



Reflection Desktop Deployment Guide

Version 17.0 SP1

Legal Notice

For information about legal notices, trademarks, disclaimers, warranties, export and other use restrictions, U.S. Government rights, patent policy, and FIPS compliance, see <https://www.microfocus.com/about/legal/>.

© Copyright 2021 Micro Focus or one of its affiliates.

The only warranties for products and services of Micro Focus and its affiliates and licensors (“Micro Focus”) are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Micro Focus shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

Contents

Introduction	7
1 Get Started	9
Exploring Deployment Options	9
Finding Information Specific to your Deployment Environment	9
Creating and Customizing Reflection Sessions	10
Customizing Reflection	11
Locking Down Settings and Controls	12
Designing and Creating Secure Solutions	13
Packaging Custom Files	14
Customizing the Installation	14
Best Practices for Securing Reflection Desktop	15
Best Practices for Planning a Reflection Deployment	19
1. Identify Technical and User Requirements, Subject Matter Resources, and Risk Mitigation	19
2. Inventory and Analyze User Requirements, Macros, Configuration Files, and Legacy Applications	20
3. Assess Which Existing Files To Carry Forward and Test	21
4. Package and Test	21
Setting up a Basic Deployment	23
Deploy Reflection out-of-the-box	24
Create and Deploy Customized session files	24
2 Install Reflection and Set Up Administrative Tools	29
Installation	29
System Requirements	30
Upgrading from Previous Versions	31
Install Reflection on a Workstation	32
Features Selection Tab	33
Set Your User Data Directory	33
Advanced Tab	33
Setting up the Reflection Administrative Tools	34
About the Tools	34
Set up an Environment for Deployment Using Reflection Administrative Tools	36
3 Create and Customize Sessions	39
Create and Customize Session Documents	39
Walkthrough: Set up and Customize a Session	41
Create SSL/TLS or SSH Session Documents	42
Digital Certificates and Reflection Certificate Manager	42
Set up SSL/TLS Connections	44
Set up Secure Shell Connections	44
Set up Session Templates	46
Create a Session Template	46
Control the Types of Sessions Users can Create	46
Configure Reflection for PKI Auto Sign-on	48

Centrally Manage VBA Macros	49
Walkthrough: Set up a Session to Reference Macros in a Session Document File	51
Walkthrough: Set up and deploy a SharedMacros file	54
4 Change Specific Session Settings	59
Best Practices, and Approaches for Applying SettingsUpdate Files to Sessions	60
Approaches for Applying SettingsUpdate Files to Sessions	61
Best Practices for Creating and Applying SettingsUpdate Files to Sessions	61
Which Types of Session Documents and Settings Can be Updated?.	62
Apply an Update to all Session Document Files	62
Setting up Local Update Files	63
Apply Automatic Updates to Select Session Document Files	65
Setting up referenced update files	66
Set up Automatic Updates for All Session Document Files	70
Setting up a combination of local and referenced update files	71
Create or Modify SettingsUpdate Files	74
Step One: Unlock access for saving update files	74
Step Two: Deploy modified update files	76
5 Protect Data	77
Add Trusted Locations	77
Configure API and Macro Security	78
6 Set up Information Privacy	79
What is PCI DSS?	80
Configure Information Privacy Settings.	80
Setting up Redaction of Primary Account Numbers.	81
Setting up and using Simple PAN Detection	82
Setting up and Using Simple PAN Detection with Preceding Text	83
Setting up and Using Reflection PAN Detection	85
Configure Information Privacy with Group policy	90
Setting up Privacy Filters	90
When to Use Privacy Filters	91
How to set up Privacy Filters.	91
More About Information Privacy.	92
7 Customize the Reflection Workspace	93
Customize Workspace Settings Directly With Reflection.	94
Configure and Automatically Package Workspace Settings.	95
Update Only Specific Workspace Settings.	98
Walkthrough: Customize Reflection Appearance and Behavior	99
8 Control Access to “Lock Down” Settings and Controls	103
Control Access to Settings and Controls with Reflection Administrative Tools	104
Specify Access Using Permissions Manager with the Installation Customization Tool	105
Specify Access Using Permissions Manager	108
Walkthrough: Restrict Access to Settings and Controls	109

Control Access to Settings and Controls with Microsoft Group Policy	111
Install Group Policy Templates	112
Set Access with Group Policy	112
9 Package Configuration Files	115
Package Sessions and Custom Settings Files	116
Create and Automatically Package Workspace and Access file Settings	117
Walkthrough: Create a Package with the Installation Customization Tool	117
Customized Files that Must be Deployed to Specific Locations	119
Workspace Settings	119
Shared Security Settings	121
Reflection FTP Client Settings	122
Regular Session Document Files, Update Files, and Session Settings	122
Custom Office Productivity Templates	123
Plus Archives	123
VBA SharedMacros Files and the VBA Common Project	123
Other Customized Files and Supported Files from Other Products	124
Other Customized Files	124
Other Supported Files	125
Supported Extra! Files	125
Supported Legacy Reflection Files	126
Supported Files from Other Products	127
10 Modify the Installation	129
Create or Modify a Transform	129
Change the Installation Directory	130
Modify Setup Properties	131
Add/Modify Registry Data	131
Select Features, Components, and Languages	132
Add (Chain) Installations and Run Programs	133
Install the Reflection Help	134
Predefined System Folders	135
Configure Shortcuts	136
Walkthrough: Create a Transform	137
Apply a Transform to Your Installation	138
11 Deploy Reflection	141
Deploy with the Reflection Setup program	141
Deploy with MSI	142
Deploy Reflection from MSI Command Line	142
Deploy Companion MSI File from MSI Command Line	144
Publish with Active Directory	144
Deploy with System Center Configuration Manager	144
Apply Software Updates (Service Packs or Patches)	145
Apply a Patch to a Workstation	145
Apply a Patch to an Administrative Installation	146
Directly Install a Patch on Workstations	147
Perform a new Installation with a Patch	147
Remove an Installation	147
Repair an Installation	149

12 Use a Centralized Management Server	151
Requirements and Best Practices for Using Centralized Management	152
Requirements for Centralized Management	152
Best Practices for using Centralized Management	152
Create and Deploy Sessions and Settings with the MSS Administrative Console	153
Create or Modify a Centrally Managed Session	153
Make Centrally Managed Sessions Available to Users	153
Use Central Management to Deploy MSI Packages	153
Enable Certificate Management for IBM Terminals	154
Enable Usage Metering	154
How can you use metering?	154
Setting up Metering.	155
Connect to Hosts using the Security Proxy Add-On	156
Requirements and Other Information for Using the Security Proxy	159
Connect using End-to-End Encryption in 3270 TLS Sessions	160
Connect using End-to-End Encryption in VT SSH Sessions	161
Set Up Terminal ID Management for Reflection Desktop Sessions.	163
Set up Automated Sign-On for Mainframe Sessions	166
Implementing Automated Sign-On for Mainframe.	167
13 Move from Extra! to Reflection	169
What's Similar?	170
What's Changed?	170
The Installation Customization Tool	171
Permissions Manager	171
Compound Session Documents	173
What's New?	174
Installation Improvements	174
Workspace Settings	175
The Trust Center.	175
User Interface Options	177
Other Features	178
Best Practices for Moving from Extra!	178
Videos for Moving from Extra!	180
Glossary	181

Introduction

This guide shows how to prepare for your Reflection Desktop deployment. It includes information for all of the phases of your deployment process.

These articles	Show how to
“Get Started” on page 9	Get started with your deployment. This includes exploring your deployment options, best practices, and a walkthrough of a simple deployment option.
“Install Reflection and Set Up Administrative Tools” on page 29	Install Reflection and make sure you meet system requirements. Review the administrative tools and set up a system for customization, testing, and deployment.
“Create and Customize Sessions” on page 39	Create and customize session document files that you can deploy to users.
“Change Specific Session Settings” on page 59	Create and deploy update files to apply specific settings in session document files without changing other settings.
“Protect Data” on page 77	Set up Reflection to add trusted locations, redact sensitive data (such as credit card numbers), and control access to the Reflection API.
“Set up Information Privacy” on page 79	Set up information privacy features that are designed to help you comply with industry and government regulations, like Payment Card Industry Data Security Standards (PCI DSS) that address data protection concerns.
Customize Reflection	Customize Reflection to change the appearance and basic functionality of the main Reflection window and to control access to controls and settings.
“Package Configuration Files” on page 115	Create an MSI package to install custom files you created when you customized the Reflection sessions or the Reflection workspace.
Modify the Installation	Create a transform (MST file) to customize how Reflection is installed on user workstations.
Deploy Reflection	Deploy Reflection, session document files, and other configuration files. Instructions are included for using the Reflection setup.exe program and deploying directly with MSI as well as deploying with Microsoft Active Directory and Microsoft System Center Configuration Manager.
“Use a Centralized Management Server” on page 151	Set up and centrally manage sessions on the Management and Security Server.

1 Get Started

You can choose from several different approaches for installing and deploying Reflection Desktop. Deciding which approach to use is typically based on a variety of factors, including your approved business processes, the scale of the deployment, your deployment tools, and whether you want to customize the installation.

The following articles provide high level information to help you get started:

[“Exploring Deployment Options” on page 9](#) highlights common deployment tasks and shows where to find more information about them.

[“Best Practices for Securing Reflection Desktop” on page 15](#) includes recommended practices to help you design a secure terminal emulation solution.

[“Best Practices for Planning a Reflection Deployment” on page 19](#) provides recommended practices to help you avoid common mistakes and problems.

[“Setting up a Basic Deployment” on page 23](#) shows how to deploy Reflection by placing the required files on a file server.

Exploring Deployment Options

The Reflection Deployment Guide includes best practices, instructions, and other information for a number of deployment environments.

Some content may apply to you regardless of your deployment environment. You may be interested in:

- ◆ [“Finding Information Specific to your Deployment Environment” on page 9](#)
- ◆ [“Creating and Customizing Reflection Sessions” on page 10](#)
- ◆ [“Customizing Reflection” on page 11](#)
- ◆ [“Locking Down Settings and Controls” on page 12](#)
- ◆ [“Designing and Creating Secure Solutions” on page 13](#)
- ◆ [“Packaging Custom Files” on page 14](#)
- ◆ [“Customizing the Installation” on page 14](#)

Finding Information Specific to your Deployment Environment

Use these guidelines to find the sections that pertain to your deployment environment.

If you are using...	See
A share that users install from	Setting up a simple deployment in this guide.

If you are using...	See
A reference image	Information about how to select installation options and how to customize settings in the Reflection Help: Installation Get Started How Do I?
Group Policy	Information about how to install and use templates for locking down Reflection and instructions for publishing: Control Access to Settings and Controls with Microsoft Group Policy Publish with Active Directory
Setup.exe	Information about setup and instructions for deploying on the Setup command line: “Setting up the Reflection Administrative Tools” on page 34 Deploy with the Reflection Setup program
MSI Command line	Information about prerequisites when deploying with MSI directly. Deploy with MSI
Host Access Management Server	Information about setting up Reflection to use centrally managed sessions. “Use a Centralized Management Server” on page 151

Creating and Customizing Reflection Sessions

Reflection stores the information required to connect to hosts in configuration files called session document files. You can create session document files that have connection information and other settings and then deploy them independently of your installation.

You can also customize session document files to use custom keyboard maps, controls, themes, and other settings. To customize these settings, see [“Create and Customize Session Documents” on page 39](#).

Using Compound Session Document Files

By default, Reflection saves custom settings for QuickPads, keyboard maps, themes, mouse maps, hotspots, and ribbons in separate configuration files that you will need to deploy along with the session document files that reference them. But you can simplify your deployment by saving your session documents as compound session files, which include all of these settings.



Compound Session Document files include all of the customized settings for QuickPads, keyboard maps, themes, mouse maps, hotspots, and ribbons. These files allow you to deploy your sessions without having to deploy dependant files for these settings.

Session Document files save these settings in separate files. If you save sessions using this default, you'll need to make sure that you deploy the custom files to the correct locations as shown in ["Customized Files that Must be Deployed to Specific Locations"](#) on page 119.

Compound files are easier to deploy because you don't have to deploy the supporting files. Regular session files offer more flexibility for sharing common configurations. For example, you can reference one custom keyboard map from several regular session files. With compound session documents, you have to recreate the custom settings in each file.

NOTE: Not all custom settings are saved in compound session files. For example, settings such as `ssh_config` and `ssh_known_hosts` required for Secure Shell connections are not saved in these files.

Creating Custom Templates

Templates provide default configurations for connecting to specified file types. Micro Focus Reflection Desktop ships with default templates for each of the supported session types included with your product. You can also create and deploy custom templates designed to meet the needs of your users. See ["Set up Session Templates"](#) on page 46.

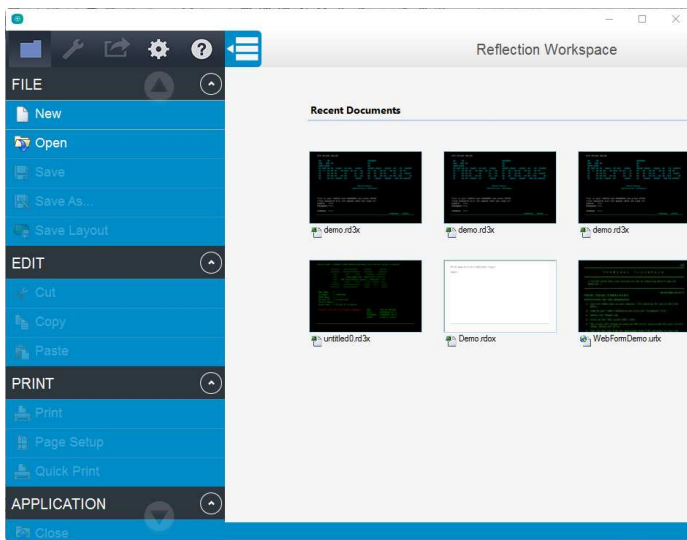
Customizing Reflection

You can customize Reflection by changing it's appearance and functionality and by locking down access to settings and controls.

Changing Reflection's appearance and functionality

You can change the appearance and basic functionality of the main Reflection window in a variety of ways, including specifying startup macros and actions.

You can also change the appearance of the window (for example, by opening Reflection in TouchUx mode).



