Upgrading VTS / VTScada Applications to Version 10

What You Need to Know

Many features have been completely re-written for VTS10. Most applications that functioned properly under previous VTS versions will continue to do so under VTS10. However, applications with customized scripting should be extensively tested to verify functionality.

Please review the following information before updating to VTS 10. For more details consult the help files at the bottom of the VTS Application Manager (VAM) or by hitting F1 within any VTS screen or dialog.

- **Backup the Application -** Prior to upgrading, backup the entire application directory on the primary configuration server. If you have saved any custom graphics to the VTS layer (C:\VTS\VTS\BitMaps), copy these into your application layer (C:\VTS\YourApplicationName\BitMaps) prior to backup. Otherwise, they will be overwritten.
- **Upgrade Primary Servers First** - If upgrading a networked application (i.e. runs on multiple workstations), upgrade the primary configuration server first, and then all clients. All computers must be upgraded to the same VTS version.
- **No More .INI Configuration Files** - Configuration variables are no longer stored in Config.INI, SecurityManager.INI, workstation.INI, etc. They are now stored in Settings.Startup (loaded only on application start) and Settings.Dynamic (can be reloaded while an application is running).
- **The Term "Configuration Variables" Replaced with "Application Properties"** - Properties can now be edited from the GUI via the Application Configuration interface. **(1 above)**
- **No Recompile Button** - Before version 10, users needed to commit real-time configuration changes using the recompile button. With VTS 10, changes are compiled automatically as they are made. 'Recompile' has been replaced by the 'Import File Changes' button **(2)** which commits and distributes imported files such as custom bitmaps.
- **User Files Now Separate from Working Files** - The files located in an application's root directory and in the Pages directory are "user files". Developers may edit these files, but must click 'Import File Changes' **(2 above)** before their edits become part of the working files.

  Working files are under version control and are stored in a hidden system folder within the application directory. Any attempt to directly edit a working file will damage your application.
- **Auto Deploy Enabled by Default** - VTS now includes an Automatically Deploy feature, which automatically sends application changes in real-time to all computers in a networked application. This is different from previous VTS versions, where the user was required to deploy changes using the Update All feature.

  Important: Automatically Deploy is turned on for all upgraded applications - If this is not preferred, refer to the VTS Developers Guide **(3)** and search for the VTS Version Control System.
- **Check that Optional Components are Enabled -** After upgrading, ensure that any optional components that you have licensed (e.g. VTS ODBC Server, VTS Internet Client) are still enabled by clicking About VTS **(4 above)**.
- **Version Upgrades -** Each VTS license entitles the owner to version upgrades for a period of time after purchase. The 'About VTS' button **(4)** will also show the end date of the paid upgrade period.
- **Application Template Directories Obsolete** - Template information is now stored within VTS Changeset files.

- **Points.MDB Depreciated** - Points.MDB has been replaced with a VTS database format. Database import/export tools in the Application Configuration Interface **(1)** enable developers to work with tag properties in Microsoft Access™ or Excel™ for offline editing and bulk import.
- **LogManager Service Obsolete** - Log storage is now handled by Historian tags and a new VTS Historian. One Historian tag is created by default for each application but an application can now have any number of Historians and Oracle, MSSQL, MySQL and Access Historians are also supported.
- **Accessing Previously Logged Data** - Because of changes to how logged data is stored, you must configure the application property, LegacyHistoryPath to point to data logged while the application was running on an earlier version.
- **If You Are Using RateOfChange Tags** - The Source Tag parameter must refer to a tag that is configured for logging.
- **Application Property OEMRestrict No Longer Supported in the VAM.**
- **For networked Applications, 'RPCManager-Inhibits Configuration' Section is Obsolete.**
- **RPCService.INI file Automatically Converted to VTS10 Format** - Note: you may need to re-enter your server lists.
- **Clipped Text in VTS Internet Clients** - If using VTS10 as a VTS Internet Server on Windows™ Vista/7/Server2008, any VTS Internet Clients that are run on Windows XP/Server2003 or older operating system will have many texts clipped. To avoid this, the server must be told to use old fonts by setting UseXPCompatibleFont to 1 in the System section of Setup.INI. If the server is XP/Server2003, the setting is not necessary.
- **Configuration Variables from Setup.INI Now Read as the Lowest Priority Settings for all Applications** - They will always be over-ridden by a matching property in the application's Settings.Startup or Settings.Dynamic.
- **These script applications are obsolete and no longer included:** ResetRemoteClients, DBConvert, ODBCBrowse

