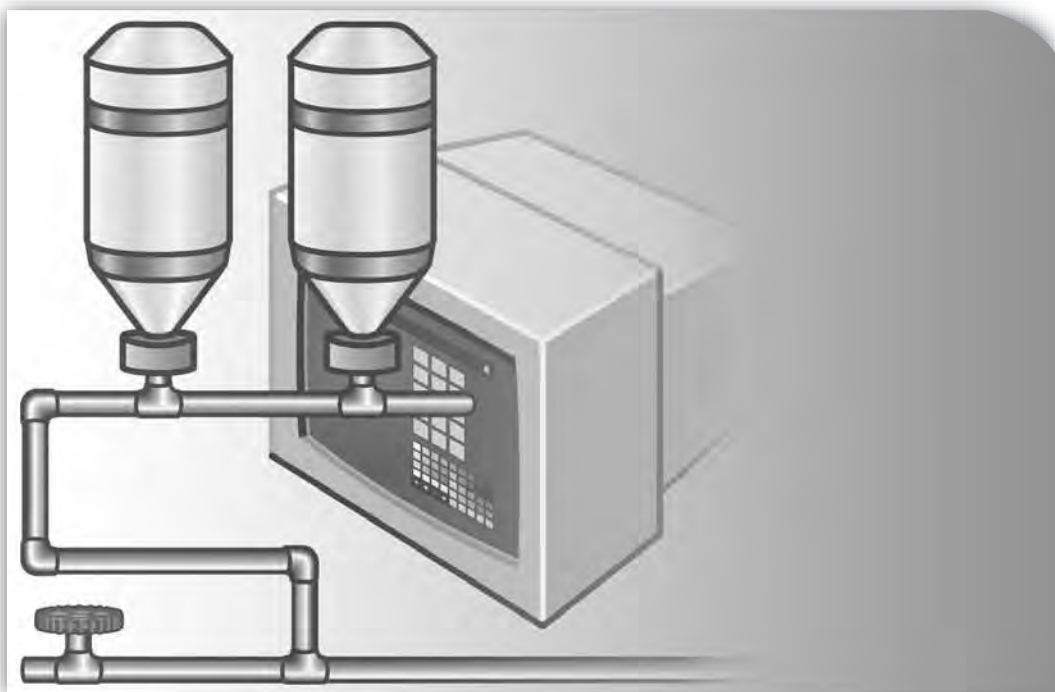


LISTEN.
THINK.
SOLVE.SM

RSView[®]32



RUNTIME USER'S GUIDE

PUBLICATION VW32RT-UM001F-EN-E-July 2011
Supersedes Publication VW32RT-UM001D-EN-E

Contact Rockwell

Customer Support Telephone — 1.440.646.3434
Online Support — <http://support.rockwellautomation.com>

Copyright Notice

© 2011 Rockwell Automation Technologies, Inc. All rights reserved. Printed in USA.
This document and any accompanying Rockwell Software products are copyrighted by Rockwell Automation Technologies, Inc. Any reproduction and/or distribution without prior written consent from Rockwell Automation Technologies, Inc. is strictly prohibited. Please refer to the license agreement for details.

Trademark Notices

Allen-Bradley, ControlLogix, FactoryTalk, Rockwell Automation, Rockwell Software, RSLinx, RSView and the Rockwell Software logo are registered trademarks of Rockwell Automation, Inc.

The following logos and products are trademarks of Rockwell Automation, Inc.:
AdvanceDDE, ControlView, Data Highway Plus, DH+, RSView32, RSView32 Resource Kit, and WINTelligent.
FactoryTalk Activation, FactoryTalk Live Data, and FactoryTalk Transaction Manager.
RSLinx Classic, SLC 5, SLC 500, and SoftLogix.

Other Trademarks

ActiveX, Microsoft, Microsoft Access, SQL Server, Visual Basic, Visual C++, Visual SourceSafe, Windows, Windows ME, Windows NT, Windows 2000, Windows Server 2003, Windows XP, and Windows Vista are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
Adobe, Acrobat, and Reader are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.
ControlNet is a registered trademark of ControlNet International.
DeviceNet is a trademark of the Open DeviceNet Vendor Association, Inc. (ODVA).
Ethernet is a registered trademark of Digital Equipment Corporation, Intel, and Xerox Corporation.
OLE for Process Control (OPC) is a registered trademark of the OPC Foundation.
Oracle, SQL*Net, and SQL*Plus are registered trademarks of Oracle Corporation.
All other trademarks are the property of their respective holders and are hereby acknowledged.

Warranty

This product is warranted in accordance with the product license. The product's performance may be affected by system configuration, the application being performed, operator control, maintenance, and other related factors. Rockwell Automation is not responsible for these intervening factors. The instructions in this document do not cover all the details or variations in the equipment, procedure, or process described, nor do they provide directions for meeting every possible contingency during installation, operation, or maintenance. This product's implementation may vary among users.
This document is current as of the time of release of the product; however, the accompanying software may have changed since the release. Rockwell Automation, Inc. reserves the right to change any information contained in this document or the software at anytime without prior notice. It is your responsibility to obtain the most current information available from Rockwell when installing or using this product.

Contents

Chapter P

Preface

About the documentation.....	P-1
Technical support services.....	P-1
When you call.....	P-2

Chapter 1

Installing RSView32 Runtime

Determining system requirements.....	1-1
Step 1 – Total your points	1-2
Step 2 – Determine the recommended system configuration	1-3
Installing RSView32 Runtime	1-4
Installing RSView32 Runtime on Windows Server 2003, Windows XP, Windows 2000 and Windows Vista	1-4
Installation summary	1-5
Ordering RSView32 Runtime on disks.....	1-5
Activating RSView32 Runtime	1-5
FactoryTalk® Activation Grace Period	1-9
Uninstalling RSView32 Runtime	1-9

Chapter 2

Running a project

Setting up and running your project	2-1
Summary of steps.....	2-1
Step 1 – Moving the project.....	2-2
Step 2 – Specifying time, date, and number formats.....	2-2
Step 3 – Starting RSView32 Runtime	2-3

Starting RSVIEW32 Runtime manually.....	2-3
Starting RSVIEW32 Runtime automatically	2-3
Step 4 – Opening the project.....	2-5
Using the Runtime Project Manager.....	2-5
Step 5 – Setting up the channel, OPC server, or DDE server	2-6
Connecting to programmable controllers using direct drivers	2-6
Connecting to OPC servers and DDE servers	2-7
Step 6 – Changing the path to files outside the project directory.....	2-7
Step 7 – Running the project.....	2-9
Running a project manually.....	2-9
Running a project automatically	2-9
Adding files to a project	2-10
Changing data logging paths	2-11
Stopping a project	2-11
When a project stops.....	2-12

Chapter 3

Using security

Editing the users for a project	3-1
Logging in at runtime	3-2
Using the electronic signature button.....	3-3
Changing passwords at runtime.....	3-3

Chapter A

RSVIEW32 commands

Index

P

Preface

RSView32™ Runtime is a package for running automation applications developed in RSView32 Works. RSView32 Runtime runs on Microsoft® Windows® Server 2003, Microsoft Windows XP, Windows 2000, Windows Vista, Windows 7, and Windows Server 2008.

About the documentation

The RSView32 Runtime documentation includes:

- Help, which contains procedures and reference information.
- Readme file, which contains additional information about RSView32. Read this file before you begin working with the software.

Technical support services

If you have questions about RSView32 Runtime, please consult the user's guide or the Help.

If you cannot find the answer, contact Rockwell Software Technical Support at:

Telephone: 440-646-3434

Internet Support: www.rockwellsoftware.com **or**
www.rockwellautomation.com

Support staff are available Monday to Friday from 8 A.M. to 5 P.M. Eastern Standard Time, except during holidays.

When you call

When you call, be at your computer and prepared to give the following information:

- the product serial number

You'll find this number on the Activation disk label and in the Help About dialog box that you access from the RSVIEW32 Project Manager.

- the product version number
- the type of hardware you are using
- the exact wording of any messages that appeared on your screen
- a description of what happened and what you were doing when the problem occurred
- how you tried to solve the problem

Installing RSView32 Runtime

This chapter describes:

- hardware and software requirements
- how to install and activate RSView32™ Runtime
- how to uninstall RSView32 Runtime

Determining system requirements

The hardware and software you use with RSView32 Runtime depends on the demands your project places on the system. The greater the demand, the more powerful a system you need.

The recommendations below are based on field experience. It is possible that your application will operate on a platform below these recommendations.

In the following tables, the demands your project places on the system are represented by points. Total your points in step 1, and then use the table in step 2 to determine the hardware and software you need.

Step 1 – Total your points

For	If you are	Add points
RSView32 project	running displays, or editing your project using RSView32 Works	2
	using more than 30,000 tags	2
	using more than 5,000 tags	1
	changing displays, on average, more than 5 times per minute	1
	using more than 200 objects in displays	2
	using ActiveX® controls in displays	2
Alarms	monitoring more than 1,500 tags for alarms	2
	monitoring more than 300 tags for alarms	1
Active Display System, DDE, or OPC® simultaneously active clients	using typically more than 5 active clients	2
	using at least 1 client	1
Data log or Historical trends	logging less than 100 tags in 1 model	1
	logging between 100 and 1,000 tags in 1 model	2
	logging to more than 1 model	2
Event detector, derived tags, or real-time trends	using any	1
VBA	using any	2
Add-ons	using any, for each one	1
Total your points here		

Step 2 – Determine the recommended system configuration

For this number of points	Minimum hardware you need	Minimum software you need
8 or more	Pentium® II 400 MHz 128 MB RAM	Windows® XP Professional* or Windows 2000 Professional** or Windows Server 2003*** or Windows Vista**** or Windows 7 Professional or Windows Server 2008*****
6 to 7	Pentium II 300 MHz 128 MB RAM	Windows XP Professional* or Windows 2000 Professional** or Windows Server 2003*** or Windows Vista**** or Windows 7 Professional or Windows Server 2008*****
3 to 5	Pentium 200 MHz 64 MB RAM	Windows XP Professional* or Windows 2000 Professional** or Windows Server 2003***

Important: The hardware should meet or exceed the minimum hardware requirements of the operating system you will run.

* With Service Pack 1 or later.

** With Service Pack 3 or later.

*** With Service Pack 1 or later.

**** With Service Pack 1 or later.

***** With Service Pack 2.

Installing RSVIEW32 Runtime

The steps below explain how to install RSVIEW32 software and user's guides from a CD-ROM to your computer. If you choose not to install the user's guides, you can view them from the CD.

Installing RSVIEW32 Runtime on Windows Server 2003, Windows XP, Windows 2000 and Windows Vista

As part of the installation process, RSVIEW32 Runtime creates program folders and modifies registry entries. To make these modifications, the currently-logged-in user must have administrative rights on the computer on which RSVIEW32 Runtime is installed.

Installing RSVIEW32 Runtime

Follow these instructions to install RSVIEW32.

To install RSVIEW32 Runtime:

1. Close all open Windows programs.
2. Place the RSVIEW32 CD-ROM into your CD-ROM drive. The CD should start running automatically.

If the CD does not start automatically, run D:\Setup.exe where D is the drive containing the CD.

3. In the RSVIEW32 window, click the button beside Install RSVIEW32 Runtime.
4. Follow the on-screen instructions.
5. During installation, you will be prompted to activate RSVIEW32 Runtime. If activation is present from an earlier version of RSVIEW32 Runtime, click Next.

If you removed the activation or if you are installing RSVIEW32 Runtime for the first time, insert the Activation disk in the diskette drive and click Activate. For details about activation, see “Activating RSVIEW32 Runtime” on page 1-5.

6. When installation is complete, restart Windows. If you choose not to restart Windows now, be sure to restart it before running RSVIEW32 Runtime for the first time.

If you plan to use RSVIEW32 as an OPC server, you must configure Microsoft® Distributed COM (DCOM). For instructions see the RSVIEW32 Help.

Installation summary

The default installation does the following:

- installs RSVIEW32 Runtime in the folder \Program Files\Rockwell Software\RSVIEW
- adds a Rockwell Software\RSVIEW32 folder to your list of programs. This folder contains the shortcut for RSVIEW32 Runtime

Ordering RSVIEW32 Runtime on disks

RSVIEW32 Runtime is available on 3.5-inch disks. To order disks, contact your local authorized Allen-Bradley® distributor or your local Allen-Bradley sales office.

Activating RSVIEW32 Runtime

Rockwell Software Inc. uses activation files to activate its Windows-based software packages. You can activate RSVIEW32 Runtime by using FactoryTalk Activation software to download and transfer activation files.