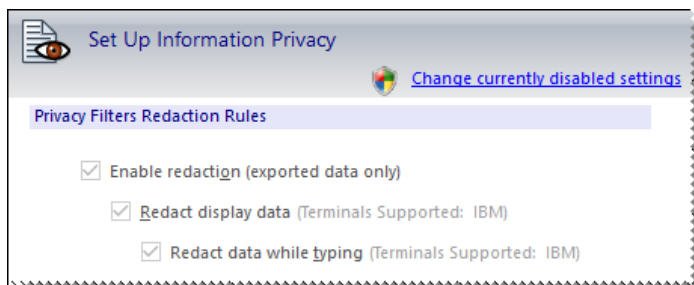
For more, see [Chapter 7, “Customize the Reflection Workspace,”](#) on page 93.

Locking Down Settings and Controls

You can lock down Reflection to limit access to settings and controls so that they are not available to users. This allows you to simplify support requirements and resolve security concerns.

To prevent a user from changing a setting, you set the permission level for that setting or control to **Restricted**. For example, you could restrict the users’ ability to modify security settings.

As shown below, the restricted settings are grayed out. The security shield and the **Change currently disabled settings** link indicate administrative access is required to change them.



For more about locking down settings and controls, see:

- ◆ [Control Access to Settings and Controls with Reflection Administrative Tools](#)
- ◆ [Control Access to Settings and Controls with Microsoft Group Policy](#)

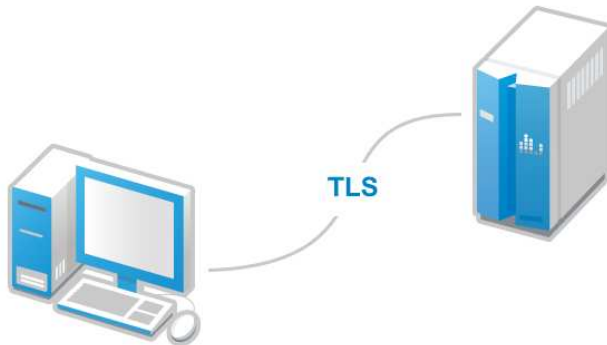
Designing and Creating Secure Solutions

Reflection provides support for a number of secure solutions. You can set up Reflection to protect your working environment from information theft, and your data from potential damage caused by opening documents from non-trusted sources, using the following methods:

- ◆ [“Creating secure sessions for data in motion” on page 13](#) enables secure, encrypted connections using secure protocols such as SSH or TLS.
- ◆ [“Setting up Information Privacy” on page 13](#) to mask sensitive data helps enable compliance with PCI-DSS requirements.
- ◆ [“Defining trusted locations” on page 14](#) from which you can safely open (and store) documents protects data from potential damage caused by opening documents from non-trusted sources.
- ◆ [“Setting up macro and API security” on page 14](#) controls access to the Reflection API.

Creating secure sessions for data in motion

Set up secure sessions for a number of secure protocols including FIPS 140-2 validation, TLS encryption, SSH, Kerberos, SOCKS, and PKI. See [“Create SSL/TLS or SSH Session Documents” on page 42](#).



Setting up Information Privacy

Set up Information Privacy to help comply with PCI-DSS requirements, including redaction of sensitive data such as credit card or social security numbers.

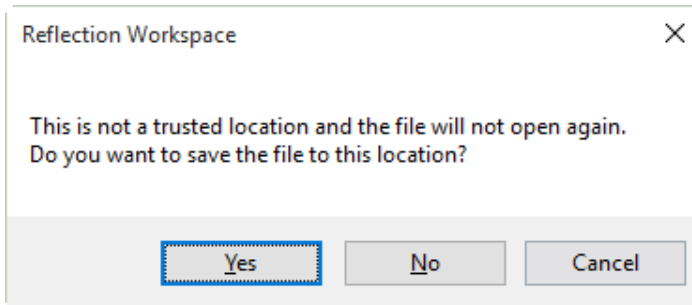
CARD TYPE:	_____
CARD NUMBER:	*****3488
EXP DATE:	_____
CARD NAME:	_____
TYPE:	_____
COMMENTS:	_____

You can also set up the Reflection API to log access to unredacted data.

See [“Set up Information Privacy” on page 79](#).

Defining trusted locations

Define Trusted Locations to differentiate safe files from potentially harmful files. When a file is in a trusted location, its files are assumed to be safe. If you try to save a file in a location that is not trusted, Reflection warns that it will not be able to reopen the file.



Reflection enforces trusted locations by default, so if you want to save sessions in directories that are not default trusted locations, you'll have to define these locations or disable the Trusted Locations feature. See [“Add Trusted Locations” on page 77](#).

Setting up macro and API security

You can enable or disable the Reflection .NET API, determine whether Reflection legacy macros are supported, and determine which legacy API has preference for the `GetObject()` method used to retrieve API COM objects.

You can also specify whether to run restricted actions that are initiated through a macro or API call without elevating permissions.

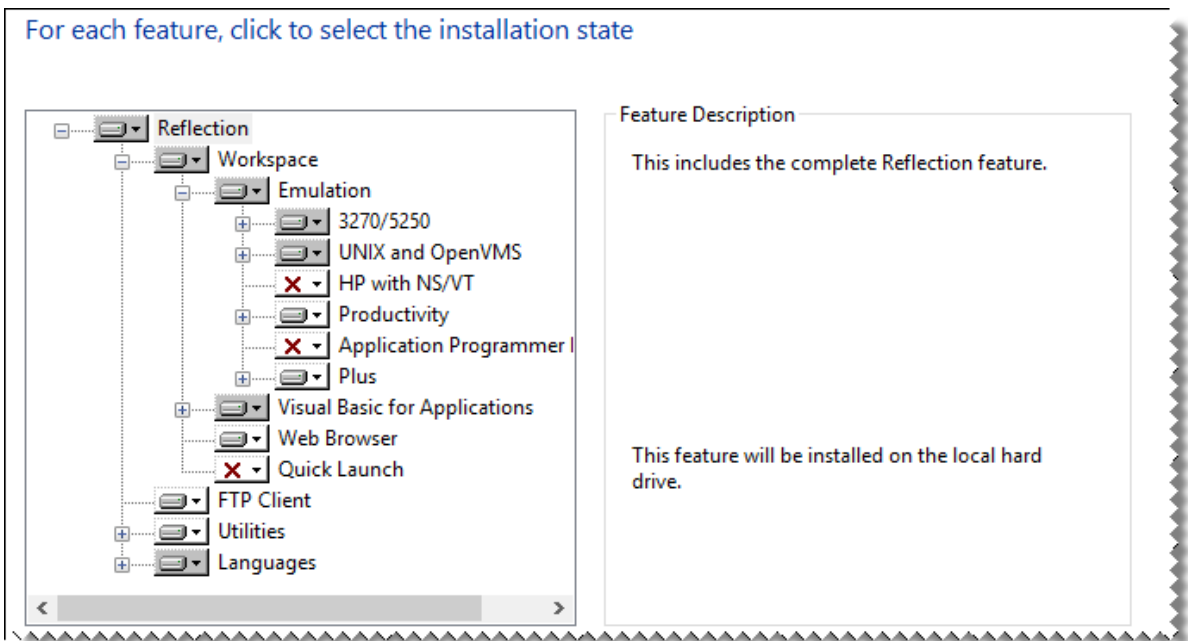
Packaging Custom Files

After you customize session files or the Reflection workspace, you can bundle the customized session and settings files into a deployable companion installer package (.msi) to make it easier to deploy to your users. You can deploy companion installer packages separately or you can add them to a customized installation. You can also create and install packages at any time after the initial installation. See [Package Sessions and Custom Settings Files](#).

Customizing the Installation

Create and deploy a transform to customize how Reflection is installed on user workstations. As defined by Microsoft, "a transform is a collection of changes applied to an installation. By applying a transform (*.mst) to a base installation package, the installer can add or replace data in the installation database."

For example, by deploying the transform with the Reflection base installation package, you can select which features to install.



You can also specify the installation directory or the user data location, change the Remove or Add commands from the Windows Uninstall or change a program list, and change other default settings.

For more about creating transforms, see [“Modify the Installation” on page 129](#)

Best Practices for Securing Reflection Desktop

Reflection Desktop has a number of security features designed to protect your personal data and prevent it from being read by unauthorized users.

Following these best practices for securing Reflection Desktop will help you design a secure terminal emulation solution. These best practices include high-level recommendations and considerations. For additional detailed information about the security features supported by Reflection Desktop, see [Secure Connections](#) in the Reflection Desktop Help.

✓ Monitor Reflection Desktop security alerts

Micro Focus regularly publishes security alerts in knowledge base articles. You can find the most recent alerts at:

[Security Alerts – Reflection Desktop \(https://support.microfocus.com/kb/security-alerts.php?id=Reflection_Desktop\)](https://support.microfocus.com/kb/security-alerts.php?id=Reflection_Desktop)

✓ Use the highest level of TLS for secure connections

Reflection Desktop supports TLS 1.2 for IBM 3270 and 5250 sessions. Reflection Desktop 17.0 and higher support TLS 1.3. If your environment supports TLS 1.3, consider using this version.

✓ Use the strongest encryption ciphers available in your environment

Reflection Desktop 17.0 contains enhanced capabilities that allow you to disable cipher suites which are less secure, and also to enable ciphers used in your environment that you consider to be more secure. See [SSL/TLS \(Security Properties Dialog Box\)](#).

✓ Stay current with versioning in Reflection Desktop

Staying current with major new releases, service packs and updates (when available) ensures you have deployed the latest security patches and fixes to your end-users. Micro Focus strives to make each new version of Reflection Desktop more secure than the last. The Host Connectivity team responsible for the development of new versions is a dedicated staff of senior engineers who have a strong focus on making the product more secure. They evaluate all security alerts against the currently released products and incorporate updates in the next versions. Micro Focus Development teams use a Secure Development Lifecycle process, where ongoing training and product review ensures that our software does not contain security vulnerabilities and that all new features are developed with security in mind.

✓ Use Certificates in a secure manner

Configure Reflection Desktop to prevent security risks associated with certificates.

- ◆ Don't allow host authentication with invalidated certificates. To prevent this security risk, make sure the **Retrieve and validate certificate chains** setting in the SSL/TLS Security Properties Dialog Box is enabled. This setting specifies whether certificates presented for host authentication are checked to determine if they are valid and signed by a trusted CA. Disabling this option can make connections vulnerable to man-in-the-middle attacks, which could compromise the security of the connection. See [SSL/TLS \(Security Properties Dialog Box\)](#).
- ◆ Consider disabling the use of the Windows certificate store for Reflection connections. Reflection applications can be configured to authenticate using only those certificates located in the Reflection store or using both the Windows and the Reflection store. Disabling use of the Windows certificate store enables you to have greater control over which certificates are used for authentication. Certificates can be added to the Windows store in a variety of ways, and you may not want to allow use of all of those certificates for authenticating Reflection sessions. When use of the Windows store is disabled, only the certificates you have imported into the Reflection store are used for host authentication. See [Enabling and Disabling Use of the Windows Certificate Store](#).

✓ Control access to product features that are not needed

Limit access to settings and controls and consider setting up custom templates with locked down settings so that users must use security settings, such as the latest TLS versions, when they create new sessions.

You can restrict access to almost any of the Reflection settings or controls to prevent users from changing values, like the host address that a session connects to. This allows you to simplify support requirements and resolve security concerns. Administrative access is required to change settings and users cannot change these options unless they elevate their access level to administrator.

Access to almost every Reflection Desktop feature can be enabled or disabled with Microsoft Group Policy or Reflection *.ACCESS files that you can create with Reflection Desktop administrative tools. See [Chapter 8, “Control Access to “Lock Down” Settings and Controls,” on page 103](#).

Control Access

Lock down or disable features which can be used in an insecure manner. For example, allowing users access to programming and macro languages could allow users to record or write automation code that includes user IDs and passwords. This code could then be freely distributed among users, creating a security risk.

- ◆ Using the Reflection Group policy settings, as documented in [Technical Information Document 7024743](#), and [Control Access to Settings and Controls with Microsoft Group Policy](#) to disable specific features leads to more secure user environments.
- ◆ Alternatively, you can use Reflection Desktop administrative tools to create and deploy .ACCESS files that lock down specific settings. See [Control Access to Settings and Controls with Reflection Administrative Tools](#).

Set up Session Templates

Deploy session templates using pre-configured settings to control the types of sessions that users can create. For example, you can create templates that have pre-configured SSL/TLS settings and then lock down these settings with Group Policy or Reflection Desktop administrative tools. Then configure Reflection to hide the built-in templates so that only the custom templates are available. See [Set up Session Templates](#).

✓ Configure the Reflection Desktop Trust Center to protect data and information privacy

Use the Trust Center to protect your working environment from information theft, and your data from potential damage caused by opening documents from non-trusted sources.

You can configure settings to protect the following types of data and information:

Trusted Locations

A trusted location is a directory that is designated as a secure source for opening files. By default, Reflection allows users to open documents only in directories specified as trusted locations and prevents them from opening untrusted documents outside of these locations.

Information Privacy

Consider protecting sensitive data such as credit card Primary Account Numbers (PANs), phone numbers, and US Social Security numbers. Information Privacy allows you to configure Reflection Desktop so that the sensitive data is not displayed on the screen or in productivity features, such as Screen History. It also allows you to require secure connections and to redact PANs in logs.

API and Macro Security

Consider the following options for handling the Reflection Desktop API and macros. You can configure Trust Center settings to:

- ◆ Enable or disable the Reflection Desktop .NET API.
- ◆ Determine if Reflection legacy macros are supported or not.
- ◆ Specify what will happen if an action that has been restricted through Group Policy or .ACCESS files is initiated through a macro or API call.

See [Protecting Data and Information Privacy](#).

✓ Do not save passwords in macros

Including user IDs or passwords in macros or other automation code creates a security risk.

When a VBA macro is recorded in Reflection Desktop, a password prompt dialog box is automatically added to the macro in place of actually recording the password. Using this password prompt in macros that require user credentials prevents security risks.

There may be circumstances where you need to consider embedding a password in a macro, although this is a security risk. Undertake this process with extreme caution and after careful deliberations of the potential for the password being compromised by others who should not have the information, as shown in [Technical Information Document 7024220](#).

NOTE: The Reflection Desktop software does not store Host usernames or passwords anywhere in the product configuration files and Reflection Workspace logs do not capture Host usernames or passwords.

✓ Consider using a centralized management server to manage host sessions

You can centrally manage, secure, and monitor users' access to host connections with the Micro Focus Host Access Management and Security Server (MSS), a separately available product that is designed to provide centralized management for Reflection sessions.

