MultiView Terminal Emulator

User Guide

MultiView User Guide

Edition 1 May 2008
Document #E-MVUG-MV2007-P053108

Last Updated: 102308

FutureSoft, Inc.
12012 Wickchester Lane, Suite 600
Houston, Texas 77079 USA

Printed in the USA
1.800.989.8908

info@futuresoft.com
http://www.futuresoft.com

Contents

Chapter 1

Introduction

Chapter 2

Installation and Configuration

Chapter 3

Secure Connections

Chapter 4

Printing

Chapter 5

Face-Lifting

Chapter 6

Scripting

Chapter 7

Integration

Chapter 8
Troubleshooting

Chapter 1

Introduction

Introduction to MultiView 2007

MultiView 2007, also called MultiView Terminal Emulator is a comprehensive PC-to-UNIX connectivity package that can be installed on desktops with one of the following:

- Windows 2000 Terminal Services
- Windows 2003 Terminal Services
- Windows 2008 Terminal Services
- Citrix Metaframe

It consists of terminal emulation, file transfer, local and remote printing and a MultiPing connectivity testing tool. UNIX applications are revitalized in less than an hour with the built-in face-lifting tools. Face-lifting enhances the user interface of UNIX applications, providing features such as mouse navigation and Windows-style buttons.

Minimum Requirements

Workstation PC:

- 32-bit Windows operating system
- 256 colors or more
- 800x600 display or above
- 15MB hard disk space (full installation)
- 16MB memory
- Processor type - Intel

Contacting FutureSoft Support

Before contacting Technical Support, please collect all information regarding your question. This should include hardware and software configurations for all systems with MultiView Terminal Emulator, the MultiView version number, any other software running when you experienced the problem, and the exact sequence of steps that preceded the problem.

For prompt and effective service, be sure to provide the following information when contacting us:

Your name
Company name
Company address
Company phone/FAX numbers

Contact us at:

- Ground: Futuresoft, Inc.
  Technical Support Services
  12012 Wickchester Lane Suite 600
  Houston, TX 77079-1222

- E-mail: support@futuresoft.com

- Phone: Monday through Friday
  8:00 am to 5:00 pm (CST)
  281.496.9400
  1.800.261.6357

- Fax: 1.281.496.1090

- Web site: http://www.futuresoft.com

Chapter 2

Installation and Configuration

Installing MultiView Terminal Emulator

Workstation Installation

MultiView can be installed in a number of ways. The easiest and most common method is a workstation installation where the product CD is loaded in the appropriate drive of each workstation and on-screen prompts are followed to install the product. This method is appropriate for a small number of workstations.

Before you begin installation, consider the following:

- Serial number

  To install an evaluation copy, do not supply a serial number.

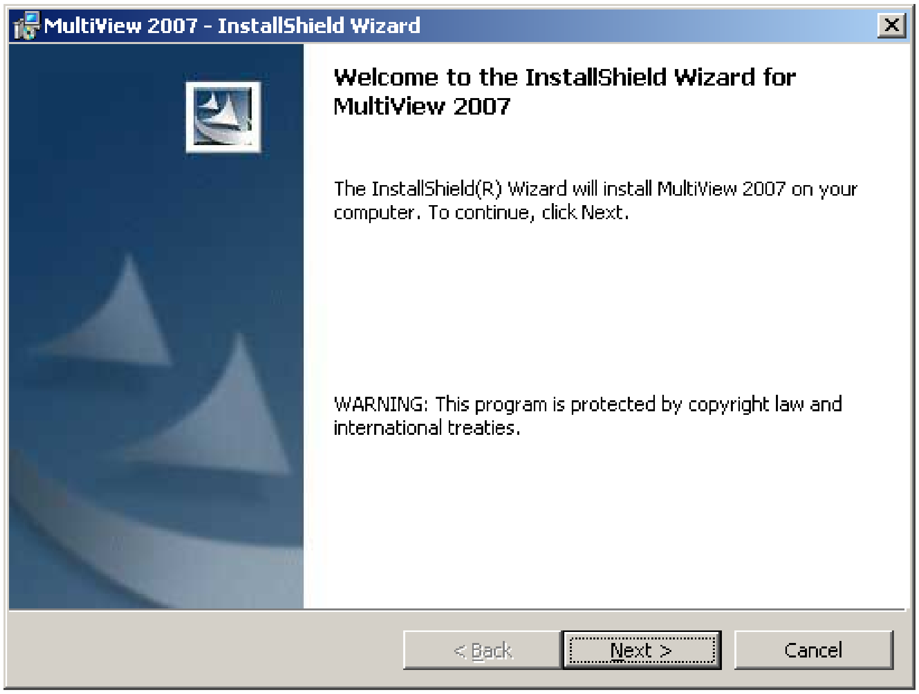
- Installation folder

  Default: `C:\Program Files\MultiView\MultiView 2007`. The installation program allows you to change this location if necessary.

Installation Steps

1 Load the MultiView CD into the appropriate drive.

The InstallShield Wizard starts and displays the Welcome dialog.



MultiView 2007 - InstallShield Wizard

Welcome to the InstallShield Wizard for MultiView 2007

The InstallShield(R) Wizard will install MultiView 2007 on your computer. To continue, click Next.

WARNING: This program is protected by copyright law and international treaties.

Figure 2.1
InstallShield Wizard Welcome dialog

2 In the Welcome dialog, click Next.

3 In the Destination Folder dialog,

- To accept the default installation location, click Next.
- To change the installation location:

  a Click Change.

  b In the Change Current Destination Folder dialog, use standard Windows navigation techniques to choose a new installation folder and click OK.

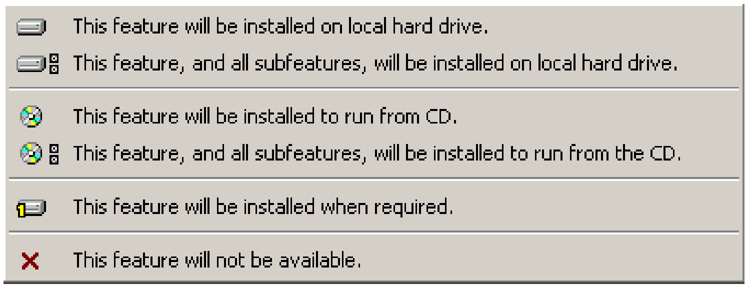
  c Click Next.

4 In the **Setup Type** dialog,

- To install all program features, enable the **Complete** option and click **Next**.
- To select program features to install, enable the **Custom** option and click **Next**.

  In the **Custom Setup** dialog, use one or more of the following to specify installation behavior:

  - Expand a feature group by clicking on the “+” to the left of the drive icon.
  - Specify installation behavior by clicking on the arrow in the drive icon of a feature group and selecting one listed option.



This feature will be installed on local hard drive.

This feature, and all subfeatures, will be installed on local hard drive.

This feature will be installed to run from CD.

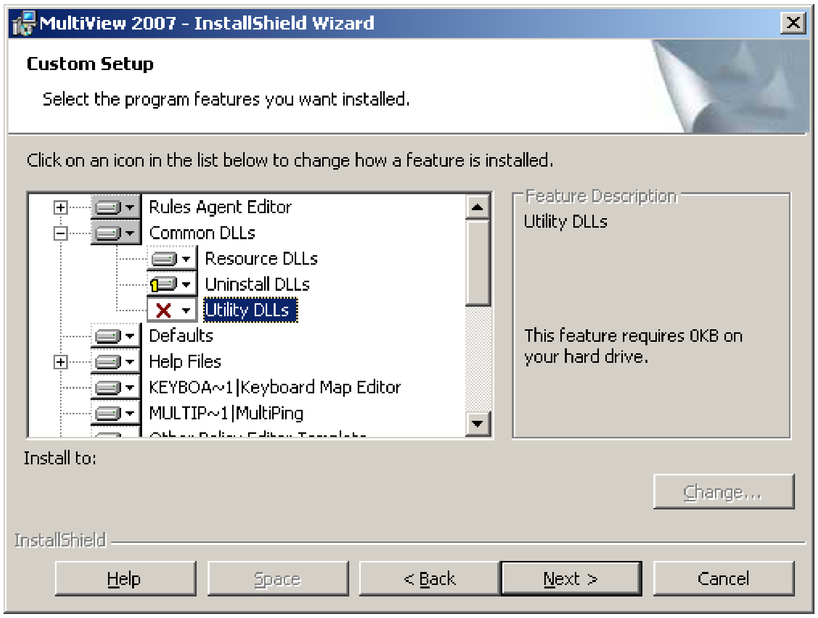
This feature, and all subfeatures, will be installed to run from the CD.

This feature will be installed when required.

This feature will not be available.

Figure 2.2
Individual program feature installation options

After selection, the drive icon changes to reflect the selected installation option.



MultiView 2007 - InstallShield Wizard

Custom Setup

Select the program features you want installed.

Click on an icon in the list below to change how a feature is installed.

```
+ Rules Agent Editor
- Common DLLs
    Resource DLLs
    Uninstall DLLs
    Utility DLLs
  Defaults
+ Help Files
  KEYBOA~1|Keyboard Map Editor
  MULTIP~1|MultiPing
```

Feature Description

Utility DLLs

This feature requires 0KB on your hard drive.

Install to:

Figure 2.3
Program feature icons indicate selected installation option

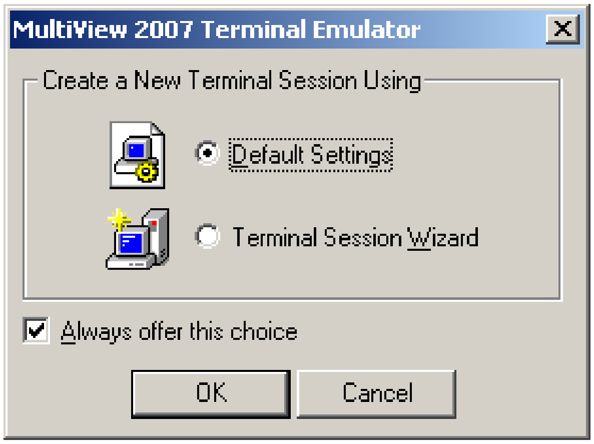
5 In the Ready to Install the Program dialog, click Install.

The Installing MultiView 2007 dialog appears.

6 In the InstallShield Wizard Completed dialog, click Finish.

Initial Configuration

When MultiView is opened the first time, the following dialog appears:



MultiView 2007 Terminal Emulator

Create a New Terminal Session Using

● Default Settings

○ Terminal Session Wizard

☑ Always offer this choice

Figure 2.4
Create a New Terminal Session dialog

To respond to this dialog

1 Choose how new terminal sessions are created:

- To create a new session file with default settings, enable the **Default Settings** option.
- To create a new session file with the Terminal Session Wizard each time, enable the **Terminal Session Wizard** option.

2 Choose to display this dialog

- To view this dialog each time MultiView is opened, select the **Always offer this choice** option.
- To suppress display of this dialog, clear the **Always offer this choice** option.

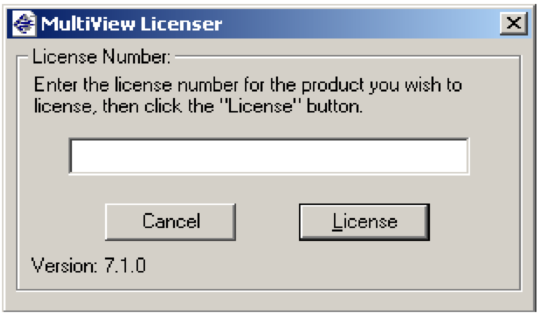
3 Click **OK**.

Serializing MultiView

The MultiView serial number is applied after installation through the licenser utility. Follow these steps to apply a serial number to an installation of MultiView:

1 In the MultiView 2007 program group (under FutureSoft), select Licenser.

The MultiView Licenser dialog appears.



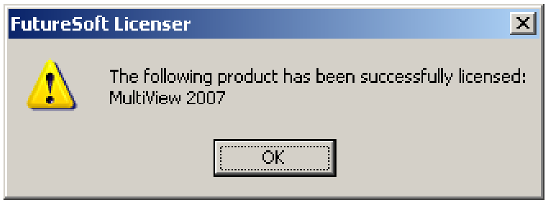
Enter the license number for the product you wish to license, then click the "License" button.

Version: 7.1.0

2 Enter the serial number in the provided field.

3 Click License.

Your license is verified with the following message:



The following product has been successfully licensed: MultiView 2007

NOTE:

Using this procedure on Microsoft Windows Vista requires adminisstrator access rights.

Configuration Files

After configuring product features, the configuration information can be saved as a file. A hierarchical structure is used for configuration files where each component application has its own configuration file:

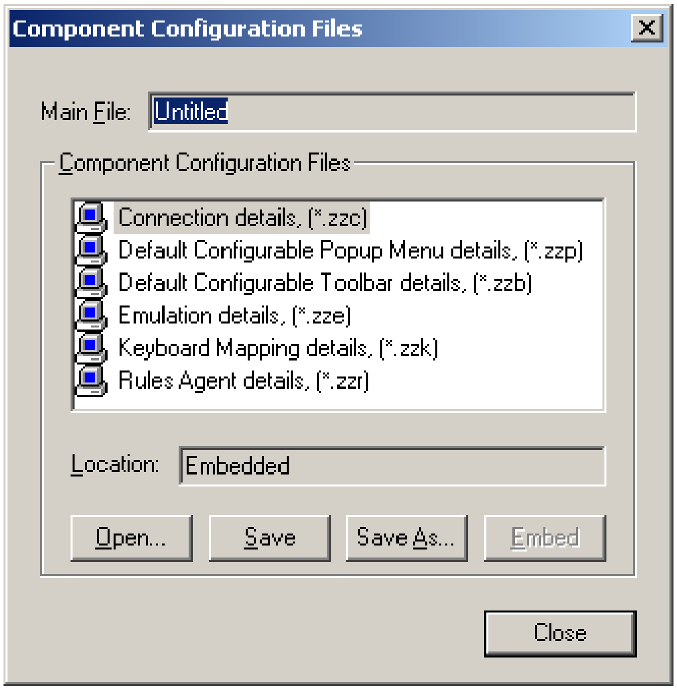
File Transfer	.zzf
Terminal Emulation	.zzt
Wizard Defaults	.zzd

The terminal emulator configuration file (.zzt) contains a number of components that can be embedded into it, or saved as separate files:

Connection Details	.zzc
Configurable Popup Menu	.zzp
Configurable Toolbar	.zzb
Emulation Details	.zze
Keyboard Mapping	.zzk
Rules Agent Editor	.zzr
Script	.zzs
Uninstall File	.zzu

The status (embedded or not) of these configuration files can be managed from within the terminal emulator from the Component Configuration Files dialog, accessed from the File menu with the Components selection.

- To save or embed one of the six component configuration files, simply select it from the list and choose either Save, Save As or Embed.



Component Configuration Files

Main File: Untitled

Component Configuration Files

Connection details, (*.zzc)
Default Configurable Popup Menu details, (*.zzp)
Default Configurable Toolbar details, (*.zzb)
Emulation details, (*.zze)
Keyboard Mapping details, (*.zzk)
Rules Agent details, (*.zzr)

Location: Embedded

Figure 2.5
Component Configuration Files dialog

- To open a component without embedding it into the .zzt file, click Open instead. This preserves the original component file.

Wizard Defaults

The Terminal Session Wizard allows you to quickly and easily create login scripts, make the terminal emulator open a connection and send commands to a remote system. It guides you through establishing a connection and configuring a terminal emulator using the default settings specified during installation.

The Terminal Session Wizard can be used with MultiView Terminal Emulator and can be run after installation, each time the terminal emulator is started or from the Start menu. Each of the parameters that can be set in the Wizard have a default value. This value is determined by the Wizard Default file (.zzd). This file can be set up so that the Wizard only asks for certain parameters such as user name and password.

Editing a Wizard Defaults (.zzd) File

The .zzd file is in plain text format and can be edited in Notepad. The file is found in the MultiView program folder. Double-click the file to open it in Notepad. By default, each step is commented out with a semicolon (;) and terminated with the expression '/skip' as shown in this short extract:

```
; Set transport to use
;Transport=modem /skip

; Name of host to use
;Host=unixhost /skip

; Name of emulator to use
;Emulator=VT320 /skip
```

Remove the semicolons to provide a default value. The /skip expression causes the Wizard skip the step and apply the specified default value. Removing this expression causes the Wizard to prompt the user for a value. If the user clicks Next, the default value specified in the file is used.

Terminal emulator configuration information is stored in a .zzt file. Some of the information can be stored in other files. The .zzt file is configured to point to these external files which can be shared between a number of machines. The .zzt file stores the following:

- Remote application help sequence
- Terminal emulator closure sequence
- Information exchange details
- Font
- Keyboard mapping details
- Connection details
- Start-up options
- .zzs file name containing scripting details

Information that can be saved externally is saved in the following component files:

.zzc configurationfile

Used by file transfer and the terminal emulator to save the connection details of remote systems.

Details saved

- Remote system name
- Username
- Protocol details
- Password request

.zze configuration file

Used by terminal emulators to save emulator properties.

Details saved

- Attribute mapping
- Number of lines in display and scroll back buffer
- Cursor style

.zzk configuration file

Used to save keyboard mappings independently of a terminal emulator. This type of configuration file can be loaded and used by a terminal emulator.

Details saved

- Key mappings
- Keys assigned to escape sequences
- Keys assigned to .zzk script files
- Keys assigned to launching applications

.zzr configuration file

Used by a terminal emulator to save details on configured rules for the application currently in use.

Details saved

- Details on all rules that have been configured

.zzs configuration file

Used by terminal emulators to save details on scripting. This type of configuration file can be loaded and used by a terminal emulator.

Details saved

- Connection and login information and commands to launch remote applications

.zzp configuration file

Used to save the popup menu item preferences.

Details saved

- List of configured popup items

.zzb configuration file

Used to save the configurable toolbar preferences.

.zzf configuration file

Used by file transfer to save settings to be used when connected to a remote system.

Details saved

- General name
- Initial directory
- Anonymous login options
- Auto connect
- Listing type
- Log details
- Listing type details
- Transfer options
- Refresh options

.zzd configuration file

Used by the Terminal Session Wizard.

Details saved

- Default values for the Wizard responses

To save configuration information into a component file

1 On the File menu, select Components.

The Component Configuration Files dialog (Figure 2.1) appears with a list of components that can be loaded or saved separately from the .zzt file.

2 Highlight a component and click Open or Save.

Note

When loading new components, you will save a new .zzt file by choosing Save on the File menu. Close the terminal emulator, then re-run it and open your .zzt file to see the changes with the external components you have added.

The Embed option allows you to incorporate component configuration files back into the .zzt file.

Post Installation Issues

Overriding Default Multinational Setting on VT terminals

For VT220, VT320 and VT420 the default terminal setting is Multinational. To override this default:

1. On the Configure menu, select Terminal.
2. In the Terminal Properties dialog, click Configure.

   The Terminal Settings dialog appears.
3. Click the Advanced tab.
4. Select the Reset sets Default Character Sets as National Mode option (apply check mark).
5. Click OK.
6. To test, reset the terminal emulator by choosing Reset Terminal on the Remote menu.

Chapter 3

Secure Connections

SSH (Secure Shell)

SSH (Secure Shell) is a command-based interface and protocol that enables you to gain access to a remote computer securely.

SSH prevents:

- Stealing of passwords - no passwords are sent in clear text
- Revealing spoofing ID - uses multiple strong authentication methods to prevent this
- ID leakage at one end of the connection - both Server and Client are authenticated

Connecting using SSH1

Connection via SSH requires an SSH Server on the UNIX host. MultiView supports protocol Version 1. This Server must support the authentication methods and encryption ciphers used by MultiView. If RSA authentication is required, the Server must also have a copy of the public part of the key the user wishes to use.

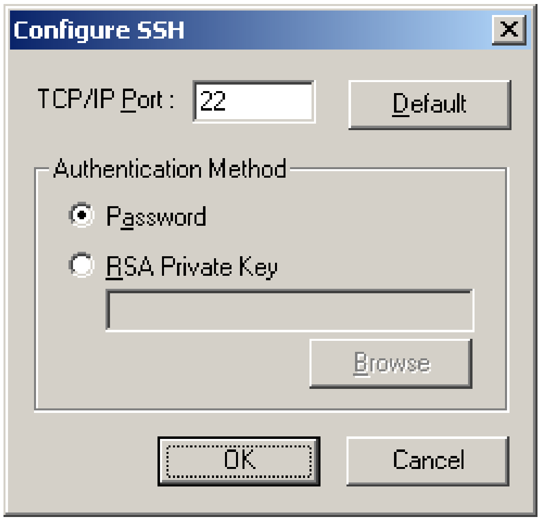
To configure an SSH1 connection from within the terminal emulator:

1 On the **Configure** menu, select **Session**.

2 In the **Session Properties** dialog, click the **Connection** tab.

3 On the **Connection** tab, select **ssh** in the **Connection Type** list and click **Configure**.

The **Configure SSH** dialog appears.



TCP/IP Port : 22

Authentication Method

● Password

○ RSA Private Key

Figure 3.1
Configure SSH dialog

4 In the **Configure SSH** dialog, click **Default**.

The number 22 appears in the TCP/IP field, which is the port that SSH should listen on to make a connection.

Two authentication methods are available, Password and RSA. To select one of these authentication methods, enable the **Username/Password Required** option in the **Connection Properties** dialog.

With the **Password** option enabled in the **Configure SSH** dialog and the **User Name/Password Required** option selected on the **Connection** tab of the **Session Properties** dialog, the extra RSA options are unavailable and the **Username/Password** dialog appears when the connection is started. Similarly if you enable the **RSA Private Key** option, the RSA options are enabled and a **Browse** button allows you to navigate to the private key file to be used. With this authentication method selected, the **Username/Password** dialog appears. However the password used is not the UNIX user name but the password associated with the private key file that the user has elected to use.

Note

If you choose an authentication method but do not enable the Username/Password required option, connection fails.