FactoryTalk® Activation

FactoryTalk Activation provides a secure, software-based system for activating Rockwell Software products and managing software activation files. With FactoryTalk Activation, there is no need for a physical “master disk” or any physical media; instead, activation files are generated and distributed electronically.

Important: EvRSI activation is replaced by FactoryTalk® Activation in this release. If you are using EvRSI activation, please contact your local Rockwell Automation Sales office or Technical Support for information on migrating your activations to FactoryTalk® Activation. For Rockwell Automation Technical Support in the U.S. call (440)646-3434. Outside the U.S. see <http://www.rockwellautomation.com/locations/>.

Finding more information about FactoryTalk Activation

You can find detailed information, instructions, and troubleshooting tips in the FactoryTalk Activation Help.

After installing and launching the FactoryTalk Activation software, click **Help** or **Learn More . . .** to open FactoryTalk Activation Help.

Using the FactoryTalk Activation Manager

The FactoryTalk Activation tool is used to obtain the activation files needed to activate your software. It can also be used to borrow activations from servers having available activations, return activations back to servers, re-host activations, etc.

To open the FactoryTalk Activation Manager:

On the Windows Start menu, click Programs > Rockwell Software > FactoryTalk Activation > FactoryTalk Activation Manager.

To use the FactoryTalk Activation Manager:

1. In the FactoryTalk Activation Manager, click the Home tab then click Get New Activations, or you may click Manage Activations tab, then click Get New Activations.
2. Select the Activation Method to be used.

To provide the software’s Serial Number and Product Key, identify the activation server’s Host ID, and download activation files, follow the instructions in the manager.

3. Save the activation file or files.

Activation files must have a .lic extension. You can save the files to the default location:

On Windows 2003 and Windows XP the default location for FactoryTalk activation files is: C:\Documents and Settings\All Users\Shared Documents\Rockwell Automation\Activations

On Windows 7 Professional and Windows Vista the default location for FactoryTalk activation files is: C:\Users\Public\Documents\Rockwell Automation\Activations"

For details, click Help in the top left corner of the page.

Protecting activation files

Activation files are simple text files that must have a .lic extension. The activation key for RSView32 is called RSVMSGR.PRO.

As long as the .lic extension is retained, you can copy or rename an activation file without harming it. However, tampering with text inside the activation file can disable your Rockwell Software products.

If an activation file is damaged or deleted, contact Rockwell Automation Technical Support.

For safekeeping, keep an original set of your activation files on backup media. Use descriptive names for the files, so that you can identify them later, and copy them back to the appropriate computers.

Activation files are locked to the Host IDs of the computers (or dongles) that need them. Activation will fail for Rockwell Software products on a computer where the specified Host ID is not recognized by the activation file.

Using borrowed activations

RSView32 supports borrowed activations for development computers. A borrowed activation is retrieved from a pool of available activations on the activation server, and expires after a specified length of time. Once a borrowed activation is checked out to a particular computer, the computer can disconnect from the network. When the time limit expires, the associated software on the computer is no longer activated, and the activation automatically becomes available again from the server's activation pool. A borrowed activation can also be returned to the activation server manually before the time limit expires.

To borrow activations from a server:

1. Connect your computer to the network where the activation server is located.
2. Run the FactoryTalk Activation Manager and then click the **Manager Activations** tab.
3. Click the Manage Activations tab, and then click Borrow Activations tab. To borrow a single activation, click the checkbox that corresponds to the activation you want to borrow, and then click **Set Borrow Term**. To borrow more than one activation, click additional checkboxes for the activations you want to borrow and then click **Borrow**. If you do not see the activations you want to borrow, click **I Don't See My Activation Here**. The Activations in gray may not be selected as they are already borrowed by your computer.
4. In the Set Borrow Term window, set the time of the return by entering a calendar date, net number of days, or approximate hours (shift). The default Calendar date is the maximum borrow term for the activation or group of activations.

To return a borrowed activation before it expires:

1. Connect to the network and run the FactoryTalk Activation Manager.
2. On the Manage Activations tab, click **Return Activations**, select the borrowed activations you want to return.
3. Click the **Return Activations** button.

To open the Activation Help:

To open FactoryTalk Activation Help. Open FactoryTalk Activation Manager, click **Help** or **Learn More . . .**

FactoryTalk® Activation Grace Period

FactoryTalk provides customers with access to critical software components for a grace period of 7 days regardless of availability of valid activations. When a product is successfully activated, all grace period records will be cleared and the product will proceed with valid activations.

- During the grace period, FactoryTalk Activation will display a warning message and log a diagnostic warning message to FactoryTalk® Diagnostics every four hours if no valid activation key is detected.
- Once the grace period expires, RSView32 Runtime will enter demo mode. However, until RSView32 is restarted, it will continue to run as in the grace period.
- RSView32 Personalized version can be successfully activated with both RSView32 Personalized activations and RSView32 Protected activations.

Uninstalling RSView32 Runtime

To uninstall RSView32 Runtime, use the Windows Control Panel.

To uninstall RSView32 Runtime:

1. Click the Windows Start button, point to Settings, and then click Control Panel.
2. Double-click Add/Remove Programs, select RSView32 7.60.00, and click Add/Remove.
3. Follow the on–screen instructions.

4. When the Uninstall program is finished, restart Windows.

Running a project

This chapter outlines:

- the steps for setting up and running your project
- how to add files to a project
- how to stop a project

Setting up and running your project

Once you have installed RSView32™ Runtime on the target computer, you are ready to set up and run the project you have created in RSView32 Works.

Summary of steps

1. Move the project to the target computer.
2. If the target computer is configured for a different locale than the development computer, specify time, date, and number formats.
3. Start RSView32 Runtime.
4. Open the project.
5. Set up the channels, OPC® server, or DDE server.
6. If applicable, change the path for any files that are outside of the project directory.
7. Run the project.

Each of these steps is discussed in detail on the following pages.

Step 1 – Moving the project

If your project will be running on a different computer than it was developed on, you must move the project using an external tool such as Windows® Explorer, My Computer, or the File Manager. Be sure you move all files referenced in your project, including any files outside of the project directory.

You do not have to move the project to the same directory in which it was created. When you move a project from one computer to another, RSView32 Runtime automatically changes the project directory's paths to the drive and directory the project has been moved to. If your project references files outside of the project directory, update the path to these files in the Project Paths dialog box. See “Step 6 – Changing the path to files outside the project directory” on page 2-7.

To move your project:

1. In Windows Explorer, My Computer, or File Manager, move or copy the project folder to the target location.
2. Move or copy any folders or files outside the project folder to the target location.

Step 2 – Specifying time, date, and number formats

Skip this step unless the target computer is configured for a different locale than the development computer, and the new locale uses different time, date, or number formats.

Use the Windows Control Panel to specify the formats. You can also change these formats after the project is running, but you must close and restart the project for the changes to take effect.

1. Click the Windows Start button, select Settings, and then click Control Panel.
2. Double-click Regional Settings or Regional Options.

3. Select the language or locale whose settings you want to use.
4. Verify or customize the settings on the Number, Time, and Date tabs.
5. Click OK.

For more information, see your Windows documentation.

Step 3 – Starting RSVIEW32 Runtime

You can start RSVIEW32 Runtime manually, or you can set up RSVIEW32 Runtime to start automatically when Windows starts.

Starting RSVIEW32 Runtime manually

To start RSVIEW32 Runtime in Windows 2000

- Click the Windows Start button, select Programs, Rockwell Software, RSVIEW32, and then click RSVIEW32 Runtime.

To start RSVIEW32 Runtime in Windows Server 2003 and Windows XP

- Click the Windows Start button, select All Programs, Rockwell Software, RSVIEW32, and then click RSVIEW32 Runtime.

Starting RSVIEW32 Runtime automatically

To automatically start RSVIEW32 Runtime each time Windows starts, include an RSVIEW32 Runtime shortcut in the Windows Startup folder. You can also open a project when Windows starts.

To add RSVIEW32 to the StartUp folder in Windows 2000

1. Click the Windows Start button, select Settings, and then click “Taskbar & Start Menu.”
2. In the Taskbar and Start Menu Properties dialog box, click the Advanced tab.
3. Click Add. The Create Shortcut wizard opens.

4. To start RSVIEW Works when Windows starts, browse to the location of the RSV32.exe file, and then click the file name.

The path is \Program Files\Rockwell Software\RSView\RSV32.exe.

5. To start RSVIEW Works and open a project when Windows starts, specify the path to RSVIEW32 and to the project. For example, to specify a project called Norm's Bakery, type:

"C:\Program Files\Rockwell Software\RSView\RSV32.exe"
"C:\Program Files\Rockwell Software\Samples\Norm's Bakery\Norm's Bakery.rsv"

If the path contains spaces, you must enclose it in double quotes.

6. Click OK, and then click Next.
7. In the Select Program Folder dialog box, select the Startup folder, and then click Next.
8. Specify a name for the shortcut (for example, RSVIEW32).
9. Click Finish, and then click OK.

For more information, see your Windows documentation.

To add RSVIEW32 to the StartUp folder in Windows Server 2003, Windows XP and Windows Vista

1. Click the Windows Start button and select All Programs.
2. Right-click the Startup folder and select Open.
3. Browse to the location of the RSV32.exe file.

The path is \Program Files\Rockwell Software\RSView\RSV32.exe.

4. Right-click the file name and select Create Shortcut. The shortcut will be created in the same location as the RSV32.exe file.
5. Drag the shortcut into the Startup folder.

For more information, see your Windows documentation.

Step 4 – Opening the project

To open a project:

1. In the RSView32 Runtime window, click Open on the File menu.
2. In the Open dialog box, locate the project, click the project's .rsv file (for example, Norm's Bakery.rsv), and then click Open.

Using the Runtime Project Manager



With the Runtime Project Manager you can:

- edit channels, nodes, user accounts, and data log paths—to open an editor, double-click the editor's icon
- view activity and alarm logs—to open the viewers, double-click the Activity Log Viewer and the Alarm Log Viewer icons

- open the command line to manually enter RSVIEW32 commands—to open the command line, double-click the Command Line icon
- run individual project components—to run a component, select the icon for the editor that created the component, and then double-click the component name
- run a project—for details about running a project, see “Step 7 – Running the project” on page 2-9

Step 5 – Setting up the channel, OPC server, or DDE server

Before you can run a project, you must load and configure drivers for the programmable controllers or servers that your project was configured to use.

Connecting to programmable controllers using direct drivers

To connect to most Allen–Bradley® devices, install and load RSLinx® and install the driver required by your RSVIEW32 project.

To communicate with SoftLogix™ 5 programmable controllers, you must use an RSLinx driver.

IMPORTANT

You must use the version of RSLinx that came with your RSVIEW32 software. This version has been tested for compatibility with RSVIEW32. Other versions may not function correctly.

To select a different RSLinx driver for your project, use the Channel editor.

For details about the RSLinx drivers, see the documentation accompanying RSLinx. For details about the Channel editor, see Chapter 2, *Setting up direct driver communications*, in the RSVIEW32 User’s Guide.

Connecting to OPC servers and DDE servers

To connect to a wide range of local and remote devices, install and configure the appropriate OPC or DDE server. For details about configuring the server, see your server documentation.

To change the OPC node's server name and access path or the application and topic name for the DDE server your project will use, use the NodeSwitch command or use the Node editor.

For details about the NodeSwitch command, see Appendix A, *RSView32 commands*, or see Help. For details about the Node editor, see Chapter 3, *Setting up OPC and DDE communications*, in the RSView32 User's Guide.

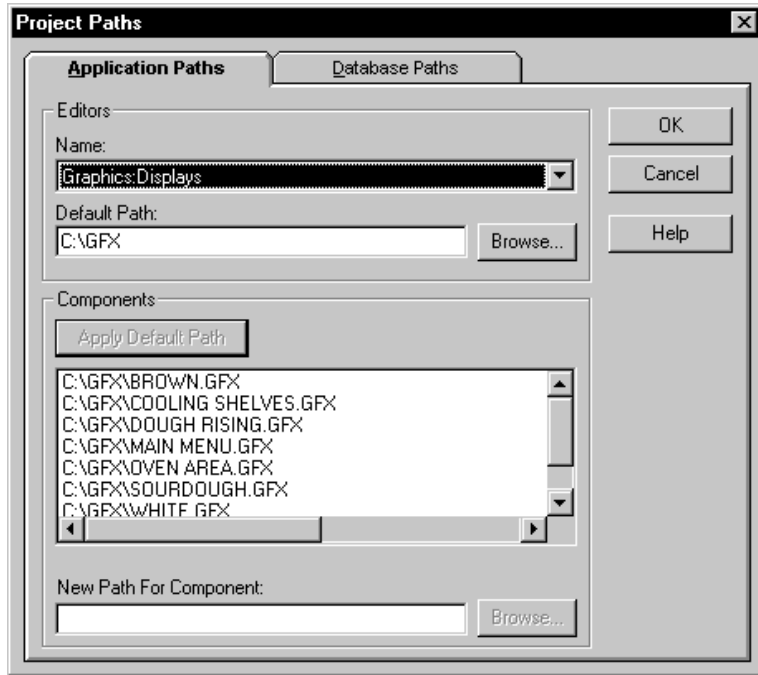
Step 6 – Changing the path to files outside the project directory

If your project references files outside of the project directory and the path to these files has changed (for example, the drive letter has changed), use the Project Paths dialog box to update the path.

