameters	REQUEST QUEUE PARAMETERS	VDEF Configuration and Status	Configures parameters for a VDEF Request Queue.
VDEF Suspend/Run Request Queue	SUSPEND/RUN REQUEST QUEUE	VDEF Configuration and Status	Suspends generation of HL7 messages for a given VDEF Request Queue.
VDEF Custodial Package Activate/Deactivate	VDEF CUSTODIAL PACKAGE	VDEF Configuration and Status	Completely turns off VDEF message queuing for all VDEF Events associated with a given custodial package.
VDEF Request Processor Schedule	REQUEST PROCESSOR SCHEDULE	VDEF Configuration and Status	Tells the Request Processor for a given Request Queue when not to run.
VDEF Status	STATUS OF VDEF COMPONENTS	VDEF Configuration and Status	Displays the current status of VDEF messaging.
VDEF Event API	VDEF EVENT API ACTIVATE/INACTIVATE	VDEF Configuration and Status	Completely turns off VDEF message queuing for all VDEF Events for a given VDEF Event.

6. ARCHIVING AND PURGING

The VDEF REQUEST QUEUE file (#579.3) has a parameter associated with it (ARCHIVAL PARAMETER field [#04]) which tells the VDEF Monitor TaskMan process how long to wait after each request was fulfilled before deleting it from the queue. It is estimated that the most that this file will grow is 100Mb per day, which is based on about 300 bytes per one VDEF request and up to 300,000 VDEF requests per day. The ARCHIVAL PARAMETER field needs to be set up based on the actual volume of VDEF transactions per day and the disk space available.

NOTE	Entries in the Request Queue file may accumulate if the Request Processor responsible for processing them is suspended for a prolonged period of time. Unprocessed (“Queued up”) entries in the Request Queue file are never deleted. Therefore, if a VDEF Request Processor is suspended indefinitely, but the VDEF Requestor that is associated with it remains Activated, then the Request Queue file will continue growing and will eventually fill up the User Class Identifier (UCI). It is important to make sure to Deactivate VDEF Requestors completely and not just Suspend their associated Request Processor if VDEF processing is to be turned off for a Requestor.
-------------	---

7. ENTRY POINTS / CALLABLE ROUTINES / APIS

There are several utilities in VDEFEL that can be used in writing extraction routines. They perform common, frequently needed functions. The Entry Points for VDEFEL are listed below:

7.1. Entry Point \$\$QUEUE^VDEFQM

7.1.1. Calling Syntax

\$\$QUEUE^VDEFQM(event_ID,name_value_pairs,[.text_message])

7.1.2. Input Parameters

Event_ID

Type	Description	Optionality	Repeatability	Constraints	Example
By Value	A string comprised of an HL7 Message Type and an HL7 Event Type separated by a "^".	Required	None	The value must be defined in the list of known VDEF Message Type/Event Type combinations in the VDEF EVENT DESCRIPTION file (#577).	"ORU^R01"

Name Value Pairs

Type	Description	Optionality	Repeatability	Constraints	Example
By Value	A string comprised of the following format: SUBTYPE=<text>^IEN=nn where the Subtype text is a code based on the application's domain and the clinical event and the IEN is the IEN of the primary file associated with the event.	Required	None	All Subtype names must be defined for this Message Type/Event Type in the VDEF EVENT DESCRIPTION file (#577.4). The name "SUBTYPE" is reserved for the purposes of uniquely identifying the request sub-type when more than one VDEF Event is defined for a Message Type/Event Type combination. For example, the HL7 message type/event ORU^R01 is used by multiple VistA domains to send unsolicited results. In these cases, the Subtype is used to differentiate the sending domain's unique message contents within the structure of the ORU^R01.	"SUBTYPE=VTLS^IEN=12345"

Text Message

Type	Description	Optionality	Repeatability	Constraints	Example
By Reference	Acceptance or rejection text	Optional	None		"Invalid Event Type"

	returned by the VDEF API.				
--	---------------------------	--	--	--	--

7.1.3. Function Return Value

Type	Description	Example
Function Value	A number whose value is 1 if the HL7 request was queued up for processing and 0 if it was not queued up for processing.	1

7.2. Entry Point SETDLMS^VDEFEL

This Entry Point sets HL7 delimiters into individual variables based either on elements of HL() array or default HL7 delimiter values.