Connecting with SSH2

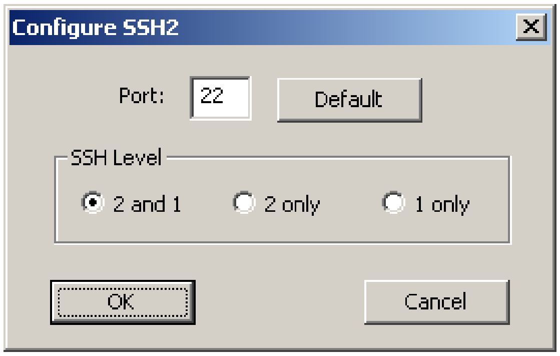
MultiView also supports SSH2 and requires an SSH2 server on the UNIX host. This Server must support the authentication methods and encryption ciphers used by MultiView 2007.

To configure an SSH2 connection from within the terminal emulator

1 On the **Configure** menu, select **Session**.

2 In the **Session Properties** dialog, click the **Connection** tab.

3 On the **Connection Type** dialog, select **ssh2** in the **Connection Type** list.



Configure SSH2

Port: 22

SSH Level: ● 2 and 1 ○ 2 only ○ 1 only

Figure 3.2
Configure SSH2 dialog

4 The default connection time-out is set to 10 (minutes). Change this value if needed.

5 Click **Configure**.

The **Configure SSH2** dialog appears.

6 The default port number used for SSH communications is 22. If needed, change this value to the port that is assigned to SSH2 communications for your network.

7 In the SSH Level group, select one SSH level.

8 Click **OK**.

With any SSH level chosen, the **Username/Password** dialog appears when connection is started.

Chapter 4

Printing

Printing Overview

MultiView offers various options for printing:

- Remote Network printing (also known as LPR/LPD) - uses industry standard TCP/IP protocol to print from a UNIX host to a printer configured on the PC or from the PC to a printer configured on a UNIX host.
- AUX (also known as pass-through printing and transparent printing) - data from the UNIX host is sent in a constant stream to the terminal and redirected to the printer.
- Screen Printing - allows use of the familiar Windows method for printing by selecting Print from the File menu; produces a print of data displayed within the terminal window.

Remote Network Printing

MultiView allows you to print using LPR functionality. Before using LPR print configure both the Client and the UNIX host. The configuration of the UNIX host for Remote Network printing depends on the type of system that you are running. To configure the UNIX host, you need the local and remote printer queue names. Consult your systems documentation for this information. For details on specific systems, there are a number of Remote Network Printing support documents available from the FutureSoft web site:

http://www.futuresoft.com/support/index.htm

PC-to-UNIX printing

Initially you need to make sure that you have enabled LPR/LPD printing on the UNIX host. To enable printing to UNIX, install a printer with these steps:

1 Click Control Panel > Settings > Printers > Add Printer.

The Add Printer Wizard dialog appears.

2 Make the appropriate choice and click Next on each Wizard dialog to complete printer set up.

UNIX-to-PC printing

Initially you need to enable LPR printing on the UNIX host and configure your PC to print to a local printer using Windows applications such as Notepad or Word. Do this by installing a printer through the Control Panel.

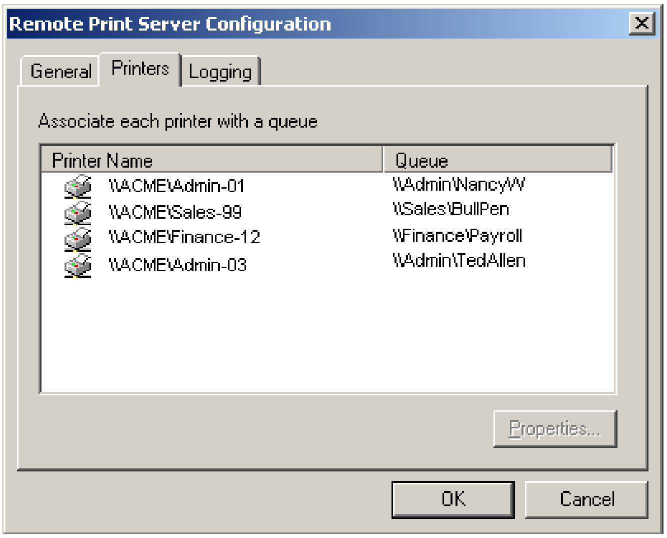
To configure the Remote Print Server

1 Click **Start** > **Programs** > **FutureSoft** > **MultiView 2007** > **Remote Print Server**.

The **Remote Print Server Configuration** dialog appears.

2 In the **Remote Print Server Configuration** dialog, click the **Printers** tab.

A list of currently available printers and the corresponding queue name is shown on the tab. Note the printer queue name that is to be used for Remote Network printing.



Associate each printer with a queue

Printer Name	Queue
\\ACME\Admin-01	\\Admin\NancyW
\\ACME\Sales-99	\\Sales\BullPen
\\ACME\Finance-12	\\Finance\Payroll
\\ACME\Admin-03	\\Admin\TedAllen

Figure 4.1
Remote Printer Server Configuration dialog

To change the printer queue name

a Select the printer in the list on the **Printers** tab.

b Click **Properties**.

c In the **Printer Queue Properties** dialog, make changes as needed.

Enabling the **Do not format received print jobs** option prevents the printer from automatically reformatting jobs that have already been formatted by the UNIX host.

d Click **OK**.

3 Install an appropriate printer on the UNIX host that is set to print to a Windows PC using the queue name noted in step 2.

AUX Printing

UNIX prints by diverting all of the data in a constant stream from the terminal to the printer. The printer starts printing as soon as the data arrives and stops when the job is finished. Windows requires an indication that all data has been sent before it will start the print job. To allow for these two differences in functionality, an end-of-print-job time-out has been added to enable printing functionality.

To enable print job time-out

1 On the Configure menu, select Terminal.

The Terminal Properties dialog appears with the Terminal tab displayed.

2 In Terminal Type select the appropriate terminal type from the drop-down list.

3 Click Configure.

The Terminal Settings dialog appears with the Display tab shown.

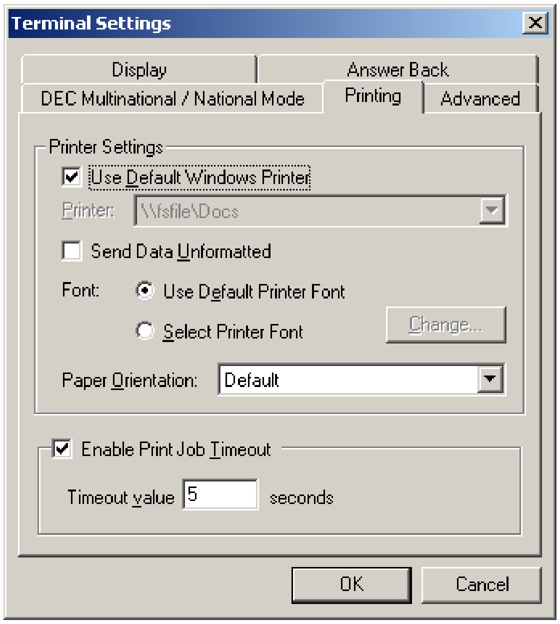
4 In the Terminal Settings dialog, click the Printing tab.

5 On the Printing tab enable the Enable Print Job Timeout option (apply check mark).

6 If needed, enter a new value for the time-out time length.

7 Click OK to close the Terminal Settings dialog.

8 Click OK to close the Terminal Properties dialog.



Printer Settings

☑ Use Default Windows Printer

Printer: \\fsfile\Docs

□ Send Data Unformatted

Font: ● Use Default Printer Font

○ Select Printer Font Change...

Paper Orientation: Default

☑ Enable Print Job Timeout

Timeout value 5 seconds

Figure 4.2
Terminal Settings dialog - Printing tab

To enable sending unformatted data

1. On the Configure menu, select Terminal.

   The Terminal Properties dialog appears with the Terminal tab displayed.

2. In Terminal Type select the appropriate terminal type from the drop-down list.

3. Click Configure.

   The Terminal Settings dialog appears with the Display tab shown.

4. In the Terminal Settings dialog, click the Printing tab.

5. On the Printing tab select to enable the Send data unformatted option.

6. Click OK to close the Terminal Settings dialog.

7. Click OK to close the Terminal Properties dialog.

Because the Server does not know which printers or fonts are available to the Client, changing either one is not possible. Printing tab options provide for setting print orientation to either Portrait or Landscape.

To set the print orientation

1. On the Configure menu, select Terminal.

   The Terminal Properties dialog appears with the Terminal tab displayed.

2. In Terminal Type select the appropriate terminal type from the drop-down list.

3. Click Configure.

   The Terminal Settings dialog appears with the Display tab shown.

4. In the Terminal Settings dialog, click the Printing tab.

5. On the Printing tab select a page orientation from the drop-down list next to the Paper Orientation option.

   To use settings that have been previously configured, choose Default from the list.

6. Click OK to close the Terminal Settings dialog.

7. Click OK to close the Terminal Properties dialog.

While a print job is being received from the Server, a print icon appears on the status bar of the client. If the print job is rejected, the icon appears with a red "X" while the data is being transmitted but the data is not sent to the Client. Due to the nature of this type of printing, the data continues to be sent even though the print job has been rejected. When printing with ActiveX only a default printer can be used.

AUX printing with Java

When the first print job for the session has been sent, a message dialog appears that asks the user if they wish to print now or later. The dialog includes the **Do not show this message again** option. If this option is enabled (check mark applied), this dialog does not reappear for that particular session thus preventing the box from appearing each time a print job has been sent. This allows the user to continue working without interruption.

If the user does want to print, he should

1. Click **Yes.**

   A print preview appears.

2. Click **Print.**

3. In the **Print** dialog, click **Print.**

If the user does not want to print, he should click **No.** The print jobs are placed in a queue and an icon appears in the status bar to show that print jobs are waiting.

To print all outstanding print jobs

1. Click **Commence Printing** from the menu.

   A print preview appears.

2. Click **Print.**

3. In the **Print** dialog, click **Print.**

To disable this printing facility, a new property should be added to your HTML code. See the HTML sample **Enable printing** property. Printing functionality can also be disabled by the user in a configuration dialog. A user may wish to disable this feature in order to reduce network traffic, thus perhaps improving performance.

Print Screen with Java using Netscape

When the user first selects Print from the pop-up menu button in the session window, if printing has not been disabled, a Netscape dialog appears asking them to grant/deny printing. Once this privilege has been granted, the Print dialog appears. Clicking OK shows a print preview of the job from which the user can select Print.

To edit security privileges

1 Click the Security button in the Netscape browser and select Java/javascripts from the resulting dialog.

2 Select your session from the list in the Edit Privileges dialog and click Edit Privileges.

A Java Security dialog appears to show the enabled privileges.

3 Select Printing from within Communicator and click Delete. This removes the printing privilege.

4 Return to the Session window and select Print from the popup menu. The Java Security dialog appears in which you can disable printing by clicking Deny and then selecting (apply check mark) the Remember this decision option.

The next time that Print is selected from the popup menu the user will be unable to print.

To turn printing functionality back on, follow the same procedure as above. The printing functionality is shown in a Never list. Highlight Never, click Remove, and then click OK. The next time that Print is selected the Netscape dialog appears and settings can be changed as needed.

Because of the limitations of the Java control, the only style of printing from the session window when using Java is that of raw text such as would be seen in an application such as Notepad. However, the session window can be set to “print to file” and then this file can be reformatted before being printed. Unfortunately, even this method does not allow for the printing of graphics. The Send Data Unformatted option is not supported in Java.

Chapter 5

Face-Lifting

Face-Lifting in MultiView 2007

Face-Lifting is a powerful set of tools that allow you to give your applications a Windows or Web look-and-feel. The Face-Lifting Wizard is accessed from the **Configure** menu and creates pre-set samples, using a step-by-step approach that can then be configured to your own requirements. The Face-Lifting Wizard can also be accessed after running the Terminal Session Wizard.

Note

The Face-Lifting Wizard will not be available if either Rules Markup or the Rules Agent Editor are active. These must be turned off to run the Face-Lifting Wizard.

The first stage in face-lifting an application is to plan out the changes that are going to be made. Here are three important factors to take into consideration:

1 The cosmetic appearance that is required.

2 The features of the application that is to be face-lifted.

3 The level and type of interaction that will occur with other Windows applications.

When face-lifting an application, you need to be aware of how and when the appearance of your application changes. The following facilities are available to face-lift an application.

Application Appearance

Most remote applications and terminal types support some character attributes to distinguish important items from plain text. These range from simple underlining or reverse video to bold and color. The Attribute Mapping engine can convert existing attributes to new, user-definable ones. The Application Appearance settings allow you, for example, to set the font used in the terminal emulator, replace line-drawing characters with three dimensional alternatives and even add custom bitmap images.

Active Rules

Rules have two key actions. First, they can be used to change the appearance of areas of an application by changing text attributes or adding graphics. Second, rules can make areas of the terminal emulator respond to mouse clicks, performing a range of predetermined actions. This section covers the use of Rules Markup, which is a relatively straight forward method of rules-based face-lifting. Advanced users may wish to create rules directly using the Rules Agent Editor.

Mouse Mapping

Mouse Mapping allows you to make your remote application respond to mouse movement. This works by sending character sequences to the remote system in response to mouse movement or clicks.

Popup Menu

The terminal emulator includes a Windows-style popup menu that, by default, includes functions such as copy and paste. The Popup Menu editor allows you to customize this menu and add functions of your own that control remote applications or launch local applications.

Configurable Toolbar

The Configurable Toolbar works in a very similar way to the popup menu. The key difference is that functions are presented as buttons on a toolbar rather than items on a menu. Functions are customized by choosing an icon from the built-in library or from a bitmap file.

Application Appearance

The Attribute Mapping engine can convert existing attributes such as reverse video, bold or underline to user-definable attributes. Windows-style attributes such as Emboss and Button are used to add some depth to the appearance of the terminal emulator. Where a terminal type and the remote application support color, the colors used can be mapped to a different attribute or color. In addition, other features of the applications appearance can be controlled. An alternative font can be chosen. Existing line-drawing characters can be replaced with three dimensional alternatives. These alternatives can even be used in conjunction with fonts that do not have line drawing characters. A border can be added around the workspace to customize applications further.

Attribute Mapping

Attribute Mapping is managed from the **Attribute Mapping** tab of the **Application Appearance** dialog.

To open the Application Appearance dialog, do one of the following:

- On the **Configure** menu, select **Appearance**.
- On the toolbar, click the **Attribute Mapping Properties** button.

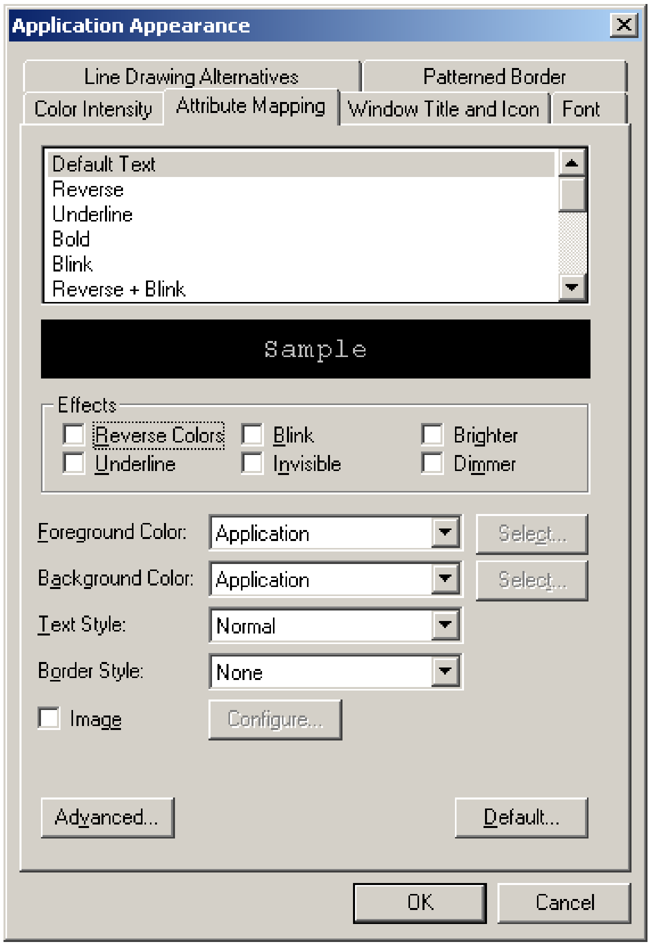
A list of the attributes than can be mapped is shown at the top of the **Attribute Mapping** tab. When an attribute is selected in the list, its "Sample" appearance is shown in the area below the list. The appearance is changed with the options offered below the Sample area.

Options in the **Effects** group are enabled/disabled as needed. The **Invisible** attribute overrides all other settings.

Foreground Color and **Background Color** can be set from one of four options in the pull down lists:

- **Application** uses the application's default color.
- **Windows 3D Objects** uses the color chosen in your Windows color scheme for 3D objects.
- **Windows Window** uses the color chosen in your Windows color scheme for windows.
- **Custom** allows you to choose a specific color from the Windows palette.

Text Style and **Border Style** add 3D attributes. Text styles can be bold, embossed or shadowed. A range of 3D borders are available too, such as button and edit-field.



Effects

☐ Reverse Colors ☐ Blink ☐ Brighter

☐ Underline ☐ Invisible ☐ Dimmer

Foreground Color: Application

Background Color: Application

Text Style: Normal

Border Style: None

☐ Image

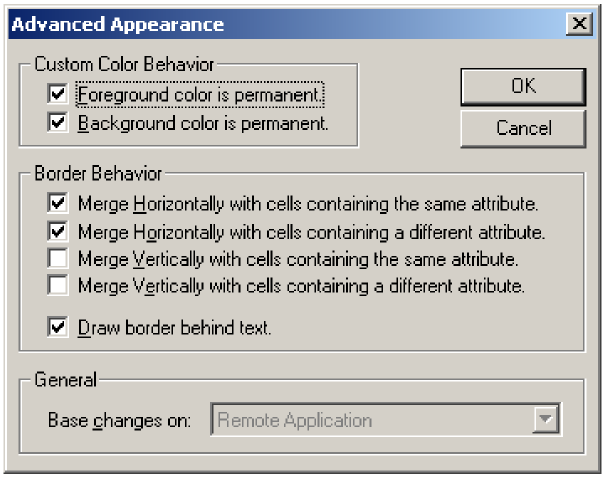
Figure 5.1

Application Appearance dialog - Attribute Mapping tab

The **Advanced Appearance** dialog fine tunes the operation of the Attribute Mapping engine. To open this dialog, click **Advanced** on the **Attribute Mapping** tab.

Custom Color Behavior options set the permanence or exclusivity of the Foreground and Background colors chosen in Attribute Mapping. When the **Foreground/Background color is permanent** option is enabled, your color choices override any color attributes used by the remote application. If these options are not enabled, the application color is used.

Border Behavior options apply specifically to the use of the Border Style attribute. If, for example, you wanted to make a defined area of the terminal emulation window appear as a button or an edit field, you would set the Attribute Mapping engine to merge the border/field horizontally and vertically with cells containing the same attribute. If all merging was switched off, the area would appear as a collection of single character sized buttons or fields.



Advanced Appearance

Custom Color Behavior

☑ Foreground color is permanent.

☑ Background color is permanent.

Border Behavior

☑ Merge Horizontally with cells containing the same attribute.

☑ Merge Horizontally with cells containing a different attribute.

□ Merge Vertically with cells containing the same attribute.

□ Merge Vertically with cells containing a different attribute.

☑ Draw border behind text.

General

Base changes on: Remote Application

Figure 5.2
Advanced Appearance dialog

Note

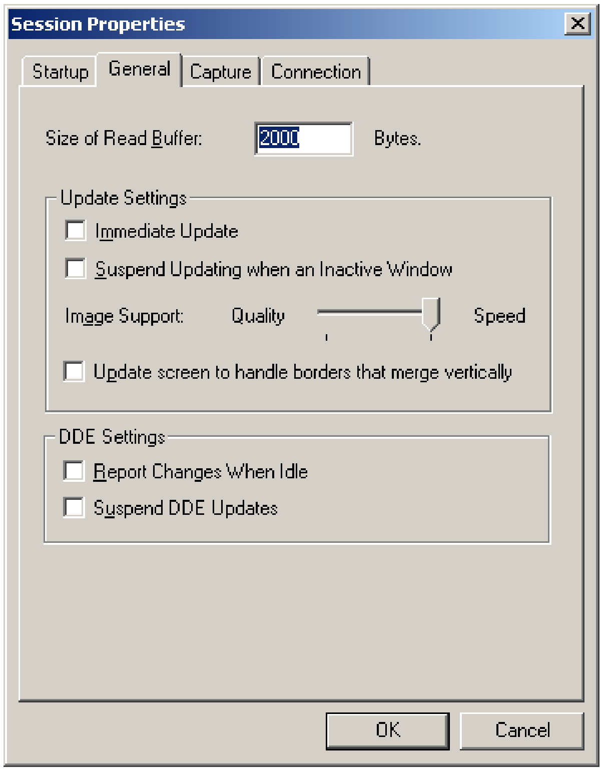
Applying heavy border styles such as Windows Button to certain text sizes can result in the border overwriting the text. The Draw Border Behind Text option makes the terminal emulator draw any border style behind the text, so as not to obscure. it.

Although this border merging facility is extremely useful for giving buttons and edit fields a smooth, Windows look, the extra processing required to make this possible can have an adverse effect on performance in MultiView SE. Therefore, if the face-lifting rules do not require vertically merged borders, this additional processing should be disabled.

To disable border merging:

1 On the **Configure** menu, select **Session.**

The **Session Properties** dialog appears.



Session Properties

Size of Read Buffer: 2000 Bytes.

Update Settings

□ Immediate Update

□ Suspend Updating when an Inactive Window

Image Support: Quality — Speed

□ Update screen to handle borders that merge vertically

DDE Settings

□ Report Changes When Idle

□ Suspend DDE Updates

Figure 5.3
Session Properties dialog - General tab

2 Click the General tab.

3 If your face-lifting rules require vertically merged border, enable the Update screen to handle borders that merge vertically option. Otherwise, disable this option to improve MultiView SE performance.

Note

In older versions of MultiView, this option was enabled by default. If you are using an older .zzt file and need to improve MultiView SE performance, disable this option.

The Image option allows you to add an image which replaces either the foreground or background color.

