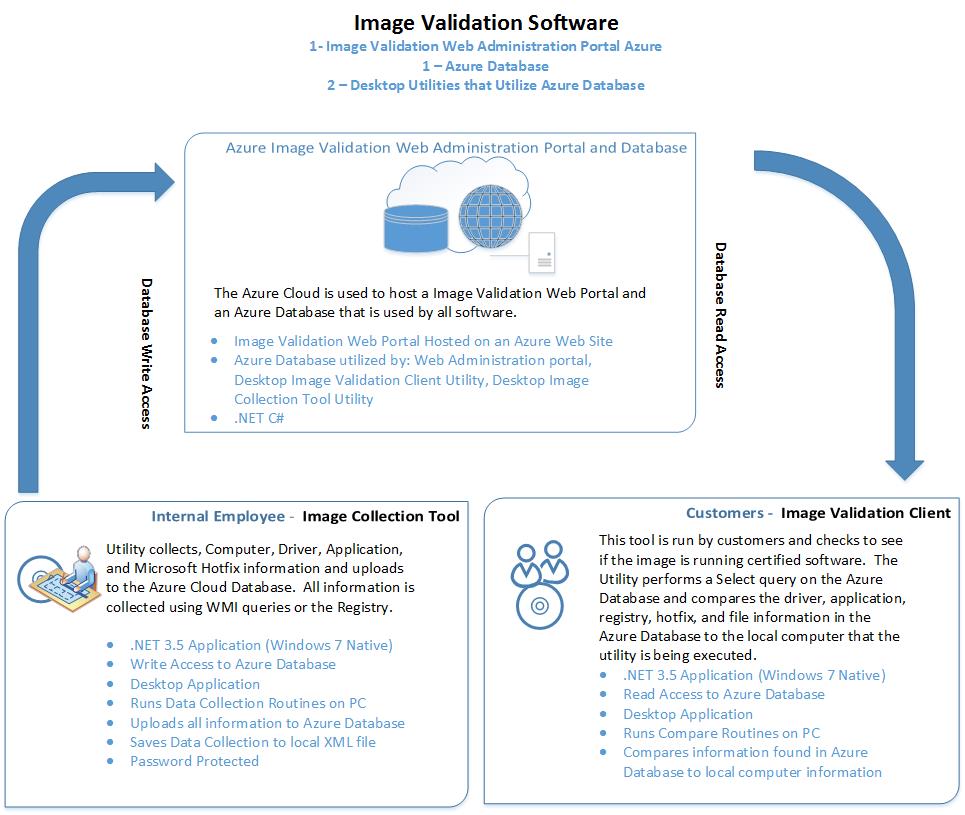
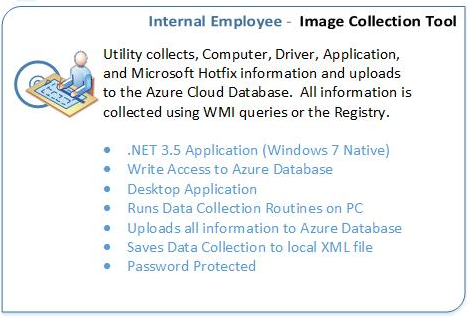
**IMAGE VALIDATION COLLECTION UTILITY**

**12/28/2012**

**Full Project Summary:**

****

**Image Validation Collection Tool:**

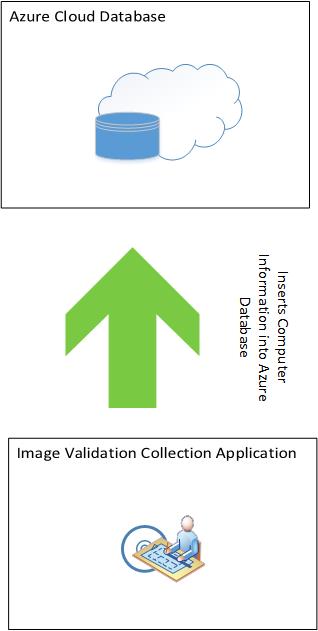
****

**Specifications:**

1. Inserts Data into a Microsoft Azure Database
2. .NET 3.5 for native compatibility on Windows 7
3. Programmed in C#
4. Implements User and Role Based Security
5. WPF Desktop Application

**Application Overview:**

This application runs information gather routines which utilizes WMI queries and registry information collection and uploads the information to the cloud Azure Database.

****

**User and Role Based Security:**

**User Security:**

1. Each user can be assigned one security role.
2. Password should be at least 6 characters long and contain the following
   1. 1 Number
   2. 1 Uppercase Letter
   3. 1 Lowercase Letter

**Role Security:**

1. Client: This account is used by the Validation Client Tool and only has read permission to the Azure Database. This account is unable to log into the Web Portal
2. Validator: This account has permission to use the Image Validation Collection Tool. The Image Validation Collection Tool uploads Computer Information data to the Azure Database. This account does not have permission to log into the web portal.
3. WebUser:This account has read permission to the Web Administration portal. This account is unable to access User Administration Page.
4. PowerUser: This account has all the permission of the Validator plus it has read and update permission to the Web Administration Portal. This account is unable to perform any delete operations. Unable to access the User Administration Page.
5. Admin: This account has full permissions. It can perform any operation.

