

The logo for Fortra, featuring the word "FORTRA" in a bold, white, sans-serif font. The letter "A" is replaced by a white triangle pointing to the right. The logo is centered within a dark teal rectangular box.

With GoAnywhere (GAMFT) + VERA, you can automatically apply digital rights management (DRM) to every file that lands or is sent within the GAMFT system, for total protection wherever the files go.

Prepared by: John Tkaczewski and Heath Kath

Date: Nov 24, 2022

Version: 2.0

Table of Contents

Pre-install Check List	3
Quick Introduction	3
Installation Instructions	3
Windows CLI Test	3
Linux CLI Test.....	4
Deployment of Projects in GAMFT.....	4
VERA Secure Form for GAMFT	6
Overview	6
Deployment of the VERA form and the corresponding project.....	7
How to Demonstrate the GAMFT + VERA Integration.....	10
Descriptions of files in this package	11
General Flow of Windows CLI (for Reference).....	13
VERA Command LINE (CLI) Reference Documentation	13
VERA CLI on Windows	13
VERA CLI on Linux.....	14

Pre-install Check List

- GoAnywhere 6.8.6 or higher installed
- Root or Administrator access to the operating system hosting GAMFT (no support for MFTaaS)
- Vera Instance (or Vera Tenant) with Admin access <https://xxxxx.vera.com/> where xxxxx is your company name
- In GAMFT, SMTP settings defined and working under System - Global Settings – SMTP
- Download or otherwise be sent **vera-gamft-integration-package-vXX.zip**
- Access to the public Internet from GAMFT machine, or at a minimum access to <https://xxxxx.vera.com/> where xxxxx is your company name
- To use the Secure Form Project, JDBC URL to the GAMFT Database as well as DB username/password

Quick Introduction

The vera-gamft-integration-package-vXX.zip contains the following utility GAMFT Advance Workflows projects:

- VeraSecureUtilityLinuxWindows.xml
- VeraUnsecureUtilityWindowsLinux.xml

Utility projects can be called to secure/unsecure files from any other GAMFT project; See examples: VeraFromTrigger.xml and VeraFromMonitorToFTPS.xml as examples. These examples are an illustration on how to call the utility projects. Create your own GAMFT projects that can be called from any trigger, monitor, secure form or schedule that use those utility projects.

Installation Instructions

1. Unzip **vera-gamft-integration-package-vXX.zip** into c:\veragamft folder on Windows or /veragamft on Linux → make sure that it's all lower case. – This zip file should have been emailed to you
2. Login to your **VERA Cloud** instance (Vera Tenant – need admin rights)
 1. <https://xxxxx.vera.com/> where xxxxx is your company name with a Vera instance already created for you
 2. On the left, click on Settings / APPS
 3. Click on Create App - For application name: provide your name or system ID where the SDK (this Software Development Kit) will be used. For username, provide the email address of your VERA admin user and press SAVE to create.
 4. Once the App is created, click on the Down Arrow all the way to the right and Download Config file

App Name	User Name	App ID	App Secret
<input type="checkbox"/> MFTPartner	john_brown@xxxxx.com	*****	*****
<input type="checkbox"/> MFTDemo	john_brown@xxxxx.com	*****	*****

3. Rename the downloaded file to *conf.json* and copy it to c:\veragamft folder on Windows or /veragamft on Linux
4. Test your CLI (Command-Line Interface). CLI TEST

Windows CLI Test

Open Command Prompt cd \veragamft or to your install package folder default: c:\veragamft

GoAnywhere + Vera DRM Integration – Installation Instructions

type in: *VeraSdkSample.exe get-policies*

Example output should be similar listing the policies defined in the Vera Tenant

```
Command Prompt
Microsoft Windows [Version 10.0.19043.1766]
(c) Microsoft Corporation. All rights reserved.

C:\Users\john.tkaczewski>cd \veragamft

C:\veragamft>VeraSdkSample.exe get-policies
Allow Decrypt
Co-Owner
Collaborate
Full LockDown
Full LockDown minus web viewing
Track
View Only
Wrap
```

Linux CLI Test

1. `cd /veragamft`
2. `export LD_LIBRARY_PATH=.`
3. `chmod a+x VeraSDKSample`
4. `./VeraSDKSample get-policies`