Additional Information for Programmers

- In 9.1 and earlier, if you included code for another module in the same source file as the Appmod.SRC, it would be used. Group modules were typically done this way. As of 10.0, the code for other modules must exist in a separate file other than AppRoot.SRC or it will be ignored without warning.
- All application-level configuration files (those with names ending in .INI) have been replaced with the two files, Settings.Startup and Settings.Dynamic. Workstation-specific configuration files have been replaced by Workstation.Startup and Workstation.Dynamic in the WorkstationSettings folder of the application directory.
- StartTag has a new flag which, if set, will update the tag database. By default, this flag is not set.
- For custom tag types, ensure that each tag's parameter metadata is in place. (May not exist in legacy applications.)
- The SecurityManager is now in the VAM layer and overrides to it will not take effect for VAM access to security.
- SecurityManager RPC service is obsolete. Security accounts and settings now synchronized by Configuration service.
- AppRoot.SRC replaces AppMod.SRC & AppMod.Web. The AppMod file in an application that is upgraded to release 10 will not be used, but will be retained for use if the application is converted back to an older version.
- PlugIns that use default string values (as shown in the following example) will no longer work.

```
[ (POINTS)
MyTag = "VarForGraphicsInMyTag";
]
```

- EditLockoutManager functions such as "MarkTagForEdit" are now obsolete. The distributed version control system replaces the Edit Lockout Manager. Any custom code that calls the EditLockoutManager will result in an error dialog.
- Libraries no longer combine code across layers and now only use the library at the highest defined layer. Drawing methods that link into libraries are not affected.
- Web services have a different script code interface. ExternalValue is no longer supported in input tags.
- OEM code references to Logger, LogManager or LogObjVar need to be changed to use the Historian interface. Of special note is any code that waits for LogManager\Started.

- Modules defined outside the scope of an application directory will need to be moved to within the application directory.
- Security Manager plug-ins only work when the application is running.
- Plugins which have references to variables in Code must be preceded by \Code.
- Template directories must be converted manually to template ChangeSets.
- DSNName no longer exists as a variable in Code
- Security Manager components OEMEncryptKey and SerEncryptOEM replaced by integrated higher level encryption.
- The Notebook tag's AddNote module interface has changed to support the new Historian.
- SelectObject and PSelectObject have a new parameters, adding new options for your code.
- SQL module calls no longer allow writing to the default tag database.
- EditIni and EditIniCheckbox library drawing methods always update the RAM copy and ignore the "Update RAM Copy" parameter.
- RPCManager\Register no longer supports specifying a file name to read the server list from.
- RPCService.INI file contents have been transferred to Servers.RPC.
- ToolExt.INI has been changed to ToolExt.CSV.
- WriteIni first acquires the working copy lock and update the file asynchronously. Use Layer\RecordProperty instead.
- ReadIni does not acquire the working copy lock. It is better to use ReadPropertiesFile instead.
- LogFileName PLUGIN no longer supported.
- LogAlarm PLUGIN no longer supported.
- .DAT and .LOG files accessed from custom code need to be moved to the DataPath directory and the code modified.
- The default window title will now be the application name rather than "Display".
- The RPC manager now uses the VTS IANA registered port, 5780 instead of 1160.
- The application property LegacyHistoryPath required in order to access older data from upgraded legacy applications.
- The RPCManager-Inhibits configuration file section is no more.
- VTS Internet Servers on Vista with clients running XP will need to set the Tahoma font in their Setup.INI files.
- OEMRestrict is no longer supported on the VAM.
- Tag names that consist of only the period and space characters will no longer be considered valid.
- RPCService.INI files will not be converted to the VTS 10 format automatically. Server lists may need to be re-entered.
- Setup.INI variables are now read as the lowest priority settings for all applications.
- After rolling back an upgraded application to a previous version, you will need a full re-compile in the previous version.
- Unless AutomaticDeploy is added to the Layer section of Config.INI prior to conversion, all local changes will be deployed automatically when the application is converted to version 10.
- Applications that use VTS as a DDE server and rely on the application window being called "Display" need to either:
  - Declare DisplayManagerTitle = Display in Settings.Dynamic or
  - Update the links to refer to the application name.
- Updates to ODBC to use ODBC3.0 drivers may cause changes in returned data types and SQLState return codes.
- Changes to ODBCStatus function to take the ODBC handle to query for status. Otherwise, you will just get status of the last operation to complete which, given the concurrent nature, might not be the operation you have just executed.