**UML Database Design Diagram:**

File: [UMLDiagram.jpg](file:///C:\Users\Administrator\Desktop\balsmiq\UMLDiagram.jpg)

The UML Database Design gives a rough draft of a proposed database design for the Image Validation Software System. This can be updated / corrected based on the Developers experience in Database design (normalization), Security and MVC Development.

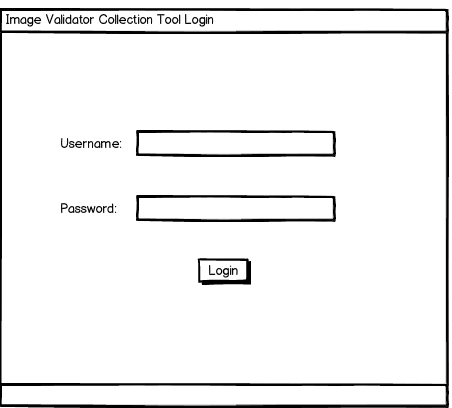
**GUI Design and Layout:**

**Login Page:**

**Page Summary:**

The Collection Tool has built in authentication that authenticates against the Azure database. The user logging in must have one of the security roles:

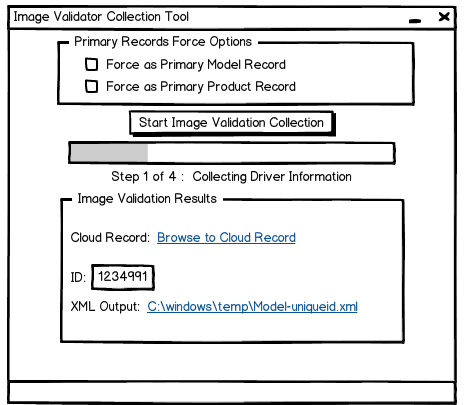
1. Admin
2. PowerUser
3. Validator

****

**Image Validation Client Tool Page:**

**Page Summary:**

This is the core of the Image Validation Client Tool Utility. It performs information collection on the Computer it is running on and uploads it to the cloud.

****

**GUI Information:**

**Force as Primary Model Record CheckBox**: If this CheckBox is checked it will automatically force the record that gets uploaded to the Azure Database to set the Computer Table IsPrimaryModel field to 1 (True). It will also check if any other record exists for the same model and ensure that the IsPrimaryModel is set to 0. Only one record can be set to 1.

Steps:

1. Check to see if that model exists already in the database and if it is set to a primary record.
   1. Select \* from Computer where IsPrimaryModel = 1 and Model=”CF-53JVUZK1M”
   2. If a record is returned update the record and set that record to 0
      1. Update Computer SET ISPrimaryModel = 0 Where Model=”CF-53JVUZK1M”
   3. Insert Computer Set IsPrimaryModel=1 etc… all the other information that needs to be inserted

**Force as Primary Product Record CheckBox**: If this CheckBox is checked it will automatically force the record that gets uploaded to the Azure Database to set the Computer Table IsPrimaryProduct field to 1 (True). It will also check if any other record for the same model is set to IsPrimaryProduct and set to 0. Only one record can be set to 1.

Steps:

1. Check to see if that model exists already in the database and if it is set to a primary record.
   1. Select \* from Computer where IsPrimaryProduct = 1 and Model=”CF-53JVUZK1M”
   2. If a record is returned update the record and set that record to 0
      1. Update Computer SET ISPrimaryProduct = 0 Where Model=”CF-53JVUZK1M”
   3. Insert Computer Set IsPrimaryProduct=1 etc… all the other information that needs to be inserted

**Start Image Validation Collection Button:**

This will start the collection routines to start collecting data and then eventually upload the information to the Azure Database.

**Progress Bar and Label Information:**

The Progress Bard keeps the user updated on which step the script is running. It goes through 6 primary steps.

1: User Clicks "Start Image Validation Collection" Button

2: Once Button is clicked a Progress Bar and label tracks the steps of the Image Validation

1. Collect Computer Information
2. Collect Driver Information
3. Collect Application Information
4. Collect Microsoft Hotfix Information
5. Save Final Output to XML File
6. Upload information to the Cloud

**Image Validation Collection Results Group**

This reports whether the upload was successful or failed. If it failed it would report the error. If it was successful it would report the following:

1. Direct Link to the record in the Administrator Portal
2. Computer ID Record:
3. XML Output File – All the information that is uploaded to the Azure database is also written to a local file.