To add an image

1 On the Attribute Mapping tab of the Application Appearance dialog, enable the Image option (apply check mark).

2 Click Configure.

3 In the Image dialog, do the following:

a Enter a file name or use Browse to use standard Windows navigation techniques to select a file.

b In the Placement drop-down list, select to place the image in either the foreground or background. Drawing an image in the foreground hides the text.

c In the Style drop-down list, select a display style for the image.:

- Stretch stretches the image to fill the entire terminal emulator window.
- Tile per character cell stretches the image to fit a single character cell and tiles it over the entire terminal emulator window.
- Tile with scroll does not stretch the image, but does tile where required. The image scrolls with the text.
- Tile without scroll does not stretch the image, but does tile where required. The image does not scroll with the text.

Note

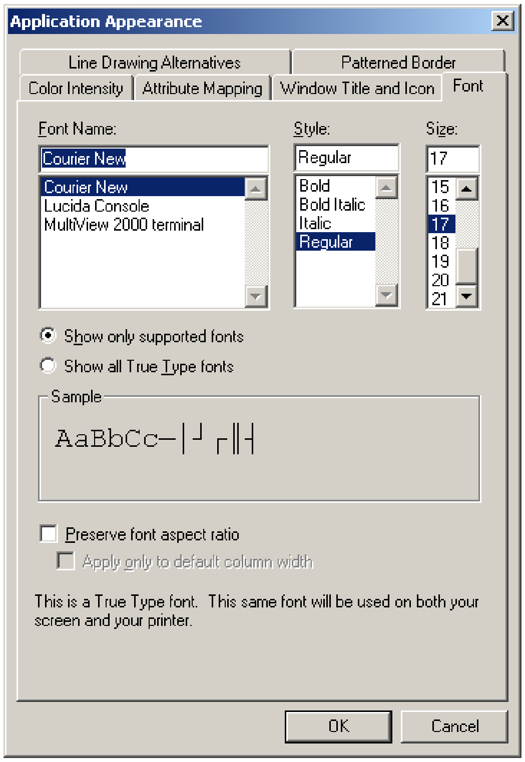
Images are only supported on true-color displays with 65536 colors or more.

By default the MultiView screen is redrawn at the maximum possible speed. At times, this can compromise the appearance of large bitmaps. To ensure that large images always appear correctly, increase the Image Quality option. This option can be found on the General tab in the Session Properties dialog (Configure > Session > General tab). See Figure 5.3.

Fonts

The font used by MultiView is selected on the Font tab in the Application Appearance dialog. MultiView supports Courier New (default font), the MultiView bitmap font and Lucida Console (if installed). Although any TrueType font can be used, MultiView performs best with fixed width fonts because MultiView draws characters within cells whose width is equal to that of the widest character so that the layout of the MultiView window is fixed. The effect that this has on appearance varies from font to font.

To optimize performance, enable the Preserve Font Aspect Ratio and Apply only to Default Column Width options. Preserving the font aspect ratio ensures that the original relationship between the height and width of characters is maintained regardless of the MultiView window size. If your remote application changes column width during a session, try selecting Apply only to Default Column Width to maximize the usage of MultiView. The MultiView bitmap font may provide better results for users with low screen resolutions.



Font Name: Courier New
Courier New
Lucida Console
MultiView 2000 terminal

Style: Regular
Bold
Bold Italic
Italic
Regular

Size: 17
15
16
17
18
19
20
21

● Show only supported fonts
○ Show all True Type fonts

Sample
AaBbCc─│┘┌║┤

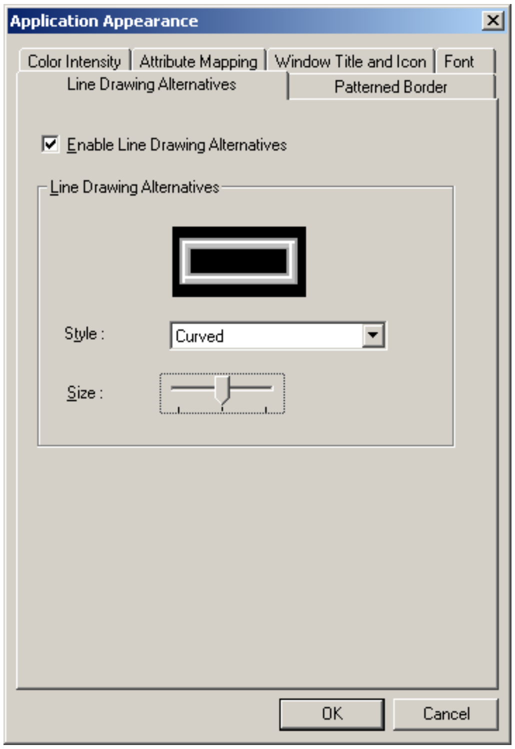
□ Preserve font aspect ratio
□ Apply only to default column width

This is a True Type font. This same font will be used on both your screen and your printer.

Figure 5.4
Application Appearance dialog - Font tab

Line-Drawing Characters

Line-drawing characters can have a 3D appearance. Add this look with the options on the Line Drawing Alternatives tab in the Application Appearance dialog. On the Line Drawing Alternatives tab, a selection of characters from the chosen font are displayed including some line-drawing characters. Standard lines are used by default and give an appearance similar to that of line-drawing characters. Curved lines are available from the pull-down menu. These give a 3D appearance that is commonly associated with Windows applications. The etched lines give an appearance used inside Windows dialogs. The Invisible option can be used to hide line drawing. The size of current and etched lines may be adjusted using the Size slide below the Style edit field.



☑ Enable Line Drawing Alternatives

Line Drawing Alternatives
Style : Curved
Size :

Figure 5.5
Application Appearance dialog - Font tab

Color Intensity

The Color Intensity tab in the Application Appearance dialog provides options to change text brightness and contrast. Disable the Restricted effect on base colors option to not affect base colors or tones. This means that the controls only affect colored items. These controls are useful when the attributes applied by the remote application result in dark colors appearing on a black background. These colors can be lightened without affecting the standard gray text. The color intensity features are also available on a toolbar (View > Color Intensity Bar).

Note

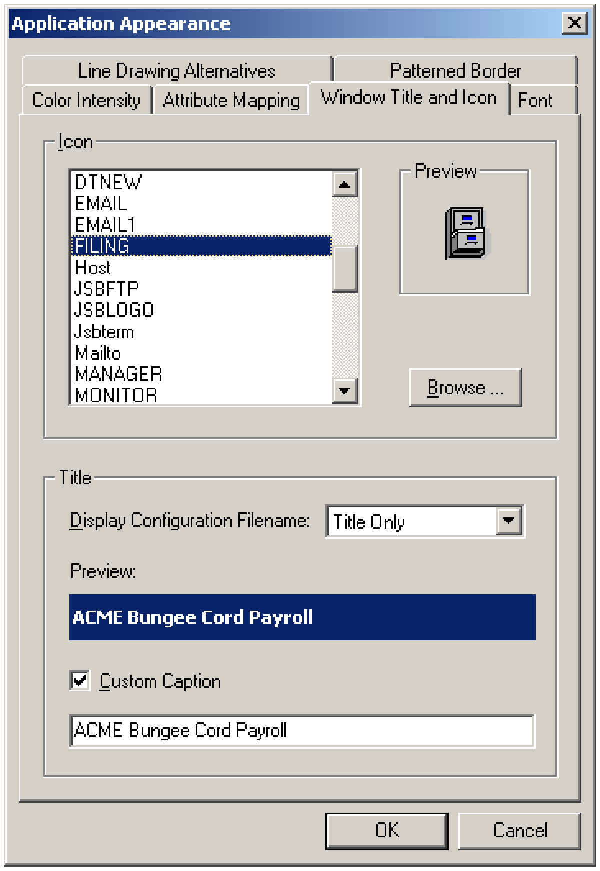
Color intensity only affects application colors and does not affect attributes applied through face-lifting.

Borders

Sometimes the screen text does not completely fill the screen area allocated to it, leaving a border. An image can be chosen to fill this border. In the Application Appearance dialog, select the Patterned Border tab and choose a border from the list or use Browse to use standard Windows navigation techniques to select a Windows bitmap file (.BMP). The MultiView configuration file (.zzt) contains only a pointer to the border bitmap file, not the bitmap itself. When distributing the .zzt file to others, also include the border bitmap file.

Window Icon and Title Bar

The MultiView window icon and title bar text can be changed with options provided on the Window Title and Icon tab of the Application Appearance dialog. To change the icon, select a new icon from the list provided or use Browse to choose a custom icon file. Custom icon files must have the .ico extension and conform to the Windows specification that at least one icon is 32-bits high or more. Change the MultiView display of the current configuration (.zzt) file name by selecting an option from the Display Configuration Filename list. Or, enable the Custom Caption option and enter text in the provided entry field.



Display Configuration Filename: Title Only

ACME Bungee Cord Payroll

☑ Custom Caption

ACME Bungee Cord Payroll

Figure 5.6
Application Appearance dialog - Window Title and Icon tab

Rules Markup Engine

Rules have two key actions. First, they can be used to change the appearance of areas of an application by changing text attributes or adding images. Second, rules can make areas of remote applications respond to mouse clicks, performing a range of predetermined actions.

Rules can apply to:

- All applications - SystemWide Rules
- One single application - Common Rules
- Specific areas of an application screen - Page Rules

Rule Types

SystemWide Rules

SystemWide Rules are stored in the SysWide.zzr file which is located in the MultiView installation folder. Each file is associated with an installation of MultiView. All other rules are saved in a .zzr file that is usually embedded into the .zzt terminal session file. This file can be used to store rules which are common to all terminal sessions. However, in each individual .zzt you can choose to not include these global rules.

Page Based Rules

The user interface of a typical UNIX application is composed of a number of screens or pages of text. Page Switching rules can be used to identify which page a remote application is displaying and to activate rule sets specific to that page, independent of screen contents.

Conditional and Non-conditional Rules

Non-conditional rules apply to a specific area of a remote application. A Non-conditional Rule allows you to change the appearance of an area of text and/or make it mouse-clickable. Conditional rules only become active if a specified text string or attribute property appears in a specified screen location. They can change the appearance of the string and/or make it mouse-clickable.

Rules Markup

There are two ways of creating rules:

- Rules Agent Editor - allows you to manually create rules and to refine rules created using the Rules Markup Engine.
- Rules Markup mode - allows you to face-lift applications quickly and easily with an intuitive mouse-driven interface.

This section covers rule creation using Rules Markup. The following section covers advanced rules design using the Rules Agent Editor. Rules Markup allows you to create rules by simply marking up areas of text in the terminal emulator and selecting the rule type.

Getting Started with Rules Markup

To open the Rules Markup Engine

- On the **Configure** menu, highlight **Rules Agent** and then select **Rules Markup**.

  Two new toolbars appear as shown in Figure 5.7.

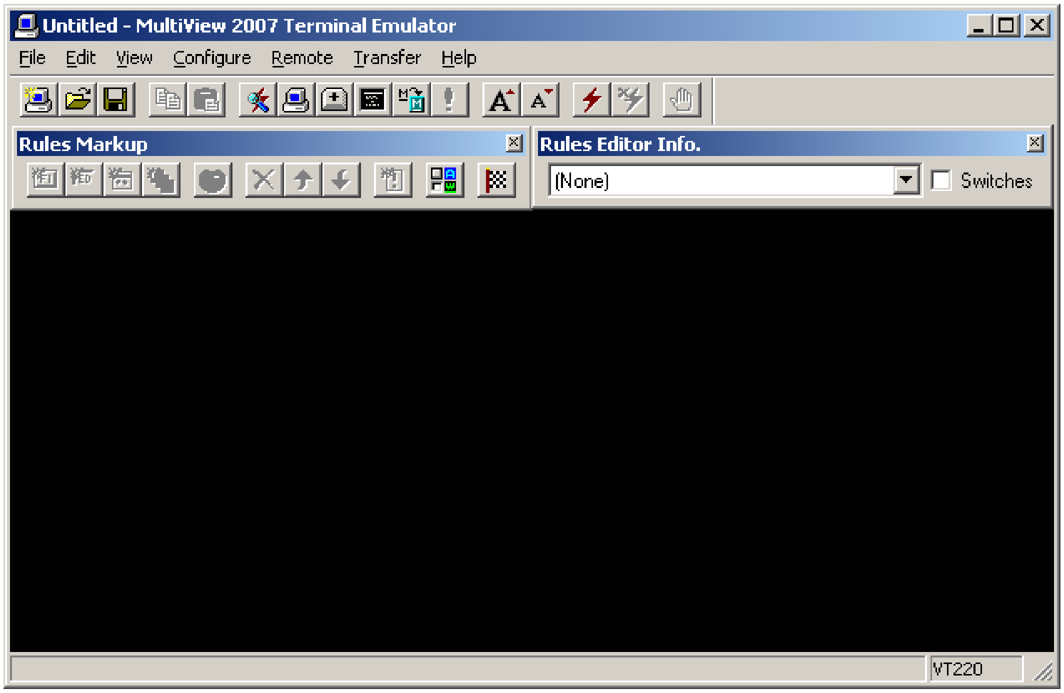


Figure 5.7
Rules Markup window

Unless you are face-lifting a very simple application it is best to start by identifying and defining the pages of your remote application using Page Switching rules. A little planning at this stage pays dividends. It is worth spending some time navigating through all of the areas of your remote application's interface and noting how the appearance of the interface changes. Some elements of the interface, a menu bar for instance, may remain in the same location throughout the application. Other elements may only appear on a single page or may move. Static elements can be face-lifted that are specific to particular pages. Elements can be put into one of these categories:

- Global A specified area will be face-lifted for all screens of the application. For example, if your top line displays your company name you may need this to appear throughout your application.
- Page Based Face-lifting rules which are specific to a particular page. For example an area in the top right of the screen can be changed into a button while another area contains the word "help."

Using the Rules Markup Engine

The Rules Markup Engine creates Non-conditional Rules. That is, rules that are applied based on screen position and not screen content. You can make sets of Non-conditional Rules that apply according to screen content by using Page Switching.

Marking Up Text

There are several ways to mark up text in the terminal emulator. using the following techniques:

Single Word	Double-click the word to select.
Single Line	Double-click a word to select it, then double-click the word again to select the line.
Ragged Right	Simply click and drag the I-beam pointer over the terminal emulator window.
Free-Form (Block Mode)	Hold down **SHIFT** and then click-and-drag an area.
Entire Contents	Either use Free-form markup or choose **Select All** from the **Edit** menu.

Creating Pages

You can create rule sets that become active depending on screen contents. Mark up the text (on one line) for which you wish to activate a page rule set and click **Create Page** on the toolbar. You are prompted for a name for the new page rule set. The default name matches the text you have highlighted. The application's current page is displayed in the Page Switching toolbar. If there is no page defined for the part of the application that is being displayed (None) is shown in the toolbar. When a page name is displayed in the toolbar, any new rules created apply only to that page (Page Rules). If no page name is displayed, Common Non-conditional rules will be created that apply to the whole application.

Creating Rules

Rules are created by marking up text or an area and clicking the appropriate rule button:

Create Button Rule

When you create a Button rule, you are prompted to enter a sequence. This sequence is sent to the remote application when Create Button Rule is clicked. From the Define Sequence dialog you can manually enter characters in the edit field or click on the buttons provided to enter common control characters. The sequence for special terminal keys such as the function keys can be entered by clicking the Terminal Key button and choosing the appropriate key or key combination from the list.

Create Edit Field Rule

An edit field is an area of the screen with the appearance of a Windows edit field. The edit field has default black text on a white background and a 3D border, giving a recessed look. Create an edit field by marking up the area required and clicking the Create Edit Field Rule button. No parameters are required.

Create Application Rule

Application Rules allow you to create links between your remote applications and your local Windows applications. Application Rules can invoke and pass data to local applications using active URLs. The default SystemWide Rules are actually Application Rules that find URLs such as web links (e.g., www.futuresoft.com) and make them mouse-clickable. When a user clicks on the face-lifted area the rule invokes the corresponding local application (e.g., a web browser) and passes the URL to it.

When you create your own Application Rules by marking up an area of text and clicking the Create Application Rule button. A Define Action dialog appears into which you can enter a URL. The URL can be composed entirely of text entered in this dialog. Alternately, the rule can take a parameter from the marked up text. For example, if "www.futuresoft.com" appears and you want to send "www.futuresoft.com" to the local application you would enter:

http:// into the dialog and a parameter.

Create Visual Effect Rule

Visual Effect Rules only affect the appearance of an application and do not make an area mouse-clickable. They are great for adding finishing touches to your face-lifted application. To create a Visual Effect Rule, mark up the area of text required and click the Visual Effect Rule button.

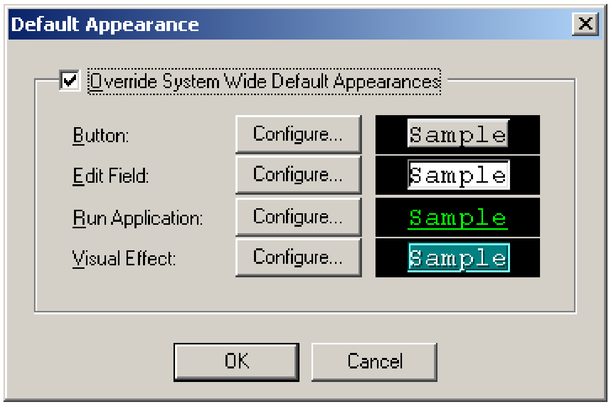
Rule Precedence

When rules are created in Rules Markup, they are added to the top of the list. The Rules Agent then processes the rules in the order that they appear in the list. Rule appearance is applied in a similar way to layers in a drawing package. Rules overwrite the appearance of rules further down the list.

Rule Appearances

All four rule types have a default appearance. If you change the appearance and want to return the default settings, click **Default** on the **Attribute Mapping** tab in the **Application Appearance** dialog (Figure 5.1). Change rule appearance defaults in the **Default Appearance** dialog (**Configure** > **Rules Agent** > **Rules Editor** > **Tools** > **Default Appearances**).

In the **Default Appearance** dialog, click **Configure** to the right of the rule type to make changes. Attributes and images are defined in the same way as attributes and images in Attribute Mapping. You can also change the appearance of individual rules by marking up the area affected by the rule and clicking the **Rule Appearance** button.



Default Appearance

☑ Override System Wide Default Appearances

Button:	Configure...	Sample
Edit Field:	Configure...	Sample
Run Application:	Configure...	Sample
Visual Effect:	Configure...	Sample

Figure 5.8
Default Appearance dialog

Deleting a Rule

To delete a rule, mark up the area affected by the rule and click the **Delete** button.

Display the Euro Symbol

The Euro symbol is not normally available as a character on a terminal. To display the Euro symbol, MultiView's face-lifting capabilities allow for text substitution through the Rules Agent Editor. The Euro symbol's Unicode decimal value of 8364 is used. If this does not give the required results you may need to update your system fonts. For updated information on using the Euro symbol, visit:

> http://www.futuresoft.com/support/index.htm

Mouse Mapping

MultiView provides two levels of mouse support. Server-side Mouse Messaging is the most advanced and requires specifically written code running on the Server. Configure the action of the mouse buttons on the **Mouse Mapping** tab of the **Application Properties** dialog.

Options are offered for Drag, Single-click and Double-click. The default settings provide some immediate mouse control over a remote application. For example, holding down the right button and dragging within the terminal emulation window sends cursor key strings to the remote application, allowing navigation of menus and windows. All of these actions can be customized or deactivated as required.

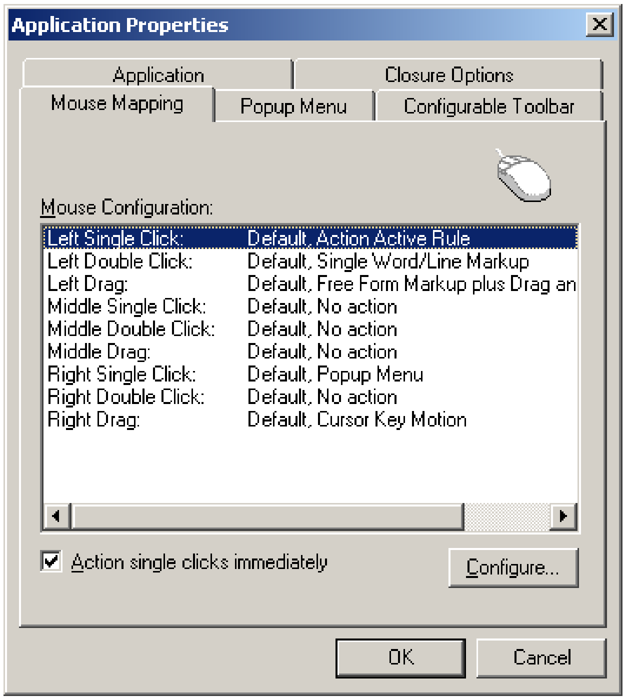
To define a sequence

1 Highlight an action in the **Mouse Configuration** list.

2 Click **Configure**.

3 In the corresponding **Application Properties** dialog, select or disable desired options.

4 Click **OK**.



Mouse Configuration:

Left Single Click:	Default, Action Active Rule
Left Double Click:	Default, Single Word/Line Markup
Left Drag:	Default, Free Form Markup plus Drag an
Middle Single Click:	Default, No action
Middle Double Click:	Default, No action
Middle Drag:	Default, No action
Right Single Click:	Default, Popup Menu
Right Double Click:	Default, No action
Right Drag:	Default, Cursor Key Motion

☑ Action single clicks immediately

Figure 5.9
Application Properties dialog - Mouse Mapping tab

Disabling the **Action single clicks immediately** option forces MultiView to wait for a double-click action to occur. The wait time is set in the Windows Mouse Control Panel.

Popup Menu

The popup menu that appears when you right-click in the terminal emulator can be configured to suit a specific application. The configuration file, .zzp, can be saved in its own unique file, as well as embedded in the .zzt terminal configuration file. Configuration is controlled from the Popup Menu tab in the Application Properties dialog. Existing items can be hidden by clearing the option if required and new items can be added and removed.