Example output should be similar listing the policies defined in the Vera Tenant

```
[root@ip-ec2-user]# cd /veragamft/
[root@ip-veragamft]# chmod a+x VeraSDKSample
[root@ip-veragamft]# export LD_LIBRARY_PATH=.
[root@ip-veragamft]# ./VeraSDKSample get-policies
Allow Decrypt
Co-Owner
Collaborate
Full LockDown
Full LockDown minus web viewing
Track
View Only
Wrap
[root@ip-veragamft]#
```

Deployment of Projects in GAMFT

1. Login **GoAnywhere** MFT with your Admin account
 - i. Go To Projects (Will be Importing XX Projects and one secure form to GAMFT)
 - ii. Create a new project folder (called *vera* – lower case) off your root projects folder by right clicking on ROOT Folder and selecting Add option
 - iii. Select the vera folder, Click on Import Projects – Import the Projects from XML.
Import the following projects from c:\veragamft on Windows or /veragamft on Linux:

- VeraSecureUtilityLinuxWindows.xml
- VeraUnsecureUtilityLinuxWindows.xml
- VeraSecureFromTrigger.xml
- VeraFromMonitorToFTPS.xml
- VeraUnsecureFromMonitor.xml

iv. **Optional Step:** Create Folder level variables for workingDir and veraExe

The screenshot shows the 'Edit Folder' dialog box. On the left, a tree view shows the folder structure: (Root) > icap > stress-Test > templates > toSecureFolder > vera > vera1.1. The 'vera' folder is selected. The 'Edit Folder' dialog has two tabs: 'Folder' and 'Variables'. The 'Folder' tab is active, showing 'Folder Name' as 'vera' and 'Description' as 'vera-projects'. Below this, a section titled 'Variables Defined on this Folder' shows a table with two variables: 'veraExe' and 'workingDir'. Both have red 'X' icons in the 'Name' column, indicating they are not set or are invalid. The 'Value' column shows 'VeraSdkSample.exe' for veraExe and 'c:\veragamft' for workingDir. The 'Description' column has links to 'Click here to edit'.

Name	Value	Description
✗ veraExe	VeraSdkSample.exe	Click here to edit
✗ workingDir	c:\veragamft	Click here to edit

These variables will apply to all the projects under the vera folder and you will not need to set workingDir and veraExe variables in each individual project.

2. Edit the following parameters for all imported projects
 1. Set the 'workingDir' variable in all imported projects to your local Vera path like c:\veragamft\ folder on Windows or /veragamft/ on Linux ---- **trailing slash is required!**
 2. If the variable is present, set 'veraExe' to the correct CLI executable. Default on Windows: VeraSdkSample.exe default on Linux: VeraSDKSample
9. Edit each non-utility projects
 1. Edit the following Project Variables as needed: veraOwnerEmail, veraRecipientEmail, veraPermission , pathForManualRun, and targetPath as needed (i.e. veraOwnerEmail to your Vera admin email address)
10. Edit the VeraFromMonitorToFTPS project
 1. Set the FTPS server you wish to use to upload the file after the files are wrapped with VERA (optionally, an SFTP Server resource could be used, but you will have to update the project accordingly); make sure the folder path is correct and your web user has the correct permissions before running.

VERA Secure Form for GAMFT

Overview

GAMFT Secure Forms allow end-users to fill out and submit information online through secure custom-made form with one or more input values, and to upload and attach files through GoAnywhere's HTTPS Web Client. When a form is submitted, a Project in GoAnywhere is executed to automatically process the submitted values and files.

This package comes with a GAMFT Secure Form that allows the end user to share and automatically protect files using VERA. Using the form is optional and may not be required for your specific application.

Example of the Vera Secure Form:

The screenshot shows the 'Forms > Protect files and Email Using VERA' interface. On the left is a sidebar with a logo for 'ACME INDUSTRIES' (a red circle with a black anvil and the text 'It Might Just Work!') and a menu with 'Files', 'Mail', and 'Forms'. The 'Forms' section is expanded, showing 'Available Forms' (selected), 'Drafts', and 'Submitted'. The main content area has a blue header with the title and buttons for 'Submit', 'Save', and 'Cancel'. Below the header, it says 'Select Security Level Recipients and Files'. A list of instructions is provided: 'Select Security Level and Recipients (max of 5)', 'Select File(s) To Send and protect (max 4 files, XML and HTML files can't be protected)', 'Enter short text message to all the recipients', and 'Click submit. The secured files will be sent via email to the recipients'. The 'File Protection and Tracking' section contains two rows for recipients. The first row has 'Security level' set to 'View Only' and 'Recipient Email' as 'user2@demo.com'. The second row has 'Security level' set to 'Full Lockdown' and 'Recipient Email' as 'user1@demo.com'. Below this is a text area for 'Message to all the recipients' with the text 'here are Vera protected files' and a '971 Characters Remaining' indicator. At the bottom, there is a 'Files to share' section with a dashed box for dropping files or a link to 'browse to attach files'. Two files are listed: 'Legal Contract - DRAFT.docx' (20.36 KB) and 'TJX application.pdf' (146.51 KB).