- ◆ Using this centralized management server, you can grant or deny access based on group or role, quickly apply security updates and configuration changes to align with changing regulatory or business needs, and make post-install adjustments on the fly. MSS allows you to configure and lock down large numbers of desktops with ease. See [Use a Centralized Management Server](#).
- ◆ Using the MSS Advanced Authentication Add-on, you can configure a multi-factor authentication solution that enables you to protect your sensitive data by using a more advanced way of authentication on top of the typical username and password authentication. With Advanced Authentication, you can authenticate on diverse platforms by using different types of authenticators such as Fingerprint, Card, and OTP. Advanced Authentication provides a single authentication framework that ensures secure access to all your devices with minimal administration. See [Micro Focus Advanced Authentication](#).
- ◆ Using the Automated Sign-On for Mainframe Add-On, you can enable a user to authenticate to a front-end system using a modern form of authentication (such as a smart card, certificate, LDAP password, Kerberos, etc.) and then be automatically logged on to a z/OS mainframe application. See [Set up Automated Sign-On for Mainframe Sessions](#).

✓ Consider encrypting session documents

You can encrypt 3270, 5250, and Open Systems session documents to protect them against unauthorized changes. Encryption effectively scrambles the data in a session document, helping to prevent unauthorized users from reading and changing the file's contents. For best results, use document encryption in conjunction with the encryption options in Reflection Permissions Manager. See [Encrypt a Session File](#).

Best Practices for Planning a Reflection Deployment

Follow these best practices to avoid common problems and make sure your deployment meets customer and technical requirements without disrupting users.

- ◆ [“1. Identify Technical and User Requirements, Subject Matter Resources, and Risk Mitigation” on page 19](#)
- ◆ [“2. Inventory and Analyze User Requirements, Macros, Configuration Files, and Legacy Applications” on page 20](#)
- ◆ [“3. Assess Which Existing Files To Carry Forward and Test” on page 21](#)
- ◆ [“4. Package and Test” on page 21](#)

1. Identify Technical and User Requirements, Subject Matter Resources, and Risk Mitigation

Large-scale terminal emulation deployments have significant risks in terms of delays, cost, and user acceptance. Performing a high level assessment allows you to identify risks at an early stage, and plan mitigation strategies to address them. Be sure to:

✓ **Develop communication plans**

Develop plans to communicate with user groups and key personnel throughout the process to avoid the communication problems that are common to many deployments. User organizations may not be aware of their licensing options, when they have to upgrade, or which resources and information to provide. IT staff are sometimes unaware of desktop macros and other customization files that are required for user groups.

✓ **Coordinate to make sure all stakeholder needs are met**

Make certain all user groups, IT, and stakeholders are “on the same page” and are prepared regarding what is needed, schedules, and expectations.

✓ **Define high level requirements**

Define high level requirements by collecting and analyzing all the information required to define and prioritize needs, address user concerns, and improve user acceptance.

✓ **Define resource requirements**

Define resource requirements to determine how many and what type of resources your deployment requires.

✓ **Assess customization requirements**

Assess your current environment to determine which customizations are required for user acceptance. Be sure to address special security requirements or other needs such as file transfer.

✓ **Assess risks**

Assess the risks of application compatibility and user acceptance.

✓ **Perform a high level analysis of your solution**

Make sure your deployment solution complies with new security mandates, reduces maintenance, meets productivity requirements and still has good user acceptance.

2. Inventory and Analyze User Requirements, Macros, Configuration Files, and Legacy Applications

Conduct a detailed inventory for each user group to determine which user applications and configurations are critical, used, or not used and whether your terminal emulator requires integration with custom applications. Performing an inventory of your current emulation collateral provides the data you need to make sound technical and business decisions about what to carry forward. It also helps identify needs for customization of Reflection and for integration with other applications. Be sure to:

✓ **Define user requirements**

Define user requirements for each user group to determine the priorities for this group and special needs such as file transfer capability or security requirements.

✓ **Inventory User Desktops**

Inventory desktops to assess how many vendor products and files are being used in the existing configuration.

✓ **Identify integration needs**

Identify needs for integration with HLLAPI or other applications that use your terminal emulation software.

3. Assess Which Existing Files To Carry Forward and Test

The success of your deployment depends on careful assessment and planning. After you determine which files to carry forward, you'll need to convert them to Reflection Desktop files and then test these new files.

✓ **Assess which existing files are required**

Assess which macros and configuration files are required for your new solution.

✓ **Analyze inventory data**

Analyze your inventory data to make key technical and business decisions about what to carry forward and deploy.

✓ **Convert Reflection 14 or Extra! files to Reflection Desktop files**

If you plan to carry forward any existing files, open the files in Reflection Desktop and save them (and any related supporting files) as Reflection Desktop files.

If any of your Reflection 14 or Extra! session files have macros that are locked and password-protected, these macros are lost when you open those files in Reflection Desktop. To import the macros, you'll need to unlock the macro projects before you open the files.

NOTE: Reflection Desktop cannot import Extra! or legacy Reflection macros when they are locked because it needs to access the macro projects to convert them.

✓ **Test all files in Reflection Desktop**

Make sure that the macros, configurations, and other files work with Reflection Desktop.

4. Package and Test

Package, test, and deploy to selective user groups and conduct pilots to minimize user disruption. Be sure to test for both technical issues and user acceptance.

Use this checklist to avoid common problems with Reflection deployments.

✓ **Deploy files to trusted locations.**

The Trusted Locations feature provides a way to differentiate safe files from potentially harmful files. When a file is in a trusted location, its files are assumed to be safe. Reflection enforces trusted locations by default, so if you want to save sessions in directories that are not default trusted locations, you'll have to define these directories or disable trusted locations. See [“Add Trusted Locations” on page 77](#)

✓ **Deploy customized files to the correct locations**

Reflection looks for session files and supporting custom files in specific locations. If you deploy one of these files to another location, Reflection will be unable to find it. See [“Customized Files that Must be Deployed to Specific Locations” on page 119](#) in the Reflection Deployment Guide.

✓ **Consider using the Reflection setup.exe**

Setup.exe is the recommended tool for installing and deploying Reflection. This tool uses the primary Reflection MSI file to install Reflection but it also installs prerequisite software (if needed) and has several other features that provide a smoother deployment than installing directly with the primary Reflection MSI file.

It determines whether each workstation has the required .NET Framework and Microsoft Windows Installer version and automatically installs them if necessary. It also automatically uses the correct language for the installation and removes previous versions of Reflection.

If the Visual Basic feature is selected, Setup.exe also installs the Visual Basic core MSI, along with the appropriate VB language MSI.

✓ **If you are deploying with MSI directly, make sure all prerequisites are installed.**

If you use msiexec.exe for your install, prerequisites are not installed automatically. If the required prerequisites are not already on your users' workstations, you need to install them separately. You can find installers for the required prerequisites in the Prerequisites folder in the distribution media, or in your administrative installation. The prerequisites you need to install depend on which programs and features you are installing:

- ◆ All Reflection Workspace features require Microsoft .NET Framework 4.7.1. If you attempt an install using msiexec.exe and this prerequisite is not found, a message displays and the installer stops. To install the .NET Framework, run the executable file in `Prerequisites\DotNet471`.
- ◆ The Visual Basic for Applications feature requires Microsoft VBA 7.1. Use the core and language-specific *.msi packages in the `Prerequisites\VB71` folder.

✓ **Ensure your user systems meet Reflection's Requirements**

To make sure users have the Reflection hardware and software requirements, see [“System Requirements” on page 30](#).

✓ **If your users have macros created with other products, install macro compatibility features**

To run a macro created with another product, the Micro Focus compatibility feature for that type of macro must be installed. This feature is available on the Reflection Setup program Features tab, under 3270/5250 | Compatibility. See [“Select Features, Components, and Languages” on page 132](#).

✓ **Make sure that all VBA projects in referenced session documents and SharedMacros files have unique project names**

If you are referencing session documents or using SharedMacros files to share macros, make sure the projects in those files have unique project names. Each project name in the VBA Project editor must be unique to avoid errors caused by naming conflicts. You can change project names by modifying the project properties in the VBA editor or by creating and saving these files in Reflection Desktop 16.2 or greater.

✓ **Follow guidelines for setting up security certificates**

You can configure certificate authentication for both Secure Shell and SSL/TLS connections.

All SSL/TLS sessions require certificates for host authentication; without the necessary certificate, you cannot make a host connection. Depending on the host configuration, you may also need to install certificates for user authentication.

Secure Shell sessions typically require both host and user authentication. Certificates can be used for either host and/or user authentication, but are not required by default.

For more, see [“Digital Certificates and Reflection Certificate Manager” on page 42](#).

✓ **If you are using Citrix or AppV, Follow best practices for these platforms**

Whenever possible, save sessions that have customized settings for QuickPads, keyboard maps, themes, mouse maps, hotspots, and ribbons as compound session document files (see [“Create and Customize Session Documents” on page 39](#)). This prevents problems caused by sessions not being able to access or find configuration files.

If you specify a user data directory, do not use the Setup user interface to set the directory and do not use a drive letter as part of the directory. Setting the user data directory in this way on a Citrix server causes all users to have the same data directory. Instead, create a transform that sets the WRQ_USERDIR property to the location you want to use for the user data directory (see [“Add/Modify Registry Data” on page 131](#)) and install Reflection with the transform. Be sure to use a Reflection property (such as [PersonalFolder]) to specify the directory instead of a drive letter.

Setting up a Basic Deployment

This video and walkthrough show a way to deploy Reflection out-of-the-box by installing an image of Reflection on a file server and then sending a link to its Setup.exe program to users. When users are ready to install, they click on the link to the Setup.exe program.

It also shows how to use the Installation Customization Tool to create a package of customized files that can be installed in the same way.



Video: Setting up a Basic Deployment

Deploy Reflection out-of-the-box

- 1 Download the Reflection installation files.
- 2 Create a network share on a network file server.
- 3 From the root directory of the installation files, double-click `setup.exe`.

This starts the Reflection Setup program.

- 4 Click **Continue** and accept the license.
- 5 From the **Advanced** tab, click **Create an Administrative install image on a server**.
- 6 Click **Continue**.

The **File Location** tab is selected.

- 7 Browse to the network share you want to use for the administrative install image and then click **Install Now**.

NOTE: Be sure to specify a UNC path for the network share. For example:

```
\\share_name\administrative_install_point
```

- 8 To deploy to users, distribute the network path to the Reflection Setup.exe file through e-mail or another type of notification:

```
\\share_name\administrative_install_point\setup.exe
```

NOTE: All the files required to install Reflection are on the administrative installation point. This location also includes the administrative tools provided to help you customize reflection.

Create and Deploy Customized session files

After you create an administrative installation point, you can use it along with a local installation of Reflection to create and deploy customized session files to your users.

First, you'll need to install Reflection on a workstation so you can create and customize your configuration files.

Install Reflection on a workstation

- 1 On the workstation, navigate to the network share where you have created the administrative installation point, and double-click `setup.exe`.
- 2 From the Reflection Setup program, click **Continue**, and then accept a license.
- 3 To select which features, components, or languages are installed, click the **Feature Selection** tab. The default installation does not include all features and components.

NOTE: If you are working with legacy macros or other legacy files, select the compatibility features that support the products you are upgrading or migrating from.

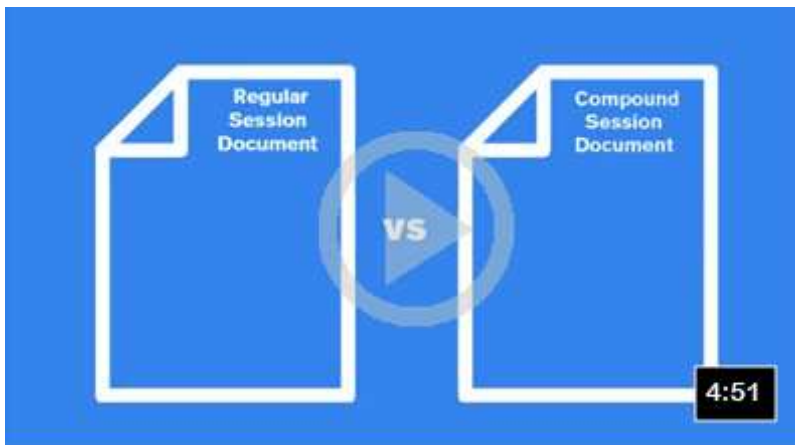
- 4 Click **Install Now**.

Create a custom compound session document file


After you install Reflection on the workstation, you can use it to create custom session document files.

You can simplify your deployment by saving your customized settings in a compound session document file. Compound session document files save all of the settings for QuickPads, keyboard maps, themes, mouse maps, hotspots, and ribbons in a single file.

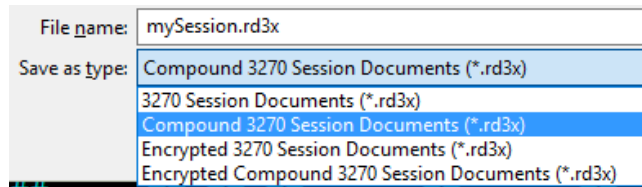
This video and the instructions that follow show how to create a session with a custom keyboard map, save all of the settings in a compound session document file, and then package that file in an .msi file.



Simplifying Deployment with Compound Session Document Files

- 1 On the workstation, open Reflection. If the Create New Document box doesn't open automatically, select **File | New** to open it.
- 2 Select 3270 terminal and then click **Create**.
- 3 In the **Host Name /IP address** box, enter the host name. To use the Reflection onboard demo, enter
`demo:ibm3270.sim`
- 4 After the session opens, click the Document Settings button  on the Quick Access toolbar and then select **Manage Keyboard Map**.

- 5 In the **Manage Keyboard Map** dialog box, select **Create a new keyboard map from an existing keyboard map file**.
- 6 In the **Create a New Keyboard Map file** dialog box, select to base the new keyboard map on the default map.
- 7 In the **Keyboard Mapper**, press the key combination to map (for example, Ctrl+F8) and then click **Select Action**.
- 8 In the **Select Action** dialog box, select an action.
- 9 Enter a new file name for the keyboard map and save it.
- 10 From the **File** menu, choose **Save As** and then save the session as a compound session file type:



NOTE: Saving a session in a compound session document file simplifies your deployment. When you save the session as a compound session document, the custom keyboard map settings are saved in your session file. If you were to save the session as a regular session document, these settings would be saved in a separate file that you would need to deploy along with your session file.

The Installation Customization Tool is designed to be opened from a command line. However, you can create a desktop shortcut that opens this tool. This will save time when you are working with the tool.