For example, imagine your project directory is C:\Norm's Bakery but you have stored graphic files in C:\Gfx. You move your project from C:\ to D:\. The paths to the files in \Norm's Bakery are automatically updated, but the path to \Gfx is not—you must update the path in the Project Paths dialog box.

To change the path:

1. With the Project Manager active, click Project on the menu bar, and then click Project Paths.



2. Click the Application Paths tab.
3. In the Name field, select the editor that was used to create the component.

Any components associated with the editor are displayed in the Components field.
4. In the Default Path field, type the new path or click Browse to select the new path.
5. Click Apply Default Path.

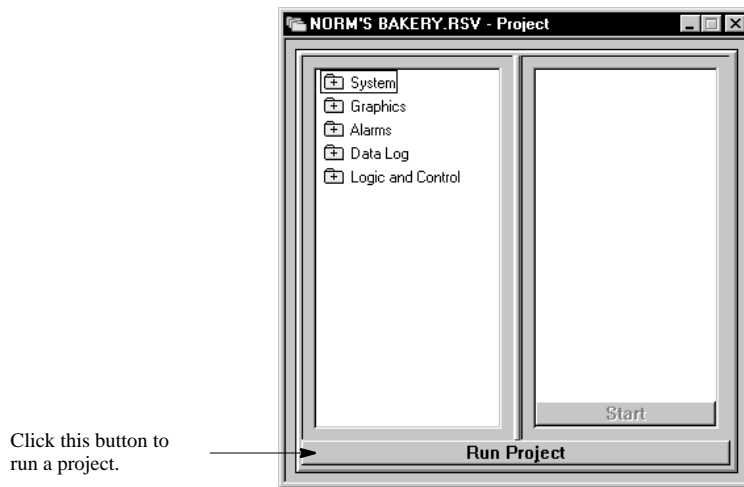
The components are updated to the new path.
6. Click OK.

Step 7 – Running the project

You can run the project manually or you can set up the project to run automatically when Windows starts.

Running a project manually

In the Project Manager, click Run Project.



Running a project automatically

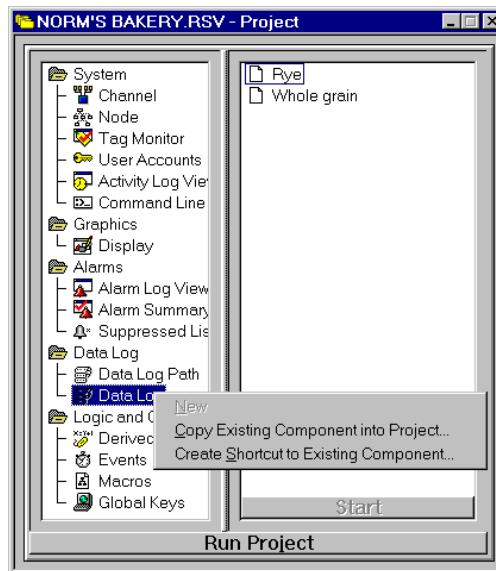
To automatically start RSVIEW32 Runtime and run a project each time Windows starts, include an RSVIEW32 Runtime shortcut in the Windows Startup folder, as described on page 2-3.

Adding files to a project

You can add new files to a project any time by copying or moving the physical file to the target computer and then adding the file's name to the Project Manager.

To add files:

1. In Windows Explorer, My Computer, or File Manager, copy or move the file you want to add into the appropriate folder in the project. For example, add a graphic display file to the Gfx folder.
2. In the RSView32 Project Manager, select the icon for the editor that was used to create the file (for example, select the Data Log icon for a data log model), right-click, and then click Copy Existing Component into Project or click Create Shortcut to Existing Component.



3. In the dialog box, locate the file that you moved, and then click Open. The file name will be added to the right-pane of the Project Manager.

Changing data logging paths

You can change the data logging paths at runtime. You can change the primary and secondary paths for dBASE® IV data log models, and the backup path for ODBC data log models. You cannot change the ODBC database using RSVIEW32 Runtime. After you change the logging path you must stop and restart data logging for the changes to take effect.

To edit the logging paths:

1. In the Project Manager, open the Data Log folder.
2. Open the Data Log Path editor.
3. Specify paths as described in “Specifying Logging Paths” in Chapter 7, *Configuring data logging*, in the RSVIEW32 User’s Guide.

You can also change logging paths at runtime using the DataLogPath command. For details about the DataLogPath command, see Appendix A, *RSVIEW32 commands*, or see Help.

Stopping a project

To stop a project, do one of the following:

- in the Project Manager, click the Stop Project button
- on the command line, type ProjectStop

When a project stops

When a project stops, the following occurs:

- open displays and RSVIEW32 components close
- running data log models stop
- running derived tag files stop
- running event files stop
- alarm monitoring stops
- OPC/DDE (real-time) data server stops

If a shutdown macro has been specified in the Startup editor, the macro runs once the above components have closed and stopped running.

Using security

This chapter describes how to:

- add, delete, or change users
- log into a running project
- change passwords

This chapter is applicable only for projects that use security.

Editing the users for a project

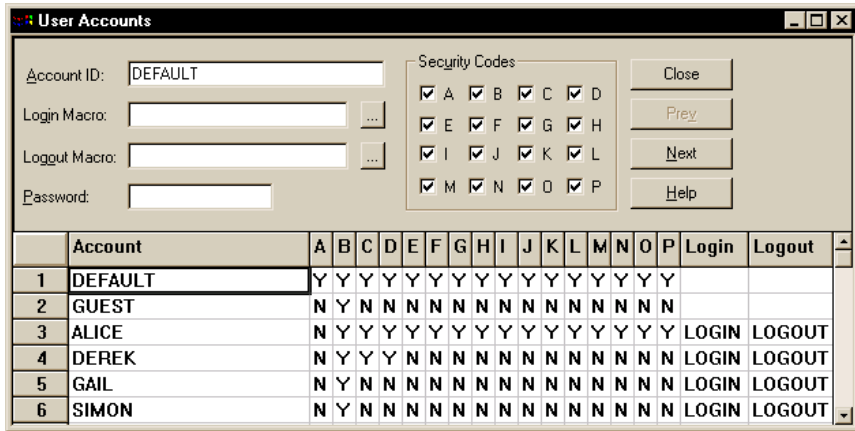
If your project has been set up with security, only users or groups of users listed in the User Accounts editor can log in and out of the project.

You can enter user names and passwords in the User Accounts editor, or, if you are using Windows® Server 2003, Windows XP, Windows 2000, or Windows Vista, you can use the Windows user list instead of creating a custom RSVIEW32™ list. No matter which method you choose to create the user list, you still have to assign security access to the individual users.



To open the User Accounts editor:

1. In the Project Manager, open the System folder.
2. Open the User Accounts editor.

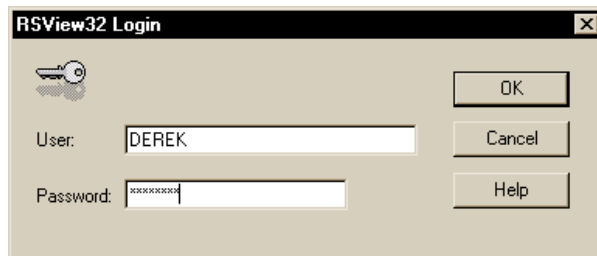


For details about the fields in the User Accounts editor, see Chapter 10, *Adding security*, in the RSVIEW32 User's Guide.

Logging in at runtime

If you want users to log in, your project must include a way for users to access the RSVIEW32 Login dialog box.

One way is to create a button in RSVIEW32 Works. When configuring the button, use the Login command as the press action. When the button is pressed, the RSVIEW32 Login dialog box appears.



To successfully log in, users must enter a name, and a password (if defined), that exists in the User Accounts editor.

Using the electronic signature button

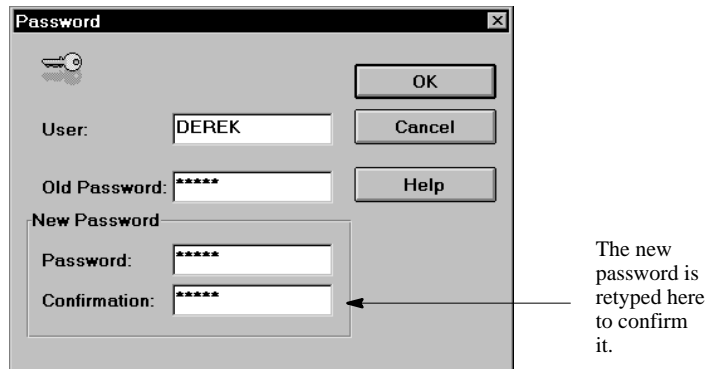
If your project uses the electronic signature button, users must enter their names and passwords in order to initiate the button's action. Optionally, you can set up the button so that a supervisor must also enter a name and password to authorize the transaction.

For more information about using the signature button at runtime, see Chapter 10, *Adding security*, in the RSVIEW32 User's Guide.

Changing passwords at runtime

If you want users to be able to change their passwords, your project must include a way for them to have access to the Password dialog box.

One way is to create a button in RSVIEW32 Works. When configuring the button, use the Password command as the press action. When the button is pressed, the Password dialog box appears.



The new password is also added to the Password field of the User Accounts editor.

IMPORTANT

If your project is using the Windows Security Options, and you change your password using the RSVIEW32 Password command, your password for the Windows Server 2003, Windows XP, or Windows 2000 domain will automatically be changed as well.

A

RSView32 commands

This appendix lists the RSView32™ commands that are supported in RSView32 Runtime. Commands that are not listed here will not run in the runtime package.

For more information about commands and their usage, see the *RSView32 User's Guide*.

= (Equal)

[&<tag_name> = <expression>

Writes the value resulting from an expression to a tag.

[&] Forces the command to be executed asynchronously, which makes the command faster.

<tag_name> The name of the tag that will store the result of the expression.

<expression> A value, string, tag name, or a more complex expression.

Enclose tag names that contain dashes or start with a number in braces { } when you use them in an expression. This distinguishes the characters in the tag name from the characters in the expression. Also use braces when using wildcards (* or ?) to represent multiple tags in an expression.

Enclose strings in quotes. The string can contain any character, and can include spaces.

Do not use braces for the tag name *before* the equal sign.

You cannot nest braces.

You can attach security to the = (Equal) command just as you can for any RSVIEW32 command. For more information about security, see Chapter 10, *Adding security*, in the RSVIEW32 User's Guide.

For more information about expressions, see Chapter 14, *Creating expressions*, in the RSVIEW32 User's Guide.

Examples: The = (Equal) command

&Tag1 = Tag1 + 1

Evaluates the command asynchronously. Increases the value of Tag1 by 1.

Tag1 = Tag2

Sets the value of Tag1 to be the same as Tag2.

Tag1 = Tag2 + Tag3

Adds the values of Tag2 and Tag3 and stores the result in Tag1.

1Pump = {Industry-2} + {2Pump}

Adds the values of Industry-2 and 2Pump and stores the result in 1Pump. Braces surround Industry-2 because of the dash in the name. Braces surround 2Pump because the name starts with a number. No braces are used for 1Pump because this name is on the left side of the equal sign.

Tag1 = if (Tag1 < Tag2) then 3 else 4

If Tag1 is less than Tag2, Tag1 is set to 3, but if Tag1 is equal to or greater than Tag2, Tag1 is set to 4.

Tank1\Message = "Tank1 Overflow"

Writes the string Tank1 Overflow to the Tank1\Message tag.

Abort

Abort [parameter]

Closes one or more windows. Without a parameter, the Abort command closes the window that has focus.

[*parameter*] One of the following:

me Closes the window from which the command is executed.

editor Closes the specified editor, such as a graphic display or an alarm summary. To name the editor, use the RSVIEW32 command that opens the editor. See the examples below.

* Closes all windows.

Ctrl-F4 also closes the active window.