Forms > Protect files and Email Using VERA

✓ Submit 📁 Save ✕ Cancel

Select Security Level Recipients and Files

- Select Security Level and Recipients (max of 5)
- Select File(s) To Send and protect (max 4 files, XML and HTML files can't be protected)
- Enter short text message to all the recipients
- Click submit. The secured files will be sent via email to the recipients

File Protection and Tracking

✕ Security level: *	View Only ▼
Recipient Email: *	user2@demo.com
✕ Security level: *	Full Lockdown ▼
Recipient Email: *	user1@demo.com

+

Message to all the recipients * 971 Characters Remaining

Files to share: *

Drop files here, or [browse to attach files](#)

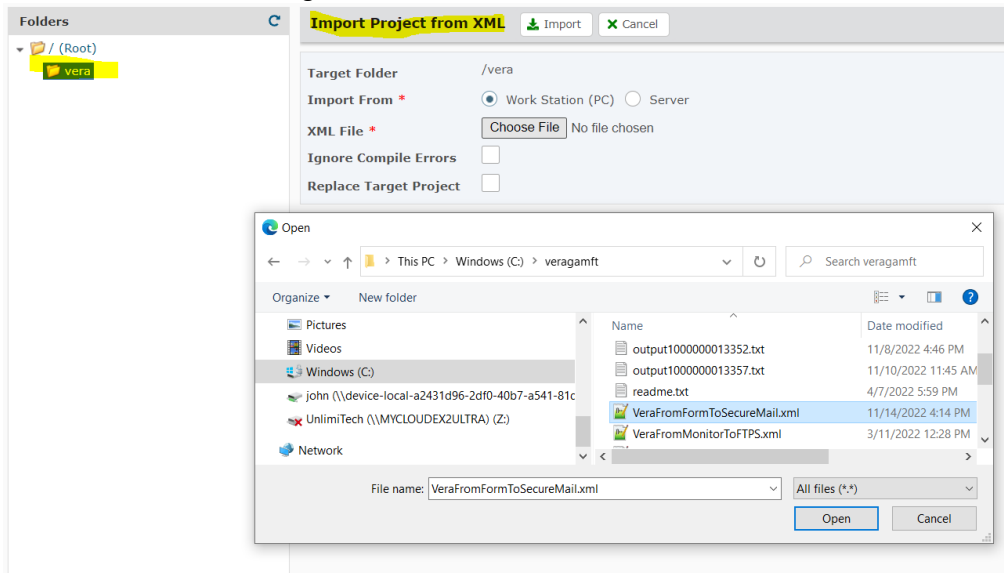
✕ Legal Contract - DRAFT.docx	20.36 KB
✕ TJX application.pdf	146.51 KB

Using the VERA secure form, end users can provide the email address of the recipients as well as individual permissions policy for each user and upload the file(s). The recipients will obtain an email with a link to download the encrypted files with the corresponding VERA policy specified by the sender.

GoAnywhere + Vera DRM Integration – Installation Instructions

Deployment of the VERA form and the corresponding project

1. Import VeraFromFormToSecureMail.xml project into the existing /vera project folder in GAMFT from c:\veragamft\ folder on Windows or /veragamft/ on Linux