Set up a shortcut that opens the Installation Customization Tool

- 1 On your administrative installation point, right-click the setup.exe file, and choose **Create Shortcut**.
- 2 Right-click the shortcut and choose **Properties**.
- 3 In the **Target** box, add the /admin option to the end of the command line. For example:

```
\\myServer\adminInstallPoint\setup.exe /admin
```

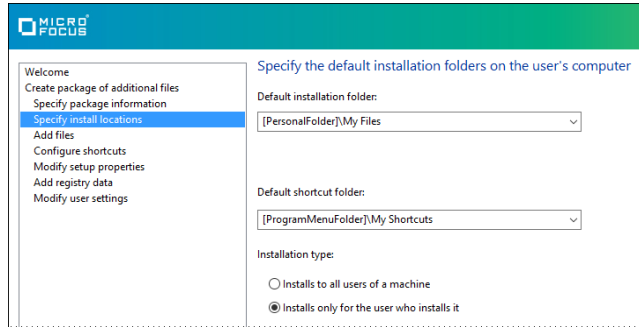
NOTE: The path in the **Target** box must conform to the Uniform Naming Convention (UNC) format and cannot include drive letters. Drive letters can cause problems if you try to use the shortcut on other workstations.

- 4 Rename the shortcut and save it on the desktop of your workstation.

Create a companion installer package

- 1 To start the Installation Customization Tool, double-click the shortcut on your workstation, and in the **Select Customization** dialog box, select **Create a new Companion installer**.

- 2 On the left pane, click **Specify install locations**, and specify the **Installation type**.
 - ◆ **Installs to all users of a machine** installs the file in a common folder so that the settings apply to all users of the machine.
 - ◆ **Installs only for the user who installs it** installs the file in a personal folder so that the settings apply only to a single user.



- 3 On the left pane, click **Add files** and then click in the **Add files to** box and type the rest of the path. If you previously selected **Installs only for the user who installs it**, The final entry should be:

```
[PersonalFolder]\Micro Focus\Reflection\
```

NOTE: By default, this folder is a trusted location. If you use another folder, you will need to add it to the trusted location list and deploy that setting. Reflection does not open files unless they are in trusted locations.

- 4 Click the **Add** button, and browse to the file you added earlier.
- 5 From the navigation pane, click **Specify package information**.
 - ◆ In the **Add/Remove** name box, type the package name to display in the Windows **Programs and Features** control panel.
 - ◆ In the **Organization** name box, enter the name of your department.
- 6 From the **File** menu, save the `.msi` file on the administrative installation point.

Now that you have created the companion MSI file, distribute a link to the file to users. When they run this MSI, the session document file is installed to their workstations in the location specified in the MSI.

2 Install Reflection and Set Up Administrative Tools

You can choose from several different approaches for installing and deploying Reflection, ranging from “out-of-the-box” installations to heavily customized deployments.



Setting Up Your Environment for Deployment

Reflection provides several administrative tools you can use to aid in your installation and deployment. You can use these tools to “lock down” Reflection, package customized configuration files for deployment, and create transforms to modify your installation.

This article	Describes
“Installation” on page 29	Hardware and software requirements, information about upgrading, and the Setup tool tabs.
“Setting up the Reflection Administrative Tools” on page 34	The Reflection administrative tools and how to set up an environment where you can use the tools to customize Reflection, package settings configuration files, and customize how Reflection is installed.

Installation

You can install Reflection from a downloaded distribution, an administrative installation image, or from a CD. If you are an administrator looking for instructions on how to create a customized installer, see the Reflection Deployment Guide.

NOTE: Reflection is typically distributed electronically. If your installation requires a CD, you will need to request it when you place your order.

- ◆ [“System Requirements” on page 30](#)
- ◆ [“Upgrading from Previous Versions” on page 31](#)
- ◆ [“Install Reflection on a Workstation” on page 32](#)
- ◆ [“Features Selection Tab” on page 33](#)
- ◆ [“Set Your User Data Directory” on page 33](#)
- ◆ [“Advanced Tab” on page 33](#)

System Requirements

Requirements for Reflection Desktop can vary, depending on your hardware and other software components present.

Hardware Requirements and Supported Operating Systems

Processor	2 GHz, 64-bit (1.5 GHz or higher multi-core recommended)
System memory (RAM)	2 GB (4 GB recommended)
Operating system and platform	One of the following: Microsoft Windows 10 Pro 64-bit Microsoft Windows 10 Enterprise 64-bit Microsoft Windows 8.1 Pro 64-bit Microsoft Windows Server 2019 with Remote Desktop Services (for multi-user environments) Microsoft Windows Server 2016 with Remote Desktop Services (for multi-user environments) Microsoft Windows Server 2012 R1 or R2 with Remote Desktop Services (for multi-user environments)
Additional software requirements	To use the Office integration features in Reflection, Microsoft Office 2013 or later must be installed.

NOTE: Where listed in the Hardware Requirements and Supported Operating Systems table, 64-bit operating systems are supported by the 32-bit Reflection Desktop product. The 64-bit .NET Framework Runtime is used on Windows 64-bit operating systems, which also supports native 64-bit applications calling the Reflection .NET API.

Prerequisite software — Microsoft .NET Framework

Reflection Desktop requires .NET Framework 4.7.1 or later. On desktop systems, if the required .NET Framework isn't installed, the Reflection Setup program installs version 4.7.1. On Microsoft Windows servers, the .NET Framework must be manually installed if version 4.7.1 or later is not installed.

Notes:

- ◆ **If you install Reflection directly with MSI** and the product features you install do not require .NET 4.7.1, you can add the SKIPDOTNET=1 property to the command line to bypass this test condition.

Microsoft Windows Installer 4.5

The Microsoft Windows Installer (MSI) version 4.5 is distributed with Reflection Desktop.

Microsoft Visual Basic for Applications (VBA)

Reflection Desktop supports Microsoft VBA 7.1. If you select to install this feature in the Reflection Setup program, it is automatically installed. If you install Reflection Desktop directly with MSI or with a deployment tool and you want to install this feature, you must install it directly, using the core VBA 7.1 MSI and an appropriate language specific MSI (these MSIs are in the Prerequisites folder, in the distribution media).

Windows 10 “Windows as a Service” Support Statement

Reflection Desktop is supported on the following editions of Windows 10: Windows 10 Education, Windows 10 Pro, and Windows 10 Enterprise. Reflection Desktop is supported on the in-market supported servicing branches (<https://technet.microsoft.com/en-us/windows/release-info.aspx>) of Windows 10 including: Current Branch, Current Branch for Business, and Long-Term Servicing branches.

Supported Virtualization Platforms

Reflection Desktop is supported with the following virtualization platforms:

- ◆ Citrix XenApp

NOTE: Note: Reflection Desktop TouchUx User Interface Mode is supported with Citrix 6.5 or higher. Reflection is supported only on Citrix versions currently supported by Citrix.

- ◆ Microsoft Application Virtualization (App-V)
- ◆ VMware Workstation/Player
- ◆ VMware ThinApp

Upgrading from Previous Versions

When you install Reflection Desktop, it upgrades the following Micro Focus products:

- ◆ Reflection for IBM 2007, 2008, 2011, and 2014
- ◆ Reflection for UNIX and OpenVMS 2008, 2011, and 2014
- ◆ Reflection for IBM v. 8.0 - 14.x
- ◆ Reflection for UNIX and OpenVMS v. 8.0 - 14.x

- ◆ Reflection for the Multi-Host Enterprise, Standard Edition v. 8.0 - 14.x
- ◆ EXTRA! X-treme v. 8.0 - 9.3
- ◆ Extra! X-treme v. 9.4 or greater
- ◆ myEXTRA! Enterprise v. 7.0
- ◆ myEXTRA! v. 7.x
- ◆ EXTRA! Enterprise 2000

NOTE: Reflection 2007, 2008, 2011, and 2014 are automatically removed when you upgrade. Extra! and the other Reflection products are automatically removed when you upgrade using setup.exe. If you upgrade these products by deploying the .msi file directly, you will get a message telling you to uninstall the older software first.

Supported Reflection Configuration Files

Reflection Desktop represents an upgrade over previous versions of both Reflection and Extra!. Most configuration files used with these products are supported by the new version.

For a complete list of Extra! and legacy Reflection files supported by Reflection Desktop, see [Knowledge Base Article 7021411 \(https://support.microfocus.com/kb/doc.php?id=7021411\)](https://support.microfocus.com/kb/doc.php?id=7021411).

Install Reflection on a Workstation

NOTE: You must log on with administrator privileges to install Reflection. If you do not have the necessary access rights, ask your system administrator to elevate your privileges.

To install on a workstation

- 1 Run the Reflection Setup program.

If you install from

A download site

An administrative installation image

Do This

Click the download link, and then run the download program. Select a location for the installer files, and then click **Next**. This extracts the files to the specified location and starts the Reflection Setup program.

From the administrative installation point, double-click the `setup.exe` file.




- 2 From the Reflection Setup program, click **Continue**, and then accept a license.
- 3 (Optional) To change the default installation folder, click the **File Location** tab and browse to the folder in which you want to install Reflection.
- 4 (Optional) To select which features, components, or languages are installed, click the **Feature Selection** tab.
- 5 Click **Install Now**.

NOTE: Use the **Advanced** tab of the installer only if you want to modify the installer log settings, or if you are an administrator configuring a Reflection deployment. An administrative installation does not actually install the product — instead, it creates an installation image that administrators can use to customize and deploy Reflection to end users.

Features Selection Tab

Use the **Feature Selection** tab to select which features you want to install.

Click the icon to the left of the feature name and select from the options below.

Option	Description
 Feature will be installed on local hard drive	Installs the selected feature. NOTE: Some features listed under a selected feature may not be included when you select to install the higher-level feature. The features that are included are the recommended defaults. If you select the higher level feature a second time, all sub-features will be included.
 Feature will be installed when required	Installs the feature when you first use it (for example, when you click the Start menu shortcut for this feature).
 Feature will be unavailable	Leaves the feature uninstalled.

Set Your User Data Directory

You can specify any location as the user data directory. The default directory is Documents\Micro Focus\Reflection\. The directory you specify will be added as a Trusted Location.

To set the user data directory

- 1 Run the Reflection Setup program as shown in “[Install Reflection on a Workstation](#)” on page 32.
- 2 On the Reflection Setup program, click **File Location**, and then under **Default user data directory**, browse to the directory you want to use.

NOTE: This change only affects those features that you include in your initial installation. If you add a feature later using Programs and Features (or Add/Remove Programs), that application will use the original default directory (Documents\Micro Focus\Reflection\), not the value you specified for **Default user data directory**.

Advanced Tab

Use the **Advanced** tab of the installer only if you want to modify the installer log settings or you are an administrator configuring a Reflection deployment.

Install to this PC	Installs Reflection to your computer.
Create an Administrative install image on a server	<p>NOTE: An administrative install image does not actually install the product — instead, it creates an installation image that administrators can use to deploy Reflection to end users.</p> <p>When you create an administrative install image, an image of Reflection is copied to a network location for later installation to multiple workstations. This network location can be used by deployment tools to access and create packages that are deployed to workstations. Also, end users can perform installations by running <code>setup.exe</code> from this location.</p>
<i>Log file settings</i>	<p>By default, an installation log file is created and then deleted after installation successfully completes. (This configuration avoids accumulation of large log files after successful installations.) To save a log file for all installations, including successful ones, select Create a log file for this installation, and clear Delete log file if install succeeds.</p> <p>The installation log file, which provides details about the installation, is saved in the user's Windows temporary folder (<code>%tmp%</code>) with a generated name that begins with <code>atm</code>. To open this directory, launch the Start menu Run command and enter <code>%tmp%</code>.</p>

Setting up the Reflection Administrative Tools

Reflection uses the Microsoft Windows Installer application installation and configuration service. The Windows Installer gets installation information from a relational database, which is saved and deployed as a Microsoft MSI file. When you deploy an MSI file to a user workstation, the Windows Installer on the workstation accesses the information in the MSI file to perform the installation.

Use the following tools to customize and deploy Reflection. These tools are installed as part of the administrative install image.

About the Tools

Reflection Setup program

The Reflection Setup program (`setup.exe`) is the recommended tool for installing and deploying Reflection. This tool uses the primary Reflection MSI file to install Reflection but it also installs prerequisite software (if needed) and has several other features that provide a smoother deployment than installing directly with the primary Reflection MSI file. When the Setup Program installs Reflection, it determines whether each workstation has the required .NET Framework and Microsoft Windows Installer version and automatically installs them if necessary. It also automatically uses the correct language for the installation and removes previous versions of Reflection. (This is required to install the new version.) If the Visual Basic feature is selected, the Attachmate Installation Program also installs the Visual Basic core MSI, along with the appropriate Visual Basic language MSI.

NOTE: If you install with MSI directly, you will need to install the .NET Framework and Microsoft Windows Installer version directly and remove any previous versions of Reflection. The MSI installer uses English for the installation unless you specify another language on its command line. The Reflection MSI does not install Visual Basic. If you install with MSI directly, you must run the Visual Basic core and language MSIs (in the Prerequisites folder) directly.

The Setup program has a command-line interface that you can run from a command line, a batch file, or a deployment tool. You can set command-line options to preset installation parameters and limit user interaction as Reflection is installing. You can also suppress installation dialog boxes to provide an unattended installation or use command-line options to prepare Reflection for installation by users. In general, any of the MSI command-line options can be used from the Setup program command line.

To see a list of available command-line parameters, enter:

```
setup.exe /?
```

Installation Customization Tool

You can use the Installation Customization Tool (ICT) to customize Reflection or customize the way it is installed. The tool lets you create the following files:

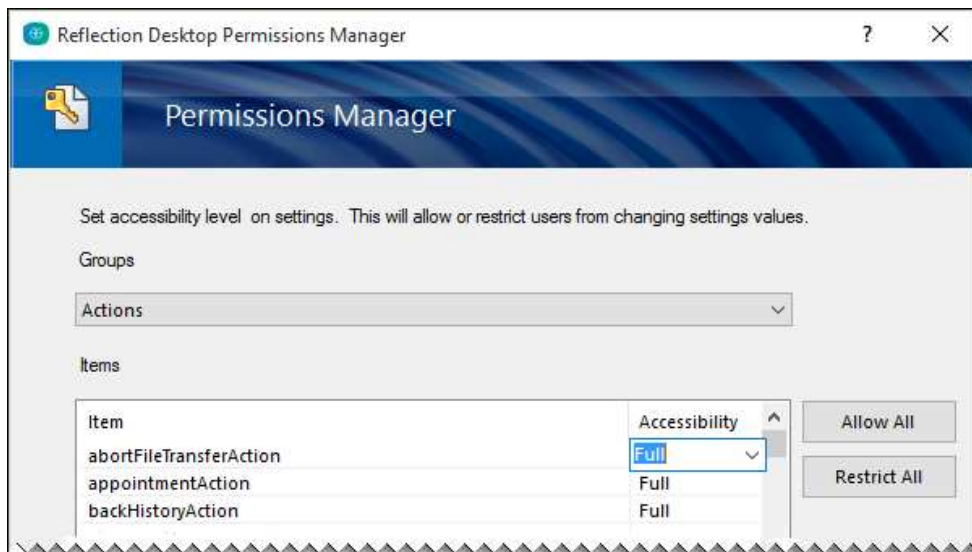
- ♦ **Companion installer package**(MSI). This package contains the customized configuration settings and files that you choose to deploy with the installation (or independently). Companion packages show up as independent entries in the Windows list of installed applications. To customize Reflection, see [“Customize the Reflection Workspace” on page 93](#).
- ♦ **Transform file**(MST). This file modifies the default installation to change how Reflection is installed (For example, remove a command button in the Windows Uninstall or change a program list). When the transform is deployed along with the Attachmate Reflection MSI file, it changes the default installation settings to the settings specified in the MST file.