7.2.1. Calling Syntax

SETDLMS^VDEFEL

7.2.2. Input Parameters

Type	Description	Optionality	Repeatability	Constraints	Example
Passed through symbol table	HL() as set by INIT^HLFNC2().	Optional	None		

7.2.3. Output Value

Type	Description	Default
SEPC	Component Separator	~
SEPS	Subcomponent Separator	&
SEPR	Repetition Separator	
SEPE	Escape Character	\
SEPF	Field Separator	^

7.3. Entry Point \$\$XCN200^VDEFEL()

This Entry Point is given an IEN from the NEW PERSON file (#200) and the source of the NEW PERSON file, then returns an XCN data type. A source of the data is passed to the API to identify the HL7 field where the person information comes from.

7.3.1. Calling Syntax

\$\$XCN200^VDEFEL(DUZ,SRC)

7.3.2. Prerequisite

HL7 delimiter (SEPC="~") was set by SETDLMS^VDEFEL or \$E(HL("ECH"),1).

7.3.3. Input Parameters

Type	Description	Optionality	Repeatability	Constraints	Example
DUZ - By Value	Internal Entry Number from the NEW PERSON file (#200).	Required	None		12345
SRC - By Value	Pass a string to the second parameter to identify where the data originates.	Optional	None	Default value of 'VistA200' is used if nothing is passed in this parameter.	PHARMACIST

7.3.4. Function Return Value

Type	Description	Example
Return Value	Function returns XCN data type.	<p>W \$\$XCN200^VDEFEL(520637377) 520637377~LASTNAME~FIRSTNAME~~~~VistA200</p> <p>W \$\$XCN200^VDEFEL(520637377,"PHARMACIST") 520637377~LASTNAME~FIRSTNAME~~~~PHARMA CIST</p>

7.4. Entry Point \$\$TS^VDEFEL()

This Entry Point returns an HL7 TS data type.

7.4.1. Calling Syntax

\$\$TS^VDEFEL(datetime)

7.4.2. Input Parameter

Type	Description	Optionality	Repeatability	Constraints	Example
By Value	Date/Time Value	Required	None	Format must be \$H, internal FileMan format, or one of the external formats supported by FileMan	<p>(\$H)</p> <p>("3140820.154420")</p> <p>("AUG 23, 2004@100923")</p>

7.4.3. Function Return Value

Type	Description	Example
Return Value	Function returns a TS data type, including time zone. No time zone is returned if no time is included in input.	W \$\$STS^VDEFEL(\$H) 20040820182643-0400 W \$\$STS^VDEFEL("3140820.154420") 20140820154420-0400 W \$\$STS^VDEFEL("AUG 23, 2004@100923") 20040823100923-0400

7.5. Entry Point POSTKID^VDEFVU()

This Entry Point is used in a post-init to add an entry to the VDEF EVENT DESCRIPTION file (#577) and, if necessary, to the VDEF EVENT SUBTYPE (#577.4) and VDEF CUSTODIAL PACKAGE (#579.6) files. An individual call contains information for a single entry in the VDEF EVENT DESCRIPTION file (#577). If necessary, an Event Description is added for a new entry in the VDEF EVENT SUBTYPE file (#577.4).

7.5.1. Calling syntax

POSTKID^VDEFVU(MSGTYP,EVNTYP,SUBTYP,PROTO,CUSTPKG,EXTROUT,EVDESC,SUBDESC)

7.5.2. Input Variables

1. **MSGTYP**: External value for the MESSAGE TYPE field (.06) in the VDEF EVENT DESCRIPTION file (#577). (Corresponds to the .01 field of an existing entry in HL7 MESSAGE TYPE file [#771.2].)
2. **EVNTYP**: External value for the EVENT TYPE field (.02) in the VDEF EVENT DESCRIPTION file (#577). (Corresponds to the .01 field of an existing entry in the HL7 EVENT TYPE CODE file [#779.001].)
3. **SUBTYPE**: External value for the EVENT SUBTYPE field (.03) in the VDEF EVENT DESCRIPTION file (#577). (Corresponds to the .01 field of an entry in the VDEF EVENT SUBTYPE file [#577.4]). If the entry does not exist in the VDEF EVENT SUBTYPE file (#577.4), a new entry will be added. An application should only add new VDEF Event Subtypes after the subtype has been authorized in writing by DaIS.
4. **PROTO**: External value for the VISTA HL7 PROTOCOL field (.07) in the VDEF EVENT DESCRIPTION file (#577) (Corresponds to the .01 field of entry in the PROTOCOL file [#101].) A new entry in the Protocol file can be sent as part of the KIDS build.