Examples: The Abort command

Abort

Without a parameter, closes the window that has focus.

Abort me

Closes the window from which the command is executed.

Abort Display

Closes all open graphic displays.

Abort Display Pumps or Abort Pumps

Closes the graphic display called Pumps.

Abort Monitor

Closes all open tag monitors.

Abort *

Closes all windows.

Account

Account

Opens the User Accounts editor. To prevent security from being changed at runtime, restrict access to this command.

Acknowledge

Acknowledge [tag_name]

Acknowledges an alarm or a group of alarms. This command gives no indication that it has run but will display a message if not executed properly.

If an acknowledge bit is associated with an alarm, acknowledging the alarm sets the acknowledge bit.

[tag_name] The name of the tag to be acknowledged. This can be a tag name, a name with wildcards, or the [tag] literal string.

If no tag is specified, this command acknowledges the most-severe, most-recent unacknowledged alarm.

[tag] Specifying the word “tag” inside square brackets acknowledges alarms for the tag associated with the highlighted object in the active graphic display.

IMPORTANT If alarms are occurring rapidly, don't run the Acknowledge command without a tag name. The Acknowledge command could acknowledge a new alarm rather than the intended alarm.

Examples: The Acknowledge command

Acknowledge Hopper1\Flow

Acknowledges all outstanding alarms for the tag Hopper1\Flow.

Acknowledge Hopper1*

Acknowledges all outstanding alarms for all tags in the folder called Hopper1.

Acknowledge *

Acknowledges all outstanding alarms.

Acknowledge [tag]

Acknowledges the alarm for the tag associated with the highlighted object in the active graphic display.

AcknowledgeAll AcknowledgeAll

Acknowledges all outstanding alarms. This command runs more quickly than Acknowledge *. It gives no indication that it has run but will display a message if not executed properly.

If an acknowledge bit is associated with an alarm, acknowledging the alarm sets the acknowledge bit.

There is no limit to the number of tags that can be acknowledged with this command.

ActivityBarOff ActivityBarOff

Hides the activity bar.

ActivityBarOn ActivityBarOn

Shows the activity bar.

You can undock the activity bar and move it anywhere on the screen. To move the bar, click between the Clear and Clear All buttons and drag. To redock the activity bar, click the title bar and drag the bar until it touches the RSVIEW32 status bar.

To resize the activity bar, drag any corner or edge.

ActivityLogSend ToODBC

ActivityLogSendToODBC <data_source>
<target_table_name> [/Wn] [/User_name] [/Ppassword]

Exports activity log data from DBF files to the ODBC database. The connection to the ODBC database is maintained for the length of time specified by the wait (/W) parameter.

<data_source> A valid ODBC data source name. If the name has a space embedded in it, enclose the name in quotes ("").

<target_table_name> The name of a table in the ODBC database where the data will be exported. If the table does not exist RSVIEW32 attempts to create it.

[/Wn] The time the connection to the ODBC database will be maintained. If nothing is specified, the wait time defaults to about 5 minutes. A wait time of zero keeps the connection open until the project is stopped.

This parameter allows frequent exports to the database without having to connect every time. Once the connection is established, RSVIEW32 waits the specified time and, if another command is received to export data, the command is executed and the wait timer is reset. If no command is received within the wait time, the connection is closed.

[/User_name] A user ID that is valid on the data source.

[/Ppassword] A password that is valid on the data source.

ActivityOff

ActivityOff

Stops activity logging.

ActivityOn	ActivityOn Starts activity logging. Activity logging is on by default.
ActivityPrintOff	ActivityPrintOff Turns off the printing of activity logging. By default, activity log printing is turned on when you start an RSVIEW32 project.
ActivityPrintOn	ActivityPrintOn Turns on the printing of activity logging if it has been turned off by the ActivityPrintOff command. By default, activity log printing is turned on when you start an RSVIEW32 project.
ActivityViewer	ActivityViewer Opens the Activity Log Viewer.
AlarmEvent	AlarmEvent <EventName> <EventType> [TagType] [/VTagValue] [/SSeverity] [/H"Timestamp"] [/L"LogMessage"] Creates an alarm event. Alarm events are not processed unless the AlarmOn command is issued, and alarm events stop being processed when the AlarmOff command is issued. You cannot specify the threshold for an alarm. <EventName> The name of the alarm event, up to 255 characters long. The event name must follow the syntax of a tag name, and can, but need not, be a tag name in the tag database. The alarm event name <i>cannot</i> be the name of an alarm tag. <EventType> The type of alarm transaction, which must be one of the following: [IntoAlarm] Indicates that the tag has gone into or [In] alarm.

of	[OutOfAlarm] Indicates that the tag has gone out or [Out] alarm.
of	[InAndOut OfAlarm] Indicates that an alarm has occurred, but the tag is again immediately out or alarm, such as a digital change-of- [InAndOut] state alarm.
[TagType]	The type of tag, which must be one of: [/A] Analog [/D] Digital If you don't specify either /A or /D, the alarm is assumed to be analog.
[VTagValue]	A floating-point value associated with the alarm event. If the floating-point value is not specified, the tag value is 0.0. If the name of an alarm event is the name of a tag in the tag database, the value specified by this parameter will not update the tag's value in the value table.
[SSeverity]	The alarm severity. The severity is an integer from 1 to 8. Alarm severity can be specified only for alarm events of type IntoAlarm, or InAndOutOfAlarm. If a value is not specified, the alarm severity is 1.

[/HTime-
stamp] The time stamp associated with the alarm transaction, in the format: <HH:MM:SS>[Date]:

<HH:MM:SS> The 24-hour military format for time. You must use this format for indicating the time.

[Date] A date that can be specified in the same format as the Windows® date style configured for your computer. If you want to use a VBA program to set the date, you can use any date format that conforms to the MFC class COleDateTime. If you do not specify a date, the current date is used.

If you specify a time stamp for an alarm, the alarm may not appear as the most recent alarm in the \system\AlarmBanner tag, even if it was logged after an alarm with a more recent time stamp.

[/LLog
Message] The alarm message, up to 132 characters long, to be logged to disk and/or printer. The log message can contain any of the placeholders available to alarm messages. If you do not specify a log message, the user default message is used.

AlarmLogOff AlarmLogOff
Stops alarm logging.

AlarmLogOn AlarmLogOn
Starts alarm logging.
Alarm logging is on by default.

AlarmLog Remark

AlarmLogRemark ["Text"] [/P] [/Sn] [/R] [/Ttagname]

Adds the specified text string as a transaction in the alarm log file.

[*Text*] A text string, up to 132 characters long. The text can contain the following placeholders:

[\D] the current date

[\T] the current time

[\N] the tag name. If the tag is in a local tag database, you can also use the placeholders [\S] tag description, and [\U] tag units.

[/P] Prompts the operator for a remark at runtime by displaying a dialog box containing a text field. The operator can type a remark up to 132 characters long. The remark can also include the placeholders shown above.

If both the Text parameter and the /P parameter are specified, the contents of the Text parameter will appear in the text field at runtime, and the operator can modify or add to the contents of the Text parameter before it is logged to the alarm log file. If the tag name (/T parameter) is specified, the prompt dialog box will display the tag name at runtime, but the operator cannot change the tag name.

[/Sn] Associates an alarm severity with the remark. The severity is an integer from 1 to 8. This value is shown in the Severity column of the alarm log file. If the remark is logged to a printer, the alarm severity determines which printer will print the remark.

If both the /P parameter and the /Sn parameter are specified, the prompt dialog box will display the alarm severity at runtime, but the operator cannot change the severity.

[/R] Logs the remark to a printer as well as to the alarm log file. If the alarm severity is not specified (/Sn parameter), the printer for Severity 1 is used. If no printer is assigned to the specified severity, the alarm log remark is not printed.

[/T*tagname*] The string that is logged in the Tagname column of the alarm log file. This string can be a tag name, the name of a user-generated alarm event, or any other string that matches the syntax of a tag name. You can use this parameter to correlate remarks with specific alarm transactions for generating reports later.

AlarmLogSend ToODBC

AlarmLogSendToODBC <data_source> <target_table_name>
[/W*n*] [/U*user_name*] [/P*password*]

Exports alarm log data from DBF files to the ODBC database. The connection to the ODBC database is maintained for the length of time specified by the wait (/W) parameter.

<*data_source*> A valid ODBC data source name. If the name has a space embedded in it, enclose the name in quotes (“”).

<*target_table_name*> The name of a table in the ODBC database where the data will be exported. If the table does not exist RSVIEW32 attempts to create it.

[/W*n*] The time the connection to the ODBC database will be maintained. If nothing is specified, the wait time defaults to about 5 minutes. A wait time of zero keeps the connection open until the project is stopped.

This parameter allows frequent exports to the database without having to connect every time. Once the connection is established, RSVIEW32 waits the specified time and, if another command is received to export data, the command is executed and the wait timer is reset. If no command is received within the wait time, the connection is closed.

[/User_name] A user ID that is valid on the data source.

[/Password] A password that is valid on the data source.

AlarmOff

AlarmOff

Stops alarm monitoring. Once this command has run, it displays a message indicating that alarm monitoring has stopped.

AlarmOn

AlarmOn [/H]

Starts alarm monitoring. Once this command has run, it displays a message indicating that alarm monitoring has started.

[/H] Turns on handshaking the moment alarm monitoring starts and sets the handshake bit for any tags in alarm at that moment.

To have alarming start when a project starts, open the Startup editor and click the Alarming check box.

AlarmPrintOff

AlarmPrintOff

Stops alarms from printing.

AlarmPrintOn

AlarmPrintOn

Starts alarm printing.

Alarm printing is on by default.

AlarmViewer

AlarmViewer

Opens the Alarm Log Viewer.

AppAbort

AppAbort <application>

Closes the specified Windows application.

<application> The name of a Windows application exactly as it appears in the application's title bar.

Example: The AppAbort command

If Notepad is open and contains an untitled file, the Notepad title bar will read Untitled – Notepad. To close Notepad you must type exactly what's in the Notepad title bar as follows:

AppAbort Untitled - Notepad

AppActivate

AppActivate <application>

Activates (pulls forward) the specified Windows application. The application must already be running. (You can use the AppStart command to start the application.)

<application> The name of the Windows application you want to activate. Typically, this is the name that appears in the application's title bar.

AppStart AppStart <application>

Runs the specified application, which can be another Windows application or an “application extender” you’ve programmed.

<*application*> The path and executable required to start the program.

Examples: The AppStart command

AppStart c:\windows\notepad c:\autoexec.bat

Opens Notepad and displays the autoexec.bat file.

AppStart c:\Program Files\Plus!\Microsoft Internet\Iexplore.exe

Opens Microsoft® Internet Explorer.

Beep Beep

Runs a wave file to produce a sound from the computer speaker. The sound is a wave file assigned to the Default Beep in the Windows Control Panel.

Channel Channel

Opens the Channel editor.

CommandLine CommandLine

Opens the command line.

ComStatus ComStatus

Obsolete. Use ComStatusOff or ComStatusOn.

ComStatusOff ComStatusOff

Stops logging of communication errors.

ComStatusOn

ComStatusOn

Starts logging of communication errors if RSVIEW32 is set up to do so.

To log communication errors, ensure the Communications category is selected in the Activity Log Setup editor.

To have error logging automatically start when a project starts, open the Startup editor and click the Communication Status check box.

DatabaseSync

DatabaseSync

Makes RSVIEW32 convert the tag database into binary format the next time the project opens. Use this command if you change the tag database that a project uses (using the Project Paths dialog box), after changing a node's data source, or after changing a direct driver's device type.

To synchronize the tag database:

1. On the command line type DatabaseSync, and then press Enter.
2. Close the project.
3. Open the project. The database will be synchronized as the project opens.

DataLogChangeRate

DataLogChangeRate <file> <value> [unit]

Changes the periodic log rate. The change affects the current logging session only, and won't be retained if data logging is stopped and restarted.

- <file> The name of a data log model, without a file extension.
- <value> The numeric portion of the time interval for the log rate. For example, if you want to log data every 20 seconds, the value is 20. The value must be an integer from 1 to 64,000.
- [unit] The time unit of the log rate: hundredths, tenths, seconds, minutes, hours, or days. If you omit the [unit] parameter, the default is seconds.

DataLogMergeToPrimary

DataLogMergeToPrimary <parameter>

Moves data from the secondary or backup path to the primary path or ODBC database, for a specified model or for all models that are currently running.

- <file> The name of a data log model, without a file extension.
- * Moves data for all models that are currently running.