This tool is accessed by running `setup.exe` from the command line with the admin switch (`setup.exe /admin`). To simplify working with this tool, you can create a desktop shortcut to the Setup program and add the admin switch on the command line as shown in [“Set up a Shortcut to the Installation Customization Tool” on page 38](#).

Permissions Manager

Permissions Manager is used to restrict access to Reflection Desktop settings and features. It creates special configuration (`.access`) files that can be deployed as part of an MSI package. There are different access files for mainframe, AS/400, UNIX/OpenVMS, and application-wide settings.

You can edit `.access` files by running Permissions Manager as shown in [“Specify Access Using Permissions Manager” on page 108](#).



Set up an Environment for Deployment Using Reflection Administrative Tools

If you plan to use the Reflection administrative tools to customize session documents, the Reflection workspace, or the way that Reflection is installed, you'll need to set up an environment that you can use to access these administrative tools.

NOTE: There are two types of Reflection installations. The Administrative install image installs the files required for installation but does not install any values in the registry required to open and run Reflection. You cannot run Reflection from an Administrative installation image. The workstation (or PC) installation enters the values in the registry required to run the product.

Create an Administrative Installation Point

You create an administrative installation point by installing an administrative install image of Reflection on a network share (typically on a file server). An administrative install image is a source image of the application, similar to an image on a CDROM. It includes all the files required to install Reflection as well as the administrative tools used for customization.

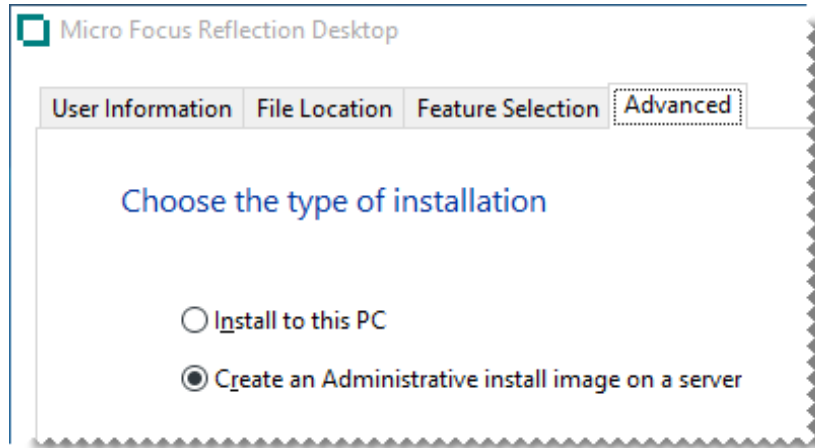
Use this procedure to create an administrative installation point on a networked file server. This places all of the administrative tools and installation files you need to customize and install Reflection to a single location.

NOTE: Micro Focus recommends to create an administrative installation point before you install Reflection on a workstation. This allows you to use the administrative installation point for the workstation installation. If you are setting up the administrative point and the workstation installation on a single workstation for testing purposes, you must perform the administrative installation first.

To create an administrative installation point

- 1 Download the Reflection installation files.

- 2 Create a network share on a network file server.
- 3 From the root directory of the installation files, double-click `setup.exe`.
This starts the Reflection Setup program.
- 4 Click **Continue** and accept the license.
- 5 From the **Advanced** tab, click **Create an Administrative install image on a server**.



- 6 Click **Continue**.
The **File Location** tab is selected.
 - 7 Browse to the network share you want to use for the administrative install image.
-
- CAUTION:** Important! Be sure to specify a UNC path for the network share. For example:
`\\share_name\administrative_install_point`
-
- 8 Click **Install Now**.
 - 9 The next step in preparing a test environment is to [install Reflection on a workstation \(page 24\)](#).

NOTE: Administrative install images are typically created in a file server folder but you can create them in any folder on a local hard drive. This is useful for testing purposes.

If you prefer to install Reflection on your workstation first, you cannot use the installation program graphical interface to create the administrative install image. Instead, you must install it from the command line as follows:

```
path_to_setup_file \setup.exe /install /admin TARGETDIR=  
UNC_path_to_administrative_installation_point
```

Install Reflection on a Workstation

After you create an administrative installation point, you'll need to install Reflection on a workstation so that you can open and run Reflection. If you plan to customize Reflection, you will use this installation to create custom configuration files.

To install Reflection on a workstation

- 1 On the workstation, navigate to the network share where you have created the administrative installation point, and double-click `setup.exe`.
- 2 From the Reflection Setup program, click **Continue**, and then accept a license.
- 3 The following steps are optional:
 - 3a To personalize the installation, click the **User Information** tab and enter the name, organization, and Volume Purchase Agreement (VPA) number, if you have one. (VPA numbers, which are issued by Micro Focus, are used by customer support to expedite service requests.)
 - 3b To change the default installation folder or the default user data directory, click the **File Location** tab and browse to the folder you want to use.
 - 3c To select which features, components, or languages are installed, click the **Feature Selection** tab. The default installation does not include all features and components.
- 4 Click **Install Now**.

Set up a Shortcut to the Installation Customization Tool

The Installation Customization Tool is designed to be opened from a command line. However, you can create a desktop shortcut that opens this tool. This will save time when you are working with the tool.

NOTE: To start the Installation Customization Tool from a command line, change to the administrative installation point and enter:

```
<path_to_setup> \setup.exe /admin
```

To create a shortcut that opens the Installation Customization Tool

- 1 On your administrative installation point, right-click the `setup.exe` file, and choose **Create Shortcut**.
- 2 Right-click the shortcut and choose **Properties**.
- 3 In the **Target** box, add the `/admin` option to the end of the command line. For example:

```
\\ myServer\adminInstallPoint \setup.exe /admin
```

CAUTION: Make sure that the path in the **Target** box conforms to the Uniform Naming Convention (UNC) format and does not include drive letters. Drive letters can cause problems if you try to use the shortcut on other workstations.

- 4 Rename the shortcut and save it on the desktops of your workstation and the server you use for your administrative installation point.
- 5 To start the Installation Customization Tool, double-click the shortcut, and then in the **Select Customization** dialog box, choose which mode you want to open.

3 Create and Customize Sessions

Reflection provides a number of options for customizing and deploying sessions.

This article	Describes
Create and Customize Session Documents	How to create session document files and customize them to configure text input, appearance, macros, and other settings. This article also describes the advantages and of saving sessions in compound session document files that include all of these settings in a single file.
Create SSL/TLS or SSH Session Documents	How to create secure session document files that include settings for SSL/TLS or SSH connections.
Set up Session Templates	How to create a custom session template configured to use specific settings (for example, an SSL/TLS terminal session with a specific TLS level of encryption).
Configure Reflection for PKI Auto Sign-on	How to use the PKI Auto Sign-on Add-On Client product with Reflection to allow the use of a Common Access Card (CAC) or other smart card for authentication.
“Centrally Manage VBA Macros” on page 49	Shows two approaches for centrally managing VBA macros.

Create and Customize Session Documents

Basic connection and terminal settings for Reflection sessions are saved in session document files that you can configure and deploy independently of the product installation.

After you create a session document, you can customize it to configure text input, appearance, macros, and other settings. You can also specify or customize the files referenced in a session document, such as QuickPads, keyboard maps, themes, mouse maps, hotspots, and ribbons.

You can save these session settings as compound session files or as standard session document files (the default).



Compound Session Document files include all of the customized settings for QuickPads, keyboard maps, themes, mouse maps, hotspots, and ribbons. When you Save a session as a compound session document file, you can deploy the session without having to deploy dependant files for these settings.

Session Document files save these settings in separate files. When you save a session in a regular session document file, all custom settings for keyboard maps, themes, mouse maps, hotspots, and other items are saved in separate files that must be deployed with the session file. You'll need to make sure that you deploy these custom files to the correct locations as shown in ["Customized Files that Must be Deployed to Specific Locations"](#) on page 119 .

Compound files are easier to deploy because you don't have to deploy the supporting files. Regular session files offer more flexibility for sharing common configurations. For example, you can reference one custom keyboard map from several regular session files. With compound session documents, you have to recreate the custom settings in each file.

Use these procedures in the Reflection Desktop Guide to create and customize a session document file (that is, change its default settings for text input, appearance, macros, Hotspots and other settings).

To	See
Create and customize session settings that specify whether to automatically connect sessions, encrypt sessions, and configure other host connection options.	Set up Sessions and Connections Set up a SOCKS or HTTP Proxy Server Session
Create custom keyboard or mouse maps	Select and Map the Keyboard and Mouse
Set options for using Microsoft Office Tools such as Recent Typing, Auto Complete, or Screen History.	Set up Productivity Features
Create custom themes and set background and foreground colors, cursor shape, and other display options.	Change the Look and Feel of a Session
Create custom Quick Access Toolbars or context menus, set up Micro Focus Plus or Hotspots features, and import legacy toolbars	Set up Custom Controls for your Program Screens

To	See
Configure Reflection to initiate Reflection actions, such as Reflection macros, and menu and terminal commands when an event is encountered during a host session.	Set up Actions for Reflection Events
Add or remove tabs, groups, buttons, and menus to the Ribbon to create custom Ribbons.	Customize the Ribbon
Perform common tasks such as editing the translation table or setting up customized host files.	Perform Other Common Tasks


Walkthrough: Set up and Customize a Session

The following example shows how to create a custom keyboard map file and set up a session document file to use this keyboard map. This keyboard map assigns the Ctrl+E key combination to the EraseEOF Send Key. The session is saved as a regular session file so you can see the additional work you need to perform to deploy it (as opposed to saving it as a compound session file).

To walk through the process for creating and deploying a compound session file, see [“Setting up a Basic Deployment” on page 23](#).

First, create and save the session.


Create a session

- 1 Open Reflection on your workstation.
- 2 In the Create New Document dialog box, under **Built-in Templates**, choose a type of session (for our example, we will choose 3270 terminal) and click **Create**.
- 3 In the Create New Terminal Document dialog box, in the **Host name/IP address** box, enter the host name (for example, IBM390 or 10.9.1.151).
- 4 From the Reflection Quick Access toolbar, click the Save button  and save the session as `mySession.rd3x`.

This file is saved in the `My Documents\Micro Focus\Reflection` folder.

Then configure the session with a keyboard map.

Customize the session to reference a custom keyboard map

- 1 With a session open in Reflection (for example, `mySession.rd3x`), on the Quick Access toolbar, click the Document Settings button .
- 2 In the Settings dialog box, under **Input Devices**, click **Manage Keyboard Map**.
- 3 In the Manage Keyboard Map dialog box, click **Create a new keyboard map from an existing keyboard map file**.
- 4 In the Create a New Keyboard Map file dialog box, select a keyboard map file to use as a template for your new file (for example, `Default 3270.xkb`) and then select **Use the new file in the current session document** and click **OK**.
- 5 In the Keyboard Mapper dialog box, click in the **Press the key or key combination that you want to map** box. Then hold down the Ctrl key and press the E key to enter Ctrl+E in this box.

- 6 Click the **Select Action** button.
- 7 In the Select Action dialog box, in the **Action** list, select **Send Key**.
- 8 Under **Action parameters**, in the **Key** list, select **Erase EOF** and click **OK**. (The Ctrl+E key combination is displayed as a key combination in the Keyboard Mapper table.)
- 9 Click **OK**. When prompted, save the new keyboard map in the Keyboard Maps folder as `myKeyboardMap.xkb`.
- 10 If you are prompted that this is not a secure location, accept to save it in the default location. (The file is saved in your `Documents\Micro Focus\Reflection\Keyboard Maps` folder.)
- 11 To “lock down” the settings to create a keyboard map, see [“Control Access to “Lock Down” Settings and Controls” on page 103](#).
- 12 To package the session file and the keyboard map, see [“Walkthrough: Create a Package with the Installation Customization Tool” on page 117](#).

Create SSL/TLS or SSH Session Documents

When you create a Reflection session document, configure it to use the security protocols your organization requires.

This connection type	Supports these protocols
3270 terminal or printer	SSL/TLS, SOCKS
5250 terminal or printer	
6530 terminal*	SSL/TLS, Secure Shell
VT terminal or FTP Client	SSL/TLS, Secure Shell, Kerberos, SOCKS, HTTP

- ♦ [“Digital Certificates and Reflection Certificate Manager” on page 42](#)
- ♦ [“Set up SSL/TLS Connections” on page 44](#)
- ♦ [“Set up Secure Shell Connections” on page 44](#)

Digital Certificates and Reflection Certificate Manager

You can configure certificate authentication for both Secure Shell and SSL/TLS connections.

- ♦ All SSL/TLS sessions require certificates for host authentication; without the necessary certificate, you cannot make a host connection. Depending on the host configuration, you may also need to install certificates for user authentication.
- ♦ Secure Shell sessions typically require both host and user authentication. Certificates can be used for either host and/or user authentication, but are not required by default.

Certificate authentication solves some of the problems presented by public key authentication. For example, for host public key authentication, the system administrator must either distribute host keys for every server to each client's known hosts store, or count on client users to confirm the host

identity correctly when they connect to an unknown host. When certificates are used for host authentication, a single CA root certificate can be used to authenticate multiple hosts. In many cases the required certificate is already available in the Windows certificate store.

Digital certificates are maintained on your computer in certificate stores. A certificate store contains the certificates you use to confirm the identity of remote parties, and may also contain personal certificates, which you use to identify yourself to remote parties. Personal certificates are associated with a private key on your computer.