5. **CUSTPKG:** External value for the CUSTODIAL PACKAGE field (.09) in the VDEF EVENT DESCRIPTION file (#577) (Corresponds to the external value of the .01 of entry in the VDEF EVENT DESCRIPTION file [#579.6] that, in turn, corresponds to the .01 field in the PACKAGE file [#9.4].) If the entry does not exist in the VDEF Custodial Package file, a new entry will be added.
6. **EXTROUT:** Value of the EXTRACTION PROGRAM field (.3) in the VDEF EVENT DESCRIPTION file (#577), the routine that will perform the extraction.
7. **EVDESC:** Value of the EVENT DESCRIPTION field (1) in the VDEF EVENT DESCRIPTION file (#577).
8. **SUBDESC:** Value of the EVENT DESCRIPTION field (.02) in the VDEF EVENT SUBTYPE file (#577.4). The parameter is required only if a new entry is being made in the VDEF EVENT SUBTYPE file (#577.4).

All input variables are required except for SUBDESC (parameter 8).

7.5.3. Output

Other than the creation of the desired file entries, this call's only output are the error messages that are passed to the KIDS system for display when a problem is detected with the call.

7.6. Entry Point ERR^VDEFREQ()

This Entry Point is used to record an error encountered during message building being done as the result of a request passed to the VDEF queue. Use this API only inside a message building routine that is invoked by VDEF. The string passed to the API will be filed with the queued request that encounters the error in file 579.3.

7.6.1. Calling syntax

ERR^VDEFREQ(ERROR_STRING)

7.6.2. Input Parameter

Type	Description	Optionality	Repeatability	Constraints	Example
By Value	Error String	Required	None	A string containing the error message to be filed.	The requested entry in the Patient file does not exist.

7.6.3. Output

There are no return values. The input error message is filed in File #579.3.

8. EXTERNAL INTERFACES

The VDEF package includes new HL7 Logical Links: VDEFVIE_n. The VDEFVIE_n links are used to deliver HL7 messages triggered by maintenance activities inside the *VISTA* application to the local site's Interface Engine. See the VDEF Installation Guide, §3 for further HL7 Link configuration details.

9. EXTERNAL RELATIONS

There are numerous Integration Agreements (IAs) covering the interdependencies between VDEF and the individual clinical domains from which VDEF extracts data to build its HL7 messages. See Section 9.1 for the complete list.

9.1. Integration Agreements

The message building routines can make use of the following IAs to VDEF APIs:

IA#	Access to VDEF Feature
4253	VDEF message building request
4248	Message-building utilities
4571	Error recording
4447	Adding entries to VDEF files in KIDS builds

The VDEF routines make use of the following IAs:

Routine	IA#	Access to
VDEFREQ	3552	\$\$PARAM^HLCS2
VDEFQM	4271	Lookup into #771.2
VDEFQM	4321	Lookup into #779.001
VDEFMNU	1373	File #101 access
VDEFMNU	10063	\$\$JOB^%ZTLOAD
VDEFMNU	4322	(\$P(^HLCS(870,link,0),U,12)
VDEFREQ1	4316	Access to #869.3
VDEFCONT	10063	\$\$S^%ZTLOAD \$\$ASKSTOP^%ZTLOAD
VDEFKIDS	10063	\$\$ASKSTOP^%ZTLOAD RTN^%ZTLOAD
VDEFKIDS	10103	\$\$FMADD^XLFDT
VDEFUTIL	10103	\$\$HTE^XLFDT

9.2. Package Dependencies

The following packages must be properly installed in order for the VDEF to function properly:

Required Applications	Minimum Version Number	Required Patches
Kernel	8.0	257 SEQ #234
FileMan	22.0	105 SEQ #111
MailMan	8.0	13 SEQ #13
HL7	1.6	98 Seq #85
MailMan	N/A	XM*DBA*152

10. INTERNAL RELATIONS

VDEF options and TaskMan processes are mostly independent of each other except as stated below. If there is no Request data to be processed, the Request Processor goes into a wait loop before checking the request queue for more entries to be processed.

11. PACKAGE-WIDE VARIABLES

All non-*VISTA* standard local M variables used by VDEF are NEWed at the beginning of each program. There are no package-wide variables.