You can use the DataLogMergeToPrimary<file> command whether or not the specified model is running. If a model is running when you issue the DataLogMergeToPrimary command, RSVIEW32 switches back to the primary path or ODBC database. If a model uses the .dbf format, RSVIEW32 moves all files on the secondary path (including the current file set) to the primary path, begins a new file set on the primary path, and continues logging to the new file set. If a model uses the ODBC format, RSVIEW32 merges the ODBC backup files into the ODBC database and continues logging to the ODBC database.

DataLogNewFile DataLogNewFile <parameter>

Creates a new data log file for the specified model or for all models, on the path RSView32 is currently logging to (either the primary path or the backup path). If RSView32 is logging to an ODBC database, RSView32 logs an End snapshot and then a Begin snapshot when you issue this command.

<file> The name of the data log model, without a file extension.

* Creates new files for all models that are currently running.

DataLogOff DataLogOff <parameter>

Stops data logging for a specified model or stops data logging for all models.

<file> The name of a data log model, without a file extension.

* Stops data logging for all models.

DataLogOn DataLogOn <file>

Starts data logging for the specified model.

<file> The name of a data log model, without a file extension.

To have data logging automatically start when a project starts, open the Startup editor, click the Data Logging check box and specify a file.

DataLogPath

DataLogPath <file>

Opens the Data Log Path editor. You can use the editor to change the primary and secondary paths for dBASE® IV data log models, and the backup path for ODBC data log models. You cannot change the ODBC database using the Data Log Path editor.

<*file*> The name of a data log model, without a file extension.

DataLogRename File

DataLogRenameFile <file> <LogFileIDString>

Changes the log file identifier string that is used as part of the file name for dBASE IV log files and ODBC backup files that use long file names. The change affects the current logging session only, including the current set of log files.

<*file*> The name of a data log model, without a file extension.

<*LogFileID* The log file identifier string, up to 20 characters.

String>

DataLog Snapshot

DataLogSnapshot <parameter>

Logs one snapshot of data to the data log file, for the specified model or all models.

<*file*> The name of the data log model, without a file extension.

* Logs a snapshot of data for all models that are running.

IMPORTANT

The data log model must be running before the DataLogSnapshot command is issued.

DataLogSwitch Back

DataLogSwitchBack <parameter>

Switches data logging back to the primary dBASE IV file path or ODBC database for the specified model or for all models. This command switches data logging for a model only if the model is running, RSView32 is logging data to the secondary or backup path, and the primary path or ODBC database is available. For .dbf files, RSView32 creates a new set of files when it switches back to the primary path.

<*file*> The name of the data log model, without a file extension.

* Switches data logging for all models that meet the conditions outlined above.

DDEExecute

DDEExecute <application>|<topic> <command>

Sends a command or series of commands to an application through a DDE channel.

<*application*> The name of the server application you want to send a command to. This is usually the name of the application's .exe file without the .exe extension.

<*topic*> The name of the subject of the communication, preceded by the | character. Many applications that support DDE recognize a topic named System, which is always available.

<*command*> A command or series of commands recognized by the server application. Multiple commands must be in one string.

If the command string contains an exclamation mark (!), enclose the string in quotes. If the string includes quotes, use double quotes.

IMPORTANT

Before using the DDEExecute command, the server application must be running (but it can be minimized).

Examples: The DDEExecute command

DDEExecute RSLinx|__Drivers [Who_Active()]

Sends the Who_Active command, which displays the default RSLinx® RSWho window.

Drivers must be preceded by two underscores.

DDEExecute WINAB5|SYSTEM .SR 3

Sends WINTelligent™ LOGIC 5 (WINAB5) the Dot command SR 3, which means search for Rung 3.

DDEExecute WINAB5|SYSTEM .UI

Sends WINTelligent LOGIC 5 (WINAB5) the Dot command UI, which displays the System Information window.

DDEExecute Winrecip|IcomDdeExe [CONTROL_PANEL_DEFAULTS ()]

Sends WINTelligent RECIPE the CONTROL_PANEL_DEFAULTS command, which opens the default WINTelligent RECIPE Control Panel.

DDEPokeDisable	DDEPokeDisable Obsolete. Use RTDataWriteDisable.
DDEPokeEnable	DDEPokeEnable Obsolete. Use RTDataWriteEnable.
DDEServerOff	DDEServerOff Obsolete. Use RTDataServerOff.
DDEServerOn	DDEServerOn Obsolete. Use RTDataServerOn.

Define

Define <symbol> [string]

Creates a symbol. A symbol is an abbreviation for a command or a command with parameters. The symbol can be used anywhere a command can be used. You cannot nest symbols.

Symbol definitions are valid only during the current session; they must be re-defined each time RSVIEW32 is restarted. Symbols are typically defined in a startup or login macro.

<*symbol*> The abbreviated command. It cannot contain spaces.

[*string*] An existing command with or without parameters. It can contain spaces and other non-alphanumeric characters. Omitting the [*string*] parameter deletes the symbol definition.

See also Undefine.

Examples: The Define command

Define Di Display

Creates the symbol Di for the command Display. Typing Di as a command in any valid command syntax will have the same effect as typing Display.

Define Show Display Overview /CC

Creates the symbol Show for the command Display Overview /CC. Whenever Show is used, the graphic called Overview is displayed in the center of the screen.

Define Di

Deletes the symbol Di.

DerivedOff	DerivedOff <file>
	Stops running the specified derived tag file.
	< <i>file</i> > The name of a derived tag file without a file extension.
DerivedOn	DerivedOn <file>
	Starts running the specified derived tag file.
	< <i>file</i> > The name of a derived tag file without a file extension.
	To have a derived tag file automatically start when a project starts, open the Startup editor, click the Derived Tags check box and specify a file.
Display	Display <file> [/B] [/E] [/U] [/O] [cache] [/Pfile] [/Ttag_name] [/Hnnn] [/Wnnn] [/Min] [/Max] [position]
	Runs the specified graphic display file. Parameters specified here override settings in the Display Settings dialog box in the Graphic Display editor.
	If you are using the Cache After Displaying option in the Display Settings dialog to cache displays, use the position parameters with the Display command to ensure that displays open in the correct position after caching. Otherwise, if a user moves a display at runtime, the new position is remembered by the cache option.
	Do not cache more than 200 displays using the [cache] parameter and/or the Cache After Displaying option.
	< <i>file</i> > The name of a graphic display file without a file extension.
	[/B] Displays the specified graphic display in the background.
	[/E] Disables the Enter key so it does not download values in numeric input fields to the programmable

controller or server, unless the Display On-Screen Keyboard option is selected (in the Behavior tab of the Display Settings dialog box).

[/U] Updates tag values in all input fields when the display first opens.

[/O] Suppresses the display of the key list.

[*cache*] Specifies how to load the specified graphic display into the display cache, as follows:

/Z Loads the specified graphic display into the cache (but does not make it visible), so the display appears quickly when it is first used.

/ZA Loads the specified graphic display into the cache (but does not make it visible) and continually updates the graphic display, even when it is not visible. For example, use /ZA for a display that contains a real-time trend, so that the trend displays data for its entire time range when you view it.

To remove all displays from the cache, use the FlushCache command. To remove a particular display from the cache, use the FlushCache [*file*] command.

[/P*file*] The name of the parameter file that contains the tag names to be substituted for placeholders in the display. If the display does not contain placeholders, do not use this parameter.

If the parameter file has a long file name, enclose the file name in quotes, for example /P“Long file name”.

[/T <i>tag_name</i>]	One or more tags, separated by commas and no spaces, to be substituted for placeholders in the display. If the display does not contain placeholders, do not use this parameter.
[/H <i>nnn</i>]	Specifies the height of the graphic display in pixels.
[/W <i>nnn</i>]	Specifies the width of the graphic display in pixels.
[/Min]	Runs the graphic display minimized (as an icon).
[/Max]	Runs the graphic display maximized (full-screen size).
[<i>position</i>]	Specifies the position of the window as follows: /Q1 top right corner /Q2 top left corner /Q3 bottom left corner /Q4 bottom right corner /CT centered in the top half /CB centered in the bottom half /CL centered on the left side /CR centered on the right side /CC centered in the screen /X <i>nnn</i> <i>nnn</i> pixels from the left edge /Y <i>nnn</i> <i>nnn</i> pixels down from the top

Examples: The Display command

Display Sample

The first time the graphic display called Sample is opened, it is positioned and sized as specified in its Display Settings dialog box. When the display is closed, its size and position will be saved, so the next time it is opened it will be the same size and position as when it was last closed.

Display Picture /PNames /CB

Positions the graphic display called Picture in the bottom half of the

screen, using the file called Names to replace tag placeholders with tag names.

Display Picture /H300 /W400 /CC

Makes the graphic display called Picture 300 pixels high and 400 pixels wide, and positions the display in the center of the screen.

Example: Replacing tag placeholders in a graphic display

If a graphic display called Canning uses three tag placeholders, and you want to substitute tags for canning corn, you would type:

Display Canning /Tcorn\off,corn\on,corn\weight

Placeholder #1 is replaced by the corn\off tag, placeholder #2 is replaced by the corn\on tag, and placeholder #3 is replaced by the corn\weight tag.

Download

Download

Writes the value in the selected input field of the active graphic display to the programmable controller or server.

This command operates on the active graphic display. If no display is active, this command is ignored.

Ctrl-PgDn also downloads the value in the selected input field.

The Enter key also downloads the value in the selected input field.

However, if the /E parameter is used with the Display command, the Enter key is disabled, unless the Display On-Screen Keyboard option is selected (in the Behavior tab of the Display Settings dialog box). If the selected input field is a recipe field, pressing the Enter key opens the Recipe dialog box.

DownloadAll

DownloadAll

Writes the values in all input fields of the active graphic display to the programmable controller or server.

This command operates on the active graphic display. If no display is active, this command is ignored.

PgDn also downloads all the values in the input fields.

DriverPrimary

DriverPrimary <channel>

Switches from the secondary driver to the primary driver on the specified channel.

<*channel*> The number of the channel (1 through 4) that is being switched from its secondary to primary driver.

DriverSecondary

DriverSecondary <channel>

Switches from the primary driver to the secondary driver on the specified channel.

<*channel*> The number of the channel (1 through 4) that is being switched from its primary to secondary driver.

DriverToggle

DriverToggle <channel>

Switches from the current driver to the one not being used on the specified channel.

<*channel*> The number of the channel (1 through 4) that is being switched from one driver to another.

EchoOff

EchoOff

Stops logging commands to the activity log file. Other activities, such as errors and tag values, are still logged.

This command is normally used in macros to prevent the contents of the macro from being logged.

EchoOn

EchoOn

Restores logging to normal after an EchoOff command.

EventOff

EventOff <file>

Stops running the specified event file.

<file> The name of an event file without a file extension.

EventOn

EventOn <file>

Starts running the specified event file.

<file> The name of an event file without a file extension.

To have an event file automatically start when a project starts, open the Startup editor, click the Event Detector check box and specify a file name.

FlushCache

FlushCache [file]

Without the parameter, unloads all graphic displays from the display cache. With the parameter, unloads the specified graphic display from the display cache.

[file] The name of a graphic display file without a file extension.

Displays can be added to the cache by using the [*cache*] parameter with the Display command or by selecting the Cache After Displaying option in the Display Settings dialog box of the Graphic Display editor. You can cache up to 200 graphic displays.

If a display uses the Always Updating option with the Cache After Displaying option, the display's shutdown command is executed when you issue the FlushCache command.

FTDataServerOff	FTDataServerOff	Stops running the FactoryTalk live data server. This command has no parameters.
FTDataServerOn	FTDataServerOn	Runs the FactoryTalk live data server, allowing FactoryTalk clients to read RSVIEW32 tags and other live data. This command has no parameters.
FTDataWrite Disable	FTDataWriteDisable	Stops FactoryTalk Clients from writing values to RSVIEW32 tags. This command has no parameters.
FTDataWrite Enable	FTDataWriteEnable	Allows FactoryTalk clients to write values to RSVIEW32 tags. This command has no parameters.
HandshakeOff	HandshakeOff	Disables alarm handshaking, regardless of whether the individual handshake bits are configured. This command gives no indication that it has run but will display an error message if not executed properly.
HandshakeOn	HandshakeOn	Enables alarm handshaking. By default this setting is off. Handshaking can be disabled with the HandshakeOff command. This command gives no indication that it has run but will display a message if not executed properly.

IMPORTANT

This command will not set the handshake bit for any tag already in alarm when the command is executed.

Help

Help [word] [/Ffile]

Displays a Windows help file.

[*word*] A word you want to search for in the help file. When you specify a search word, the command will either:

- open the help file at a topic if the word uniquely identifies that topic
- open a list of related topics

[/F*file*] The name of a Windows help file. The default help file is for RSView32.