To add a new menu item

1 Click **Add.**

2 In the **Popup Menu Type** dialog, choose an action type from the following:

- **Action As** - allows you to define an action using URL and Parameter tokens. Click the popup button to choose from a list of actions. Text selected by the mouse is used as the parameter. Select **Browse** from the popup menu to select a file or application to open as the action.

- **Send Sequence** - allows you to assign a character sequence to a menu selection. Use the **Define Sequence** dialog to create the character sequence.

- **Play Script** - allows you to assign a popup menu entry to play a script file.

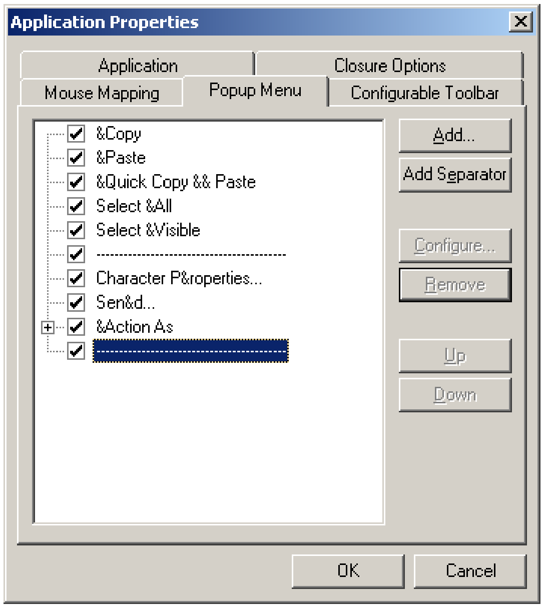
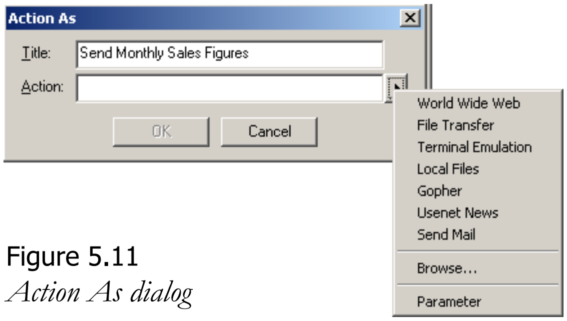
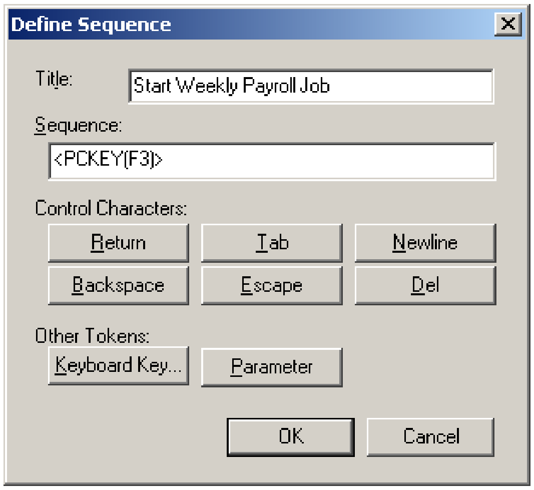


Figure 5.10
Application Properties dialog - Popup Menu tab


Title: Send Monthly Sales Figures

Action:

Figure 5.11
Action As dialog


Title: Start Weekly Payroll Job

Sequence:
<PCKEY(F3)>

Figure 5.12
Define Sequence dialog

Configurable Toolbar

A dockable toolbar is available in the terminal emulator that is fully user configurable. You can control the appearance of the buttons and the action that they have. The toolbar buttons are shown on the Configurable Toolbar tab in the Application Properties dialog (Configure > Application > Configurable Toolbar tab). A title can be specified for the toolbar in the Title field. This is the title that appears above the toolbar when it is not docked. Using this tab, buttons can be added and removed.

To add a new button

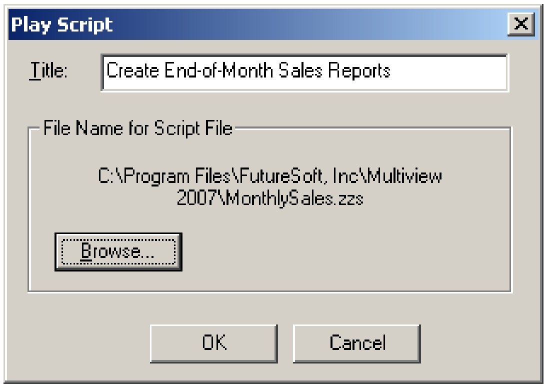
1 On the Configurable Toolbar tab, click Add.

2 Choose one of the following:

- Action As - allows you to define an action using URL and Parameter tokens in the same way that you did for popup menu. Click the popup button to choose an action of a parameter. Text selected by the mouse is used as the parameter. Use Browse to select a file or application to open as the action.

- Send Sequence - allows you to assign a character sequence to a toolbar button. Use the Define Sequence dialog to create the character sequence.

- Play Script - allows you to assign a script file to a toolbar button. When the user clicks the toolbar button the script is executed. Use the Play Script dialog to assign a script file to the toolbar button.

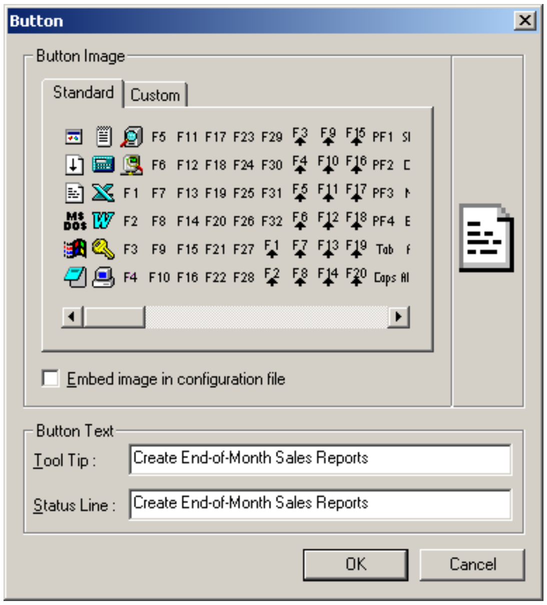


Title: Create End-of-Month Sales Reports

File Name for Script File

C:\Program Files\FutureSoft, Inc\Multiview 2007\MonthlySales.zzs

Figure 5.13
Play Script dialog


Button
Button Image
Standard
Custom
F5 F11 F17 F23 F29 F3 F9 F15 PF1
F6 F12 F18 F24 F30 F4 F10 F16 PF2
F1 F7 F13 F19 F25 F31 F5 F11 F17 PF3
F2 F8 F14 F20 F26 F32 F6 F12 F18 PF4
F3 F9 F15 F21 F27 F1 F7 F13 F19 Tab
F4 F10 F16 F22 F28 F2 F8 F14 F20 Caps
Button Text
OK
Cancel

☐ Embed image in configuration file

Tool Tip : Create End-of-Month Sales Reports

Status Line : Create End-of-Month Sales Reports

Figure 5.14
Button dialog

3 When you have defined an action for the button, click OK.

4 In the Button dialog:

a Select a button. You can select a button provided on the Standard tab or you can select a custom bitmap file from the Custom tab. The bitmap file must:

- Have a size of 16 pixels high X 15 pixels wide.
- Use the standard 1- color Windows palette
- Have a transparency color of magenta (255,0,255 in RGB)

Use the Browse button on the Custom tab to use standard Windows navigation techniques to select a file. The image appears in the right pane of the Custom tab.

b Enter tool tip text for the new button.

c Enter status line text for the new button.

d Click OK.

To embed the image files for button icons into the Configurable Toolbar configuration file (.zzb), enable the Embed image in configuration file option.

A toolbar created using this facility behaves in the same way as other toolbars. It can be dragged around the screen or docked. The appearance and behavior of any of the buttons can be changed at any time by displaying the Configurable Toolbar tab, choosing a button and clicking Button or Configure. The view of the configurable toolbar is off by default and must be turned on (View > Configurable Bar). The configurable toolbar is embedded in the .zzt file by default but you may wish to save the configuration file (.zzb) separately from the terminal emulator file (.zzt). By doing this, you can create more than one Configurable Toolbar. This allows you to use the Rules Agent to change the Configurable Toolbar based on changes in the appearance of the remote application.

Rules Agent Editor

The Rules Agent effectively enhances an existing text based user interface of a remote application with some features of a Graphical User Interface (GUI). In the previous section we discussed rule creation using Rules Markup. If you use a large, complex application you may find Rules Markup limited. The Rules Agent Editor allows you to create rules that can do any of the following:

- Change the appearance of the terminal emulator by:
  - Changing Attributes
  - Adding bitmap graphics
  - Substituting text
- Create Windows dialogs
- Play sounds
- Make areas of the terminal emulator respond to mouse clicks
- Create popup menus

To open the Rules Agent Editor, do one of the following:

- Click the Rules Agent Editor toolbar button.
- On the Configure menu, highlight Rules Agent and then select Rules Editor.

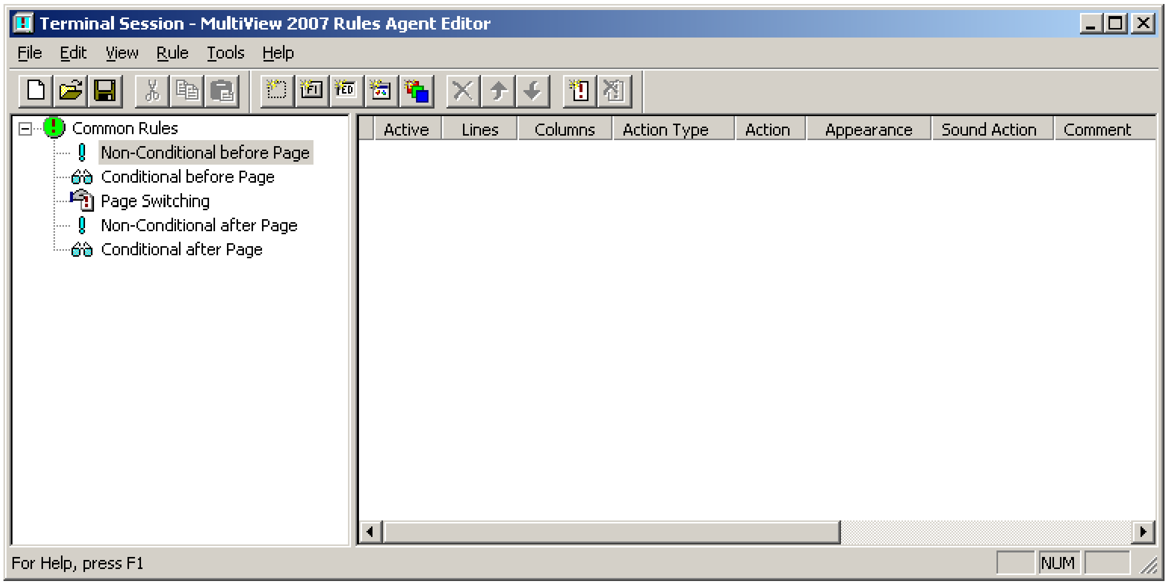


Figure 5.15
Rules Agent Editor window

All rules are organized into a tree that is similar in appearance to Windows Explorer. The tree is located to the left of the rules in the Rules Agent Editor. The first main branch contains Common Rules. These rules are applied to an entire application in the order that they appear in the tree. Branches of Page rules appear below this branch. Page rules apply to a specific part of an application. There are seven classes of rules:

- Non-conditional Before Page Rules - these rules apply to part or all of the terminal emulator based on position, regardless of its contents.
- Conditional Before Page Rules - Conditional Rules become active when the Rules Agent finds a text string in the terminal emulator that matches the rule's Find section. The Find section is composed of a set of parameters designed to match text strings that appear in the remote application.
- Page Switching Rules - Like Conditional Rules, Page-Switching rules also contain a Find specification. When the Rules Agent finds a text string that matches the Find specification, the rule becomes active. Page-Switching rules do not, however, affect the application directly. Instead, they activate groups of Page Rules and apply any Page Functions that have been configured. In this way, different sets of rules can be made to apply to different parts of an application.
- Non-conditional Page Rules - These rules work in the same way as Non-conditional Rules but they are only made active by a Page-Switching Rule.
- Conditional Page Rules - These rules work in the same way as Non-conditional Rules but they are only made active by a Page-Switching Rule.
- Non-conditional After Page Rules - These rules apply to part or all of the terminal emulator, regardless of its contents, on all parts of the application following the present page.
- Conditional After Page Rules - These have the same action as Conditional rules but only apply to pages of the application which follow the present one.

Anatomy of a rule

Non-conditional rules and Non-conditional Page rules

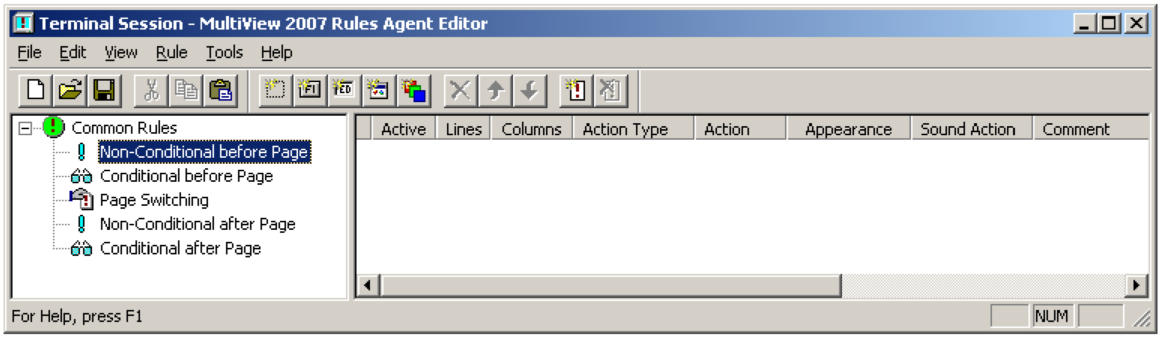
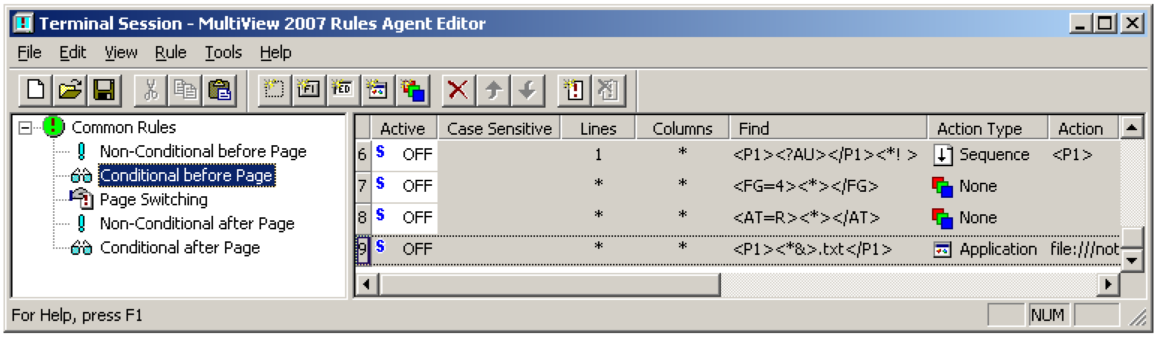


Figure 5.16
Non-conditional and Non-conditional Page rules window

Non-conditional rules include three sections:

- First section has two parameters that define the area that the rule affects.
- Second section, called the Action Section, defines what effect the rule will have. Action Type defines the action. Action contains the necessary parameters.
- Third section defines the appearance of the rule. The appearance is controlled in the same way as Attribute Mapping. Like Attribute Mapping, images can be used to replace foreground or background colors.

Conditional rules and Conditional Page rules



	Active	Case Sensitive	Lines	Columns	Find	Action Type	Action
6	$ OFF		1	*	<P1><?AU></P1><*! >	Sequence	<P1>
7	$ OFF		*	*	<FG=4><*></FG>	None	
8	$ OFF		*	*	<AT=R><*></AT>	None	
9	$ OFF		*	*	<P1><*&>.txt</P1>	Application	file:///not

Figure 5.17
Conditional and Conditional Page rules window

Conditional rules have three sections.

- The first section has two cells that define the area that the rule affects. By default, these cells contain wildcards that make the rule apply to the whole terminal emulator.

- The second section has a cell labeled Find that contains an expression that the Rules Agent uses to match against screen text. If the Rules Agent finds a text string that matches the Find expression, it applies the rule. Once activated, Conditional rules apply actions in the same way as Non-Conditional rules.
- The third section defines the appearance of the rule. The appearance is controlled in the same way as Attribute Mapping. Like Attribute Mapping, images can be used to replace foreground or background colors. Alternative text can be substituted for text found by the rule.

Page Switching rules

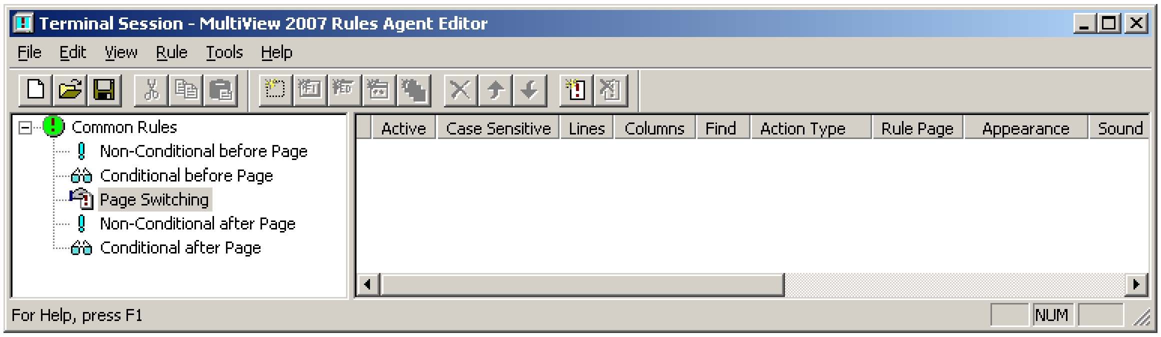


Figure 5.18
Page Switching rules window

Page Switching rules are different from Conditional and Non-conditional rules in that the only action they can perform is to enable a set of Page rules and Page Functions. They perform this action when the Rules Agent matches text to their Find expression. Page Switching rules do not affect the appearance of the terminal emulator by themselves. All rules also have an Active cell that enables or disables the rule. To make a change, double-click in the cell. You can also specify an area of the terminal emulation window in which to search.

The Find Section

Non-conditional Rules are applied to the terminal emulator regardless of what is displayed within it. Page Switching and Conditional rules are only applied when the Rules Agent finds a match between the Find section of a rule and the text string in the remote application. Find sections are composed of three key elements. The first two cells define the area that the Rules Agent should search within. The Line and Column cells contain an "*" by default, telling the rules agent to search the entire window. To search only a specific line or column, enter the line number or column number that you would like to search from. Ranges of values can also be specified using a hyphen, for example 1-15 would cover all values between 10 and 15 inclusive. The rule can also be made case sensitive by enabling the Case Sensitive option (column).

Find Tokens

Tokens provide a powerful method of searching for certain types of strings. You might, for example, want to search for all of the menu titles in an application. Using tokens you could create an expression that will match the exact characteristics of a menu title: "Any upper-case letter followed by any number of lower-case letters followed by a space."

A wide range of tokens are available by right-clicking in the Find cell and choosing a token from the popup menu. Choose the type of token required from the list and it will be placed at the cursor position in the cell. Alternatively, if you are familiar with tokens, you may enter them directly into the Find cell using the keyboard. If you use the keyboard to insert tokens, be aware that the Rules Agent Editor is case-sensitive.

Single Character Tokens

Single Character tokens match individual characters of the type specified.

<?> Any single character (including a space).

<?! > Any character *except* the character inserted in the space. You can enter any keyboard character in the space.

<?A> Any alphabetical character (i.e., only a letter, not a number or other symbol).

<?AU> Any uppercase alphabetical character.

<?AL> Any lowercase alphabetical character.

<?U> Any URL character. Typical URL strings would be "www" or "ftp" addresses.

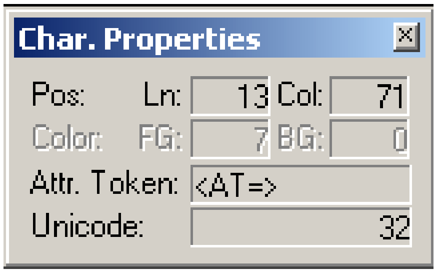
<?#> Any numeric character.

<?&> Any alphanumeric character (i.e., only a letter or a number).

<?C=> The Unicode Character token allows you to specify any unicode character that is supported by the terminal type being used. Specify the unicode value in the space. e.g., <?C=0149> matches "•".

To find the unicode value of any character in the terminal emulator, right-click the character and select Character Properties from the popup menu. The unicode value is shown:

- In the Character Cell Properties dialog. This dialog also shows the exact location of the character and its attributes.
- Character Properties toolbar (View > Character Properties Toolbar)



Pos: Ln: 13 Col: 71
Color: FG: 7 BG: 0
Attr. Token: <AT=>
Unicode: 32

Figure 5.19
Character Properties toolbar

The Single Character tokens can also be preceded by a numeric parameter that specifies the number of single characters to searched for. For example, <20?> would find 20 characters of any type. Alternatively, <20?&> would find sequences of 20 alphanumeric characters.

URL Tokens

The URL tokens search specifically for URL strings such as “www” or “ftp” addresses and can be used in both the Find and Action cells.

Wildcard Tokens

Wildcard Tokens match strings of one or more characters containing the type specified:

<*> Any characters (including spaces).

<*! > Any string of characters *except* the character or string inserted in the space. You can type any keyboard character(s) in this space. Other tokens can also be used in the space.

<*A> Any string of alphabetical characters (i.e., alpha letters only, no numbers or symbols).

<*AU> any string of uppercase alphabetical characters.

<*AL> Any string of lowercase alphabetical characters.

<*U> Any URL string. Typical URL strings would be “www” or “ftp” addresses.

<*#> A string of any numeric characters (numbers only).

<*&> A string of any alphanumeric characters (i.e., letters and numbers only).

