



Upgrading VTScada

What You Need to Know

Your VTScada license entitles you to upgrades for a period of time after purchase. (*Maintaining a Support Plus contract provides unlimited upgrades.*) In each new version of VTScada, new tools are introduced and will sometimes change how older features work. Most applications can be upgraded by simply downloading and running the installer for the most recent version of VTScada. Applications should continue to function normally (applications with customized scripting should be extensively tested).

Notes: Please review the list of changes for each version of VTScada between the one you are upgrading from and the current one to find additional tasks that you may need to perform. To consult the VTScada Help files click the [?] at the bottom of the VTScada Application Manager or hit F1 from any VTScada screen or dialog. In all cases, create a backup of your application before starting. In general, it is better to install a new version on top of an older version.

General Upgrade Procedure:

In a networked system, upgrade one workstation at a time, starting with the primary server.

1. Shut down VTScada.
2. Run the installation program, installing to the existing folder.
3. Restart VTScada and run your application. Confirm that the application runs as expected.
4. Proceed to the next workstation.

Upgrading to VTScada 11

- Windows XP™ is no longer supported.** Version 11 requires features that are available only in more modern operating systems.
- The VTScada layer has been fully merged into the standard layer.** The legacy VTScada layer still exists as a hidden layer, and will be used by legacy VTScada-based applications that are being updated to version 11. One widget (Duplexes) and one font (Notes) have not been merged into the standard layer, but can be found in the legacy layer if required.
- If you have an OEM layer, and if that layer has custom widgets** (formerly called "drawing methods"). You must run the OEM layer once in version 11 before running applications that use it. This will happen automatically, even if the OEM layer is flagged as "never run." This is required to allow VTScada 11 to create menu item tags for the widgets, thus making them available within the palette. The OEM layer should not be left running. When started automatically, it will also be stopped automatically.
- The Call-Out List page is no longer included.** Any page, of your own creation, may be used in its place. Legacy applications that have the Call-Out List page will continue to have that page when imported into VTScada 11.
- Many legacy images have been dropped from the palette in favor of modern images.** The original image files are still included for the sake of backward compatibility.
- The Mobile Browser interface has several improvements, including the ability to show page graphics.** With this new functionality, the mobile client is now licensed as being equivalent to an Internet Client (VIC) connection. Customers who need to limit page graphics in order to conserve bandwidth or server CPU may set the application property, MobileBrowserDisablePageGraphics. Existing customers who wish to continue using the older, limited mobile interface and its matching license calculation should contact Trihedral's Technical Support for relevant information.
- The Internet Client Monitor log is now stored in the file VICMonitorLog.txt.** This is located in the installation folder instead of Log.txt located in the BrowserMon folder.
- Hand-coded, custom widgets may not look or work correctly until they have been opened in the Idea Studio.** Click the widget's Set Parameters button, within its Menu Item tag's configuration folder, then click OK. The recommended method for defaulting the parameters of a custom-coded widget is to add default values to the parameters in the widget's .SRC code. Alternatively, a GetParmDefaults subroutine could be added to the widget, which will return an array of parameter defaults.
- Pipes are no longer drawn as a separate function (GUIPipe).** A variant of the Line (GUIPolygon) command is used instead. Pipes in legacy applications will be converted automatically when either their color or their width is changed. Legacy pipes are not converted upon import, or when moved or re-aligned.
- The lexicon file, Lexicon.vlx, will not be copied from your older installation during the upgrade process.** If you are installing to a new folder, and you have made changes to your lexicon that you want to keep, you must copy the file Lexicon.vlx from the old installation to the new.