If no search word is specified, the command opens the RSView32 help file and displays the Contents topic.

Identify

Identify [tag_name]

Runs the command or macro associated with the named tag, whether or not the tag is in alarm.

[*tag_name*] The name of the tag in alarm. The command or macro associated with this tag will run. If no tag name is specified, the Identify command runs the command or macro associated with the most recent, most severe unacknowledged alarm.

[tag] Specifying the word “tag” inside square brackets runs the identify command for the tag associated with the highlighted object in the active graphic display.

Running the Identify command is not the same as acknowledging an alarm.

IMPORTANT

When many alarms are occurring rapidly, do not use the Identify command without a tag name. A new alarm could become the current alarm before the Identify command runs, and the command or macro that runs might not be the one expected.

Examples: The Identify command

Identify Hopper1\Divider

Runs the Identify command or macro for the tag Hopper1\Divider, whether or not the tag is in alarm.

Identify [tag]

Runs the Identify command or macro for the selected tag in the active graphic display.

Identify

Runs the Identify command or macro for the most recent, most severe unacknowledged alarm.

Invoke

Invoke <parameter>

Use this command with ActiveX objects. Use the command to:

- call an object's method
- assign the value returned by a method to a tag
- set an object's property to a tag value or a constant
- set a tag to the value of an object's property

The easiest way to specify the <parameter> string for the Invoke command is to use the Command Wizard.

The syntax for this command has four variations:

- *file.object.method(parameter1, parameter2, . . .)* calls an object's method

file The name of the graphic display that contains the ActiveX object

me An alternative to using *file*. At runtime it resolves to the graphic file that has focus.

object The name of the ActiveX object as specified in the Object Name dialog box in the Graphic Display editor.

method The name of a function or sub-routine in the ActiveX object. The method is initiated by an external event such as the Invoke command.

parameter The tag name or constant that the method will use. You must specify all of the parameters, even those that are optional.

- *tag_name=file.object.method(parameter1, parameter2, . . .)* assigns the value returned by a method to the specified tag

tag_name The name of the tag to which the value returned by the method will be assigned.

- *file.object.property(tag_name or constant)* sets an object's property to the specified tag's value or to a constant value
- *tag_name=file.object.property* sets the specified tag's value to the value of the object's property

Key

Key <parameter>

Starts or stops running the global key file.

<*parameter*> One of the following:

file The name of a global key file without a file extension.

/R Stops running the global key file.

To have a global key file automatically start when a project starts, open the Startup editor, click the Global Key File check box and specify a file.

LInsertStringTag [sheet] [tagname series]

Use this command to select one or more string tags from the tag database and insert the tag names to an Excel spreadsheet if the tags do not exist in the spreadsheet.

[*sheet*] The spreadsheet name of Excle file.

[*tagname series*] A series of tag names.

Make sure Microsoft Excel has been installed before using this command.

If the inserted tags have already existed in the Excel file, there will be a warning message logged into Activity Bar.

System string tags cannot be inserted in the spreadsheet by executing LInsertStringTag command.

Login

Login [username] [password]

Logs users into the system. To log in, users must have an account in the User Accounts editor.

[*username*] The user's name as defined in the User Accounts editor.

[*password*] The user's password as defined in the User Accounts editor. If this parameter is omitted, a window appears prompting the user to enter the password.

Using this command with both parameters logs the user into the system. Using this command with only one parameter displays the login dialog box.

Logout

Logout

Logs the current user off the system.

LTagSubstitute

LTagSubstitute [language] [sheet]

Use this command to substitute string values of all the tags specified in the selected spreadsheet with the selected language.

[*language*] The selected language name.

[*sheet*] The spreadsheet name of Excel file.

Space is not allowed in Excel spreadsheet name.

The Excel file is created automatically with the project name and located in the current project folder by default. Its name and location can not be changed, or the commands LTagSubstitute and LInsertString will not work.

Monitor

Monitor [file][/Xnnn] [/Ynnn][/Ttag_name]

In edit mode, without the [*file*] parameter, opens the Tag Monitor editor; with the [*file*] parameter, opens the specified tag monitor file.

In run mode, with the [*file*] parameter, opens the specified tag monitor file. With the [/T*tag_name*] parameter, opens a tag monitor containing the specified tags.

[*file*] The name of a tag monitor file without a file extension.

[/X*nnn*] Positions the tag monitor *nnn* pixels from the left edge of the screen. The width depends on screen resolution.

[/Y*nnn*] Positions the tag monitor *nnn* pixels from the top edge of the screen. The height depends on screen resolution.

[/T*tag_name*] The name of a tag. You can use wildcards and can name more than one tag. Precede each tag name with a space and /T.

You can monitor up to 100 tags in one file.

Examples: The Monitor command

Monitor /THopper1\Flow

Opens a tag monitor displaying the tag Hopper1\Flow.

Monitor /THopper1\Flow /THopper\Temp

Opens a tag monitor displaying the tags Hopper1\Flow and Hopper\Temp.

Monitor /T*

Displays information on the first 100 tags in the tag database.

Monitor /THopper*\Divider

Displays information on the first 100 tags called Divider from all folders beginning with Hopper.

Monitor Bread

Displays the tag monitor file called Bread.

NextPosition

NextPosition

Moves focus to the object with the next highest index number.

This command operates on the active graphic display. If no display is active, this command is ignored.

Tab also moves focus to the object with the next highest index number.

See also Position and PrevPosition.

NextWindow

NextWindow

Moves focus to another open graphic display.

Ctrl-Tab and Ctrl-F6 also move focus to another open graphic display.

See also PrevWindow.

Node	<p>Node</p> <p>Opens the Node editor.</p>
NodeDisable	<p>NodeDisable <node name></p> <p>Disables the specified node.</p> <p><<i>node name</i>> The name of the node you want to disable.</p>
NodeEnable	<p>NodeEnable <node name></p> <p>Enables the specified node.</p> <p><<i>node name</i>> The name of the node you want to enable.</p>
NodeSwitch	<p>NodeSwitch <node name> <parameter></p> <p>Permanently changes a device node's address, an OPC[®] node's computer name, server name, and access path, or a DDE node's application and topic.</p> <p><<i>node name</i>> The name of the node that you want to switch to another programmable controller address, server, or application.</p> <p><<i>parameter</i>> One of the following:</p> <p><i>station</i> For device nodes, the physical address for the programmable controller you want to switch to.</p> <p><i>\\computer name\server name[access path]</i> For OPC servers, the computer name, server name, and access path (if in use) for the OPC server you want to switch to. Separate the access path from the server name with the character.</p> <p><i>application </i> For DDE servers, the application and topic, separated</p> <p><i>topic</i> by the character, for the DDE server you want to switch to.</p>

When you use the NodeSwitch command to change a node's address or server information, the change appears in the Node editor's spreadsheet. If the Node editor is open when you use the command, the change won't appear until you've refreshed the display by closing and re-opening the editor.

Password

Password

Displays the Password dialog box so users can change their password. Users must already be logged in to use this command.

Pause

Pause <seconds>

Specifies a pause between the execution of two commands. You can use this command in macros, from the command line, or as an action for a button.

All commands or macros that follow the Pause command must not be asynchronous (preceded by the & sign), otherwise the Pause command has no effect.

<seconds> The number of seconds that will elapse between the execution of the first and second command.

PlayWave

PlayWave <file>

Plays the specified wave file.

<file> The complete path to the wave file, including the .wav extension.

Position

Position <n>

Moves focus to the object with the specified index number.

<n> The object's index number.

These are the objects that use index numbers:

- numeric and string input objects
- buttons
- objects to which you have assigned object keys

This command operates on the active graphic display. If no display is active, this command is ignored.

See also PrevPosition and NextPosition.

PrevPosition

PrevPosition

Moves focus to the object with the previous index number.

This command operates on the active graphic display. If no display is active, this command is ignored.

Shift-Tab also moves focus to the object with the previous number.

See also Position and NextPosition.

PrevWindow

PrevWindow

Moves focus to another open graphic display.

Ctrl-Shift-Tab and Ctrl-Shift-F6 also move focus to another open graphic display.

See also NextWindow.

PrintDisplay

PrintDisplay [parameter] [/Wait=tttt] [/U] [/Pfile] [/Ttag_name]

Prints the specified graphic display to the default printer. If no display is specified, prints the display that has focus.

[<i>parameter</i>]	Specifies which graphic display to print, as follows:
<i>file</i>	The name of a graphic display file without a file extension.
<i>me</i>	The current graphic display (the

display from which the command is executed).

- [/Wait=*time*] Specifies the amount of time to wait before starting to print, where *time* is the time in milliseconds. If you use the /U parameter, specify enough time to allow for the upload of tag values. If you do not specify a time, the default is 2000 milliseconds.
- [/U] Initiates an update of tag values in all input fields before starting to print. This parameter is not necessary with input fields that are updated continuously.
- [/P*file*] Specifies the name of a parameter file that contains tag names to be substituted for placeholders in the display.
- [/T*tag_name*] Specifies one or more tag names to be substituted for placeholders in the display. Separate multiple tag names with commas; do not use spaces.

If the specified display is not currently visible, RSView32 prints the display without making it visible. If this command is issued without specifying a display, and no display has focus, RSView32 logs an error to the activity log.

PrintDisplay prints the entire runtime display, even if parts are covered by other displays. However, PrintDisplay does not necessarily print ActiveX or OLE objects in their runtime state.

The ScreenPrint command prints an image of whatever shows on the monitor. Use ScreenPrint to ensure that the display is printed showing all objects in their current, runtime state (provided they are not covered by other displays).

If you issue the PrintDisplay command when RSView32 is in Edit mode, RSView32 sends the last-saved version of the display file to the printer. Any changes you make after saving the display are not

reflected in the printout. To ensure the printout is up-to-date, save the display before printing it.

Project	Project Obsolete. Use ProjectHide, ProjectRun, ProjectShow, or ProjectStop.
ProjectHide	ProjectHide Hides the Project Manager so it is not visible on the screen. Ctrl-Alt-P also hides the Project Manager if it is visible.
ProjectRun	ProjectRun Starts running the project. This is the same as choosing the Run Project button in the Run Mode tab of the Project Manager.
ProjectShow	ProjectShow Displays a hidden Project Manager. Ctrl-Alt-P also shows the Project Manager if it is hidden.
ProjectStop	ProjectStop Stops running the project. This is the same as choosing the Stop Project button in the Run Mode tab of the Project Manager.
PullForward	PullForward <file> Moves the specified graphic display in front of all other windows. If the specified graphic display is of the Replace or Overlay type, and a display of the On Top type is open, PullForward positions the specified display behind the back-most On Top display and gives the specified display focus. <file> The name of an open graphic display file without a file extension.

If the specified display is of the Replace or Overlay type, and if no other Replace or Overlay displays are open, nothing happens when you issue this command.

PushBack

PushBack <file>

Moves the specified graphic display behind all other windows. If the specified graphic display is of the On Top type, PushBack positions the display behind any other open On Top displays, and in front of any open displays of the Replace or Overlay type.

<file> The name of an open graphic display file without a file extension.

If the specified display is of the On Top type, and if no other On Top displays are open, nothing happens when you issue this command.

Quit

Quit [/R]

Quits RSVIEW32, stopping all project components, and returns to Windows.

[/R] Restarts the computer.

Ramp

Ramp [/V] <tag_name> <value>

Increases or decreases a tag value by a particular value or by another tag's value.

Use this command only with analog tags.

[/V] Performs a read immediately after the write to verify that the value was altered in the programmable controller or server.

This parameter is useful if the network is in poor condition or susceptible to noise interference.

If the verification fails, the error message is logged to the activity log, activity bar, or printer using the

Tag Write category. Specify where to send Tag Write errors using the Categories tab of the Activity Log Setup editor.

- <tag_name>* The name of an analog tag.
- <value>* The amount to add or subtract from the current programmable controller value, as follows:
- plus (+) or minus (-) a numeric value
 - plus (+) or minus (-) a percentage of a numeric value in the form: +value%
 - plus (+) or minus (-) a tag

If Ramp calculates a value that is outside of the tag's minimum and maximum range, it will write the highest or lowest allowable value to the programmable controller or server.

IMPORTANT

If you use the Ramp command in a macro, you must use two percent signs (%%) instead of one. A single percent sign is used in a macro to indicate a parameter. For example, to create a macro that increases the value of tag1 by 75%, you would type: Ramp tag1 + 75%%

Examples: The Ramp command

Ramp Hopper1\Level +50%

If the value of Hopper1\Level is 100, Min = -100 and Max = 900, the command writes the value 600 to the programmable controller.