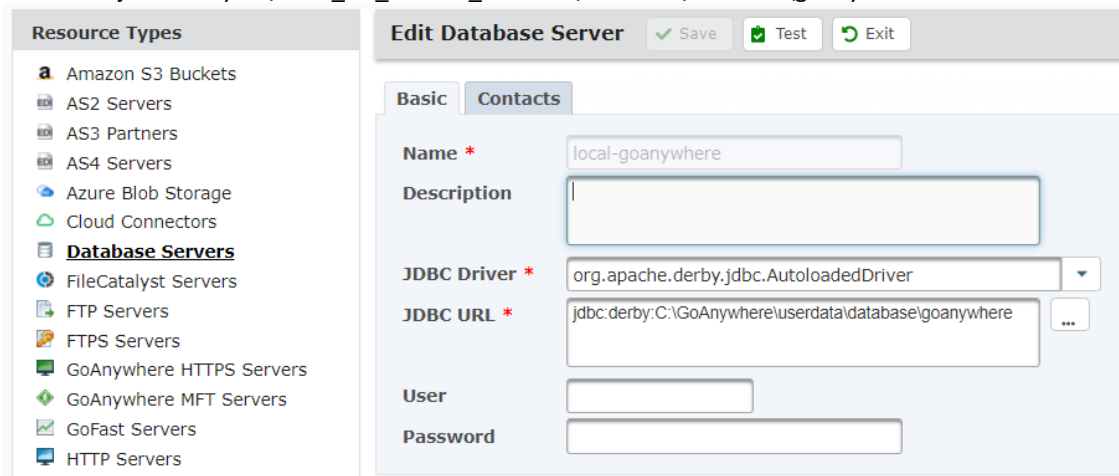


3. Create New or Verify whether the following GoAnywhere MFT Resources exist and are working:
 - i. Database back to GAMFT – connects to your own GoAnywhere database **hint:** see System / Database Configuration

Example on Windows using default Internal Apache Derby DB for GAMFT

JDBC Driver: org.apache.derby.jdbc.AutoloadedDriver

JDBC URL: jdbc:derby:C:\PATH_TO_GAMFT_INSTALL\userdata\database\goanywhere



Example on Linux with Externalized DB to MariaDB:

JDBC Driver: org.mariadb.jdbc.Driver

JDBC URL: jdbc:mariadb://127.0.0.1:3306/GADATA?useCursorFetch=true&defaultFetchSize=20&characterEncoding=utf8

GoAnywhere + Vera DRM Integration – Installation Instructions

Resource Types

Amazon S3 Buckets

AS2 Servers

AS3 Partners

AS4 Servers

Azure Blob Storage

Cloud Connectors

Database Servers

FileCatalyst Servers

FTP Servers

FTPS Servers

GoAnywhere HTTPS Servers

GoAnywhere MFT Servers

GoFast Servers

HTTP Servers

HTTPS Servers

Edit Database Server

Save

Test

Exit

Basic

Contacts

Name *

local db

Description

JDBC Driver *

org.mariadb.jdbc.Driver

JDBC URL *

jdbc:mariadb://127.0.0.1:3306/GADATA?
useCursorFetch=true&defaultFetchSize=20&characterEncoding=utf8

User

GADATA

Password

Change Password

- ii. Test your Secure Mail configuration in GAMFT. Check your Domain and make sure the Web Client Site is correct (not the default of localhost). Test GAMF Secure Mail by logging in as a regular user to the web client portal and click on Mail -> Compose. Send your-self a file and make sure that the email sending works and that the download link to download files works.

Files

/ (Home)

Shared Files

GoDrive

Mail

Inbox

+

Compose

Request Files

Drafts

Templates

Outbox

Sent Items

Forms

Mail >Compose

Send

Save

Cancel

From

john.tkaczewski@helpsystems.com

To *

CC

BCC

Subject *

Message

Options

☐ Require Registered Users

☐ Expire Package

☐ Read Receipt

Attachments ▾

Drop files here, or [browse to attach files](#)