- Four font files will be installed with VTScada.** These are for use by widgets and built-in text styles. Review labels and titles in your application pages for scaling and alignment issues after upgrading to version 11. The new fonts are:
 - Open-Sans
 - Sansation
 - Crystal
 - Register
- The Gradient Color Change is deprecated.** This was used to make pipes change color according to pump or valve state, but GUIPipes are no longer created. To replace this feature, link the color property of the pipe (or any other object) to an expression that monitors the value of the tag that was monitored by the gradient color change. For example, given a digital status tag, "Motor1_Status" and a color that should be green when on, red when off, the expression would be: [`<Motor1_Status>`] ? "`<FF00FF00>`" : "`<FFFF0000>`"
- The \Editing flag is no longer used.** The flag has been replaced by `ParentWindow(Self)\Editing`. All scripts that used the former version must be updated.
- The toolbox has been replaced by ribbons.** Custom toolbox buttons are no longer available.
- The F11 / F12 key.** These now open the FindPage dialog in all applications.
- Custom driver code.** Code that used the group name, "VTScadaDriver" must now use the group name "LiftstationDrivers".
- Changes to default property values:**
 - AnswerCalls - Now defaults to True
 - AlarmLogFreq - Now defaults to monthly
 - OperatorNotesSecurity - Defaults to none
 - SiteRetries - Defaults to 3
 - ModemManagerLogSize - Defaults to 1000
 - AlarmEventDesc labels - Default labels are those used by VTScada rather than VTScada prior to version 11
- The following properties are obsolete:**

AlarmEventDescX	AlmDBPointName	Cycles
AlarmRevUnack	AlmDBPriority	DataFlowModuleName
AlarmStateDesc0 through AlarmStateDesc5	AlmDBTimeStamp	HTTPWAPport
AlmColumn1 through AlmColumn7	AlmDBStatus	ReportXPos
AlmDBArea	AlmDBType	ReportYPos
AlmDBHPUnits	AlmHdg1 through AlmHdg7	ReportXSize
AlmDBHPValue	AlmPgLineStyle	ReportYSize
AlmDBMessage	AlmTagsOnly	TrendDuration
	ClientAlarmSoundOn	

Upgrading to VTScada 10.2

- Tags created in 10.2 will not run properly in 10.1 - In order to allow for the renaming of tags, 10.2 assigns tags a "UniqueID" and a few other new internal parameters in the .tag files. 10.1 applications cannot understand UniqueIDs which are made up of 6 to 11 characters that include letters, digits, and special symbols. You can export tags to an Excel spreadsheet and manually modify them to be 10.1 compatible. However, page references, history, etc., would have to be reworked as well.
- 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 VTScada layer (C:\VTScada\VTScada\BitMaps), copy these into your application layer (C:\VTScada\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), all computers must be upgraded to the same version. To do this, first upgrade the primary configuration server. Then, on each of the other computers, manually remove the application folder from the VTScada folder, update VTScada to 10.1, and then retrieve the updated VTScada application from the primary by selecting 'Add New' and then 'Get from Server'. This ensures that VTScada Application Version Control will continue to synchronize correctly. If the server is a log manager, copy the history folder from the original application folder into the new application folder.
- StartTag Deprecated - When creating 'Child Tag' structures, you should now only use the VTScada Tag Browser rather than write code.
- 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.
- No Recompile Button - Before version 10, users needed to commit configuration changes using the recompile button. With VTScada 10, changes are compiled automatically as they are made. 'Recompile' has been replaced by the 'Import File Changes' button 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' 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 - New Automatically Deploy feature automatically sends application changes in real-time to all computers in a networked application. In previous VTScada versions, the user had 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 VTScada Developers Guide and search for the VTScada Version Control System.
- Check that Optional Components are Enabled - After upgrading, ensure that any optional components that you have licensed (e.g., VTScada ODBC Server, VTScada Internet Client) are still enabled by clicking About VTScada.
- Version Upgrades - Each VTScada license entitles the owner to version upgrades for a period of time after purchase. The 'About VTScada' button will also show the end date of the paid upgrade period.
- Application Template Directories Obsolete - Template information is now stored within VTScada Changeset files.
- Points.MDB Deprecated - Points.MDB has been replaced with a VTScada database format. Database import/export tools in the Application Configuration Interface let you 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 VTScada 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.
- 'Storage Limiting Settings' for logging - VTScada allows you to setup a logger within the configuration dialog of every input tag. Before version 10, users who needed to limit the number of records saved could do so within this dialog. However, when upgrading to 10.1, these loggers are automatically converted to VTScada Historian Tags which keep all logged data by default. After you upgrade, it is important to re-add these settings in each VTScada Historian Tag; otherwise, your storage drive may fill sooner than you anticipated. If this applies to you, we recommend you contact Trihedral before upgrading.
- Accessing Previously Logged Data - Due to changes in how VTScada stores logged data, you must configure the application property, LegacyHistoryPath, to point to the data logged by the earlier version of VTScada.
- If You Use 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.
- 'RPCManager-Inhibits Configuration' Section is Obsolete - For networked applications.
- RPCService.INI File Automatically Converted to VTScada10 Format - You may need to re-enter your server lists.
- Clipped Text in VTScada Internet Clients - If using VTScada10 as a VTScada Internet Server on Windows® Vista/7/Server2008, any VTScada Internet Clients that are run on Windows XP/Server2003 or older operating system will have many texts clipped. To avoid this, set the server to use old fonts by setting UseXPCompatibleFont to 1 in the System section of Setup.INI. Not necessary if server is XP/Server2003.
- Configuration Variables from Setup.INI Now Read as the Lowest Priority Settings for all Applications - They will always be overridden by a matching property in the application's Settings.Startup or Settings.Dynamic.
- These script applications are obsolete - ResetRemoteClients, DBConvert, ODBCBrowse