You can use digital certificates located in all of the following stores:

- ◆ **The Windows Certificate Store**

This store can be used by a number of applications, web browsers, and mail clients. Some certificates in this store are included when you install the Windows operating system. Others may be added when you connect to internet sites and establish trust, when you install software, or when you receive an encrypted or digitally signed e-mail. You can also import certificates manually into your Windows store. Manage the certificates in this store using the Windows Certificate Manager.

- ◆ **The Reflection Certificate Manager Store**

This store is used only by Micro Focus applications. To add certificates to this store, you must import them manually. You can import certificates from files and also use certificates on hardware tokens such as smart cards.

- ◆ **Centralized Management Server**

The Centralized Management Server provides an administrator the means to centrally manage, secure, and monitor users' access to host applications. Administrators can deploy centrally managed sessions and certificates to the user. Digital certificates through the centralized management server can only be enabled if the centralized management server is configured to provide users' access to host applications.

Reflection Certificate Manager

Use the Reflection Certificate Manager to manage certificates for use exclusively by Reflection. You can deploy certificates and settings per-user or for all users of the system.

- ◆ User-specific location: [\[PersonalFolder\]](#)\Micro Focus\Reflection\.pki
- ◆ Global location: [\[CommonAppDataFolder\]](#)\Micro Focus\Reflection\.pki

NOTE: These settings are not included in compound documents.

The procedures for opening the Certificate Manager depend on your product and session type.

To open the Reflection Certificate manager from the Secure Shell Settings dialog box

- 1 Open the [Reflection Secure Shell Settings](#) dialog box.
- 2 On the **PKI** tab, click [Reflection Certificate Manager](#).

To open the Reflection Certificate manager from the Security Properties dialog box

- 1 Open the [Security Properties](#) dialog box.
- 2 On the **SSL/TLS** tab, select [Use SSL/TLS Security](#).

- 3 Click **Configure PKI**.
- 4 Click **Reflection Certificate Manager**.

Set up SSL/TLS Connections

SSL/TLS connections use digital certificates for authentication. Depending on how your certificate was issued and the way your host is configured, you may need to install a host and/or personal certificate before you can connect using SSL/TLS.

- ◆ In 3270, 5250, and VT sessions, SSL/TLS connection settings are saved to the session document.
- ◆ In the FTP Client, SSL/TLS connection settings are saved to the FTP Client settings file (*.rftw).

To configure SSL/TLS in 3270, 5250, or VT terminal sessions

- 1 Open the **Create New Document** dialog box, select a session template and click **Create**.
- 2 Select **Configure additional settings**, and then click **OK**.
- 3 Do one of the following:
 - ◆ If you are setting up a 3270 and 5250 terminal session, under **Host Connection**, click **Set Up Connection Security**. Then, in the Configure Advanced Connection Settings dialog box, click **Security Settings**.
 - ◆ If you are setting up a VT terminal session, click **Configure Connection Settings**, confirm Network Connection Type is set to **Telnet**, and click the Back arrow button. Then, under **Host Connection**, click **Set Up Connection Security**.
- 4 From the **Security Properties** dialog box, select the **SSL/TLS** tab, and select **Use SSL/TLS security**.
- 5 Click **Configure PKI** to configure certificate settings.
- 6 To lock down these settings, see [“Control Access to “Lock Down” Settings and Controls” on page 103](#).

To configure SSL/TLS in FTP Client Sessions

- 1 Start the FTP Client.
- 2 In the **Connect to Site** dialog box, select a site and click **Security**.
- 3 Click the **SSL/TLS** tab and select **Use SSL/TLS security**.
- 4 Click **Configure PKI** to configure certificate settings.

Set up Secure Shell Connections

Secure Shell connections are available for VT terminal sessions and to configure SFTP transfers using the FTP Client.

By default, Secure Shell connections use public key authentication for the host and username/password authentication for the user. If you configure non-default settings, they are saved for each host (or ssh configuration scheme) to the ssh configuration file. This file is used for all connections (VT sessions and the FTP Client). You can deploy these settings per-user or for all users of the system. These settings are not included in compound documents.

- ◆ User-specific configuration: [PersonalFolder]\Micro Focus\Reflection\.ssh\config
- ◆ Global configuration: [CommonAppDataFolder]\Micro Focus\Reflection\ssh_config

To configure a secure terminal session using Secure Shell (SSH)

- 1 Open the **Create New Document** dialog box, select the **VT Terminal** template and click **Create**
- 2 In the **Create New** dialog box, under **Connection**, select **Secure Shell** and click **OK**.
- 3 Click **OK**.
- 4 To lock down these settings, see [“Control Access to “Lock Down” Settings and Controls” on page 103](#).

To configure non-default Secure Shell settings

- 1 Open a session that you have configured to use Secure Shell. Disconnect if you are connected.
- 2 Open the **Document Settings** dialog box.
- 3 Under **Host Connection**, click **Set up Connection Security**.
- 4 In the **Reflection Secure Shell Settings** dialog box, configure any non-default settings and then click **OK**.

When you click **OK**, changes to the default settings are saved in the Secure Shell `config` file in [\[PersonalFolder\] \Micro Focus\Reflection\.ssh](#)

- 5 To lock down these settings, see [“Control Access to “Lock Down” Settings and Controls” on page 103](#).

To configure username and password prompts to appear in the terminal window

- 1 Open a session that you have configured to use Secure Shell. Disconnect if you are connected.
- 2 Under **Host Connection**, click **Configure Connection Settings**.
- 3 Under **Connection Options**, select **Handle SSH user authentication in terminal window**.
- 4 To lock down these settings, see [“Control Access to “Lock Down” Settings and Controls” on page 103](#).

Known Hosts

Host authentication (performed with public key authentication) enables the Secure Shell client to reliably confirm the identity of the Secure Shell server. If the host public key is not installed on the client, the host fingerprint is displayed and users are prompted to contact the system administrator to verify the fingerprint. This confirmation prevents risk of a "man-in-the-middle" attack, in which another server poses as the host. If you select **Always** in response to this prompt, the host key is saved in a file called `known_hosts`, which is created in [\[PersonalFolder\] \Micro Focus\Reflection\.ssh](#). After the host key is added, Micro Focus Reflection Desktop can authenticate the server without requiring user confirmation, and the unknown host prompt does not appear again.

To prevent end-users from seeing the unknown host message you can deploy a known hosts file per-user or for all users of the system. These settings are not included in compound documents

- ♦ User-specific file: [\[PersonalFolder\] \Micro Focus\Reflection\.ssh\known_hosts](#)
- ♦ Global file: [\[CommonAppDataFolder\] \Micro Focus\Reflection\.ssh_known_hosts](#)

Set up Session Templates

Templates provide an untitled copy of the original, giving you a quick and easy way to create pre-configured documents, while ensuring that your original file remains unchanged.

You can also use templates to control the types of sessions that users can create. For example, you can create templates that have pre-configured SSL/TLS settings. These templates appear in the **Create New Document** dialog box, under **User-defined**.

Create a Session Template


After you configure a session document, you can share and reuse your settings by saving the document as a template.

To create a session template

- 1 Open the session document that you've configured.
- 2 Save the session as a template.

The steps depend on your user interface mode.

Ribbon

On the **File** menu or the Reflection button  (if using the Office 2007 Look and Feel), choose **Save As** and then **Save Template**.

Classic MDI

From the **File** menu, select **Save As Template**.

- 3 Name the template file with an `.rsft` extension.
- 4 To lock down settings you have configured in the template, see [“Control Access to “Lock Down” Settings and Controls” on page 103](#).

To make changes to the template, you must replace the template file — save the file that contains your changes using the same filename and extension as the template.

NOTE: To deploy templates, install the `.rsft` files to the `templates` folder:

`[AppDataFolder]\Micro Focus\Reflection\Desktop\v17.0\templates\`

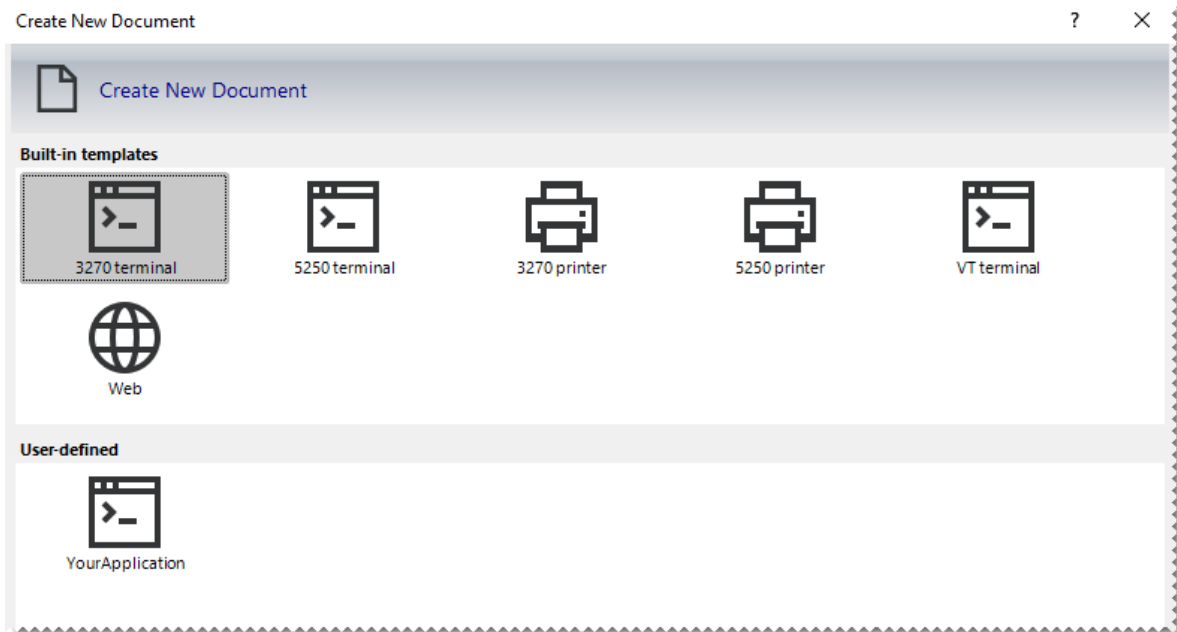
Templates saved to this location appear in the **Create New Document** dialog box, under **User-defined**.

Control the Types of Sessions Users can Create

You can use templates to control the types of sessions users can create. First you'll need to create custom templates and lock down their settings as shown in [“Create a Session Template” on page 46](#).

Then you'll need to configure the workspace to hide the built-in templates that are distributed with Reflection, as shown below.

When you open Reflection after creating a custom template, the built-in templates are displayed along with your custom template.



You'll need to configure the workspace to hide these templates.

To hide built-in templates

- 1 From the **File** menu, choose **Reflection Workspace Settings**.
- 2 On the Reflection Workspace Settings window, click **Workspace Defaults**.
- 3 In the Configure Workspace defaults dialog box, select **Hide built-in templates for new documents**.
- 4 Test the configuration by creating a new document. When the Create New Document dialog box opens, the built-in templates are not displayed.



- 5 Deploy this workspace setting using one of the methods described in [Chapter 7, "Customize the Reflection Workspace,"](#) on page 93.

Configure Reflection for PKI Auto Sign-on

You can configure a Reflection Desktop session to use the PKI Auto Sign-on Add-On Client product, which allows the use of a Common Access Card (CAC) or other smart card for authentication.

To use PKI Auto Sign-on, the PKI Auto Sign-on host module must be installed on your host server. This module can be used to verify that a client is in control of a CAC or other smart card, and to extract the Distinguished Name (DN) from the certificate used for authentication. The DN, or some substring contained in the DN, can then be used to provide service to the authorized user. PKI Auto Sign-on is designed to provide a validated identity even via a shared host login, that is, the identity comes from the smart card itself, not from the host user ID.

When a session is configured to use PKI Auto Sign-on:

- ◆ System administrators can set up an OpenVMS or UNIX session to use a shared log-on that provides the host application with a strongly validated identity directly from a CAC.
- ◆ Host programmers can get the strongly validated DN of a user in control of a CAC. The programmers can then extract information from the DN and use it as an identifier to authorize access (for example, to the CAC-bearer's health records).

Prerequisites

- ◆ The Reflection PKI Auto Sign-on host module must be installed on the host server.
- ◆ You can use PKI Auto Sign-on with Reflection Desktop or Reflection 2014 R1 SP1 VT terminals using the SSH protocol. All of the client-side functionality required for PKI Auto Sign-on is included only in these product versions.

To create an SSH-enabled Reflection session that uses PKI Auto Sign-on

- 1 Create a new VT session document.
- 2 Click **Configure additional settings** and then click OK.
- 3 In the Settings dialog box, under **Host Connection**, select **Set up Connection Security**.
- 4 On the Reflection Secure Shell Settings dialog box General tab, under **User authentication**, deselect **Public Key**.
- 5 On the PKI tab, click **Reflection Certificate Manager**.
- 6 On the Reflection Certificate Manager dialog box PKCS #11 tab, click **Add**.
- 7 In the PKCS #11 Provider dialog box, browse to the Provider DLL required to access your CAC.
- 8 In the `.ssh/config` file for this session document, add the appropriate PKIC prompt string configured on the server. The following example shows an entry for a prompt "Starting PKI Validation..."

```
PKICPrompt "Starting PKI Validation..."
```

When you are done, the file should look like this:

```
Host myHostName
```



```
RSAAuthentication no
PubkeyAuthentication no
connectionReuse no
PKICPrompt "Starting PKI Validation..."
#EndHost
```

9 To lock down settings, see [“Control Access to “Lock Down” Settings and Controls”](#) on page 103.

Centrally Manage VBA Macros

With Reflection Desktop 16.2 (and greater), you can use two different approaches to centrally control and manage Reflection Desktop VBA macros.

- ◆ You can save VBA macros in the `SharedMacros.rd3x`, `SharedMacros.rd5x`, or `SharedMacros.rdox` files and deploy them to the user data folder. When a session opens, it automatically loads the macros from the `sharedMacros` file that corresponds to its file type. (See [“Walkthrough: Set up and deploy a SharedMacros file”](#) on page 54)
- ◆ You can save VBA macros in session document files that reside on a network share or other location and then configure local session documents to reference these shared documents. When the local session documents open, they load the macros from the documents that they reference. (See [“Walkthrough: Set up a Session to Reference Macros in a Session Document File”](#) on page 51)



Sharing Macros in Session Documents

Reflection Desktop supports using `SharedMacros` files and referencing session documents that contain VBA macros for IBM (`.rd5x` and `.rd3x`) and VT (`.rdox`) session documents. You'll need to create at least one session document file for each type of terminal you are using. Session document files can only reference macros in the same type of session document. For example, IBM3270 sessions can only reference session documents with an `.rd3x` extension.