Finding Characters with Specific Attributes

It is also possible to find characters according to the attribute(s) that have been assigned by the remote application. This can provide a versatile way of searching for specific features in applications that use a wide variety of attributes.

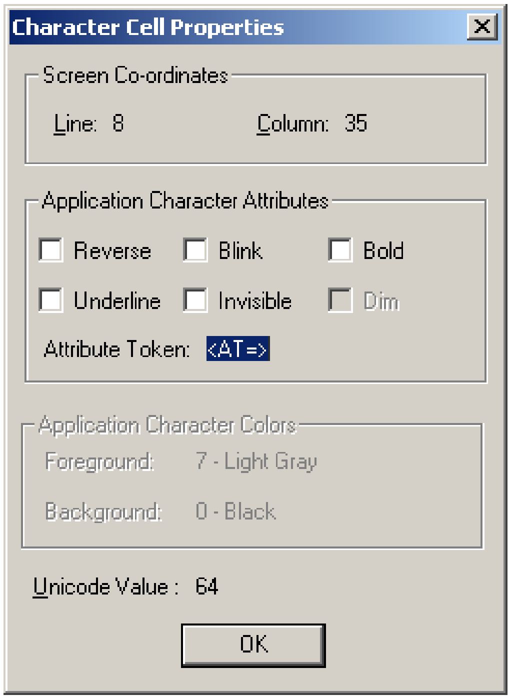
To search for a specific attribute

1 Right-click over the character containing the desired attribute and choose Character Properties from the popup menu. Highlight the character attribute token on the dialog that is displayed (see Figure 5.20).

2 Copy the token to the Find cell of your rule.

3 Insert the characters that you are interested in.

4 Remember to put the 'finish' </AT> at the end of your parameter, e.g., <AT=R><20?></AT> where <20?> is your specification for the characters that you are interested in.


Character Cell Properties
Screen Co-ordinates
Application Character Attributes
Application Character Colors
OK

Line: 8 Column: 35

□ Reverse □ Blink □ Bold
□ Underline □ Invisible □ Dim

Attribute Token: <AT=>

Foreground: 7 - Light Gray
Background: 0 - Black

Unicode Value : 64

Figure 5.20
Character Cell Properties toolbar

Parameter Token

Parameter tokens are not part of the search expression but they are used within the Find cell to extract data from the terminal emulator. The Parameter tokens come in start/end paris. They are typically inserted around a Single Character or Wildcard Token. Any text that matches the expression between the Parameter Tokens becomes the parameter. More than one set of parameter tokens can be used at a time, e.g., <P1><?></Pa><P2><?></P2>.

This search specification would find two single characters of any type. The first character would become P1 and the second one would become P2. Parameters are extremely versatile and can be used in many places. For example, they can be used as part of a sequence sent to the remote system or as part of the text used in a message box or in text substitution.

Like Parameter Tokens, Do Not Highlight Tokens also come in start/end pairs and they do not form part of the search specification. They are inserted around part of the search specification. When a match is found in the terminal emulator the Rules Agent does not change the appearance of any text matching the token(s) enclosed by Do Not Highlight Tokens: e.g., <P1><?><NH><?></NH></Pa>.

Definable Set Tokens

If you cannot find a token that meets the find criteria that you wish to search for, you can define your own tokens. Definable Set Tokens are composed of a series of characters that the Rules Agent searches for. To define a new Definable Set Token, select Definable Set Tokens from the Tools menu in the Rules Agent Editor. Select an undefined token from the list then click Define. Into this dialog you can enter a series of keyboard and/or Unicode characters to search for. You can also choose to make the token case-sensitive.

Definable Set Tokens are used in the same way as the default tokens. When they are defined they appear in the right-click popup menu under either System Defined or Session Defined tokens. Definable Set Tokens can be used on their own or in conjunction with other tokens.

Using Tokens

If, for example, you wanted to identify the words in this menu title bar:

File View Options

Each menu title has some characteristic properties:

- It starts with an uppercase alphabetical character
- It contains a number of lowercase alphabetical characters
- It is followed by a space

In each case, the first letter of each menu title must be sent to the remote system to open that item. This situation is quite common in UNIX applications and it makes the application easy to face-lift. The following search expression searches for the menu titles: `<?AU><*AL>`

This expression searches for any single uppercase alphabetical character followed by any lowercase alphabetical characters. To make the first character of each menu title a parameter that can be sent to the remote application when required, Parameter tokens must be inserted around the token that matches the first character: `<P1><?AU></P1><*AL>.` The first character of any string that matches this expression then becomes the parameter P1. To send the captured parameter to the remote system when the user clicks one of the face-lifted menu titles it is necessary to configure an action. Set the Action Type to Send Sequence and add the parameter P1 to the Action cell. Some UNIX applications need control characters to be followed by pressing Enter. This is easy to implement too. Simply right-click in the cell and choose Return from the Control Characters section of the Popup menu. The contents of the Action cell would look like this: `<P1>\.`

Rules Agent Search Engine

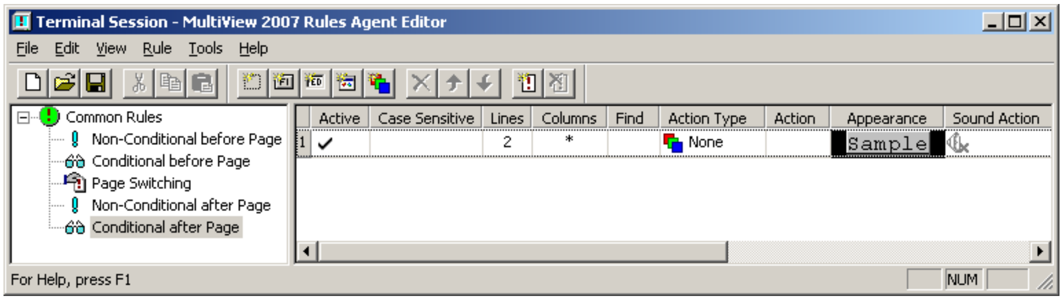
In order to achieve fast, efficient face-lifting of a UNIX application it is important to understand the basic operation of the Rules Agent Search Engine. There are two important points to note here:

- When an area of the terminal emulator has been face-lifted it cannot be face-lifted again by a subsequent Common rule. This is because the terminal emulator is only scanned once by the Rules Agent at each refresh cycle.

- The first rule to become active on any one line of the terminal emulator takes precedence over any other rule, regardless of the order it appears in the rules list.

The reason for the second point is that the Search Engine scans the terminal emulator one line at a time, starting at the top line and scanning each line from left to right. At each character cell it tests all of the Conditional Before Page rules for a match in the sequence that they appear in the rules list. Then it repeats the procedure with the Page Switching rules and then the Conditional After Page rules. This action is illustrated here. If a Page Switching rule becomes active, the Page rules associated with it are applied and processed before the Rules Agent returns to the main list.

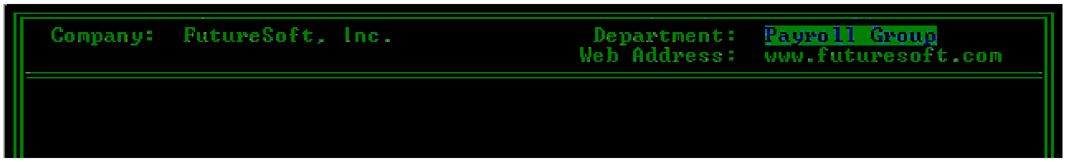
1. The rule shown in Figure 5.21 was created in the Conditional after Page section to face-lift “MultiView Group”.



	Active	Case Sensitive	Lines	Columns	Find	Action Type	Action	Appearance	Sound Action
1	✓		2	*		None		Sample	

Figure 5.21
Conditional after Page rule

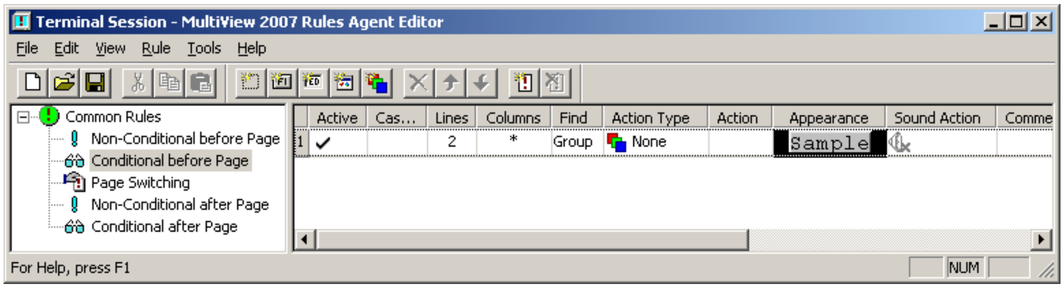
2 Its effect is shown in Figure 5.22.



```
Company:  FutureSoft, Inc.              Department:  Payroll Group
                                       Web Address:  www.futuresoft.com
```

Figure 5.22
Results of Conditional after Page rule

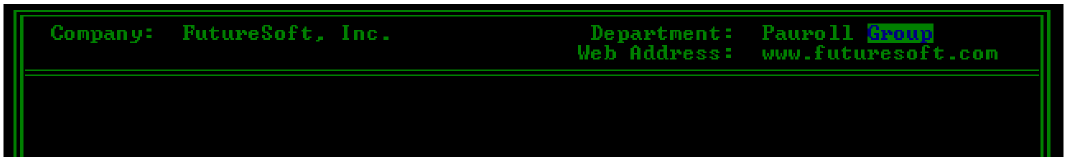
3 A second rule is then added in the Conditional before Page section. This rule affects the word “Group”.



	Active	Cas...	Lines	Columns	Find	Action Type	Action	Appearance	Sound Action	Comme
1	✓		2	*	Group	None		Sample		

Figure 5.23
Conditional before Page rule

4 This rule overwrites the Conditional After Page rule we created because all Conditional Before Page rules are applied to the line before the Conditional After Page rules. Since the word Group has been face-lifted the Rules Agent does not attempt to apply the rule in the Conditional After Page.



```
Company:  FutureSoft, Inc.                Department:  Payroll Group
                                         Web Address:  www.futuresoft.com
```

Figure 5.24
Results of Conditional before Page rule

When the Rules Agent finds a match between the search specification of one of the rules and some text in the terminal emulator it applies the rule, the rule appearance is applied to each subsequent character cell until the search specification no longer matches. At this point the Rules Agent moves on to the next character. This factor can lead to some unexpected results when face-lifting applications with Conditional rules. Provided you understand the searching mechanism used by the Rules Agent, however, conflicts are easily avoided.

One exception to this behavior can occur with Page Switching rules. An option is available (described below) to overwrite previous face-lifting when a new page becomes active. This feature is enabled by the Rules Agent scanning the terminal emulator twice in each refresh cycle, once for the Common rules and a second time from the Page rules.

Rule Actions (Conditional and Non-conditional rules)

When an active rule is clicked by the user the rule can invoke an action. The type of action is specified in the Action section of the rule.

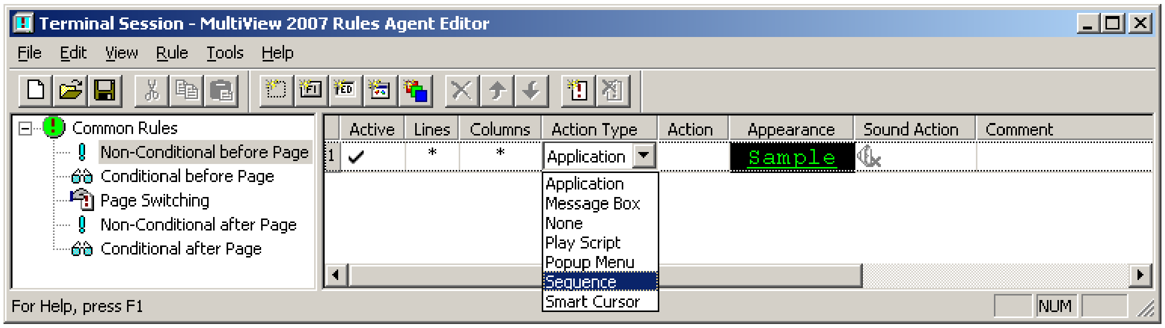


Figure 5.25
Rule Action Types

There are several types of Action that can take place:

Application
Starts a local application. It can also pass a parameter to the application such as a URL.

Message Box
Create a Windows message dialog with an OK button in it. Use this rule to give the user additional information about part of an application.

None
Performs no action at all but is very useful for when you just want to apply an attribute or add an image to a specific area of the terminal emulation window.

Play Script
Plays a script (.zzs) file.

Popup Menu
Creates a popup menu that opens in response to the user clicking an area of the screen. This provides a similar functionality to the Windows list box.

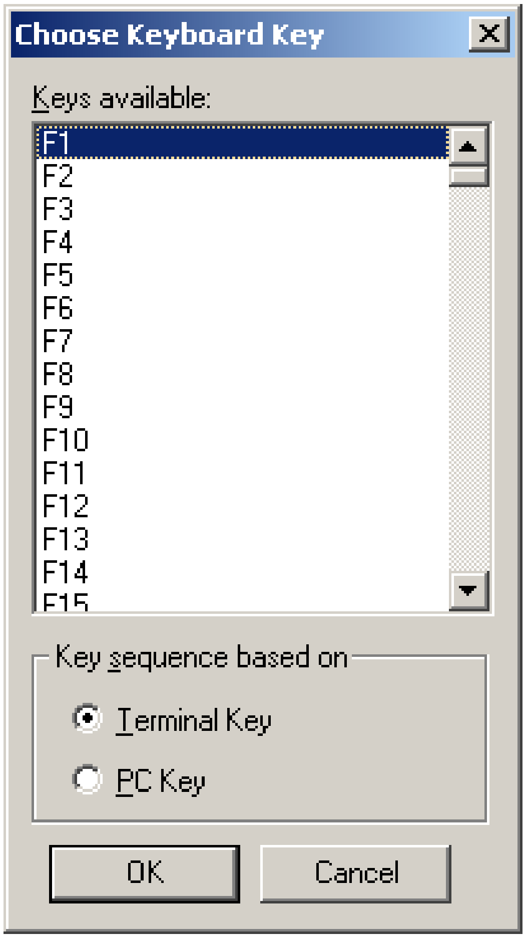
Sequence
Sends a character sequence to the remote application. Send a keyboard key, such as a function key, whose string may be unknown for a specific terminal type.

Smart Cursor
Creates an object that has focus when it is clicked on by the mouse. This mirrors the Windows user interface feature although unlike Windows the cursor cannot be positioned within an edit field by clicking between characters.

The Application, Sequence and Message Box Actions can make use of parameters extracted using the Parameter token(s) in the Find cell. This functionality allows you to make versatile rules that can adapt their actions to suit the features that are being face-lifted.

Specifying a Sequence

When specifying a sequence to send to a remote application you can use a combination of keyboard characters, parameters (found with the Parameter token), and control characters. Parameters and control characters are easily inserted by right-clicking in the Action column and choosing the appropriate item from the list that appears. Terminal keys such as the "F" key are added with Terminal Key from the popup menu as shown in Figure 5.26.



Key sequence based on

● Terminal Key

○ PC Key

Figure 5.26
Choose Keyboard Key dialog

The Choose Keyboard Key dialog appears. Select the key to use whether this is a terminal key or a PC keyboard. The terminal emulator sends the appropriate character sequence to the remote system depending on the terminal type selected and any key mapping you may have enabled.

Specifying an Application Action

URL tokens can be specified in the Action cell in association with the Application action type. Choosing, for example, the World Wide Web token in the Find and Action cells creates clickable URLs that appear in the remote application. When they are clicked, MultiView launches the default local browser (if installed) and goes to that URL.

Specifying a Windows Message Dialog

You can specify a Windows message dialog to appear when the Rules Agent matches a defined string. Within the box, you can specify a window title, contents text and choice of icon. You can also control the behavior of the window: Enable the Restore Terminal Window option so that the main terminal window is restored (if it is minimized or in the background) when a Windows message dialog is opened by the Rules Agent.

Specifying a Popup Menu

The rules-driven popup menu action allows you to create Windows list-box style functionality in a remote application. In order to use this action type you must define at least one popup menu using the popup menu procedure discussed earlier in this manual and save the popup menu configuration file(s) (.zzp) in a suitable location.

To add a popup menu action, double-click in the Action Type cell of a rule and choose Popup Menu from the list. Select the Action cell, right-click in the Action cell and select Browse File. Select the appropriate popup menu configuration file. Popup menu actions that require parameters can be used in conjunction with rules. Any parameter that is specified in the Find field is used as the parameter for the popup menu. It is, of course, possible to specify more than one parameter in a rule and more than one parameter can be used in the popup menu too. There are two important point to note:

- Popup menu actions that require more than one parameter are not available if the menu is used as an ordinary right-click popup menu.

- If the required number of parameters are not available for a given popup menu action, then it will not be available.

When a popup menu action is not available for either of the above reasons it appears grayed in the menu. Parameters can also be added to the Action field of the rule, either before or after the popup configuration (.zzp) file and the parameter(s) are sent before or after the popup menu action has been performed.

Specifying Play a Script

The Play Script function allows you to specify a script to be played when a rule is clicked. Scripts are a perfect choice when you need to send more information to the remote system than can be easily accommodated in a Sequence action. Using a script file also allows you to use the extra operators that are described in the Scripting section. To specify a script, choose Play Script in the Action Type field. Then, select the Action field and choose Browse File from the right-click popup menu to choose the appropriate script (.zzs) file. Parameters can also be added to the Action field of the rule, either before or after the script file and the parameter(s) are sent before or after the Play Script action has been performed.

Specifying Smart Cursor

Smart Cursor enables you to give focus to a new or previously configured edit field, button or menu when it is clicked on using the mouse.

Enabling debugging of Smart Cursor

Because the application uses a direction key such as **Tab**, to navigate around the screen and give an area focus, in the Smart Cursor scenario it must still visit every field between its starting position and its final destination. By default this is hidden, but you may find it useful while you are face-lifting your application, to see exactly which fields the cursor is visiting as it moves to the required position. Before you start, it is a good idea to enable the **Shows screen updates during Smart Cursor actions** facility. Enabling this option shows every position of the cursor on its journey round, enabling you to see if any of the fields are being skipped.

To enable Smart Cursor debugging

1 On the **File** menu in the Rules Agent Editor, select **Properties**.

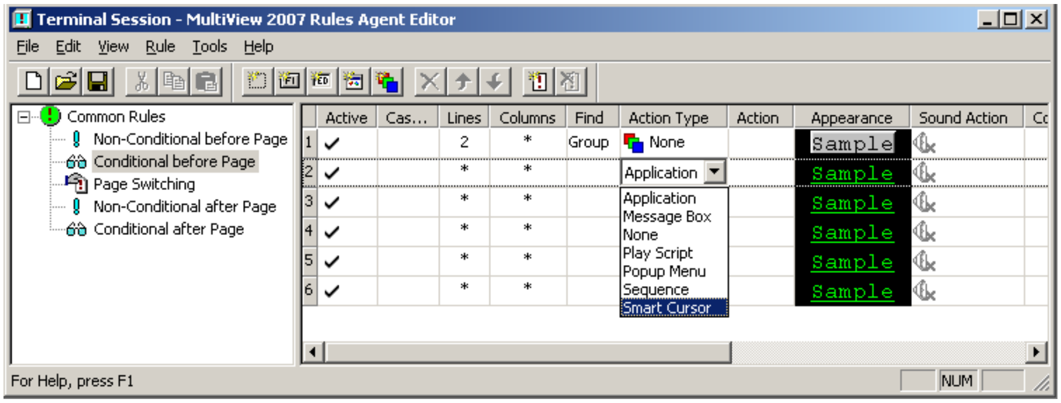
2 In the **Rule File Properties** dialog, click the **Advanced** tab.

3 On the **Advanced** tab, enable the **Show screen updates during Smart Cursor actions** option (apply check mark).

4 Click **OK**.

Configuring Smart Cursor

Before you start to configure Smart Cursor you must identify the loop to which the Smart Cursor functionality is to apply. This loop must be cyclic in that once you have visited each of the sites in this loop, you arrive back where you started. You then need to create face-lifting rules to cover each of these sites using either conditional or non-conditional rules. All of these rules need the Action Type set to **Smart Cursor**. For each area showing a Smart Cursor you will need to configure three options: Focus Detection, Action Sequence and Group ID.



	Active	Cas...	Lines	Columns	Find	Action Type	Action	Appearance	Sound Action	Cc
1	✓		2	*	Group	None		Sample		
2	✓		*	*		Application		Sample		
3	✓		*	*				Sample		
4	✓		*	*				Sample		
5	✓		*	*				Sample		
6	✓		*	*				Sample		

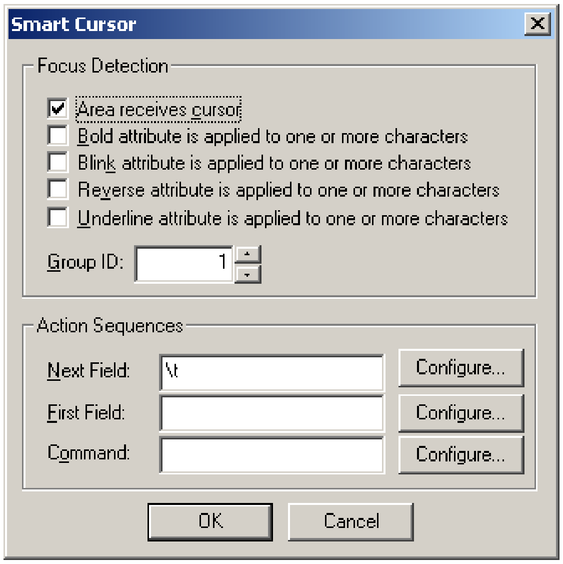
Figure 5.27
Smart Cursor selection on the Action Type pulldown menu

Focus Detection

Focus Detection defines what attribute the UNIX application applies to an area that currently has focus. This could be that selected text is reversed for example or that the cursor blinks when it is in the face-lifted area. Focus Detection is defined by the UNIX application itself and may be different depending on which remote application you are using.