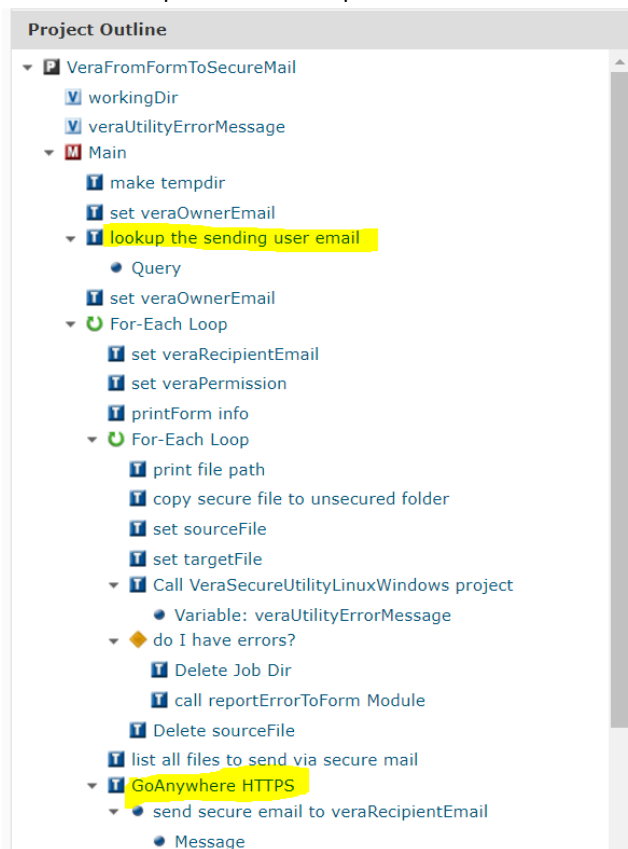
Example of a Secure Email

The screenshot shows an email client window. At the top, there's a header bar with navigation icons like Reply, Forward, Archive, Junk, Delete, and More. Below this, the email details are displayed: From: Me <john@filecatalyst.demo>, To: Me <john@filecatalyst.demo>, Subject: testing. The main body of the email has a blue header section with the text "Secure Delivery". Below this, it states: "The following file(s) have been sent to you from john.tkaczewski@helpsystems.com". A single attachment is listed: "Structural-Sanitization-pdf-active-content (1).pdf" with a size of "1.65 KB". There is a blue button labeled "Download Files". At the bottom, a small disclaimer reads: "If the link above does not open, please copy and paste the following URL into your browser: http://localhost:6443/doc?token=9d4a953f-04d0-m7g-j0ba-s4y5z4v793v". The footer indicates it is "Powered by Gdnovate".

- iii. GoAnywhere HTTPS Server – connect to your local GA HTTPS server for Secure Mail. The username and email specified here will be used as the email address from which all VERA emails will be going out as.

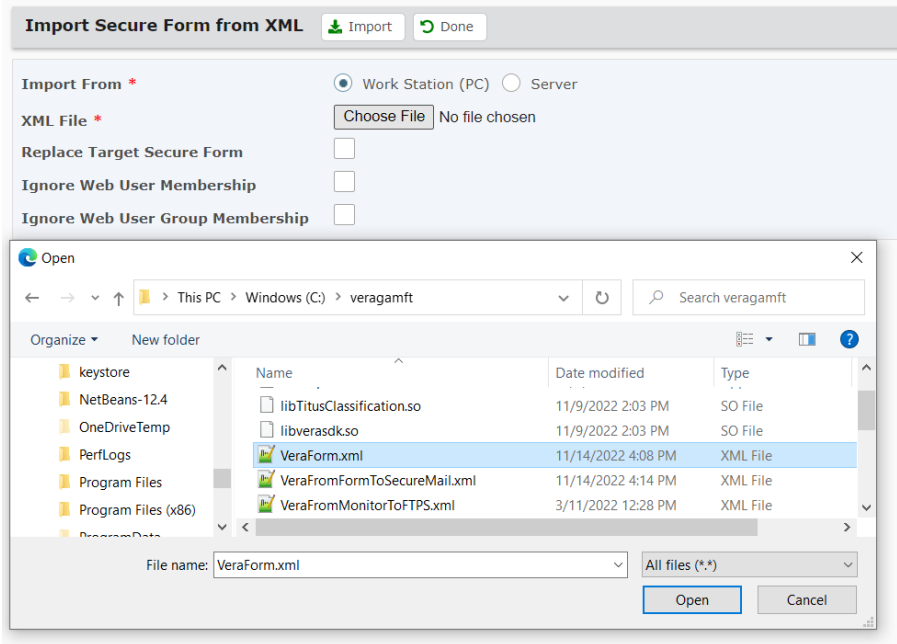
The screenshot shows the 'Edit GoAnywhere HTTPS Server' dialog in the GoAnywhere interface. On the left, a 'Resource Types' list includes various cloud and server resources, with 'GoAnywhere HTTPS Servers' highlighted. The main dialog has five tabs: 'Basic', 'Connection', 'Proxy', 'Certificates', and 'Contacts'. The 'Basic' tab is selected, displaying the following fields: 'Name' (local), 'Description' (empty), 'Host' (localhost), 'Port' (6443), 'User' (johnhk), and 'Password' (Change Password). At the top right of the dialog are buttons for 'Save', 'Test', and 'Exit'.