**Database Collection Information:**

1. Collect Computer Information

|  |  |
| --- | --- |
| WMI Query Information | ("SELECT \* FROM Win32\_OperatingSystem", |
| SOURCE | **DESTINATION** |
| WMI WIN32\_Win32\_OperatingSystem | **Azure Database Computer Table** |
| BuildNumber: | BuildNumber: |
| Caption: | Caption: |
| CSDVersion: | CSDVersion: |
| InstallDate: | InstallDate: |
| MUILanguages: | MUILanguages: |
| OSArchitecture: | OSArchitecture: |
| OSLanguage: | OSLanguage: |
| OSProductSuite: | OSProductSuite: |
| OSType: | OSType: |
| ServicePackMajorVersion: | ServicePackMajorVersion: |
| ServicePackMinorVersion: | ServicePackMinorVersion: |
| SystemDirectory: | SystemDirectory: |
| SystemDrive: | SystemDrive: |
| Version: | Version: |
| WindowsDirectory: | WindowsDirectory: |

|  |  |
| --- | --- |
| WMI Query Information | ("SELECT \* FROM Win32\_BaseBoard", |
| SOURCE | **DESTINATION** |
| WMI WIN32\_Baseboard | **Azure Database Computer Table** |
| pRODUCT | Product |
| Manufacturer | Manufacturer |

|  |  |
| --- | --- |
| WMI Query Information | ("SELECT \* FROM Win32\_ComputerSystem", |
| SOURCE | **DESTINATION** |
| WMI WIN32\_COMPUTERSYSTEM | **Azure Database Computer Table** |
| Manufacturer | Manufacturer2 |
| Model | Model |

1. Collect Driver Information

Run a WMI Query on the Win32\_PnPSignedDriver Namespace and only store driver information for OEM drivers. These are drivers that start with a OEM file name prefix (E.g OEM1.inf, OEM2.inf, OEM3.inf, etc) Use a query that searches for only OEM\*.inf files.

|  |  |
| --- | --- |
| WMI Query Information | ("SELECT \* FROM Win32\_PnPSignedDriver WHERE infname LIKE OEM\*.inf", |
| SOURCE | **DESTINATION** |
| WMI Win32\_PnPSignedDriver | **Azure Database Driver Table** |
| CompatID | CompatID |
| Description | Description |
| DeviceClass | DeviceClass |
| DeviceID | DeviceID |
| DeviceName | DeviceName |
| DriverDate | DriverDate |
| DriverProviderName | DriverProviderName |
| DriverVersion | DriverVersion |
| FriendlyName | FriendlyName |
| HardWareID | HardWareID |
| InfName | InfName |
| IsSigned | IsSigned |
| Manufacturer | Manufacturer |
| Name | Name |
| PDO | PDO |
| Signer | Signer |

1. Collect Application Information

|  |  |
| --- | --- |
| Registry: |  |
| x64 OS Registry Keys: (These Registry KEYS MUST BE USED on X64 OS)  HKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Uninstall  HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows\CurrentVersion\Uninstall  x86 OS Registry Keys:  HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows\CurrentVersion\Uninstall | |
| SOURCE | **DESTINATION** |
| REGISTRY KEYS ABOVE | **Azure Database Application Table** |
| DisplayName | DisplayName |
| DisplayVersion | DisplayVersion |
| Publisher | Publisher |
| VersionMinor | VersionMinor |
| VersionMajor | VersionMajor |
| Version | Version |
| HelpLink | HelpLink |
| HelpTelephone | HelpTelephone |
| InstallDate | InstallDate |
| InstallLocation | InstallLocation |
| InstallSource | InstallSource |
| URLInfoAbout | URLInfoAbout |
| URLUpdateInfo | URLUpdateInfo |
| Comments | Comments |
| AuthorizedCDFPrefix | AuthorizedCDFPrefix |
| Contact | Contact |
| EstimatedSize | EstimatedSize |
| Language | Language |
| ModifyPath | ModifyPath |
| Readme | Readme |
| UninstallString | UninstallString |
| SettingsIdentifier | SettingsIdentifier |

4. Collect Microsoft Hotfix Information

|  |  |
| --- | --- |
| WMI Query Information | ("SELECT \* FROM Win32\_QuickFixEngineering |
| SOURCE | **DESTINATION** |
| Win32\_QuickFixEngineering | **Azure Database Driver Table** |
| CSName | CSName |
| Description | Description |
| HotFixID | HotFixID |
| InstallDate | InstallDate |
| InstalledBy | InstalledBy |

**Image Validation Collection Results Group**

**Error Results:**

1. Create errors for the following
   1. Network connection failures
   2. Information Collection Errors
   3. Validation errors with Azure Database
   4. Upload errors

**Success:**

1. Cloud ComputerID Record: Illustrate the ComputerID Record in a TextBox
2. A Direct Link to the ComputerID record in the Administration Portal.
3. Output a local XML file with all the information collected.
   1. Save the file to C:\Windows\Temp