Additional Information for Programmers

New in 10.2

- Character "]" No Longer Legal in a Tag Name - Existing tags that used this character will display using their immutable name rather than the short name. You can mitigate this by changing TagNameDelimiter in Setup.INI to use an alternative character. ">" is recommended. Short names will then have the form [ShortName> rather than [ShortName].
- Searching-and-replacing tag names in a page - While we no longer recommend this practice, if you copy graphics (including all graphics on a page using Ctrl-A) and paste the code into a text editor, the pasted text will be produced using the friendly names in the form [Digester\Pump1] instead of the UniqueID reference in the page SRC file. Pasting the text editor code back to the page will automatically translate the code back to the UniqueID.
- Code for tag handling now has a parameter named "DisplayName" - Any user-defined tags that have a parameter named, "DisplayName" will not update properly. You should change the name of that property in your tag's code.
- GetSystemColor() and option 10 of VStatus() now return an RGB color string instead of a palette number - Custom code that uses this value as a number will need to be updated to use text instead. Code that simply passes the value to other VTScada functions will continue to work.
- These functions have been removed. Code using them will compile and compute values, but functions will do nothing.

AnimateState,	HighlightState	ModuleCollapsed	MoveSibling	PickState	SetStateColor
CollapseTree	HighlightTree	Moduletree	MoveState	ReadNum	ShadowTree
DragState	LastSelectedModule	ModuleTreeSize	Palette	ReadText	StateDiagram
HighlightedModule	LastSelectedState	MoveSelState	PickModule	SelectStates	StateHighlighted

New In 10.1

- 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 apps.)
- 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/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 and replaced by the distributed application version control system. 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 the application directory.
- Security Manager plug-ins only work when the application is running.

- Plug-ins 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 high level encryption.
- The Notebook tag's AddNote module interface has changed to support the new Historian.
- SelectObject and PSelectObject have 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 RAM copy and ignore "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 updates 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 by 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 VTScada IANA registered port, 5780 instead of 1160.
- The application property LegacyHistoryPath required to access older data from upgraded legacy applications.
- The RPCManager-Inhibits configuration file section is no more.
- VTScada 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 VTScada 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 to re-compile the rolled back 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 11.
- Applications that use VTScada as a DDE server and rely on the application window being called "Display" need to:
 - 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 just executed.

Try It for Yourself

Download the 90-day Trial

Trihedral.com/trial