NOTE: Reflection does not support nested references to session documents. If a session references a session document that in turn references other session documents, the references in the second session document are ignored.

If you are referencing session documents or using SharedMacros files to share macros, make sure the projects in those files have unique project names. Each project name in the VBA Project editor must be unique to avoid errors caused by naming conflicts. You can change project names by modifying the project properties in the VBA editor or by creating and saving these files in Reflection Desktop 16.2 or greater.

	Using SharedMacros files	Referencing macros in shared session document files
Use this approach when	You want to share macros with every session file of a specific terminal type (for example, all IBM3270 session files).	<ul style="list-style-type: none"> You want to share macros only with specific session document files. You want to centrally manage macros by saving them in session document files that reside on a network share or other location. Saving the session file on a share allows you to change the macros on the user's machines without deploying any files. <p>NOTE: When a local session with a reference to a session document file opens, it downloads the latest versions of the macros from the document that it references.</p>
You'll need to	<ul style="list-style-type: none"> Develop the macros you want to share and save them in one of the following session document files: SharedMacros.rd3x SharedMacros.rd5x SHaredMacros.rdox Deploy the SharedMacros file to the user data folder. 	<ul style="list-style-type: none"> Develop the macros you want to reference and save them in a session document file on a network share or another location. Set up custom session documents that reference the documents that contain the macros and deploy them to users. If the location you save the session documents on is not a trusted location, you'll need to add it to the list of trusted locations and deploy that setting to users.
Considerations	If you want to modify any of the macros in the SharedMacros files, you'll need to deploy these files to users.	<p>To prevent users from editing the macro source in session documents referenced by other sessions, consider locking and password-protecting the macro projects or using some other method to control access to these files.</p> <p>NOTE: When you save a macro that was downloaded from a referenced session in a local session, the modified macro source is automatically uploaded to the referenced session document.</p>

NOTE: Prior to Reflection Desktop version 16-2, Visual Basic code was shared by exporting and distributing code module (`.bas`), class (`.cls`), and form (`.frm`) files that could then be imported into user session document files. This method for sharing macros is still supported. However, for new implementations, it is recommended to share macros either by including them in the `SharedMacros` files or by referencing session documents that contain the shared macros.

Walkthrough: Set up a Session to Reference Macros in a Session Document File

This walkthrough shows how to set up a local session document file to reference macros in a session document file on a network share and how to add the share as a trusted location. It also shows how to create an MSI file that can be used to deploy the local sessions and their related settings to users.

NOTE: You can set up session document files to reference session document files that contain shared macros on any location that is a trusted location. These locations are not limited to network shares. For example, you could choose to reference macros in a session document file that resides on a folder on user workstations that is available to all users.

Set up and Test References to Macros


To set up and test your references to macros, first you'll need to develop the macros you want to reference and save them in a session document on the network share. Then you'll need to specify the network share as a trusted location. And then set up a local session document to reference this session document and configure the local session with buttons or menu items that run the macros.

Make sure the VBA projects in the session document files you are referencing have unique project names. Each project name in the VBA Project editor must be unique to avoid errors caused by naming conflicts. You can change project names by modifying the project properties in the VBA editor or by creating and saving these session document files in Reflection Desktop 16.2 or greater.

To develop and test a reference

- 1 Set up a network share to save the macros on.
- 2 Create a session document and develop the macros you want to reference. Make sure the macro projects in this document have unique names to prevent naming conflicts with VBA projects in any sessions that reference this session. Then save that session document file on the network share.

NOTE: You'll need to create at least one session document file for each type of terminal you are using. Session document files can only reference macros in the same type of session document. For example, IBM3270 sessions can only reference session documents with an `.rd3x` extension.

- 3 Open the Reflection Workspace Settings window as follows:
 - ♦ If you're using the Ribbon interface, choose **Reflection Workspace Settings** from the **File** menu or the Reflection button  (if using the Office 2007 Look and Feel).

- ♦ For an IBM session in the Classic interface, choose **Settings** from the **Options** menu to open the Settings window and then choose **Reflection Workspace Settings**.
 - ♦ For a VT Session in the Classic interface, choose **View Settings** on the **Setup** menu to open the Settings window and then choose **Reflection Workspace Settings**.
- 4 In the Specify Trusted Locations dialog box, make sure the location you are saving the shared session documents on is on the list of trusted locations and add it if necessary. You must use a UNC path to specify this location. If you are using a network share, you'll also need to select **Allow trusted location on my network**.
 - 5 Open each session you want to use these macros in and configure them as follows:
 - 5a In the Settings window, select **Set Up VBA References**.
 - 5b Under **Referenced Session Documents**, enter the fully qualified paths to the session documents that have the macros you want to reference.

NOTE: If you are saving the referenced session documents on a network share, you'll need to specify the location with a UNC path. Mapped letter drives are not supported.

 - 5c If you want to set up a toolbar button or menu item to run a referenced macro, open the UI Designer on the Appearance tab and set the action for the button (or menu item) to run the macro.
-
- NOTE:** For more about setting up a button or menu item, see "Customize the Ribbon" or "Customize Classic Menus and Toolbars" in the Reflection Help.
-
- 6 Run and test the macros in your local session documents and modify them as needed. When you save your local sessions, any changes you made to the macros are automatically uploaded and saved in the document on the share.
 - 7 To prevent users from modifying the macro source in the session document on the share, consider locking and password protecting the macros or using some other method to control access.
 - 8 To "lock down" the settings for trusted locations or VBA references, see "[Control Access to "Lock Down" Settings and Controls](#)" on page 103.

Deploy Sessions that Reference Macros and Related Settings

After you have developed and tested your macros, you'll need to deploy your local session files and the related settings files to users. This typically includes creating a companion MSI file or using another method to deploy the following files:

- ♦ *The session files* (the `.rd5x`, `.rdox`, or `.rd3x` files) that are configured with references to the files that contain the macros. (For this walkthrough, we'll deploy the `mySession.rd3xfile`.)
- ♦ *A custom ribbon* (`.xuml`) file that includes any buttons or menu items you added to run the macros.
- ♦ *The `Application.Settings` file* that includes the trusted locations you added.

To create an MSI file for session documents that reference shared macros

- 1 Make sure you know where the session files and their related settings you want to add are located. For this example, we are packaging the files in the following locations:

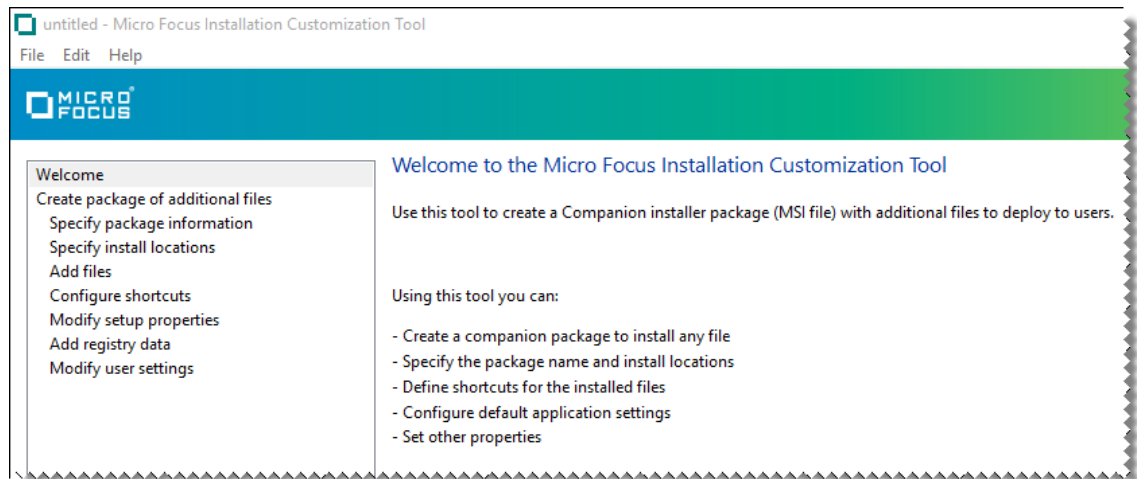
Add this file	In this directory
mySession.rd3x	%USERPROFILE%\Documents\Micro Focus\Reflection
myRibbon.xuml	%USERPROFILE%\Documents\Micro Focus\Reflection\CustomUI

- On your administrative workstation, open the Installation Customization Tool from a desktop shortcut or from a command line as follows:

```
path_to_setup\setup.exe /admin
```

- In the Select Customization dialog box, choose **Create a new Companion installer**.

The Installation Customization Tool opens in the mode used to create companion installer packages.



- From the Navigation pane, click **Specify package information**.
 - In the **Add/Remove name** box, enter the name for the package that you want to appear in the Windows **Uninstall or change a program** list.
 - In the **Organization name** box, enter the name of your department.
- From the Navigation pane, click **Specify install locations**. Then, under **Installation type**, select **Installs only for the user who installs it**.
- In the Navigation pane, select **Add Files**. Then, in the **Add files to** list, select `[PersonalFolder]` and type in:

```
\Micro Focus\Reflection\CustomUI
```

When you are done, the list entry is:

```
[PersonalFolder]\Micro Focus\Reflection\CustomUI
```
- Click **Add**. Then browse to the configuration file (`myRibbon.xuml`) and click **Open**.
- Repeat steps 6 and 7 to add the `mySession.rd3x` file to the `[PersonalFolder]\Micro Focus\Reflection` location.

NOTE: Be sure to specify the correct location for each file you add. Reflection looks for configuration files in specific locations. To find the correct file locations for any type of file see [“Customized Files that Must be Deployed to Specific Locations”](#) on page 119.

When you are finished adding files, the panel should look similar to the following:

File name	Shortcut	Location
myRibbon.xuml	No	[PersonalFolder]\Micro Focus\Reflection\CustomUI
mysession.rd3x	No	[PersonalFolder]\Micro Focus\Reflection\

Add files to: Include shortcut

[PersonalFolder]\Micro Focus\Reflection\ Add... Update Remove

- 9 On the Navigation pane, click Modify user settings. Then, in the **Make changes to user settings** panel, under **Application – Settings**, select **Reflection Desktop -Workspace Settings** and then click **Define**.
- 10 In the Reflection Workspace Settings dialog box, under **Workspace Settings**, select **Specify Trusted Locations**.
- 11 Add the path to the shared macros to the list of trusted locations and select **Allow trusted locations on my network**.

NOTE: Be sure to specify the path in the same way (either drive letter or UNC path) that you used in your session files.

- 12 Click **OK** to add the settings to the package and return to the main Installation Customization Tool window. The `Application.Settings` file is automatically added to the package in the appropriate location.
- 13 To “lock down” these settings, see [“Control Access to “Lock Down” Settings and Controls”](#) on page 103.
- 14 From the **File** menu, save the package as an `.msi` file on the administrative installation point.

Walkthrough: Set up and deploy a SharedMacros file

This walkthrough shows how to set up a `SharedMacros.rd3x` file and create an MSI file that can be used to deploy that file to users.

NOTE: Reflection Desktop supports shared VBA macros for IBM (.rd5x and .rd3x) and VT (.rdox) session documents. The SharedMacros settings files have the same extensions as their corresponding session document files:

IBM 5250: SharedMacros.rd5x

IBM 3250: SharedMacros.rd3x

VT: SharedMacros.rdox

Set up and Test a SharedMacros file

First, you'll need to set up and test your SharedMacros file. After you test the macros in this file, consider locking them and password protecting them so that users cannot change them.

Make sure the VBA projects in the SharedMacros files have unique project names. Each project name in the VBA Project editor must be unique to avoid errors caused by naming conflicts. You can change project names by modifying the project properties in the VBA editor or by creating and saving the SharedMacros files in Reflection Desktop 16.2 or greater.

To develop and test a SharedMacros file

- 1 Develop the macros you want to share.
- 2 Make sure the macro projects have unique names and save them in a session document named SharedMacros.rd3x in the user data directory.

NOTE: The default user data directory is %USERPROFILE%\Documents\Microsoft Focus\Reflection

- 3 If you want to set up a toolbar button or menu item to run a shared macro, open the UI Designer on the Appearance tab and set the action for the button (or menu item) to run the macro.

NOTE: For more about setting up a button or menu item, see "Customize the Ribbon" or "Customize Classic Menus and Toolbars" in the Reflection Help.

- 4 Run and test the macros in your session documents and modify the macros as needed.
- 5 To prevent users from modifying the shared macros, consider locking and password protecting them.
- 6 To "lock down" UI Designer settings, see ["Control Access to "Lock Down" Settings and Controls" on page 103](#).

Deploy the SharedMacros.rd3x file

After you have developed and tested your macros, you'll need to deploy the SharedMacros.rd3x file to users, along with related settings, such as toolbar or ribbon buttons. This includes:

- ♦ *The SharedMacros files* (the .rd5x, .rdox, or .rd3x files) that contain the shared macros.
- ♦ *A custom ribbon (.xuml) file* that includes any buttons or menu items you added to run the macros.

To create an MSI file for a SharedMacros file

- 1 Verify the location of the user data folder and make sure you know where the session files and their related settings you want to add are located. For this example, we are packaging the files in the following locations:

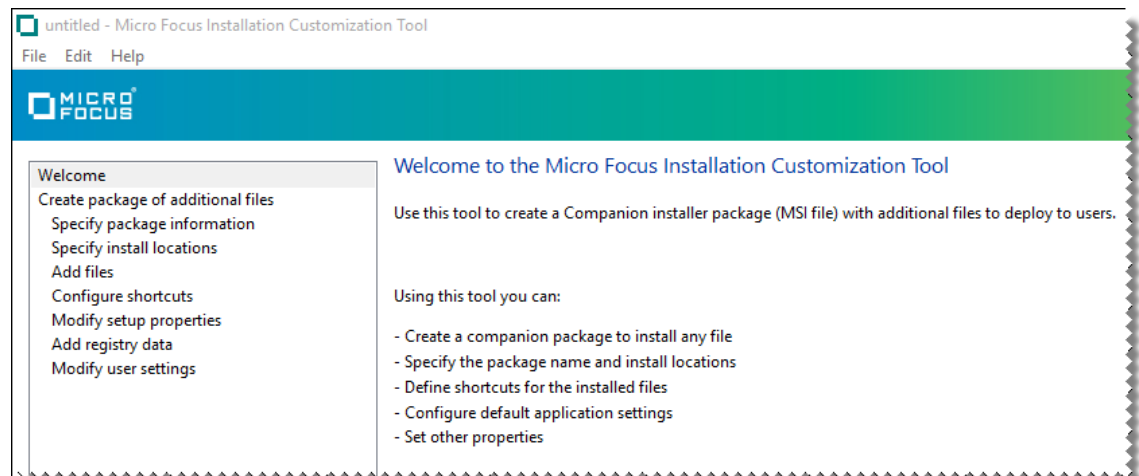
Add this file	In this directory
SharedMacros.rd3x	%USERPROFILE%\Documents\Micro Focus\Reflection\
myRibbon.xml	%USERPROFILE%\Documents\Micro Focus\Reflection\CustomUI

- 2 On your administrative workstation, open the Installation Customization Tool from a desktop shortcut or from a command line as follows:

```
path_to_setup\setup.exe /admin
```

- 3 In the Select Customization dialog box, choose **Create a new Companion installer**.