3. Edit the VeraFromFormToSecureMail project and Set the Database Resource to your GAMFT DB resource. Locate line: "Lookup the sending user email" in the project and select the correct Internal DB resource you created in the previous step
4. Still while Editing VeraFromFormToSecureMail, edit "GoAnywhere HTTPS" task and set the resource that points back to GoAnywhere HTTPS Server you defined in the previous step.
5. Illustration for previous two steps.

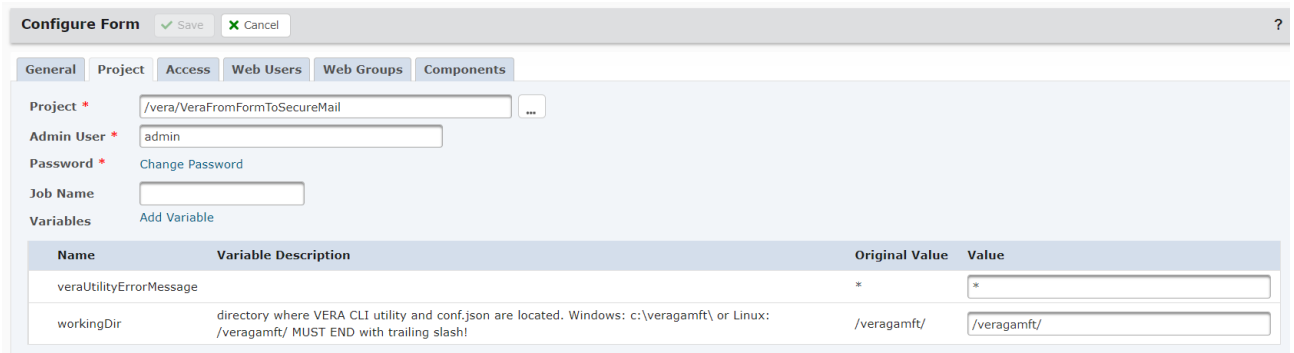


6. Within GAMFT Admin Go to Services - Secure Forms - Form Manager. Click on Import Secure Forms - Import from XML
 - i. Import VeraForm from c:\veragamft on Windows or /veragamft on Linux

GoAnywhere + Vera DRM Integration – Installation Instructions



- ii. Edit the newly imported form
- iii. Edit the Admin User, Password, and Web Users / Web Groups that are allowed to access this form
- iv. You must point the form to /vera/VeraFromFormToSecureMail project
- v. Update the workingDir under project tab of the VeraForm to c:\veragamft on Windows or /veragamft on Linux



You should now have a working **VERA** form. To use the email features of the **VERA** secure form, you must also have the **SMTP settings defined and working under System - Global Settings – SMTP**

How to Demonstrate the GAMFT + VERA Integration

1. Web User logs in via the HTTP Web Client and submits the **Protect files and Email Using VERA form**
 2. Monitor -> Run VeraFromMonitorToFTPS project manually or create a monitor to pick up the file(s) and pass to project.
 3. Trigger -> Upload a file and have the trigger run the VeraSecureFromTrigger project passing both filePath and fileName variables. Or to run project manually, you will need to create these 2 project variables and you may want to disable the delete source file task if you wish to keep the file for testing.
 4. Setup a trigger on "Delete File Successful", call the project '/vera/VeraRevokeAllAccessFromTrigger' any file deleted in GAMFT will also revoke all access to the file in VERA
- VERA can't process HTML or XML files, do not attempt to secure files of those types.