12. SOFTWARE PRODUCTS SECURITY

12.1. Mail Groups and Alerts

All alerts generated by VDEF will be routed to the VDEF NATIONAL ALERTS mail group using the Application VDEF ALERTS. IRM staff should edit this Mail Group to make sure that the appropriate local IRM staff are included in the Mail Group. A MailMan message containing the text of the alert is also sent to the VDEF developer on FORUM and Outlook.

12.2. Remote Systems

The HDR-IMS is the first system to which VDEF will transmit HL7 messages. However, VDEF is capable of supporting a number of destinations for HL7 messages.

VISTA HL7 destinations use the VISTA HL7 package to transmit the generated HL7 messages to their remote systems. When using VISTA HL7 destinations, HL7 messages may be delivered to multiple remote systems as long as the VISTA HL7 subscriber information is set up properly for the affected HL7 Event Types.

At this time, VDEF software does not use any form of encryption beyond what is currently built into VISTA HL7 and the BusinessWare Interface Engine. This approach is used because all recipients of the VDEF-generated HL7 data are behind the Department of Veterans Affairs (VA) firewall.

12.2.1. Acknowledgements

Maintenance requests. When a software trigger inside VISTA calls the VDEF API (see this document for the API specification), the API returns “1” if the request was successfully queued up for processing or “0” otherwise. If the returned value is “0,” the VDEF API also returns a free text error message in an output parameter. Once the VDEF request is fulfilled and the resulting HL7 message is passed on to VISTA HL7 for delivery to one or more remote systems, the VISTA HL7 package will be responsible for processing the acknowledgements from those systems.

12.3. Contingency Planning

VDEF has the following built-in mechanisms that prevent adverse impact on the VISTA system:

Manual suspension of Request Queue Processor using the VDEF SUSPEND/RUN REQUEST QUEUE menu option, which is designed to alleviate ad hoc performance problems that VDEF may cause.

Automatic suspension of Request Processor using the VDEF REQUEST PROCESSOR SCHEDULE option, which is designed to alleviate chronic performance problems during peak load hours.

13. APPENDIX A: VDEF SHUT DOWN PROCEDURES

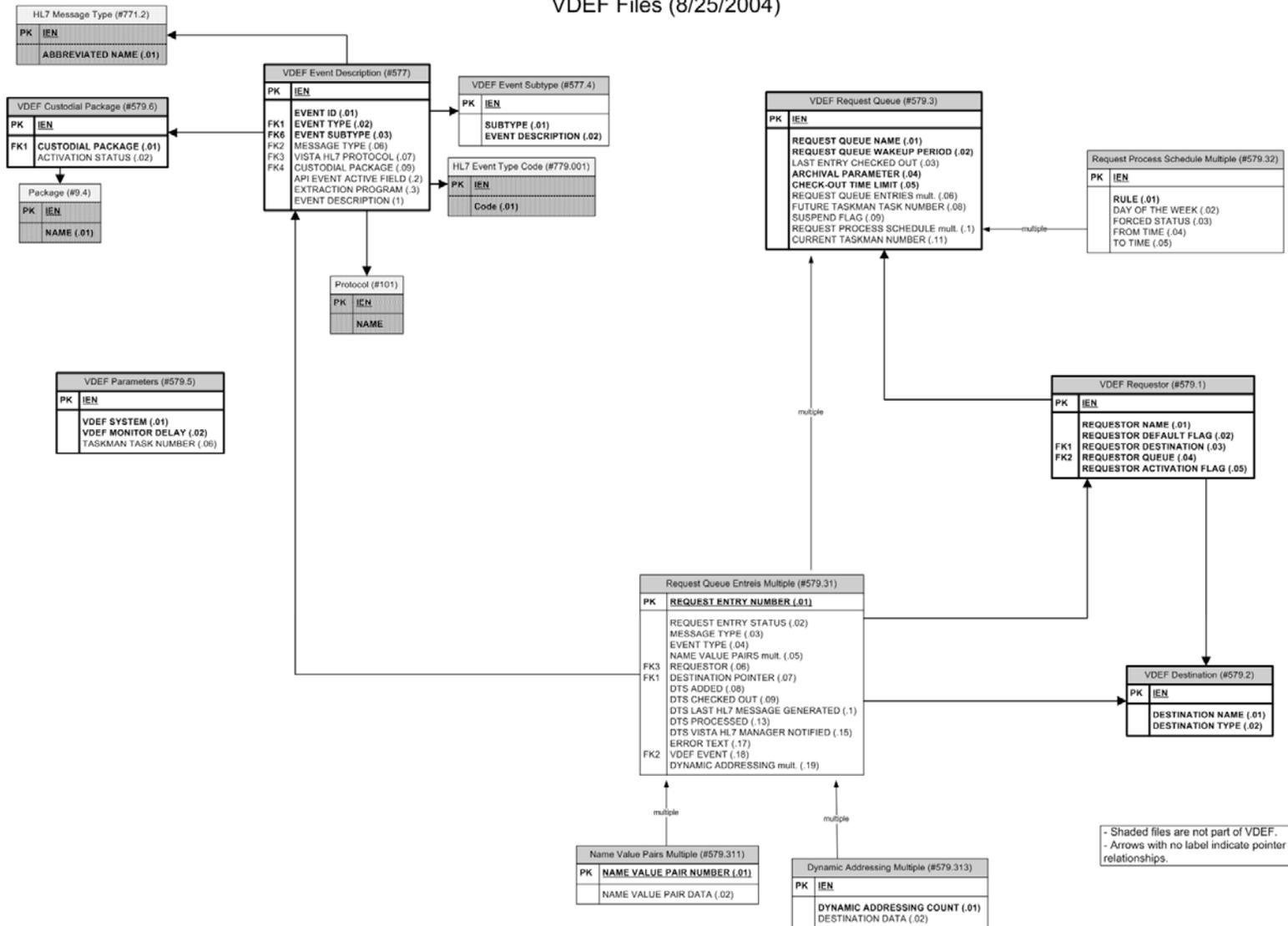
Refer to section 5 of the VDEF 1.0 Installation and User Configuration Guide for instructions on the procedures for shutting down various portions of the VDEF processing.