Ramp Hopper1\Level +75%

If Hopper1\Level is 900, Min = 0 and Max = 1000, the command writes 1000 to the programmable controller because this is the highest allowable value.

Ramp Hopper1\Level delta

If Hopper1\Level is 1000 and delta is –200, the command writes 800 to the programmable controller.

RecipeRestore

RecipeRestore <file>

Reads the values from a recipe file into all input fields in the active graphic display. This command is used with the Recipe field.

<file> The name of the file, without a file extension, tag values are read from.

This command operates on the active graphic display. If no display is active, this command is ignored.

RecipeSave

RecipeSave <file>

Saves the values in all input fields of the active graphic display to a recipe file. This command works with the Recipe field.

<file> The name of the file, without a file extension, to which the values will be saved. If a file of this name does not exist, it is created. If a file of this name does exist, this command prompts you to overwrite it.

This command operates on the active graphic display. If no display is active, this command is ignored.

Remark

Remark <comment>

Writes a text string to an activity log file, which is useful for operators to store comments.

<comment> A string up to 132 characters long. The string can contain any characters, including spaces. Enclose the exclamation mark (!) and the semi-colon (;) in single or double quotes.

To log comments, the Remarks category must be selected in the Activity Log Setup editor.

RTDataServer
Off

RTDataServerOff
Stops running the RSView32 real-time data server.

RTDataServerOn

RTDataServerOn [/Quiet] [/NetDDE]

The RSView32 OPC or DDE server is also known as the real-time data server. This command runs the RSView32 real-time data server, allowing OPC and DDE client applications to read (but not write) tag values.

To enable writes, use the RTDataWriteEnable command.

[/Quiet] Makes RSView32 use the current Remote Users accessibility setting in the Windows Registry. Without this parameter or if the setting is not configured correctly, RSView32 displays a dialog box that allows you to specify the setting when you issue the command.

[/NetDDE] Enables RSView32 as a NetDDE server to remote DDE clients. If the real-time data server is already running, this parameter is ignored.

RTDataWrite
Disable

RTDataWriteDisable [/Quiet] [/NetDDE]

Disables writes by external Windows applications running OPC or DDE, so these applications cannot change tag values in RSView32.

The RSView32 OPC or DDE server is also known as the real-time data server. If the RSView32 real-time data server is not running—that is, if the RTDataServerOn command has not been run—RTWriteDisable also starts the RSView32 real-time data server.

[/Quiet] Makes RSView32 use the current Remote Users accessibility setting in the Windows Registry. Without this parameter or if the setting is not configured

correctly, RSView32 displays a dialog box that allows you to specify the setting when you issue the command.

[/NetDDE] Enables RSView32 as a NetDDE server to remote DDE clients. If the real-time data server is already running, this parameter is ignored.

By default, writes are disabled. Therefore, the RTDataWriteDisable command is required only after the RTDataWriteEnable command has been used.

RTDataWrite Enable

RTDataWriteEnable [/Quiet] [/NetDDE]

Enables writes by external Windows applications using OPC or DDE, so these applications can change tag values in RSView32.

The RSView32 OPC or DDE server is also known as the real-time data server. If the RSView32 real-time data server is not running—that is, if the RTDataServerOn command has not been run—RTDataWriteEnable also starts the RSView32 real-time data server.

[/Quiet] Makes RSView32 use the current Remote Users accessibility setting in the Windows Registry. Without this parameter or if the setting is not configured correctly, RSView32 displays a dialog box that allows you to specify the setting when you issue the command.

[/NetDDE] Enables RSView32 as a NetDDE server to remote DDE clients. If the real-time data server is already running, this parameter is ignored.

By default, writes are disabled.

ScreenPrint

ScreenPrint

Generates a screen print on the default printer specified in Windows.

See also PrintDisplay.

SendKeys

SendKeys <"keystring">

Sends the specified keystrokes to the active window.

<"*keystring*"> A list of keys. The quotes must be included.

The following syntax rules apply:

- Use braces { } to enclose special keys, such as F3 or Enter; for example, "{Enter}" sends Enter.
- Use ^ to send a control key. For example, "^{F2}" sends Ctrl-F2.
- Use + to send a shift key. For example, "+{F3}" sends Shift-F3.
- Use % to send an alt key. For example, "%A" sends Alt-A.

Special keys can be in upper and/or lower case letters. You can type any of the following to represent the special keys on the keyboard:

Backspace, BkSp, BS	Insert
End	Left
Enter	PgDn
Escape, Esc	PgUp
Delete, Del	PrtSc
Down	Right
F1 to F12	Tab
Home	Up

Set

[&]Set [/V] <tag_name> <value>

Writes a value to a tag.

IMPORTANT

If a node has been disabled, the Set command changes the value in the value table but not in the programmable controller.

[&] Forces the command to be executed asynchronously, which makes the command faster.

[/V] Performs a read immediately after the write to verify that the value was altered in the programmable controller or server.

This parameter is useful if the network is in poor condition or susceptible to noise interference.

If the verification fails, the error message is logged to the activity log, activity bar, or printer using the Tag Write category. Specify where to send Tag Write errors using the Categories tab of the Activity Log Setup editor.

<tag_name> The name of a tag.

<value> One of the following:

For analog tags:

- Numeric value within the range specified by the tag's minimum and maximum values.
- Percentage of the total min/max range. The formula is:
value = min + percentage/100 * (max – min)
- Tag name of another analog or digital tag.

For digital tags:

- Numeric value of 0 or 1.
- Tag's on or off label specified in the tag database. Setting a digital tag to its on label writes the value 1, and setting the tag to its off label writes the value 0.
- Name of another analog or digital tag. If it is an analog tag with a value that is *not* 0, the value 1 is written to the digital tag, otherwise the value 0 is written.

For string tags:

- String enclosed in quotation marks. The string can contain any character and can include spaces.
- Name of another string tag.

Examples: The Set command

Set Hopper1\Flow 10

Hopper1\Flow is a tag. This command writes the value -10” to the programmable controller or server.

Set Hopper1\Level 50%

Hopper1\Level is a tag with the minimum specified as -100 and the maximum defined as 900. This command writes the value -400” to the programmable controller or server.

Set Valve\23 open

Valve\23 is a digital tag with its on label configured as open. This command uses the on label for the tag to write the value “1” to the programmable controller or server.

Set Hopper1\Level Preset

Hopper1\Level is a tag and Preset is a tag with a value of “90.” This command writes “90” to the programmable controller or server.

Set String\string1 “open”

String\string1 is a string tag. This command writes “open” to the programmable controller or server.

SetFocus

SetFocus <file>

Sets the focus to a particular graphic display.

<file> The name of a graphic display file without a file extension.

Silence

Silence <tag_name>

If the alarm for the specified tag is configured to use the internal bell, this command silences the computer’s sound.

If the alarm for the specified tag is configured to use the external bell, this command resets the tag associated with the external bell and silences the associated audio device.

<tag_name> The name of a tag that is in alarm. You can use wildcards.

SilenceAll

SilenceAll

Silences the internal and external bells for all tags in alarm.

Using this command is faster than using the Silence command with the asterisk (*) wildcard.

Summary

Summary <file> [/Xnnn] [/Ynnn]

Runs the specified alarm summary file.

<file> The name of an alarm summary file without a file extension.

[/Xnnn] Positions the X coordinate of the alarm summary window to *nnn* pixels from the left edge of the screen. Valid range depends on screen resolution.

[/Ynnn] Positions the Y coordinate of the alarm summary window to *nnn* pixels from the top edge of the screen. Valid range depends on screen resolution.

Suppressed

Suppressed

Opens the Suppressed List editor.

SuppressOff

SuppressOff <tag_name>

Restores alarm reporting for the specified suppressed alarm.

<tag_name> The name of a tag that no longer requires alarm suppression. You can use wildcards.

SuppressOffAll

SuppressOffAll

Restores alarm reporting for all suppressed alarms.

This command is faster than the SuppressOff * command.

SuppressOn

SuppressOn <tag_name>

Suppresses reporting of alarms for the specified tag.

<tag_name> The name of a tag, or a wildcard.

This command is useful when equipment repairs or maintenance would otherwise result in alarms being generated. All alarms for each threshold of the monitored tag are suppressed. Alarm suppression can take place before or after issuing the AlarmOn command.

Toggle

Toggle [/V] <tag_name>

Reads the tag's value and writes back a 1 or 0 as follows:

- if the tag value is 0, Toggle changes it to 1
- if the value is not 0, Toggle changes it to 0

[/V] Performs a read immediately after the write to verify that the value was altered in the programmable controller or server.

This parameter is useful if the network is in poor condition or susceptible to noise interference.

If the verification fails, the error message is logged to the activity log, activity bar, or printer using the Tag Write category. Specify where to send Tag Write errors using the Categories tab of the Activity Log Setup editor.

<tag_name> The name of an analog or digital tag.

Undefine

Undefine <symbol>

Deletes a symbol definition that was created using the Define command.

Example: The Undefine command

Undefine test

Deletes the definition for the symbol “test” that was previously created with the Define command.

Undefine *

Deletes all symbol definitions.

See also Define.

Upload

Upload

In a graphic display, reads a value from the programmable controller or server and displays it in the selected input field.

This command operates on the active graphic display. If no display is active, this command is ignored.

Ctrl–PgUp also uploads the value to the selected input field.

UploadAll

UploadAll

In a graphic display, reads values from the programmable controller or server and displays them in all the input fields.

This command operates on the active graphic display. If no display is active, this command is ignored.

PgUp also uploads all the values to the input fields.

VBAExec

VBAExec <program_name> [parameters]

Runs a VBA program.

<*program
_name*>

The name of the program as specified in the Sub statement.

[*parameters*]

Any argument that a particular program requires. Separate arguments with commas.

Index

Symbols

- .rsv file ■ 2-5
- .wav files ■ A-14, A-36
- {tag} parameter, using
 - to acknowledge alarms ■ A-4
 - with the Identify command ■ A-29
- = (Equal) command ■ A-1

A

- Abort command ■ A-3
- Account command ■ A-4
- Acknowledge command ■ A-4
- AcknowledgeAll command ■ A-5
- Activating RSView32 Runtime ■ 1-5
- Activation disk
 - activation keys ■ 1-6
- ActiveX methods ■ A-30
- ActiveX properties ■ A-30
- Activity commands
 - ActivityBarOff ■ A-5
 - ActivityBarOn ■ A-5
 - ActivityLogSendToODBC ■ A-6
 - ActivityOff ■ A-6
 - ActivityOn ■ A-7
 - ActivityPrintOff ■ A-7
 - ActivityPrintOn ■ A-7
 - ActivityViewer ■ A-7
 - ComStatusOff ■ A-14
 - ComStatusOn ■ A-15
 - EchoOff ■ A-26
 - EchoOn ■ A-27
 - Remark ■ A-42
- Activity Log Viewer
 - opening ■ 2-5
- ActivityLogSendToODBC command ■ A-6
- ActivityPrintOff command ■ A-7
- ActivityPrintOn command ■ A-7
- Adding files to a project ■ 2-10
- Alarm commands
 - Acknowledge ■ A-4
 - AcknowledgeAll ■ A-5
 - AlarmEvent ■ A-7
 - AlarmLogOff ■ A-9
 - AlarmLogOn ■ A-9
 - AlarmLogRemark ■ A-10
 - AlarmLogSendToODBC ■ A-11
 - AlarmOff ■ A-12
 - AlarmOn ■ A-12
 - AlarmPrintOff ■ A-12
 - AlarmPrintOn ■ A-12
 - AlarmViewer ■ A-13
 - HandshakeOff ■ A-28
 - HandshakeOn ■ A-28
 - Identify ■ A-29

- Silence ■ A-48
- SilenceAll ■ A-48
- Summary ■ A-49
- Suppressed ■ A-49
- SuppressOff ■ A-49
- SuppressOffAll ■ A-49
- SuppressOn ■ A-49
- Alarm Log Viewer, opening ■ 2-5
- AlarmEvent command ■ A-7
- AlarmLogRemark command ■ A-10
- AlarmLogSendToODBC command ■ A-11
- AlarmPrintOff command ■ A-12
- AlarmPrintOn command ■ A-12
- Allen-Bradley devices
 - connecting to ■ 2-6
- Application commands
 - AppAbort ■ A-13
 - AppActivate ■ A-13
 - AppStart ■ A-14
- Applications
 - Windows ■ A-13, A-14
- Asynchronous execution
 - and the = (Equal) command ■ A-1
 - and the Pause command ■ A-36
 - and the Set command ■ A-46

