

## About FactoryWiz Monitoring Custom Apps

RYM designed FactoryWiz Monitoring from the ground up to allow for the integration of custom applications. These applications may be written by the end user, by RYM for the end user, or by a 3<sup>rd</sup> party.

Common functions involving data entry or returning real time data can be performed easily using Microsoft Visual Basic .NET or C# by adding a web reference to one of two web services on the system – one for writing data and another for returning a real time snapshot of information. We refer to these web services as the “web API”. Only basic programming knowledge is required to use these functions, since all of the code to access the databases and keep things in order has been performed for you.

Access to the volumes of historical data is better performed with SQL commands, so we place all of our data in a standard MySQL database that can be accessed remotely either with a native MySQL interface or via an ODBC driver.

At the most intimate level, functionality of the various monitoring services can be altered by having RYM provide a custom DLL to your specifications. By placing your custom code in a separate DLL, you will be able to update your FactoryWiz Monitoring with the standard updates available off the RYM web site without overwriting your special features. Once the DLL is written, it is just a matter of overwriting the default “blank DLL” that does nothing with the custom DLL, restarting the system, and your new functionality will become active.

## What can be done via web API?

Examples of applications that could be written using the web API:

- Custom user interface to allow for operator entry of data, such as on a touch display
- Data collection service for an uncommon type of machine tool
- A program that displays data from FactoryWiz Monitoring in real time, such as in the Windows task bar or on a large plasma display
- Custom logic that automatically updates the machine mode (such as to ‘setup’ or ‘production’) based on data received from a PLC.
- A link between the status of the CNCs in the shop and a shop scheduling system
- Create a solution for automatic operator login or logout based on time clock output, RFID badge, or other identification.

## What specific functions are available in the web API?

<http://factorywiz/interface/displaypanel.asmx>

Generally used for retrieving real-time status information from FactoryWiz Monitoring.

Method	Description
<i>GetAllStatuses</i>	Returns all of the real time status information for all CNCs, including custom fields and everything available on the real time status page
<i>GetSingleStatus</i>	Returns all of the real time status information for a single, specified CNC, including custom fields and everything available on the real time status page
<i>GetFilteredStatuses</i>	Similar to <i>GetAllStatuses</i> , except results can be filtered by CNC name, status, CNC groups, active mode, active idle reason, and active part ID.
<i>GetShopSummary</i>	Returns basic figures such as the in cycle minutes, idle minutes, and common idle reasons for the entire shop, for a time period you specify
<i>GetShopSummaryShift</i>	Returns the same information as <i>GetShopSummary</i> only for a single shift
<i>GetMachineCount</i>	Returns the number of machines active
<i>GetMachineList</i>	Returns an array of strings with the names of every CNC

Method	Description
<i>GetGroupList</i>	Returns an array of strings with the name of every CNC Group
<i>GetPartIDList</i>	Returns an array of strings with every Part ID every used
<i>GetFilteredPortIDList</i>	(Optionally, limit results to a certain number)

<http://factorywiz/interface/dataentry.aspx>

Used for updating data in FactoryWiz Monitoring. All methods require a specific CNC to be specified.

Method	Description
<i>UpdateLink</i>	Updates the reason a CNC is currently not running (not functional while in cycle)
<i>UpdateLink</i>	Updates the link and description showed on the real-time status screen. Useful for webcams.
<i>UpdateMode</i>	Updates the CNC operating mode, such as to maintenance, setup, production, etc.
<i>UpdatePartCount</i>	Updates the part count shown in real-time status
<i>UpdatePartID</i>	Updates the currently running part name
<i>UpdatePictureURL</i>	Changes the URL specified for the image displayed on the real-time status screen
<i>UpdateRapidOverride</i>	Updates the rapid override rate %
<i>UpdateSequenceNumber</i>	Updates the currently executing N-number
<i>UpdateSpindleOverride</i>	Updates the spindle override rate %
<i>Update Status</i>	Updates the current CNC status, such as to Running, Paused, Idle, Alarm, Disconnect, etc.
<i>UpdateWatchedValues</i>	Updates a currently watched values field displayed in the real-time status. Useful for macro variables and ladder signals