14. APPENDIX B: VDEF INITIAL SETUP PROCEDURES

Refer to the VDEF V1.0 Installation and User Configuration Guide.

15. APPENDIX C: VDEF FILES

VDEF Files (8/25/2004)



16. APPENDIX D: VDEF ALERT MESSAGES

The following is the list of alerts that VDEF can generate.

1. VDEF CHECKED OUT MONITOR FAILED TO START. CHECK ERROR TRAP.
(The scheduled task MONITOR^VDEFCONT did not start as requested by the VDEF STARTUP OPTION.)
2. RECORD <IEN> IN QUEUE <QUEUE NAME> HUNG IN CHECKED OUT STATUS.
(The request specified by the IEN in the queue specified in the QUEUE NAME has been in the Checked Out status longer than the allowed time period.)
3. VDEF REQUEST PROCESS <QUEUE NAME> FAILED TO START. CHECK ERROR TRAP.
(The task EN^VDEFREQ failed to start as requested by the VDEF STARTUP OPTION)
4. VDEF QUEUE PROCESS MONITOR DID NOT START. CHECK ERROR TRAP.
(The task MONITOR^VDEFMON failed to start as requested by the VDEF STARTUP OPTION)
5. VDEF HAS REQUEUED CHECKED OUT RECORDS. NO ACTION NEEDED.
(The VDEF queue process monitor has requeued one or more requests that were hung in the Checked Out status)
6. VDEF HAS REQUEUED ERRORED OUT RECORDS. NO ACTION NEEDED.
(The VDEF queue process monitor has requeued one or more requests that were in the Errored Out status)
7. VDEF QUEUE PROCESS MONITOR HAS EXITED.
(The MONITOR^VDEFMON process has stopped running in response to a request to stop from TaskMan)
8. VDEF QUEUE ""_QUEUE_" AUTO-RESTARTED. NO ACTION REQUIRED.
(The MONITOR^VDEFMON process has found that the request queue process is not running and has restarted it by creating a new EN^VDEFREQ task)
9. VDEF QUEUE ""QUEUE"" IS SUSPENDED. PLEASE START IT.
(The MONITOR^VDEFMON process has found that a Request Queue is in the Suspended state and needs to be reset to the Running state)
10. VDEF REQUEST QUEUE PROCESSOR FOR "\$P(Queue,U)" HAS EXITED.
(The request queue process EN^VDEFREQ has stopped running in response to a request to stop from TaskMan or because the request queue status has changed to Suspended)