B

- Beep command ■ A-14

C

- Changing passwords ■ 3-3
- Channel commands
 - Channel ■ A-14
 - DriverPrimary ■ A-26
 - DriverSecondary ■ A-26
 - DriverToggle ■ A-26
- Channel editor
 - using to select drivers ■ 2-6
- Channels
 - setting up ■ 2-6
- Command line
 - opening ■ 2-6
- CommandLine command ■ A-14

- Commands. See Appendix A, Individual command entries

Communications commands

- ComStatusOff ■ A-14
- ComStatusOn ■ A-15
- DriverPrimary ■ A-26
- DriverSecondary ■ A-26
- DriverToggle ■ A-26
- FTDataServerOff ■ A-28
- FTDataServerOn ■ A-28
- FTDataWriteDisable ■ A-28
- FTDataWriteEnable ■ A-28
- NodeDisable ■ A-35
- NodeEnable ■ A-35
- NodeSwitch ■ A-35
- RTDataServerOff ■ A-43
- RTDataServerOn ■ A-43
- RTDataWriteDisable ■ A-43
- RTDataWriteEnable ■ A-44

Components

- adding ■ 2-10
- changing the path to ■ 2-8
- running ■ 2-5, 2-6

Connecting

- to device drivers ■ 2-6
- to OPC and DDE servers ■ 2-7

D

Data logging

- changing rate at runtime ■ A-16
- on demand ■ A-18
- paths ■ A-18
 - moving data between ■ A-16
 - switching at runtime ■ A-19
- setting up ■ A-18
- starting and stopping ■ A-17

Data logging commands

- DataLogChangeRate ■ A-16
- DataLogMergeToPrimary ■ A-16
- DataLogNewFile ■ A-17
- DataLogOff ■ A-17
- DataLogOn ■ A-17
- DataLogPath ■ 2-11, A-18

- DataLogRenameFile ■ A-18
- DataLogSnapshot ■ A-18
- DataLogSwitchBack ■ A-19
- Data logging paths
 - changing at runtime ■ 2-11
- DatabaseSync command ■ A-15
- Date format
 - specifying ■ 2-2
- DDE
 - and RSLinx ■ A-20
 - and WINtelligent LOGIC 5 ■ A-20
 - and WINtelligent RECIPE ■ A-20
 - connecting to devices ■ 2-7
 - RSView32 as server, enabling ■ A-43
 - for tag writes ■ A-44
 - server ■ 2-7
 - setting up ■ 2-6
 - switching applications ■ 2-7
 - switching topics ■ 2-7
- DDE commands
 - DDEExecute ■ A-19
 - NodeDisable ■ A-35
 - NodeEnable ■ A-35
 - NodeSwitch ■ A-35
 - RTDataServerOff ■ A-43
 - RTDataServerOn ■ A-43
 - RTDataWriteDisable ■ A-43
 - RTDataWriteEnable ■ A-44
- DDE nodes
 - changing at runtime ■ A-35
- Define command ■ A-21
- Derived tag commands
 - DerivedOff ■ A-22
 - DerivedOn ■ A-22
- Device nodes
 - changing at runtime ■ A-35
- Devices
 - direct driver ■ 2-6
 - OPC and DDE ■ 2-7
- Disks
 - ordering ■ 1-5
- Display command ■ A-22

- Download command ■ A-25
- DownloadAll command ■ A-26
- Driver commands
 - DriverPrimary ■ A-26
 - DriverSecondary ■ A-26
 - DriverToggle ■ A-26

E

- EchoOff command ■ A-26
- EchoOn command ■ A-27
- Editors
 - opening ■ 2-5
- Electronic signatures
 - using at runtime ■ 3-3
- Event commands
 - EventOff ■ A-27
 - EventOn ■ A-27
- Expressions
 - in the = (Equal) command ■ A-1

F

- File types
 - sound
 - .wav ■ A-14, A-36
- Files
 - .rsv file ■ 2-5
 - adding to a project ■ 2-10
 - changing paths ■ 2-7
 - referencing the physical file ■ 2-10
- FlushCache command ■ A-27
- FTDataServerOff command ■ A-28
- FTDataServerOn command ■ A-28
- FTDataWriteDisable command ■ A-28
- FTDataWriteEnable command ■ A-28

G

- Global key commands
 - Key ■ A-31
- Graphic display commands
 - Abort ■ A-3
 - Display ■ A-22
 - Download ■ A-25
 - DownloadAll ■ A-26

- FlushCache ■ A-27
- NextPosition ■ A-34
- NextWindow ■ A-34
- Position ■ A-36
- PrevPosition ■ A-37
- PrevWindow ■ A-37
- PrintDisplay ■ A-37
- PullForward ■ A-39
- PushBack ■ A-40
- RecipeRestore ■ A-42
- RecipeSave ■ A-42
- ScreenPrint ■ A-44
- SendKeys ■ A-45
- SetFocus ■ A-48
- Upload ■ A-51
- UploadAll ■ A-51

Graphic displays

- caching, with the Display command ■ A-22
- displaying in background ■ A-22
- moving among ■ A-39, A-40
- positioning ■ A-24
- printing ■ A-37
- reducing call-up time of ■ A-23
- sizing ■ A-24
- using parameter files in ■ A-23

H

- HandshakeOff command ■ A-28
- HandshakeOn command ■ A-28
- Hardware requirements ■ 1-1
- Help command ■ A-29
- Help, online ■ P-1

I

- Identify command ■ A-29
- Input field commands

- Display
 - /U parameter ■ A-23
- Download ■ A-25
- DownloadAll ■ A-26
- NextPosition ■ A-34
- Position ■ A-36
- PrevPosition ■ A-37

- SendKeys ■ A-45
- Upload ■ A-51
- UploadAll ■ A-51

Input fields

- updating tag values in ■ A-23
- using keys with
 - Shift-Tab ■ A-37
 - Tab ■ A-34

- Installing RSView32 Runtime ■ 1-4

- Internet technical support ■ P-1

- Invoke command ■ A-30

K

- Key command ■ A-31

Keyboard

- on-screen ■ A-23

Keys

- See also Special keys

L

- Log files, viewing ■ 2-5

- Logging in at runtime ■ 3-2

- Login command ■ A-32

- Logout command ■ A-32

M

- Method, ActiveX ■ A-30

- Microsoft DCOM ■ 1-5

- Monitor command ■ A-33

- Moving projects between computers ■ 2-2

N

- NetDDE ■ A-43, A-44

- NextPosition command ■ A-34

- NextWindow command ■ A-34

Node commands

- Node ■ A-35
- NodeDisable ■ A-35
- NodeEnable ■ A-35
- NodeSwitch ■ 2-7, A-35

- Node editor ■ 2-7

Number format

- specifying ■ 2-2

O

Online Help ■ P-1

On-screen keyboard ■ A-23

OPC

- connecting to devices ■ 2-7
- RSView32 as server, enabling ■ A-43
 - for tag writes ■ A-44
- server ■ 2-7
 - configuring ■ 1-5
 - setting up ■ 2-6
 - switching ■ 2-7
- switching access paths ■ 2-7

OPC commands

- NodeDisable ■ A-35
- NodeEnable ■ A-35
- NodeSwitch ■ A-35
- RTDataServerOff ■ A-43
- RTDataServerOn ■ A-43
- RTDataWriteDisable ■ A-43
- RTDataWriteEnable ■ A-44

OPC nodes

- changing at runtime ■ A-35

P

Parameter files ■ A-23

Password command ■ 3-3, A-36

Passwords

- changing at runtime ■ 3-3

Path

- to log data to, changing ■ 2-11
- to project files, changing ■ 2-7

Pause command ■ A-36

PlayWave command ■ A-36

Position command ■ A-36

PrevPosition command ■ A-37

PrevWindow command ■ A-37

PrintDisplay command ■ A-37

Project commands

- ProjectHide ■ A-39
- ProjectRun ■ A-39
- ProjectShow ■ A-39
- ProjectStop ■ A-39

Quit ■ A-40

Project directory

- changing the path to ■ 2-7

Project Manager

- adding components to ■ 2-10
- using ■ 2-5

Project paths, updating ■ 2-7

Projects

- adding files to ■ 2-10
- adding users ■ 3-1
- changing passwords ■ 3-3
- changing paths ■ 2-7
- commands
 - ProjectStop ■ 2-11
- editing users ■ 3-1
- logging in ■ 3-2
- moving from another computer ■ 2-2
- opening ■ 2-5
- running ■ 2-3, 2-9
 - automatically ■ 2-9
 - manually ■ 2-9
- stopping ■ 2-11
- updating paths ■ 2-7
- working with ■ 2-1

Property, ActiveX ■ A-30

PullForward command ■ A-39

PushBack command ■ A-40

Q

Quit command ■ A-40

R

Ramp command ■ A-40

Readme file, described ■ P-1

Real-time data server. See DDE server, OPC server

Recipe commands

- Download ■ A-25
- DownloadAll ■ A-26
- RecipeRestore ■ A-42
- RecipeSave ■ A-42
- Upload ■ A-51
- UploadAll ■ A-51

- Remark command ■ A-42
- Requirements, hardware and software ■ 1-1
- RSLinx drivers ■ 2-6
- RSLinx, and DDE ■ A-20
- RSView32 commands. See Appendix A, Individual command entries
- RSView32 Runtime
 - activating ■ 1-5
 - described ■ P-1
 - documentation ■ P-1
 - hardware and software requirements ■ 1-1
 - starting automatically ■ 2-3
 - uninstalling ■ 1-9
- RSView32 Works
 - running projects ■ 2-1
- RTDataServerOff command ■ A-43
- RTDataServerOn command ■ A-43
- RTDataWriteDisable command ■ A-43
- RTDataWriteEnable command ■ A-44
- Runtime
 - changing logging paths ■ A-18
 - changing logging rate ■ A-16

S

- ScreenPrint command ■ A-44
- Security ■ 3-1
 - electronic signatures ■ 3-3
 - logging in ■ 3-2
 - passwords, changing at runtime ■ 3-3
 - users ■ 3-1
 - Windows user list ■ 3-1
- SendKeys command ■ A-45
- Set command ■ A-46
- SetFocus command ■ A-48
- Setting up
 - channel ■ 2-6
 - DDE server ■ 2-6
 - OPC server ■ 2-6
 - project ■ 2-1
- Signature button
 - using at runtime ■ 3-3
- Silence command ■ A-48

- SilenceAll command ■ A-48
- SoftLogix 5 programmable controllers ■ 2-6
- Software requirements ■ 1-1
- Special keys
 - Ctrl-F6 ■ A-34
 - Ctrl-PgDn ■ A-25
 - Ctrl-PgUp ■ A-51
 - Ctrl-Shift-F6 ■ A-37
 - Ctrl-Shift-Tab ■ A-37
 - Ctrl-Tab ■ A-34
 - Enter ■ A-25
 - disabling ■ A-23
 - Shift-Tab ■ A-37
 - Tab ■ A-34

- Stopping projects ■ 2-11
- Summary command ■ A-49
- Suppressed command ■ A-49
- SuppressOff command ■ A-49
- SuppressOffAll command ■ A-49
- SuppressOn command ■ A-49
- Symbol commands
 - Define ■ A-21
 - Undefine ■ A-50

T

- Tag commands
 - = (Equal) ■ A-1
 - DatabaseSync ■ A-15
 - Monitor ■ A-33
 - Ramp ■ A-40
 - Set ■ A-46
 - Toggle ■ A-50
- Tag placeholders
 - parameter for Display command ■ A-24
 - using in commands ■ A-25
- Technical support ■ P-1
- Time format
 - specifying ■ 2-2
- Toggle command ■ A-50

U

- Undefine command ■ A-50
- Uninstalling RSView32 Runtime ■ 1-9

Updating project paths ■ 2-7

Upload command ■ A-51

UploadAll command ■ A-51

User account commands

Account ■ A-4

Login ■ A-32

Logout ■ A-32

Password ■ 3-3, A-36

User accounts

editing ■ 3-1

logging in ■ 3-2

maintaining in Windows Server 2003,
Windows XP, Windows 2000 ■ 3-1

passwords

changing at runtime ■ 3-3

User Accounts editor

opening ■ 3-1

V

VBA commands

VbaExec ■ A-51

Viewing log files ■ 2-5

W

Windows 2000

and RSView32 Runtime ■ 1-4

user list, using ■ 3-1

Windows applications ■ A-13, A-14

Windows Server 2003 ■ 1-4

and RSView32 Runtime ■ 1-4

user list, using ■ 3-1

Windows XP

and RSView32 Runtime ■ 1-4

user list, using ■ 3-1

WINtelligent LOGIC 5 ■ A-20

WINtelligent RECIPE ■ A-20

World Wide Web

using for technical support ■ P-1