Action Sequence

Action Sequence uses the information entered in the **Next Field** edit field of the **Smart Cursor** dialog to know how to move to the next location in the loop that is currently moving. Alternatively, you may wish to configure the application so that the area that has focus, such as an **OK** button, performs an action when clicked on. This information is entered in the **Command** field of the same dialog.



Smart Cursor

Focus Detection

☑ Area receives cursor
☐ Bold attribute is applied to one or more characters
☐ Blink attribute is applied to one or more characters
☐ Reverse attribute is applied to one or more characters
☐ Underline attribute is applied to one or more characters

Group ID: 1

Action Sequences

Next Field: \t
First Field:
Command:

Figure 5.28
Smart Cursor dialog

Group ID

Group ID gives a number to each loop. This allows for more than one loop to be active on a page at any one time. Each loop of the cursor needs a unique number so that the application knows which group of items, such as edit fields, menus and/or buttons, is having attributes applied.

For example, in a typical UNIX application navigation around the screen is achieved by using the direction keys to first navigate from side to side along the top menu and then by suing the up and down keys to move up and down the drop-down menus which result from the first movement of the cursor. Also, notice that each loop uses the reverse attribute to show which item has focus.

These two directional movements, first, side to side then second, up and down, are classed as two loops. Each loop has to have a Smart Cursor configuration assigned specifically to it. This is achieved by giving each loop a group ID. Giving each loop a group ID enables you to give the loops specific instructions and control each cycle of movement that the Cursor makes. Because of this, both loops can be active at the same time enabling you to select from the top menu and then from the drop-down one.

Using Smart Cursor

1 Create an edit field rule using Rules Markup or Rules Agent Find parameters.

2 Select **Smart Cursor** in the **Action Type** field.

3 Double-click on the **Action** field to display the **Smart Cursor** dialog.

4 Determine the attribute in the UNIX application that indicates a particular item has focus. Select the Focus Detection options accordingly.

5 After all Focus Detection option(s) are enabled, determine how many loops are included in the page that is being face-lifted. A loop is a complete cycle for the cursor where one item has focus at any one time (See **Group ID** in this chapter).

Note

You can have more than one loop, each with a side to side or up and down movement. As long as each loop has a unique group ID, they will all be treated separately by the Rules Agent. However, each location must have a unique ID.

6 Next, define the key or sequence needed to move the cursor to the next item in the loop. This is done by entering a sequence in the **Next Field** edit field, for example “tab” or “/t.”

7 Sometimes the situation occurs that the loop that you are trying to apply Smart Cursor to is not cyclic (the cursor does not automatically move back to the first field after it has reached the last field in the cycle). In this situation, to prevent applying a different rule just to move the cursor back to the top of the loop, the **First Field** edit field is used to enter a second sequence. This second sequence directs the cursor where to move when it has reached the last field in the loop. Usually, this is “Home” (cursor moves back to the first field when the “Home” key is pressed) or something similar.

8 You may wish to configure the application so that the area that has focus, such as an **OK** button, performs an action when clicked on. To do this click **Configure** by the **Command** edit field to open the **Define Sequence** dialog. Click the required button to enter the required action. Click **OK**.

9 When you have completed configuration with Smart Cursor, disable the **Show screen updates during Smart Cursor actions** option in the **Rule File Properties** dialog. You may choose to leave this option enabled for a short period of time so that users can verify that the cursor is moving to each site.

Rule Appearance

Non-conditional rules and Conditional rules (when they are active) can change the appearance of the terminal emulator. The controls used to do this are very similar to the Attribute Mapping controls. When you double-click on the appearance cell of a rule, the **Appearance** dialog appears with the **Appearance** tab displayed. Rule appearance is not Attribute Mapping. Rules change the appearance of the terminal emulator regardless of how it appeared before. Conversely, Attribute Mapping changes the appearance of the terminal emulator regardless of how it appeared before. Conversely, Attribute Mapping changes the appearance of existing attributes. Rule appearance can, however, base its attributes on either the remote application or the changes that have been made in Attribute Mapping, if applicable. The following attributes can be based on either the remote application or the Attribute Mapping settings by clicking the option to place a gray check mark in the option check box. This indicates tri-state operation:

- Reverse Colors
- Underline
- Blink
- Invisible
- Brighter
- Dimmer

As with Attribute Mapping, images can be incorporated in place of foreground or background colors.

Text Substitution

As well as changing the appearance of existing text, Rule Appearance can also change the actual text that is displayed in the terminal emulator. On the **Appearance** tab of the **Appearnace** dialog, enable the **Text Substitute** option and then click **Configure** to open the **Text Substitution** dialog. From this dialog you can enter text, a parameter, or a combination of the two. You can also enter Unicode characters, a feature that is particularly useful for producing the line-drawing characters and new characters such as the Euro symbol.

Sound Action

To play a sound when a rule becomes activated, double-click the **Sound Action** cell and choose a new sound. Sounds must be in **.wav** format and have a sampling rate that is compatible with your sound card. If you have Microsoft Text-to-Speech installed, you can choose the **Speech** option. Enter the text to be spoken into the box provided. Click **Preview** to review the sound or speech.

Note

Parameters can be used in the Speech text box.

Page Switching Control

Getting Started

Page Switching rules have one specific action — they identify the current page that the remote application is displaying to the Rules Agent. When a new page is identified, the set of page rules that is associated with it is applied. Additionally, a group of Page functions can be applied if required. Page Switching rules have two different columns in the rules list compared to Conditional rules. The Rule Page cells identify the name of the page that is activated by this Page Switching rule. The Action Type column sets the permanence of the rule. Page Switching rules can be either temporary or permanent.

A temporary rule activates page rules when and only when its Find parameter is satisfied. If the rule is set to permanent, when its Find condition has been satisfied, it permanently activates until another Page Switching rule becomes active.

To use Page Switching control, identify something unique about the part of the application to face-lift. It may be a single character or a sequence of characters that always appears in a specific place when that page is displayed. The more precisely that you can define the Find specification, the better.

To make a new Page Switching rule

1 Click Add Page on the Rules Agent Editor toolbar.

2 In the Page Details dialog, enter a new page name and click OK.

A new branch is added to the Rules tree where the Page rules to apply are placed.

Adding and Switching Pages

If you have defined more than one page in your application, when a new page is detected the new set of page rules either replace or add to those associated with the previous page. This behavior is defined in the Page Switching rule in the Action Type cell. The options are:

- Add Page Temporary
- Add Page Permanent
- Switch Page Temporary
- Switch Page Permanent

Add Page rules are advantageous when two pages need to be active at the same time, such as, when an application opens a menu list. You define each of the menu lists as additional pages to the main application page. The remainder of the application remains face-lifted when the menu lists appear. Permanent Pages are best used for face-lifting elements that are not likely to change significantly. Small content changes are permissible if you use Conditional Page rules as these can adapt to the changes. Non-conditional Page rules cannot adapt.

When you start using Page Switching rules, a new toolbar entitled Rules Editor Info displays which if any Page Switching rule is active in the terminal emulator. Checking the box when a Page Switching rule is active deactivates the rules and highlights wheat screen text is being used to activate the page switching.

Page Functions

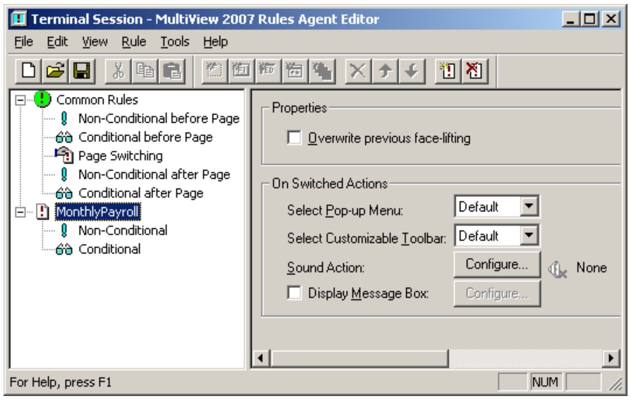
When a new page is enabled, a number of pre-defined actions can occur in addition to the associated Page rules being applied. These actions are:

- Overwrite Previous Face-lifting
- Enable a new Popup menu
- Enable a new Configurable Toolbar
- Play a sound
- Display a Windows message dialog

These actions are set up in the right pane of the Page Switching page window as shown in Figure 5.29.

Overwrite Previous Face-lifting

When any area of the remote application has been face-lifted by a rule it cannot be face-lifted further by a subsequent rule. Enabling the **Overwrite Previous Face-lifting** option (apply check mark) in the right pane of a root page allows the Page rules associated with this page to overwrite any previously face-lifted areas.



□ Overwrite previous face-lifting

Select Pop-up Menu: Default

Select Customizable Toolbar: Default

Sound Action: None

□ Display Message Box:

Figure 5.29
Page Switching page window

Page Dependent Toolbars and Popup Menus

Page Switching rules can be used to control the appearance and behavior of the user configurable toolbar and the popup menu. To use Page Switching control, you must create separate Configurable Toolbar (**.zzb**) and/or Popup Menu (**.zzp**) configuration files for the Page Switching rules to refer to. To create separate configuration files, you must first configure the toolbar or popup menu in the terminal emulator and save its configuration details as a separate file using the **Component Configuration Files** dialog (**File > Component**).

For page dependent toolbars and popup menus choose one of the following:

- None—do not display the toolbar or popup menu
- Basic
- Use the default
- Use the current
- Open a new (toolbar or popup menu)

Note

Page Functions are only activated by Switch-Page type Page Switching rules. Add Page rules do not enable any page functions. This is due to the fact that only one set of Page Functions may be active at one time.

Configuration components are managed in the **Component Configuration Files** dialog (**File > Components**). After configuring a toolbar or popup menu, open this dialog, select the appropriate configuration file and click **Save As**. Choose a file name and click **Save**. In the **Component Configuration Files** dialog, click **Close**. Open the Rules Agent Editor and select the root of the Page in the Rules tree. In the right pane, select the configurable toolbar or popup menu to associate with the page.

Play a Sound

To play a sound when a new page becomes active, click **Configure** to the right of **Sound Action** to display the **Sound Action** dialog. In the **Sound Action** dialog, enable **Sound Wave** and either enter a complete path or use **Browse** to select a sound file. Sounds must be in **.wav** format and have a sampling rate that is compatible with your sound card.

If you have Microsoft Text-to-Speech installed, the **Speech** option becomes available. Enter text to be spoken in the area provided. Click Preview to hear a preview of the sound or speech.

Display a Windows message

Enable the **Display Message Box** option to display a Windows message when the new page becomes active. Click **Configure** to the right of the option to create and customize the dialog in the **Message Box** dialog.

Using the Rules Agent Editor with Rules Markup

The most efficient way to start face-lifting an application is to use the Rules Markup Engine in conjunction with the Rules Agent Editor. The Rules Markup engine has limitations for two reasons:

- It only creates Non-conditional rules
- It only creates Add Page Temporary Page Switching rules

The absence of Conditional rules is a limitation because if the application changes in appearance very slightly, completely new rules are needed to face-lift it. conditional rules can adapt to changes.

Maximizing Rules Engine Speed

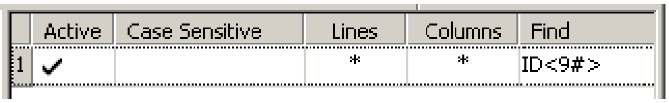
When face-lifting an application it becomes apparent that different combinations of rules can have the same effect on the user interface. Some combinations of rules are, however, vastly more efficient than others. This section explains how to write rules that maximize face-lifting speed, providing a more usable interface.

Line and Column Numbers

The more rules that have to be checked for each character location, the more time is needed to parse the entire screen. The line and column numbers can therefore be used to tighten the scope for where a particular rule is checked.

Case Sensitive

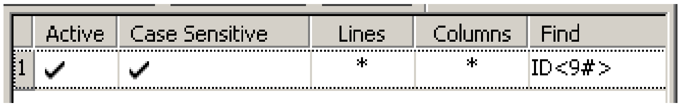
Rules have to be parsed a token at a time. Any method whereby rules can be rejected as soon as possible is going to improve performance. Sometimes this can be achieved by checking the Case Sensitive flag, (the Case Sensitive cell titled) in the Rules Editor.



	Active	Case Sensitive	Lines	Columns	Find
1	✓		*	*	ID<9#>

Figure 5.30
Page Switching rule example

For example, the rule shown in Figure 5.30 looks for a customer ID, indicated by the prefix “ID” followed by 9 numbers. Consider a screen with the word “Friday” present. The rule above gets a match on the letters “ID” within the word and then attempts to match the “nine numbers" token. This will obviously fail, but extra processing has been performed. If the case sensitive flag has been checked as shown in Figure 5.31, then the first part of the rule would have failed.

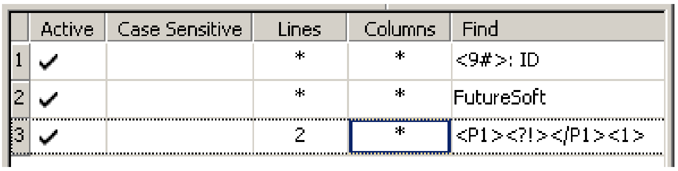


	Active	Case Sensitive	Lines	Columns	Find
1	✓	✓	*	*	ID<9#>

Figure 5.31
Page Switching rule example

Rule Ordering

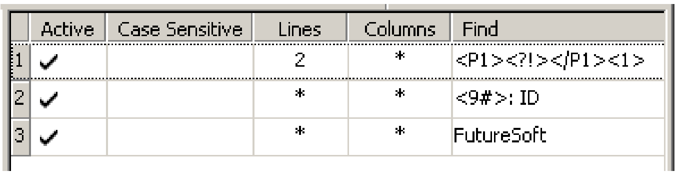
When a rule has been matched to face-lift an area of the screen, the rules ordered after the matched rule can be quickly rejected. Rules ordered before the processed rule, however, will have been processed but rejected due to the Find criteria failing to match.



	Active	Case Sensitive	Lines	Columns	Find
1	✓		*	*	<9#>: ID
2	✓		*	*	FutureSoft
3	✓		2	*	<P1><?!></P1><1>

Figure 5.32
Page Switching rule example

In Figure 5.32 rule 3 highlights a horizontal ring menu on line 2 of the terminal screen. When line 2 is face-lifted, rules 1 and 2 have to be fully checked before rule 3 does the face-lifting.



	Active	Case Sensitive	Lines	Columns	Find
1	✓		2	*	<P1><?!></P1><1>
2	✓		*	*	<9#>: ID
3	✓		*	*	FutureSoft

Figure 5.33
Page Switching rule example

Figure 5.33 shows the revised rule list where the ring menu has been moved to be the first rule. This means that when line 2 is being face-lifted, rule 1 matches first and face-lifts the line. Rules 2 and 3 are instantly rejected for those areas of the line that have already been face-lifted.

Paging

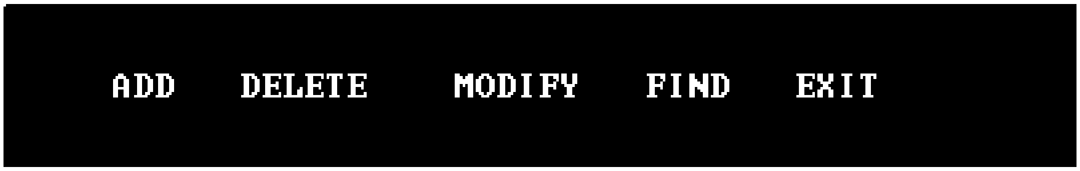
Page rules give the Rules Agent the ability to face-lift separate areas of an application independently. However, some rules can be placed in the Common rules section, these are rules that are applied constantly to all screens of the application. If this contains rules that are not needed for some areas of the remote application, you may wish to consider moving them to a relevant page of pages that require those rules. This will then speed up rule processing for other areas of your application.

Using Wildcard Tokens

As mentioned above, the sooner a rule can be rejected, the better. Therefore, using tokens that closely match the characters required can be beneficial. Say, for example, that you have a customer code consisting of 3 alphabetical characters followed by 7 numbers, using the Find rule as "<3>A><7?#>", (any 3 alphabetical characters followed by 7 numerical characters), is better than "<3?!_><7?#>", (any 3 characters except a space followed by 7 numerical characters). Although the same areas of the screen are probably face-lifted, the second token of 7 numerical characters will most probably have been processed more times with the prefix of "<3?!_>" than when the "<?3A>" token is used.

Generic Rules-versus-Strict Rules - an example

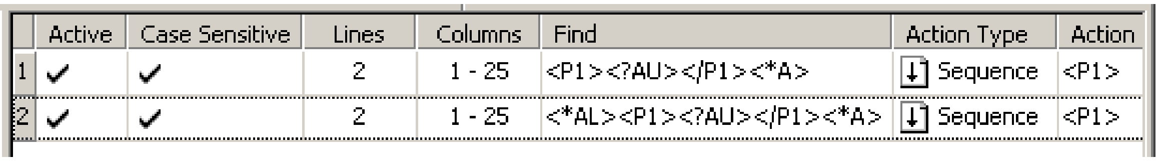
A static, horizontal ring menu that appears on line 2 of the screen and only in a specific area of an application is shown in Figure 5.34.



```
ADD   DELETE    MODIFY   FIND   EXIT
```

Figure 5.34
Horizontal ring menu example

The two generic rules to turn these into clickable buttons are shown in Figure 5.35.



	Active	Case Sensitive	Lines	Columns	Find	Action Type	Action
1	✓	✓	2	1 - 25	<P1><?AU></P1><*A>	⤓ Sequence	<P1>
2	✓	✓	2	1 - 25	<*AL><P1><?AU></P1><*A>	⤓ Sequence	<P1>

Figure 5.35
Generic rule example

Note that the column ranges have been set to improve efficiency. These rules are checked for all columns between 1 and 25 inclusive.

Column

1 Rule 1 finds the word “Add” and face-lifts it.

Rule 2 is checked and rejected as this cell has just been highlighted.

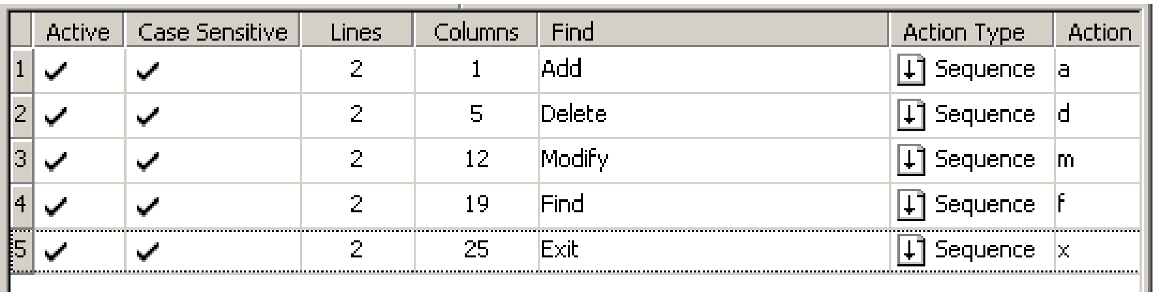
2 &3 Both rules are rejected as those character cells are already highlighted.

4 Both rules are rejected, as the space character does not match with the Any Alphabetical type character tokens.

Rule 1 matches and the word “DELETE” is face-lifted.

Rule 2 is rejected as this cell is now highlighted and so on, until column 25.

This equates to two rules being checked at 25 column positions giving 50 rule checks. Alternatively, because these rules are within a single page, we can create 5 rules that quickly face-lift this menu as buttons.



	Active	Case Sensitive	Lines	Columns	Find	Action Type	Action
1	✓	✓	2	1	Add	Sequence	a
2	✓	✓	2	5	Delete	Sequence	d
3	✓	✓	2	12	Modify	Sequence	m
4	✓	✓	2	19	Find	Sequence	f
5	✓	✓	2	25	Exit	Sequence	x

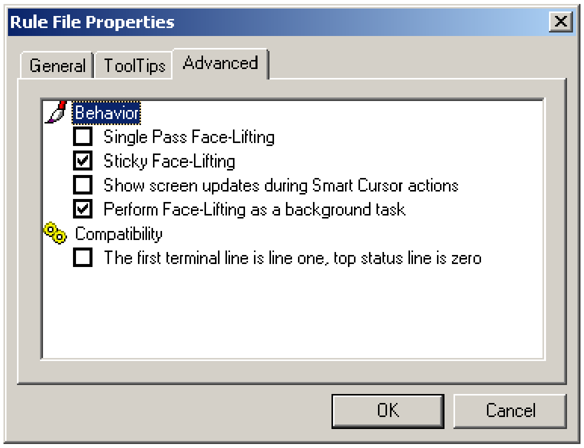
Figure 5.36
Strict rule example

Advanced Face-lifting Options

By default, when the Rules Agent face-lifts an application it causes the terminal emulator windows to be re-drawn in two passes. When using complex face-lifting on slower PCs the redraw time may be unacceptable. You can, however, control the way in which the window is redrawn using the drawing options provided. These options are accessed on the **Advanced** tab of the **Rule File Properties** dialog in the Rules Agent Editor (**Rules Agent Editor > File > Properties**).

The available options include:

- Single Pass Face-lifting
- Sticky Face-lifting (on by default)
- Perform Face-lifting as a background task (off by default)
- Show screen updates during Smart Cursor actions



Rule File Properties

Behavior

□ Single Pass Face-Lifting
☑ Sticky Face-Lifting
□ Show screen updates during Smart Cursor actions
☑ Perform Face-Lifting as a background task

Compatibility

□ The first terminal line is line one, top status line is zero

Figure 5.37
Rule File Properties dialog - Advanced tab

Enabling the **Single Pass Face-lifting** option can improve the appearance of the application by only re-drawing the window when the terminal emulator is idle. Two reasons for doing this:

- Your PCs may not be fast enough to re-draw and face-lift all changes in the remote application as they occur. The resulting lag would produce a confusing display as the remote application is changing.

- You may wish to suppress the display of certain features of the remote application such as scrolling to make the application look more Windows-like.

The **Sticky Face-lifting** option controls the way in which the face-lifted parts of the screen are updated. When enabled, Sticky Face-lifting causes features like edit fields to be drawn intelligently. If, for example, a user types into a face-lifted edit field the edit field is not re-drawn each time a character is added to the field. This re-draw can cause a distracting ghosting effect that is undesirable. In certain circumstances, however, Sticky Face-lifting may cause anomalies such as parts of the window remaining face-lifted after the text in it has changed. If this occurs, disable Sticky Face-lifting.