Descriptions of files in this package

File Name	Description
Read-Me-Instructions.pdf	This file
VeraForm.xml	Secure form export, this form can take up-to 5 recipients with different VERA permissions and multiple files. This form calls VeraSecureUtilityLinuxWindows project.
VeraFromFormToSecureMail.xml	Secure form export, this form can take up-to 5 recipients with different VERA permissions and multiple files. This form calls VeraFromFormToSecureMailUsingWindowsUtility project.
VeraFromFormToSecureMail.xml	Project that is called by VeraForm, this project in turn calls VeraSecureUtilityLinuxWindows. This project: <ul style="list-style-type: none"> takes the input from the secure form wraps the uploaded files with permissions and recipients provided by the secure form Sends a secure mail to each recipient with the secured/encrypted files
VeraFromMonitorToFTPS.xml	Project that processes a call from a monitor. This project calls VeraSecureUtilityLinuxWindows. Requires \${files} parameter passed from the Monitor. This project will loop through each file, secure it with VERA and upload the file to an FTPS resource.
VeraFromTrigger.xml	Project that processes a call from a trigger (usually upload success trigger), secures the file with VERA using VeraSecureUtilityLinuxWindows project and deletes the original file. This type of workflow is meant for automatically replacing any file that arrives to GAMFT with a VERA secured version.
VeraSecureUtilityLinuxWindows.xml	Utility Project to secure files by executing Execute Native Command in GAMFT. This project uses VeraSdkSample.exe CLI on Windows or VeraSDKSample on Linux. <p>Parameters Required:</p> <ul style="list-style-type: none"> veraOwnerEmail - Owner of the file to be encoded. This user must exist in the VERA Portal veraRecipientEmail - Recipient of the file for whom the permission will be applied. <ul style="list-style-type: none"> This user must exist in the VERA Portal – or will be asked to create an account This parameter can also be set to 'none' without the single quotes, if you need to secure a file but don't know the recipient. Files secured without a recipient will require every recipient to Request Access before the file can be accessed

GoAnywhere + Vera DRM Integration – Installation Instructions

	<ul style="list-style-type: none"> veraPermission - Textual Representation of the Permission. Permissions will look like this: Collaborate, Full LockDown Track, View Only sourceFile - Absolute Path + filename to the file to be encoded targetFile - Absolute Path + filename to the new file that will be created workingDir - Path to the CLI executable - this will also be the working directory for the project. Must end with a trailing slash. veraExe - file name the CLI tool ex: VeraSdkSample.exe on Windows or VeraSDKSample on Linux veraUtilityErrorMessage – in case of an error, the error message will be passed back to the project that called this utility projects
VeraUnsecureUtilityLinuxWindows.xml	<p>Utility Project to UNSECURE files with VERA using GAMFT. In some use cases, an automated way to unsecure VERA files might be required, use this project to unsecure VERA files. This project is cross platform.</p> <p>Parameters Required:</p> <ul style="list-style-type: none"> sourceFile - Absolute Path + filename to the file to be unsecured targetFile - Absolute Path + filename to the new unsecured file workingDir - Path to unzipped VERA package must end with trailing slash veraExe - file name the CLI tool ex: VeraSdkSample.exe on Windows or VeraSDKSample on Linux veraUtilityErrorMessage – in case of an error, the error message will be passed back to the project that called this utility projects
VeraUnsecureFromMonitor.xml	<p>Project that takes files from a Monitor, UNSECURES the files from VERA and places the unsecured files in a new unsecured folder. This project calls VeraUnsecureUtilityLinuxWindows project.</p>
verasdk.dll, Interop.VeraSdkLib.dll, VeraSdkSample.exe.config	<p>DLL and config files required to run the exe version of VERA CLI VeraSdkSample.exe in Windows</p>
VeraSdkSample.exe	<p>Windows CLI utility to wrap files with VERA. See VERA Command LINE HELP OUTPUT below for details. This file is called multiple times by VeraSecureUtilityLinuxWindows project. This file requires conf.json which must be downloaded from the VERA portal under Settings - Apps for authentication. Admin rights to the VERA portal will be required to download the json file.</p>
libTitusClassification.so, libverasdk.so,	<p>Linux library files used by the Linux CLI (VeraSDKSample)</p>
VeraSDKSample	<p>Linux CLI Executable. You must chmod a+x this file. This file requires conf.json which must be downloaded from the VERA portal under Settings - Apps for authentication. Admin rights to the VERA portal will be required to download the json file. This file is called multiple times by VeraSecureUtilityLinuxWindows project.</p>
VeraSecureTrigger.xml	<p>Export of the trigger that calls VeraFromTrigger project upon any successful file upload.</p>

General Flow of Windows CLI (for Reference)

To wrap a file with VERA, 4 CLI commands need to be executed. Both Windows and Linux CLI's require similar same inputs. This is an example of how the utility vera project VeraSecureUtilityLinuxWindows interacts with the VERA Windows CLI. You must download conf.json from the VERA portal under Settings - Apps for authentication; place it in the same folder as the executable.