Method	Description
<i>AddCycle</i>	Adds a new part cycle to the database
<i>AddEvent</i>	Adds a new event, such as an alarm, to the database
<i>AddIdle</i>	Adds a new idle time period to the database
<i>GetCustomText1</i>	Returns the current custom field #1 value for a particular CNC
<i>GetCustomText2</i>	Returns the current custom field #2 value for a particular CNC
<i>GetCustomText3</i>	Returns the current custom field #3 value for a particular CNC
<i>GetMachineMode</i>	Returns the current mode for a particular CNC
<i>GetMachineModeAndTime</i>	Returns the current mode for a particular CNC and the time it took effect
<i>GetPermissibleMachineModes</i>	Returns an array of strings with the predefined modes which an operator may select from
<i>GetIdleReason</i>	Returns a CNC's current idle reason
<i>GetPermissibleIdleReasons</i>	Returns an array of strings with the predefined idle reasons which an operator may select from
<i>GetMachineGroups</i>	Returns an array of strings with the names of every CNC group
<i>GetPartCount</i>	Returns a CNC's current part count
<i>OperatorLogin</i>	Changes which user is operating a CNC
<i>OperatorLogout</i>	Specifies that no operator is present
<i>UpdateAlarms</i>	Updates the list of currently displayed alarms
<i>UpdateBlockNumber</i>	Updates the currently executing raw block number
<i>UpdateCurrentTool</i>	Updates the currently used tool number
<i>UpdateCustomText1</i>	Updates a text field reserved for user data for the current cycle/idle
<i>UpdateCustomText2</i>	Updates a text field reserved for user data for the current cycle/idle
<i>UpdateCustomText3</i>	Updates a text field reserved for user data for the current cycle/idle
<i>UpdateFeedOverride</i>	Updates the feed override

### SQL Database Tables

The two tables of greatest interest are called *CNC\_Status* and *CombinedTimes*. *CNC\_Status* is constantly updating with the real-time status information for every CNC. *CombinedTimes* updates every time there is a new cycle, idle period, or alarm period recorded, differentiated by a value in the TYPE column of "CYCLE", "IDLE", or "NOTIFY" respectively.

It is interesting to note that you do not need to be familiar with SQL to access this data. Users who install the MySQL ODBC driver on their computer may set up a data source pointing to FactoryWiz Monitoring. Then, in a program such as Microsoft Excel or Microsoft Access, that data source can be browsed and used for reports and queries as though it were native Excel or Access data.

Additional tables may be added by the end user as desired, but as a matter of naming convention please prefix the names of any user tables with *EU\_* (EU followed by underscore to denote end user). This will ensure that no future FactoryWiz Monitoring tables will conflict with the name you choose. If you are a third party integrator, please prefix your table names with *TP\_* (TP followed by underscore to denote third party).

### Custom DLLs

Custom DLLs can be used for changing just about any behavior on the data collection side of the RYM data collection services. Examples include a DLL to plug into the ADAM collection module to interpret several signals from multiple ADAMs and combine them into a numeric value or to indicate some custom condition and then perform a custom action immediately. With a custom DLL for the FOCAS module you could send data into the CNC and update custom macro variables on the fly as the CNC runs.

### Custom Web Pages

Changing the default FactoryWiz Monitoring web pages would result in your changes getting overwritten when new updates are applied, so two reserved port numbers are available for separate web apps to be loaded on FactoryWiz Monitoring for added functionality. User custom web pages will go into a dormant web server that can be turned on at port 8080. Third party custom web pages will load to a dormant web server on port 8081. By placing files named 8080.jpg or 8081.jpg, respectively, in the root directory for those web sites, icons will become visible from the main FactoryWiz Monitoring menu for your custom sites.

### RYM Engineered Solutions

Nobody knows your FactoryWiz Monitoring better than RYM, so if you would rather not make custom FactoryWiz Monitoring apps yourself, contact us and we would be happy to provide a quote for your special need.

### About this publication

This guide was produced as a basic introduction to the methods available for creating custom solutions based on FactoryWiz Monitoring. The concepts described here are general descriptions and not meant to be exhaustive. If you have questions about a topic not covered here you are always welcome to call us and ask about your application.

### About RYM

Refresh Your Memory, Inc has been networking CNC machine tools and providing machine/event monitoring solutions since 1983. FactoryWiz Monitoring and FactoryWiz DNC, developed completely in-house, are the result of decades of hand's-on experience in shops

across the country. The company now has coast-to-coast technical support and an expanding management team to handle our increasing customer base around the world.

© 2014 Refresh Your Memory, Inc

### Contact Us

Refresh Your Memory, Inc.

6613 San Anselmo Way San Jose, CA 95119

Toll Free: (866) 796-4362

Outside USA: +1 (408) 224-9167

Or visit us online at [www.factorywiz.com](http://www.factorywiz.com)