The **Perform Face-lifting as a background task** option is useful if you wish to give applications a more steady appearance. Enabling this option (applying check mark) forces the Rules Agent to apply its rules as soon as new data arrives, rather than just when the CPU is idle, as in Single Pass Face-lifting. If your PC is not fast enough to re-draw and face-lift all changes in the remote application as they occur, you may find that performance is improved by leaving this option disabled (check mark removed).

Note

If this option is disabled and, while applying rules, the Rules Agent detects more new data, it stops applying the rules and reads the new data instead. This works on the principle that if there is new data waiting to be read then there is little point in applying rules which need to be changed anyway. The settings for this option are stored in the .zzr configuration file.

The **Show screen updates during Smart Cursor actions** option is mainly a debugging tool which enables you to see each movement of the cursor as it tabs around the screen, until it reaches its final destination. This should be the edit field that the user has clicked on. In some cases the cursor is unable to navigate to certain edit fields. Showing each tab position of the cursor as it travels around the page allows you to see where any problems lie. It also illustrates to the user that the cursor is in the process of reaching the required edit field. With a slow system, allowing users to see the cursor tabbing round the screen ensures that they do not keep clicking on an edit field, thinking that the functionality is not working.

Because the screen is redrawn with every movement of the cursor which may impede system performance, this option is designed not to be used in a permanently enabled state. It is therefore better to use this feature as a diagnostic tool rather than to use it all the time, particularly with a component such as MultiView Catalyst, where network traffic needs to be kept to a minimum.

Note

To optimize the way the terminal emulator redraws when using other face-lifting features such as Image, refer to the **General** *tab of the* **Session Properties** *dialog, described earlier in this chapter. On this tab there is a slider that allows you to optimize the redrawing for the best speed or the best quality. (The* **Session Properties** *dialog is accessible from the* **Configure** *menu in the terminal emulator.)*

To see an example of face-lifting and how it works view the demonstration in the Samples folder on your CD: **\MVW2000\Sample\Acme.** A readme describing the installation and usage of the example is provided.

Note

The Samples folder is not installed with the product. Visit the support area of the FutureSoft web site for download of the sample files (www.futuresoft.com/support/index.htm).

Chapter 6

Scripting

What is Scripting?

Scripts are a way of automating a procedure with minimal input from the user. They are used in the MultiView Software Family products to:

- Automate login procedures
- Start remote applications
- Perform routine tasks

Scripts can be created in a number of ways. For example, Multiview has advanced script-learning functionality for generating login scripts. Alternatively, scripts can be created and edited in a text editor such as Notepad. Once created, scripts can be used in pop-up menus, configurable toolbars, keyboard mapping and face-lifting rules.

Learning Scripts

The easiest way to create a new script is to use the MultiView Learn Script function. This functionality is most commonly used to create login scripts.

To use Learn Script

1 On the **Remote** menu, select **Learn Script.**

A new set of buttons appears on the toolbar.

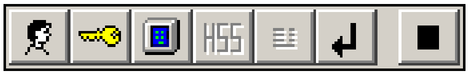


Figure 6.1
Scripting toolbar

2 After opening a new connection, send your user name, password and terminal type (if necessary) to the remote system using the toolbar buttons. Which of these you send depends on the type of connection you are making, e.g., some sessions may not require a password but would need a user name.

The HSS button is used to send a command to launch Host Support Server on the UNIX host. This remains grayed unless you have enabled the **Host Support** option. When you have entered the necessary details, click **Stop** and you are prompted to save your file.

You can configure a terminal session to open and run the script file on start up with these steps:

1 On the **Configure** menu, select **Session**.

2 On the **Startup** tab, enable **Auto Replay Script File**.

3 Click **Browse** to open the **Open** dialog.

4 Use standard Windows navigation techniques to select a script file.

Notes:

- *When using scripts with Host Support, remote commands are sent automatically.*
- *It is not necessary to press Enter after clicking the Host Support Server button.*

Creating and Editing Scripts Manually

Scripts are saved as text in a **.zzs** file and can be created and edited manually using a text editor such as Notepad. You can either edit a script by opening it from within Notepad or by choosing **Edit Script File** from the **Remote** menu within the terminal emulator and browsing to the desired script file. Scripts are within the terminal emulator and browsing to the desired script file. Scripts are composed of operators and tokens. Operators give a script functionality while tokens allow you to use data that is supplied dynamically such as user names or passwords.

Conventional scripts, such as those created using Learn Script will fail if an expected string is not received. This may occur if a change in the command prompt is made when attempting to use the script with different remote systems, or if a notification of an expired password appears. To handle this, the MultiView scripting language provides the conditional operators, **waitfor** and **goto**.

Operators

Send

Defines a sequence to send to the remote system:

```
send="sequence"
```

`"sequence"` is the text that is to be sent to the remote system and must be enclosed within quotation marks.

Receive

Defines a sequence to be received from the remote system:

receive="sequence"

`"sequence"` is the text required from the remote system and it must be enclosed within quotation marks. The scripting engine waits indefinitely for the expected sequence unless a Wait modifier is used.

Wait

Used in conjunction with the Receive command, wait defines a period to wait for the remote system to return the specified text string.

receive="OK", wait=2

This example would wait 2 seconds for the string **OK** to be returned from the remote system. **Wait** is also used with the **Waitfor** conditional operator.

Delay

Defines a pause in the continuation of a script.

```
delay="seconds"
```

`"seconds"` is the delay in the number of seconds that the script stops for.

Connect

Opens a connection between the specified terminal emulator and the remote system using the currently selected protocol.

```
connect
```

Window

Controls how the terminal emulator appears while the terminal emulator is executing all or parts of the script.

```
window=<option>
```

Where `option` is one of the following:

`blank`	suppress output to the terminal emulator window
`hide`	hide the terminal emulator window
`minimize`	minimize the terminal emulator window

You can use these options to hide the login procedure from view. The blank option shows a blank emulation screen. `Hide` does not show the emulation screen at all. For all options the screen window will become visible upon termination of the script.

`restore`	cancel all/any of the above options and display the current terminal screen - there is an implied restore function at the end of a script.

Message

Displays information on screen in a message box using a script:

```
message=<option>
```

The option can either be `on` or `off` where `on` displays the message dialog, and `off` removes it. In order to make use of the message box you will need to use the text operator:

```
text= "text to display"
```

This functionality would typically be used to indicate the progress of a procedure to the user. The message dialog will automatically re-size to accommodate the changes in the length of the text string. Use the `endmessage` operator to define the end of the message dialog procedure:

```
endmessage
```

If the `message` option was originally set to `off` the `endmessage` operator does not need to be used. In all other cases it is necessary to specify the text and `endmessage` operators after each `message` operator.

Note

Message boxes are not supported in scripts when using MultiView Catalyst.

The `window` and `message` operators can be used together while launching remote applications to hide the UNIX login procedure from the user giving the face-lifted application more of a Windows look-and-feel. If the operators are being used for this purpose, the script should look for some data from the remote application before ending, otherwise the logon text may briefly flash on the screen before the remote application clears.

Script Tokens

$LOGIN

Specifies the current user name. The manner by which the current user name is obtained is defined in the Connection tab of the Session Properties dialog.

$PASSWORD

Specifies the current password (associated with the user name above). This is set from the “login to” dialog that appears when a user connects for the first time.

$TERM

Specifies the terminal environment variable which is set in the Terminal tab of the Terminal Properties dialog.

$APPLICATION

Specifies a remote application or command to execute on the remote system once a connection has been established. This remote command is defined in the Application tab of the Application Properties dialog. Open this dialog from the Configure menu.

Conditional Operators

To enable scripts to adapt to changes in the remote system you may use conditional operators. Conditional operators are used in the following example to determine what expressions have been returned from the remote system:

```
waitfor, wait=20
      “expression1” goto label1
      “expression2” goto label2
endwaitfor
```

If the remote system returns either `expression1` or `expression2` within the `wait` period, the script will jump to the appropriate label. Labels are defined by the operator `label` followed by a space and a unique identifier for that label, such as a number.

Scripting Examples

Example 1: Login Script

Here is a typical login script:

```
connect

receive=“login”, wait=35

send=“$LOGIN”
receive=“name”, wait=10
send=“\n”
receive=“Passw”, wait=15
send=“$PASSWORD”
send=“\n”

receive=“(netw”, wait=15
send=“$TERM”
receive=“t”, wait=10
send=“\n”
receive=“$”, wait=15
send=“$APPLICATION”

receive=“notepad”, wait=5
```

Example 2: Using Window and Message Operators

The examples show the use of the window and message operators to make a remote application look more like a Windows application during logon.

```
#Connect to the remote host
connect
#turn off the remote output and blank the TE
window=blank

#display a message screen
message=on
text=“Loggin On. Please wait...”
endmessage
#login
recieve=“login”, wait=35
send=“$LOGIN”
receive=“name”, wait=10
send=“\n”
receive=”Passw”, wait=15
send=“$PASSWORD”
send=“\n”
receive=“(ansi”, wait=10
send=“$TERM”

receive=“nsic”, wait=40
send=“\n”
receive=“machine”, wait=20
#change the message screen text
message=on
text=“Starting Email Application”

endmessage

#run the app
send=“$APPLICATION”
receive=“text”, wait=20
send=“\n”

#wait for the app to display its menu

receive=“text”, wait=20
```

Example 3: Using Conditional Scripting

In the following example, the script waits for the remote system to send a command line prompt “$” to ensure that the user has been logged in before sending the application comjand. If, for example, the response from the remote system changed and the prompt became a “#” instead of a “$” the script would time-out and fail. The example below uses conditional scripting to overcome this problem.

```
connect
receive=“login”, wait=35
send=“$LOGIN”
receive=“name”, wait=10
send “\n”
receive=“Passw”, wait=15
send=“$PASSWORD”
send=“\n”
receive=“(netw”, wait=15
send=“$TERM”
receive=“t”, wait=10
send=“\n”
waitfor, wait=20
      “$” goto Finish
      “#” goto Finish

endwaitfor

label Finish

send=“$APPLICATION”
receive=“notepad”, wait=5
```

Chapter 7

Integration

Application Integration

Face-lifting can transform the appearance of your UNIX application as well as make them more accessible to the end user. Application Integration allows you to achieve a high level of integration with Windows applications. The following functionality is documented in this section.

- Browser Embedding using ActiveX and OLE/2
- Dynamic Data Exchange (DDE)
- Object Linking and Embedding (OLE)

ActiveX allows you to embed the MultiView terminal emulator or file transfer programs in an ActiveX container, such as Microsoft Internet Explorer or Microsoft Binder.

Dynamic Data Exchange improves functionality by allowing the terminal emulator to share information with Windows applications, while the actual sharing mechanisms can be hidden from the end user. Object Linking and Embedding provides an alternative method of integrating remote and local applications.

Overview of ActiveX

ActiveX is a technology that is designed to bring a greater level of functionality to web sites and applications. It is based on Microsoft's COM (Component Object Model). The technology has two components: ActiveX Controls and ActiveX Documents.

ActiveX Controls

ActiveX Controls are small, self contained modules of code that work within web pages to add functionality to them. From a users perspective, controls work like a dynamically loaded plug-in. When a user visits a site that contains an ActiveX control, the control downloads automatically. ActiveX Controls can be created in Visual Basic 5.0 or later and C++. It is probably easier, however, to use existing controls instead of creating new ones from scratch.

ActiveX Documents

ActiveX Documents allow users to view documents other than HTML through a web browser. The MultiView terminal emulator and file transfer programs are ActiveX Document Servers. This means that they can be embedded in container applications that support ActiveX. Currently, these include Microsoft Internet Explorer 3, 4, 5, 6, and 7, and Microsoft Binder.

Browser Embedding

Embedding behavior for Microsoft Internet Explorer

To embed the main components of MultiView into a browser, ActiveX Document technology used in Microsoft Internet Explorer is used. MultiView Terminal Emulator and File Transfer components are ActiveX Document Servers. This allows them to be embedded into any ActiveX Document Client such as Microsoft Binder and Microsoft Internet Explorer 3 and above.

ActiveX technology was introduced by Microsoft to allow non-HTML documents to be viewed in a a browser without the need for opening another application. A browser is used to navigate various documents, readying and responding to them, then moving on to the next. We recommend that same mode of usage for our documents. Therefore, we recommend embedding only when the session is used for a single or simple action. Using the Terminal Emulator, for example, the purpose of a session could be to perform a sales query; on completion, the session finishes and the user can navigate on. If the user spends most of their time using a single remote application they would benefit the most from a standalone Terminal Emulator that can be permanently open and therefore easily accessed without the need for navigation

- As with all ActiveX Documents opened in a web browser, only the document is downloaded, therefore, MultiView must be installed.
- When the user navigates away from the open document, any connections related to that session are closed and that instance of the application is terminated. This saves both local and remote system resources.
- As Internet Explorer is primarily a document viewer, when loading, you may be asked to hide any toolbars. If this is the case, they will be hidden. This is also the case for any other ActiveX document viewer.
- When embedded in a frame, ActiveX Documents are not allowed to merge their menus with Internet Explorer. It would clearly be very confusing if multiple applications merged their menus and as a result, no configuration controls are available.
- We recommend that when using frames the same application is NOT loaded into more than one frame as a conflict of application focus would occur.
- When using the MultiView Active Rules technology to launch URL links from within the terminal emulator, links such as telnet:// and ftp:// launch the relevant application as a separate instance and do not replace the current embedded session.

Embedding behavior for Netscape Communicator

Netscape Communicator v4.04 has the ability to open applications using OLE. As the MultiView terminal emulator and file transfer applications also support OLE, these can be embedded into this browser. ActiveX Document technology is, however, an extension to OLE. The level of integration into Netscape is consequently reduced. The following behavior applies to embedding in Netscape Communicator.

- When embedded, applications do not fill the entire Client area of the browser, Instead, a rectangle contains the application instance.
- The OLE object requires activating by double-clicking it. When this is done, the session can be used.
- With this version of Netscape, embedding into a frame does not activate in place, but instead the session appears as a stand-alone application when activated.
- When using our Active Rules technology to launch URL links from within the terminal emulator, links such as telnet:// and ftp:// launch the relevant application as a separate instance and do not replace the current embedded session.

DDE while Embedded

All DDE conversations require a topic name. This is used as a method of uniquely identifying which open document the conversation is to take place with. The MultiView terminal emulator defaults its topic name to jsbterm followed by the full filename of the open document. When opening a .zzt file, that is downloaded from a web server, Internet Explorer downloads the document to a temporary file. This means that when the file is embedded, its topic name is the temporary file name. This makes knowing the topic name rather difficult. When opening a local .zzt file, however, the topic name is that of the correct file name.

OLE while Embedded

If you need to converse with the terminal emulator using OLE while embedded in to a browser, complications arise. To manipulate any OLE object, you must have a copy of its object handle. Usually this handle is known, as you would normally create the OLE instance within the same application that talks to it. For the session to be embedded within the browser, however, the browser creates and owns this object. You need to obtain a copy of this handle by interrogating the browser itself or the operating system.

Appendices

Microsoft IE3 White Paper Extract on ActiveX Documents

ActiveX Documents enable you to open an application with its own toolbars and menus in the Internet Explorer window. This means you can open richly formatted documents, such as a Microsoft Excel spreadsheet, without having to open another application. You can do your work and then go back to the web page that you were last viewing. Like all ActiveX technologies, ActiveX Documents are available as an open published specification available to all parties. Support for ActiveX Documents enables Internet Explorer to become the frame in which users can browse and view non-HTML documents and applications. To users, it appears that they are running the stand alone application complete with toolbars, menus, and all other user interface elements, while network administrators can use existing documents and applications on their Intranet site without having to convert them all to HTML format.

The Internet Explorer frame is not only an ActiveX Document host, but it also packages the HTML viewer as an ActiveX Document. This means that other vendors can use Internet Explorer as a document viewer for files in other formats. ActiveX Documents are a set of extensions to OLE Documents, the compound document technology of OLE. The extensions are in the form of additional interfaces that allow what looks like an embeddable in-place object to represent an entire document instead of a single piece of embedded content. As with OLE Documents, ActiveX Documents involve a container that provides the display space for ActiveX Documents and Servers that provide the user interface an manipulation capabilities for Doc Objects themselves.

Dynamic Data Exchange

Most terminal emulation packages support some form of Data Exchange by means of copy and paste. MultiView provides full support for Dynamic Data Exchange (DDE). It is possible to achieve a high level of integration between remote applications and Windows Client applicaitons. Briefly, there are two levels of DDE that are supported by MultiView: One-way Links and Streams-based Links.

One-way Links

This is the easiest link to set up. MultiView works as a DDE Server, automatically updating information in a DDE Client, such as Microsoft Excel. To set up a link, simply select and copy a section of text in the emulation window, then paste it into a Windows application using the Edit and Paste Link option. As the content of the original selection changes, the data in the Client is automatically updated. Multiple links can be established between the two applications.

Streams-based DDE Links

A Streams-based link can be used as a way of establishing a conversation between the terminal emulator and a Windows application (Client). An example of this is logging on to a remote system from a Client application. The Client application receives requests for the user name and password. The application processes the requests and returns the appropriate information.

DDE Data Format

To participate in a DDE conversation, both the Client and Server need to understand the data format. This is done with the application, the data topic and the data item.

APPLICATION — The Server application or program name. This would usually be the name of the terminal emulator, jsbterm.

TOPIC — A combination of the server application name and the full path of the .zzt file which contains the remote system login details. The topic is used to identify the window of the server application which is required by the client to carry out a DDE conversation.

ITEM — For MultiView, the two item types are:

- JSBDDEStream - a circular buffer which can contain up to 4 standard screens of data (24 lines by 80 columns).
- The Item name - the screen co-ordinates of the Server data that is to be retrieved by the Client are contained within the name itself.

There are two ways to specify the area, they are:

Block — A larger area is defined by using x, y coordinates, length and height on the screen.
e.g., X11Y5L5H3

Line — A more specific area is defined, using x, y coordinates, length and height on the screen
e.g., LX11Y5L5H3

Types of DDE conversation

Three types of conversation links can be used:

COLD LINK — The Client sends messages to the server requiring specific data. The server responds by sending the requested data. The Client is not notified when data on the server changes.

HOT LINK — The Client initiates the conversation by requesting some data. When the link for a data item is established, the Server notifies the Client when the state of this item changes by sending the updated item.

WARM LINK — This link combines the features of the hot and cold links. When a link is established, the server only notifies the Client of a change in a data item but does not send the updated data. The Client must request the data item if it is required.

Starting the DDE Server

In order to establish a DDE conversation between Server and Client you must first run the terminal emulator with a configuration file (.zzt) loaded. This allows the Client to access a topic in order to initiate a DDE conversation. The command line used to start the DDE Server must use a `-s` flag to indicate that DDE scripting is taking place and the DDE "topic" is the name of the .zzt file (already loaded). This name must be enclosed in quotes.

Note

The terminal emulator in DDE mode supports DDE Execute commands. All command strings must be enclosed in angle brackets, for example [ABORT].

ABORT Exit the terminal session.

SHOW(x) Shows the terminal session window as controlled by the parameter x, where x is a value that is equivalent to the Windows ShowWindow() constants.

For your convenience, some command definitions have been reproduced here. You are advised, however, to consult your development system documentation for information relevant to your particular system.

Definition	Value
#define SW_HIDE	0
#define SW_SHOWNORMAL	1
#define SW_SHOWMINIMIZED	2
#define SW_SHOWMAXIMIZED	3
#define SW_SHOWNOACTIVATE	4
#define SW_SHOW	5
#define SW_MINIMIZE	6
#define SW_SHOWMINNOACTIVE	7
#define SW_SHOWNA	8
#define SW_RESTORE	9
#define SW_SHOWDEFAULT	10
#define SW_FORCEMINIMIZE	11

MOVE(x, y, width, height) Changes the position and size of the terminal session window.

There are two ways to start the DDE Server. You can use a full command line, such as:

```
jsbterm -s [drive][path]filename
```

However, if you are using Visual Basic for Applications to set up your DDE connection, the preferable way is to use the DDE Initiate command. The following example illustrates how to do this:

```
Chan = DDEInitiate("JSBTERM", "SYSTEM")

ConfigCommand = "[open(" + chr$(34) + ConfigFileName + Chr$(34) + ")]"

DDEExecute Chan, ConfigCommand
DDETerminate (Chan)
```

This starts jsbterm and then loads the requested **.zzt** file. A DDE conversation can then be created using the **.zzt** path name as the topic.

Supported DDE messages include:

```
DDE_REQUEST
DDE_POKE
DDE_EXECUTE
```

The following command line option may be used to run a terminal emulator from the Windows Run command:

```
jsbterm -s
```

Starts a terminal emulator which remains invisible until a DDE connection is made:

```
jsbterm -name =remotesystemname configurationfile .zzt
```

Uses the specified remote system name rather than the name specified in the configuration file:

```
jsberm -name=configuationfile.zzt
```

Prompts for a remote system name rather than using the name specified in the configuration file:

```
jsbterm telnet://remotesystemname
jsbterm telnet://username@remotesystemname
jsbterm telnet://username:password@remotesystemname
```

Allows you to specify a remote system and optionally a user name and password:

```
jsbterm -dial=phonenumber configurationfile.zzt
```

This value overrides the phone number specified in the **.zzt** configuration file.*

*The specified number must not contain any spaces. Delay characters such as commas can be used to provide a delay in the dialing. The duration of the delay depends on the configuration of your system.

Visit the Support area of the FutureSoft web site for sample script files (www.futuresoft.com/support/index.htm). The DDE samples contain a sample Visual Basic script which handles the connection and login to a UNIX Server, then loads a basic "sysadmin" application.

Note

Multiple Steams-based DDE links to the same application are not supported.

Automation Overview

Automation (formally OLE automation) Server applications allow created objects to be exposed to other applications which can then manipulate them. Interaction between the object and the automation Client is provided by a group of predefined functions. Your application code can be written in any language that supports automation, such as, Visual Basic or Visual C++. In MultiView, Automation can be used to provide an alternate connection procedure for terminal emulators.