Steps to Secure a file:

1. `cd \veragamft`
2. `VeraSdkSample.exe get-policies` (to obtain unique ID of the policy you wish to wrap the file with)
3. `VeraSdkSample.exe secure c:\path\to\inputfile.txt c:\path\to\wrappedfile.txt.html` (this operation will return DocID, note the DocID for the next operation)
4. `VeraSdkSample.exe change-owner <doc-id-from-previous-call> user_id_in_vera@emailaddress.com`
5. Optional STEP: `VeraSdkSample.exe grant-access c:\path\to\wrappedfile.txt.html user_id_in_vera@emailaddress.com <user-name-policy-id-obtained-in-Step-A>`

Step to unsecure a file:

```
VeraSdkSample.exe unsecure c:\path\to\wrappedfile.txt.html c:\path\to\unsecurefile.txt
```

VERA Command LINE (CLI) Reference Documentation

BEFORE USAGE: Download the conf.json file from the VERA Portal. The JSON file is an application connection file required by either VeraSdkSample.exe or VeraSDKSample on Linux

VERA CLI on Windows

```
C:\VERA\veragamft>VeraSdkSample.exe
```

Usage:

get-policies

secure <input-path> <output-file> [--doc-id:<docId>] [--mapping-tag:<mappingTag>]

get-doc-key <doc-id>

secure-input-stream <input-path> <output-file> [clientDocId]

secure-bulk <input-path>+

unsecure <input-path> <output-file>

unsecure-stream <input-path> <output-file>

unsecure-range <input-path> <range-start> <range-end>

grant-access <path> <user-name policy-id>+

revoke-access <path> <user-name>+

revoke-all-access <path>

get-metadata <path>

```
change-access [--doc-id:<docId> [--action:<grant|revoke> --type:<user|group|domain|anon> --name:<user-name> --
policy-id:<policy-id> --sharing-policy-id:<sharing-policy-id>]+ ]+
change-owner <doc-id> <user>
get-doc-details <doc-id>
update-mapping-tag <doc-id> <mapping-tag>
Classification/Tagging:
classify:put-metadata-tags <input-path> <output-file> <tag-name tag-value>+
classify:get-metadata-tags <input-path>
classify:get-classification-tags
classify:put-classification-tag <input-path> <output-file> <tag-id>
classify:get-classification-tag <input-path>
classify:put-tags <input-path> <output-file> <classification-tag-id> <tag-name tag-value>+
classify:get-tags <input-path>
classify:remove-tags <input-path> <output-path>
```

VERA CLI on Linux

```
[root@gamftserver veragamft]# ./VeraSDKSample
```

Usage:

VeraSDKSample get-policies

VeraSDKSample secure <input-path> <output-file> [--type:<user|group|domain|anon> --name:<user-name> --policy-
id:<policy-id> --sharing-policy-id:<sharing-policy-id>]+ [--mapping-tag:<mapping-tag>]

VeraSDKSample secure-stream <input-path> <output-file>

VeraSDKSample get-metadata <path>

VeraSDKSample get-doc-details <doc-id>

VeraSDKSample unsecure <input-path> <output-file>

VeraSDKSample grant-access <path> [user-name policy-id]+

VeraSDKSample revoke-access <path> <user-name>+

VeraSDKSample revoke-all-access <path>

VeraSDKSample change-access [--doc-id:<docId> [--action:<grant|revoke> --type:<user|group|domain|anon> --
name:<user-name> --policy-id:<policy-id> --sharing-policy-id:<sharing-policy-id> --start-time:<timebomb-start-
time(yyyy-MM-dd'T'HH:mm:ssZ)> --end-time:<timebomb-end-time(yyyy-MM-dd'T'HH:mm:ssZ)>]+]+

VeraSDKSample update-mapping-tag --doc-id:<docId> --mapping-tag:<mapping-tag>

VeraSDKSample change-owner --doc-id:<docId> --owner:<owner>

Classification/Tagging:

VeraSDKSample classify:put-metadata-tags <input-path> <output-file> <tag-name tag-value>+

VeraSDKSample classify:get-metadata-tags <input-path>

VeraSDKSample classify:get-classification-tags

VeraSDKSample classify:put-classification-tag <input-path> <output-file> <tag-id>

VeraSDKSample classify:get-classification-tag <input-path>

VeraSDKSample classify:put-tags <input-path> <output-file> <classification-tag-id> <tag-name tag-value>+

VeraSDKSample classify:get-tags <input-path>

VeraSDKSample classify:remove-tags <input-path> <output-path>