WARNING: We strongly recommend that you be familiar with Automation before attempting to use the Automation functionality available in MultiView.

Some examples of Automation usage:

- Connecting to a remote system - Automation can be used to interact with MultiView by specifying an alternate method of connecting to a remote system via a terminal emulator. In such an example, control of the login procedure is passed from the terminal emulator to an Automation application or Client which will handle requesting of data from the remote UNIX host via the terminal emulator. This method bypasses the default login procedure which may be useful, for example, when a different login front end is required. Another use would be to provide an invisible terminal emulator required to run a remote application which does not need to be apparent to the user.

- Exchanging information - MultiView allows Automation to be used to exchange information between a terminal emulator (Server) and an application using Automation (Client). For example, a specified field in a remote database could be identified using a Client application which would then also be responsible for sending and retrieving data from the Server application.

Visit the Support area of the FutureSoft web site for sample Automation script files (www.futuresoft.com/support/index.htm).

Automation Supported Functions

MultiView supports the following functions. A detailed discussion of each follows.

- VT_BOOL Connect (VTS_NONE)
- VT_BOOL Disconnect (VTS_NONE)
- VT_BOOL GetRectChars (VTS_I2, VTS_I2, VTS_I2, VTS_I2)
- VT_BOOL MoveWindow(VTS_I2, VTS_I2, VTS_I2, VTS_I2)
- VT_BOOL OpenFile (VTS_BSTR)
- VT_BOOL Quit(VTS_NONE)
- VT_BOOL ResetStreamMode (VTS_NONE)
- VT_BOOL RetrieveData(VTS_NONE)
- VT_BOOL SendData(VTS_NONE)
- VT_BOOL SetStreamMode(VTS_NONE)
- VT_BOOL ShowWindow(VTS_I21)

- PROPERTY szWriteBuffer
- PROPERTY szReadBuffer

- PROPERTY Username
- PROPERTY RemoteHostName
- PROPERTY ProtocolName
- PROPERTY BlockingConnection

Creating an Automation Server

In the following, "Server" refers to MultiView running as an Automation Server. "Client" refers to the Automation Client that created the Server.

Creating objects within an Automation application

The following prog ID can be used to create Automation objects from your development environment.

JSBTerm.Docment - creates a terminal emulator object

Here is an example using Visual Basic:

```
Dim OTE as Object
Set OTE = CreateObject ("JSBTerm.Document")
```

VT_BOOL Connect(VTS_NONE)

Parameter

None

Comments

Makes a connection to the remote system specified in the .zzt configuration file or by the RemoteHost name and ProtocolName properties.

VT_BOOL Disconnect(VTS_NONE)

Parameter

None

Comments

Disconnects from the remote system.

VT_BOOLGetRectChars(VTS_I2, VTS_I2, VTS_I2, VTS_I2)

Parameter

VTS_I2 x,y coordinates, width and height respectively

Comments

Retrieves the characters that are stored in the specified screen location from the Server. The origin for the X and Y coordinates is in the upper-left corner on the screen and is zero-biased. In other words, (0,0) is actually column 1, row 1. The data is available via the szReadBuffer property.

VT_BOOL OpenFile(VTS_BSTR)

Parameter

VTS_BSTR

Specifies a terminal emulator based configuration file (.zzt).

Comments

Loads the Server with the specified configuration file.

VT_BOOL MoveWindow(VTS_I2, VTS_I2, VTS_I2, VTS_I2)

Parameter

VTS_I2 x,y coordinates, width and height respectively

Comments

Moves the terminal emulator window in one of the above states.

VT_BOOL Quit(VTS_NONE)

Parameter

None

Comments

Forces the Server to exit.

VT_BOOL ResetStreamMode(VTS_NONE)

Parameter

None

Comments

Resets the server to normal processing mode. The Server will read data when needed and process it accordingly.

VT_BOOL RetrieveData(VTS_NONE)

Parameter

None

Comments

The Client application notifies the terminal emulator to read some data from the remote UNIX host into the property szReadBuffer. This is directly affected by the BlockingConnection property described later in this chapter.

VT_BOOL SendData(VTS_NONE)

Parameter

None

Comments

The Client application notifies the terminal emulator to send the data that is stored within the property szWriteBuffer to the communications server.

VT_BOOL SetStreamMode(VTS_NONE)

Parameter

None

Comments

Sets the Server to stream mode. In this mode, the Server does not read any connection data unless requested by the Client application.

VT_BOOL ShowWindow(VTS_I2)

Parameter

VTS_I2

One of the following flags can be passed in this function:

0 Hide the window and pass activation to another window.

1 Activate and display a window. If the window is minimized or maximized, it is restored to its original size and position (same as flag 8 below).

2 Activate a window and display it as an icon.

3 Activate a window and display it as a maximized window.

4 Display a window in its most recent size and position. The currently active window remains active.

5 Activate a window and display it in its current size and position.

6 Minimize the specified window and activate the top-level window in the system list. Display a window as an icon. The currently active window remains active.

7 Display a window in its current state. The currently active window remains active.

8 Activate and display a window. If the window is minimized or maximized, it is restored to its original size and position (the same as flag 1 above).

Comments

Shows the terminal emulator window in one of the above states.

PROPERTY szWriteBuffer

A property string that stores the data to be sent to the remote UNIX host.

PROPERTY szReadBuffer

A property string that stores the data that has been received from the remote UNIX host.

PROPERTY UserName

String property that sets/gets username to connect to remote UNIX host.

PROPERTY RemoteHostName

String property that sets/gets remote UNIX host name to connect to.

PROPERTY ProtocolName

String that specifies the connection protocol to use, i.e., rlogin, telnet.

Note

For mode, RS232 or SSH, define connection parameters via a .zzc/.zzt file.

PROPERTY BlockingConnection

A boolean property that specifies if the **RetrieveData** method is to wait for data. A setting of TRUE means that it will wait for the data while a setting of FALSE means that the function **RetrieveData** immediately returns if the data is received or not.

Further Integration

This section contains a selection of useful information to help you integrate your remote applications at a low level using the Remote Command Server and the Mouse Messaging features.

Remote Command Server

It is possible to integrate applications further by using the Remote Command Server (RCS) facility. This allows remote applications in a terminal emulation window to start PC applications. The most useful application of RCS is to transfer files from a Client to a remote host without the need for user intervention. All that is required is that the terminal emulator be open on the Client machine. The general syntax is as follows, where the name of the Windows application is {application}:

```
\ 033[x(applicaton)\033\\
```

To open File Transfer and transfer files to the PC, for example, you would use the following:

```
\ 033[xjsbxfer.exe\033\\
```

To open the Windows Notepad, for example, send this string to the terminal emulator:

```
\ 033[xnotepad.exe\033\\
```

Control Characters

The next section refers to the following control characters.

Token	8-bit Octal	7-bit Octal
<CSI>	233	033 133
<ST>	234	033 134

Client IP Address

With MultiView it is possible to obtain the Client IP address. To do this the UNIX application must send an escape sequence to the terminal emulator using the following form:

Sequence	Description
<CSI>0~Zi	Unique escape sequence
(0|1)	"0" send IP Address of direct Client "1" Send IP Address of indirect Client
<ST>	Terminating sequence

A Direct Client is a Client which accesses the terminal emulator on the same machine as itself, for example, in a Workstation installation scenario. An indirect Client accesses the terminal emulator from a different machine to that which the terminal emulator itself is installed on, for example, in a WTS environment. If there is no indirect Client, then the IP address of the direct Client will be returned, even if an indirect Client was asked for.

An example of sending an escape sequence to request an IP address:

```
<CSI>0~Zi0<ST>
```

An example of a returned escape sequence with an IP address:

```
<CSI>0~zi193.129.109.20<ST>
```

If the request fails, i.e., no IP address available, the response is:

```
<CSI>0~zi<ST>
```

Mouse Messaging

MultiView supports Mouse Messaging functionality that allows a remote application to support mouse operations. Mouse Messaging is enabled by an escape sequence sent from the remote application to the terminal emulator. The sequence is unique to all emulators and allows for flags to specify required messages. The following list shows the information that is passed to the remote application:

Messages	Flags/Data	Message Type
Left, Middle & Right Buttons	LMR	
Single-clicks		S
Double-clicks		D

Messages	Flags/Data	Message Type
Mouse x,y cursor position	######	
Is Shift key pressed?	S	
Is Control key pressed?	C	
Mouse Move		m
Button Up		u
Button Down		d

A sequence must be sent to the terminal emulator to enable Server-side Mouse Messaging. The sequence is composed from the following elements: a unique escape sequence, parameter(s), and terminating sequence. When enabled, an acknowledgment message is sent back to the Server. The remote application can check for a response to verify that the terminal emulator supports this functionality.

Enabling and disabling sequences are shown below, with the mouse messaging format.

Enabling Sequence

Sequence	Description
<CSI>0~Zw	Unique Escape Sequence
D	Enable mouse up/down messages
L	Enable left button message reporting. By default, only includes position, single and double-clicks
M	Enable middle button message reporting. By default. only includes position, single and double-clicks
R	Enable right button message reporting. By default, only includes position, single and double-clicks
K	Enable Shift and Control key reporting
+	Enable advanced messaging of mouse moves and mouse up and down messages
Q	Quiet Flag; No Acknowledgement Package
#	Numeric values of mouse pointer required 0 = MultiView cursor, used as default 1 = Windows Arrow 2 = Windows I Beam 3 = Windows Hour Glass 4 = Windows Cross Hair 5 = Windows Up Arrow
ST	Terminating Sequence

Disabling Sequence

Sequence	Description
<CSI>0~Zw<ST>	Unique Escape Sequence

Mouse Message Format

Sequence	Description	
<CSI>0~zw	Unique Escape Sequence	
(L|-|x)	“L”	Left button pressed
	“-”	Left button not pressed
	“x”	Not reporting
(R|-|x)	“R”	Right button pressed
	“-”	Right button not pressed
	“x”	Not reporting
(M|-|x)	“M”	Middle button pressed
	“-”	Middle button not pressed
	“x”	Not reporting
(S|D|m|d|u)	“S”	Single-click
	“D”	Double-click
	“m”	Mouse move
	“d”	Button down
	“u”	Button up
###	X	Cursor position padded to 3 numeric characters
###	Y	Cursor position padded to 3 numeric characters
(s|-|x)	“s”	Shift key pressed
	“-”	Shift key not pressed
	“x”	Not reporting
(c|-|x)	“c”	Control key pressed
	“-”	Control key not pressed
	“x”	Not reporting
<ST>	Terminating Sequence	

Acknowledging Mouse Message Format

Sequence	Description
<CSI>0~zw	Mouse Message Escape Sequence (as above)
D	Next numeric sequence is the Windows double-click time-out value
#	A series of ASCII digits indicating the Windows double-click time-out value
<ST>	Terminating Sequence

Behavior

As MultiView already makes use of mouse messages, some functions are disabled when Server-side Mouse Messaging is enabled. The following functions are affected:

- For left button messaging, copy/paste highlighting is disabled. This includes double-click word markup. Scroll back is disabled and repositioned to the active area.
- For right-button messaging, right-click popup menus are disabled.

Mouse Button Double-click Programming

The double-click programming function allows a fully configurable sequence to be sent to the remote application in response to a double-click of any of the mouse buttons. The function can be configured from both the terminal emulator **.zzt** file, and by an escape sequence from the Server. Similar to Server-side Mouse Messaging, it overrides any existing Mouse Mapping configuration in the terminal emulator such as double-click single word markup.

Configuring Sequence

Sequence	Description
<CSI>0~Zm	Unique Escape Sequence
(L|M|R)	Configure Left, Middle or Right mouse double-click
{delimiter}	Single character to use as sequence delimiter
{sequence}	Required sequence
{delimiter}	Previously used delimiter character

Mouse Functionality Behavior

User interface considerations include:

- If a particular mouse operation has been programmed by a remote Server, this information is displayed in the Mouse Mapping tab in the Application Properties dialog by the appropriate mouse events but will NOT allow configuration. The Configure button is disabled.

- If the user has configured a mouse event sequence, but currently has another function selected, say disabled, then the sequence should still be stored in the .zzt file. This allows them to switch back the functionality to send the already configured sequence.

- A single-click and double-click are only valid if the character positions are the same when the mouse button went down as when the mouse button came up.

- A single-click event is only sent when the time-out value for a double-click has expired. The double-click time-out can be set in the Mouse Control Panel.

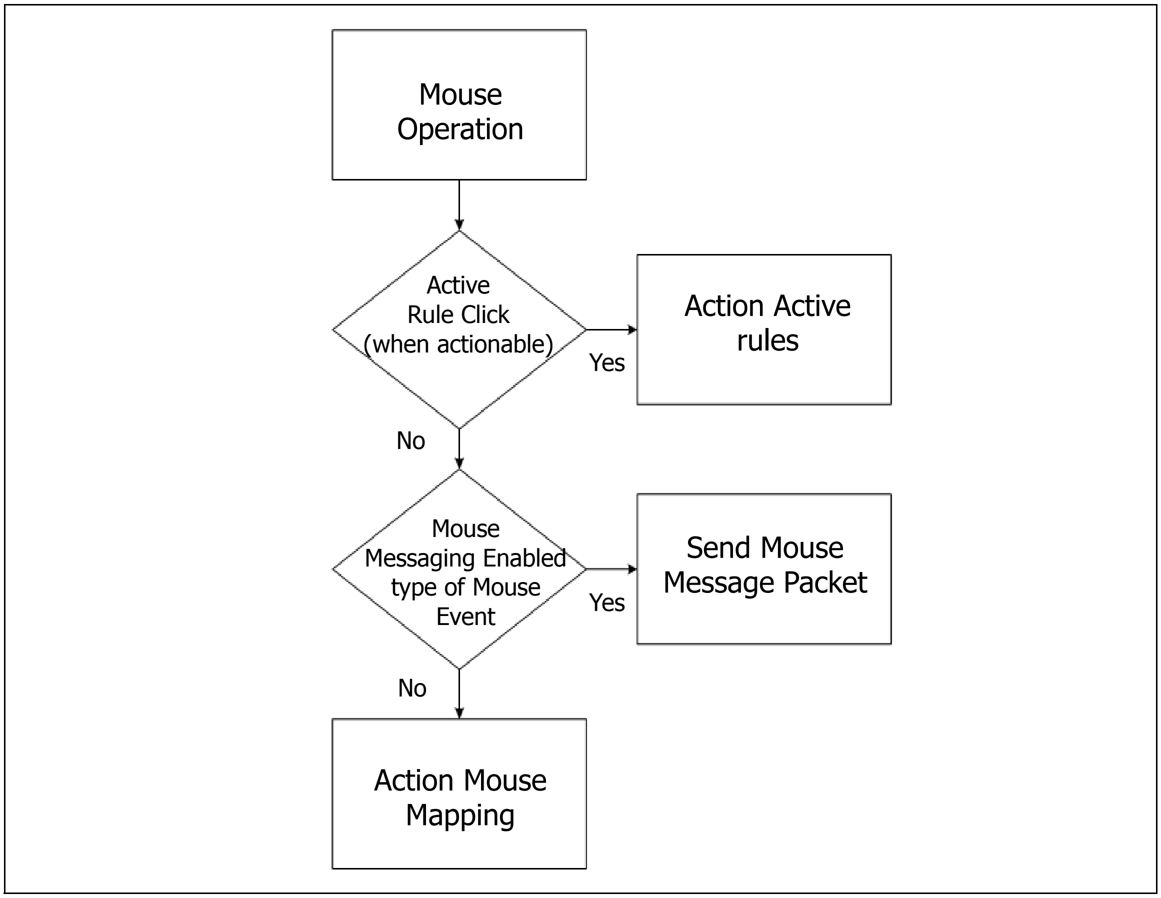

Mouse Operation
Active Rule Click (when actionable)
Yes
Action Active rules
No
Mouse Messaging Enabled type of Mouse Event
Yes
Send Mouse Message Packet
No
Action Mouse Mapping

Figure 7.1
Mouse Message behavior

The flowchart in Figure 7.1 shows the behavior Mouse Message displays when enabled. For example, since markup is part of Mouse Mapping, if Mouse Message is enabled for the left mouse button, the markup is not available.

Chapter 8

Troubleshooting

Troubleshooting MultiView Terminal Emulator

In the event that a problem occurs when using MultiView Terminal Emulator, this section provides insights into how to resolve your problem or where to find more information.

Client Display

Why am I having problems with my display on the Client?

If you are using a Java control in Internet Explorer you could be experiencing problems with your display. These could be such things as the menu button getting larger each time it is clicked, the menu flashes briefly and then disappears, or other such anomalies. If this is the case you may need to upgrade your Microsoft Virtual Machine to the most current version available. To do this, run IE setup from the Microsoft web site and make sure that you select Custom installation and then select Microsoft Virtual Machine from the list.

Printing Issues

When using auxiliary (pass-through) printing, how do I pass through formatting codes without alteration by the printer driver?

"Auxiliary" or "pass-through" printing is activated by receipt of a special control code sequence and temporarily redirects the data stream to the specified printer. Display to the screen is resumed when a corresponding "end printing" sequence is received. The destination printer may be specified as either the default Windows printer, or a specific print device or file. By default, the output is formatted for the specific device. For example, if a PostScript printer is selected as the destination, the data stream is converted to PostScript by the driver before sending it to the printer. Sometimes an application pre-formats a job with all the necessary printer control codes for a specific printer. In this case we do not want the printer driver to attempt to format the print stream any further because corrupted output results.

To suppress printer formatting

1. On the **Configure** menu, select **Terminal**.
2. In the **Terminal Properties** dialog, click **Configure**.
3. In the **Terminal Settings** dialog
   a. Click the **Printing** tab.
   b. On the **Printing** tab, enable the **Send Data Unformatted** option (apply check mark).
   c. Click **OK**.
4. In the **Terminal Properties** dialog, click **OK**.

Why does my remote print job request not appear?

- Make sure that your printer is configured correctly and that you have the most up-to-date print drivers installed. If in doubt, check with the manufacturer of your printer.

- The print job may have been formatted twice, or not at all. Set the Remote Print Server to format jobs if required. If your remote application formats jobs, formatting in the Remote Print Server should be turned off, and vice versa.

  ### To turn off formatting of received print jobs

  1. On the **Start** menu, open **Remote Print Server**.
  2. In the **Remote Print Server Configuration** dialog, click the **Printers** tab.
  3. On the **Printers** tab, select the printer and click **Properties**.
  4. In the **Printer Queue Properties** dialog

     a. Enable the **Do not format received print jobs** option (apply check mark).

     b. Click **OK**.

- Is the printer's queue name correct on your PC? Like formatting, queue names are defined in the **Remote Print Server Configuration** dialog.

  1. On the **Start** menu (or icon tray), open **Remote Print Server**.
  2. In the **Remote Print Server Configuration** dialog, click the **Printers** tab.
  3. On the **Printers** tab, select the printer and click **Configure**.
  4. In the **Printer Queue Properties** dialog, enter the correct queue name in the **Queue Name** field and click **OK**.
  5. In the **Remote Print Server Configuration** dialog, click **OK**.

Connecting to a Remote System

How do I enable users to login automatically?

Create a login script using the Learn Script function. Refer to Chapter 6: Scripting for more information.

How can I prevent a session configured to auto connect from connecting?

If a **.zzt** file is configured in the **Session Properties** window to auto connect, but you want to defeat the connection, hold down the **Shift** key immediately after double-clicking the shortcut or **.zzt** file to start the session. In most cases the window does not become the topmost window, but you will be able to make it visible by clicking its icon on the task bar.

Error Message: Cannot create sockets Windows Sockets TCP/IP version 1.1

This message may result from:

- Incorrectly configured hosts file (PC side)
- More than one “winsock32.dll” file
- No host file available
- A host IP address has been specified instead of a remote system name

Resetting the Terminal Emulator

In certain error situations it is possible for the terminal emulator to freeze. In this situation, the keyboard may be locked and printing may not be functioning, yet the remote application is still running. In this situation you can use the Reset terminal function to reset the terminal emulator without forcing a disconnection from the remote system as this may cause further problems.

To perform a hard reset of the terminal emulator

1 On the Remote menu, select Reset terminal.

The effect of this action depends on the terminal type that you are using:

SCO/AT386
- EP/DP
- Select the Primary Font, i.e., out of graphics mode
- Auto wrap
- Reset color settings (in case set to Blue on Blue)

VT100
- EP/DP
- Reset LNM Mode (Line Feed/New Line)
- Reset SCN Mode (Screen)
- Reset DECPEX (Printer extent)
- Reset DECCKM (Cursor keys)
- Set DECAWM (Auto wrap)
- Set DECTCEM (Text cursor enabled)
- Put in multinational mode
- Clear the screen
- Home the cursor
- Reset the character set mapping tables

VT220
- VT100 Settings followed by:
- Clear the downloadable character set
- Reset 8-bit character set mapping tables

VT320/TI928
- VT220 Settings followed by:
- Clear Status line
- Reset color settings

VT420
- VT320 Settings followed by:
- Clear all off-screen page memory
- Sets SGR to normal rendition
- Clears all macro definitions

Wyse60

- EP/DP
- Reset character set mapping tables
- Clear the status line text
- Clear the label line text
- Reset page memory to default page setting
- Unlock keyboard

Warning: Use of the Terminal reset function is not recommended unless all options have been explored.

Note

Resetting the terminal emulator can be used to test "Override Multinational by default." See Chapter 3: Installation and Configuration for more information.

Capture Utility

MultiView provides an advanced capture utility to help you communicate any problems to Technical Support. It is possible to capture text, as it appears in the terminal emulation window, to a file or to a printer.

The use the Capture utility

1 On the **View** menu, select **Capture Bar**.

2 On the **Capture** toolbar, click **Capture Properties**.

3 On the **Capture** tab of the **Session Properties** dialog

a In the **Default Capture** group, select the type of capture output.

b If you chose **File** in step 3a, select an option in the **File** group and supply a file name, if required.

c If you chose **Printer** in step 3a, select an option in the **Printer Settings** group and set up printer options, if required.

4 In the **Session Properties** dialog, click **OK**.

5 On the Capture toolbar, click **Start Capture**.

6 When you are ready to stop the capture process, click **Stop Capture** on the Capture toolbar.

Index