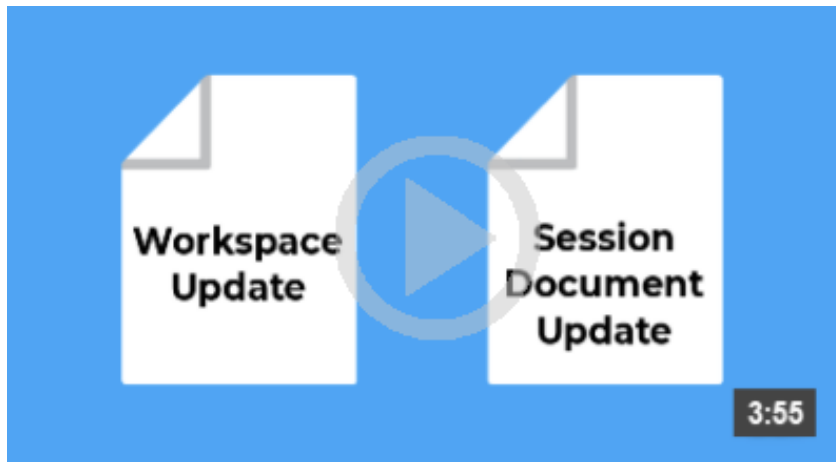
The Installation Customization Tool opens in the mode used to create companion installer packages.



- 4 From the Navigation pane, click **Specify package information**.
 - ◆ In the **Add/Remove name** box, enter the name for the package that you want to appear in the Windows **Uninstall or change a program** list.
 - ◆ In the **Organization name** box, enter the name of your department.
- 5 From the Navigation pane, click **Specify install locations**. Then, under Installation type, select **Installs only for the user who installs it**.
- 6 In the Navigation pane, select **Add Files**. Then, in the **Add files to** list, select [PersonalFolder] and type in:
`\Micro Focus\Reflection\CustomUI`
When you are done, the list entry is:
`[PersonalFolder]\Micro Focus\Reflection\CustomUI`
- 7 Click **Add**. Then browse to the configuration file (`myRibbon.xml`) and click **Open**.
- 8 In the Add file to list, select [PersonalFolder] and type in:

4 Change Specific Session Settings

You can use the SettingsUpdate feature to change one or more settings in session document files without changing any other settings users may have configured independently. For example, you could create and deploy a SettingsUpdate file that specifies a security setting and port. When users open a session document file, the update is applied and the security setting and port are updated without affecting any other settings.



Deploying Specific Settings

Unlike session document files, update files do not contain all the information required for a session. When you open a session document file, Reflection loads all the settings that specify every detail for a session. By contrast, update files contain only information about specific settings. You can use them to change a number of settings for several types of session documents (see [“Which Types of Session Documents and Settings Can be Updated?”](#) on page 62).

There are two types of update files used to update session documents: local update files that reside in the user data directory, and referenced update files that typically reside on a server. Depending on your requirements, you may choose to use one of these files or an approach that uses a combination of both types of files.

Local update files

Local update files apply settings to all existing and new user session document files that have the same terminal type as the update file. These files are deployed to the user’s data directory and have a special naming convention; they must be named “update. `xxxxu`” (for example `update.rd3xu`). When a session with the same type of terminal as the update file is opened (for example, the session is a 3270 session and the update is a 3270 update file), the update file applies its settings to that session.

You could use local update files when you need to apply a setting to all existing user session documents as well as new session documents that the user creates and you don’t want to write over settings (such as screen colors) that the user has customized in these files. To set up local update files for sessions, see [“Apply an Update to all Session Document Files”](#) on page 62.

Referenced update files

Referenced update files apply settings only to session document files that reference them. These files have no special naming requirements and have an .xxxxu extension (for example `regular-updates.rd3xu`). They are typically placed on a server but can also be placed in a local folder, such as a public documents folder on a user machine.

You could use referenced update files to modify session settings when you know you will need to periodically update settings in session documents that you have deployed and you don't want to write over settings (such as screen colors) that users have customized in these files. To set up referenced update files, see [“Apply Automatic Updates to Select Session Document Files” on page 65](#).

This topic	Shows how to...
“Best Practices, and Approaches for Applying SettingsUpdate Files to Sessions” on page 60	Decide which approach to use to apply the updates in your environment and follow best practices for creating update files. This topic also shows which settings and types of session documents can be modified by update files.
“Apply an Update to all Session Document Files” on page 62	Use a local update file to apply settings to all session documents with the same terminal type as the update file (for example, <code>update.rd3xu</code> applies updates to <code>myIBMSession.rd3x</code>). These updates are applied both to session documents you deploy and to sessions created by users.
“Apply Automatic Updates to Select Session Document Files” on page 65	Use referenced update files to apply settings only to specific sessions that you deploy and centrally manage the updates with an referenced update file on a server.
“Set up Automatic Updates for All Session Document Files” on page 70	Use a combination of local and referenced update files to apply settings to all user session documents and centrally manage the updates with an update file on a server.
“Create or Modify SettingsUpdate Files” on page 74	Create, modify and deploy update files that you have configured to modify sessions.

Best Practices, and Approaches for Applying SettingsUpdate Files to Sessions

This topic includes:

- ♦ [“Approaches for Applying SettingsUpdate Files to Sessions” on page 61](#)
- ♦ [“Best Practices for Creating and Applying SettingsUpdate Files to Sessions” on page 61](#)
- ♦ [“Which Types of Session Documents and Settings Can be Updated?” on page 62](#)

Approaches for Applying SettingsUpdate Files to Sessions

You can set up automatic updates for a specific type of session document file (for example, an IBM 3270 .rd3x session document file) in several ways:

Option	Advantages	Considerations
Use a local update file to update all Session Documents —You can deploy updates to all session documents by creating a local update file and deploying it to the user data directory of each user.	You can update all existing and new sessions.	You need to deploy the local update file every time you need to modify session settings.
Use a referenced update file to apply updates for select session document Files —You can configure and deploy sessions that reference an update file on a server.	You can easily change and apply updates by changing the update file on the server. You have more control over which files are updated.	These updates apply only to session documents that you have configured to reference the update file on the server. They do not apply to new session documents that users create or other existing session documents.
Use a combination of both approaches to apply updates for all session document files —Deploy a local update file that adds a reference to another update file on a server.	You can set up all of your user's existing and new session documents to receive updates. You can easily change and apply updates by changing the update file on the server.	The first time users open their session, the local update files sets a reference to the update file on the server. The next time the session is opened, the settings in the server's update file are applied. (After the session document is configured to use a referenced update file, it doesn't use the local update file.)

Best Practices for Creating and Applying SettingsUpdate Files to Sessions

- ◆ Make sure that the update file changes only the settings you intend to change. A good approach is to create an update file that perfectly matches the session document defaults before you change the settings you intend to update.
- ◆ Do not enter any connection settings or other default settings in update files unless you want to update those settings.
- ◆ Do not try to create an update file by renaming a session file. You can create an update file for sessions only by saving the file from the Reflection Desktop interface.

Which Types of Session Documents and Settings Can be Updated?

You can create and apply updates to the following types of session documents: IBM3270 (.rd3x), IBM5250 (.rd5x), Open Systems (.rdox), and 6530 NonStop (.rd6x) terminals.

In addition to updating standard session document files, you can update encrypted session documents, compound session documents, and encrypted compound session document files.

SettingsUpdate files do not apply to sessions that are not opened in the workspace, such as HP, FTP, and printer sessions.

Settings you can change with SettingsUpdate files

You can change these settings	You cannot change...
<ul style="list-style-type: none">◆ connection settings◆ terminal settings◆ clipboard settings◆ security settings◆ productivity settings◆ mainframe file transfer settings◆ some printer settings and settings that reference printers◆ update settings and references to update files◆ references to VBA macros◆ hotspot display settings	<ul style="list-style-type: none">◆ keyboard maps◆ mouse maps◆ ribbons◆ themes◆ hotspot maps◆ quickpads

NOTE: You cannot modify Macros with SettingsUpdate files.

Apply an Update to all Session Document Files

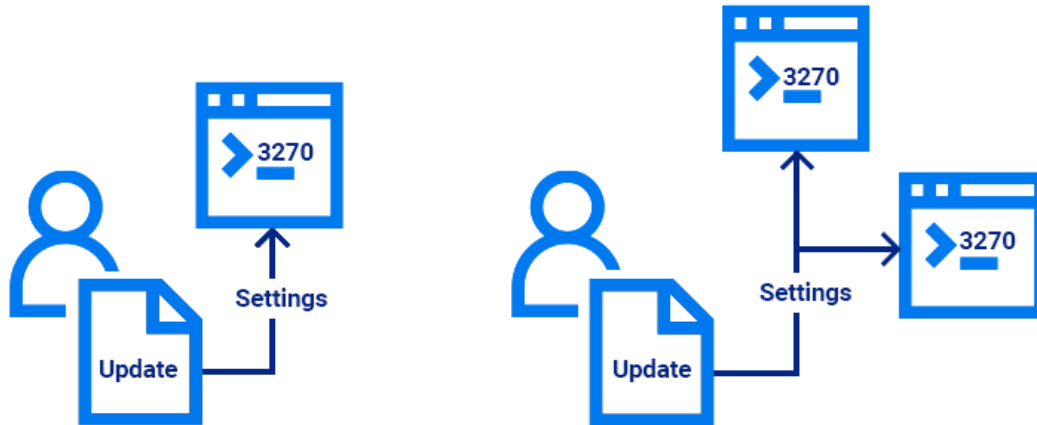
You can use the SettingsUpdate feature to change specific settings in all session document files of a specific type. For example, you could deploy a .rd3xu update file to modify specific settings for all existing and new IBM 3270 sessions. The update file includes only those settings that are changed from the default value.

You might use these special local update files when:

- ◆ You need to apply a setting to all user sessions for a terminal type, both for existing sessions and for new sessions.
- ◆ Your environment is set up to allow users to create and customize their own session documents. (You don't know the names of session documents or how users have customized them.)
- ◆ You don't want to change customizations (such as screen colors) in sessions that users have created.

For example, let's say you need to enforce a security mandate that specifies all user sessions must use TLS 1.2 to connect to your mainframe. You need to ensure this type of secure connection is used, not only for existing sessions but also for any new sessions users may create.

You could create a 3270 update file that has only the security settings required for the TLS 1.2 connection and then deploy it to each user's data directory as shown below.



When users open a 3270 session or create a new 3270 session, the settings in the local update file are applied. None of the other settings in the users' session files are changed. Other types of session documents (for example 5250 sessions) are not changed.

Setting up Local Update Files

To set up a local update file, you'll need to create the update file and then deploy it to users.



"Step One: Unlock access for saving update files" on page 64

"Step Two: Create a local update file" on page 64

"Step Three: Deploy the local update file" on page 65

Step One: Unlock access for saving update files

Before you create an update file, you'll need to unlock the access settings that allow you to save it. You can unlock these settings with Permissions manager as follows. You can also lock/unlock them with Group Policy as shown in [“To unlock access for saving update files with Group Policy” on page 75](#).

To unlock access for saving update files with Permissions Manager

The following instructions show how to use Permissions Manager to unlock access for saving update files.

- 1 On a workstation on which you have installed Reflection, log on as administrator and in the Reflection Desktop [install folder](#), run `AccessConfig.exe`.
- 2 When prompted to create a new permission file, or edit an existing one, choose **Create new permission file**.
- 3 When prompted with a list of access file templates, select the `.access` option for the terminal for which you want to create the update file.
- 4 In the **Groups** panel, click **Next**.
- 5 In the **Additional Security options** panel, select **User can create session update files** and click **Next**.
- 6 Click **Finish** to save the file in the default folder.
- 7 Repeat these steps for each type of terminal that you want to create an update for.

Step Two: Create a local update file

Create the local update file that includes only the settings you want to modify as follows.

To create a local update file

- 1 On the **File** menu, choose **New**.

NOTE: All the settings you configure that are different from the default settings or are not part of the default settings are included in the update. This includes connection settings such as the host address.

- 2 In the Create New Document window, select the terminal for which you want to apply the update and click **Create**.
- 3 In the Create New Terminal Document dialog box, select **Configure additional settings** and click **OK**. (Do not enter the host name unless you want to include it as part of the update.)
- 4 In the Settings window, click **OK**.
- 5 If prompted for a host name or IP Address, click **Cancel**.
- 6 Configure the settings you want to change on user desktops.
- 7 On the **File** menu, choose **Save As**, name the file `update` and then, in the Save as Type list, select an **Encrypted Session Update** file extension and save the file. (For example, a 3270 update file should be named `update.rd3xu`.)

NOTE: SettingsUpdate files that apply automatically to all sessions must be named with the special `update` name. Unlike regular session files, update files contain only the settings you changed (the deltas).

You can create update files that apply to sessions only by saving the file from the Reflection Desktop interface. You cannot create an update file by renaming a session file.

To prevent file tampering, update files that apply to sessions can be saved only as encrypted files.

Step Three: Deploy the local update file

After you create the file, you'll need to deploy it to the users data directory.

To deploy the local update file to users

- 1 Create an MSI companion installer package that includes the update file as shown in [“Walkthrough: Create a Package with the Installation Customization Tool” on page 117.](#)

NOTE: Session update files are deployed to the same directory as regular session files.

- 2 Deploy the MSI package as shown in [“Deploy Companion MSI File from MSI Command Line” on page 144.](#)

Apply Automatic Updates to Select Session Document Files

You can use the SettingsUpdate feature to apply updates only to sessions that you configure to receive them by referencing a update file.

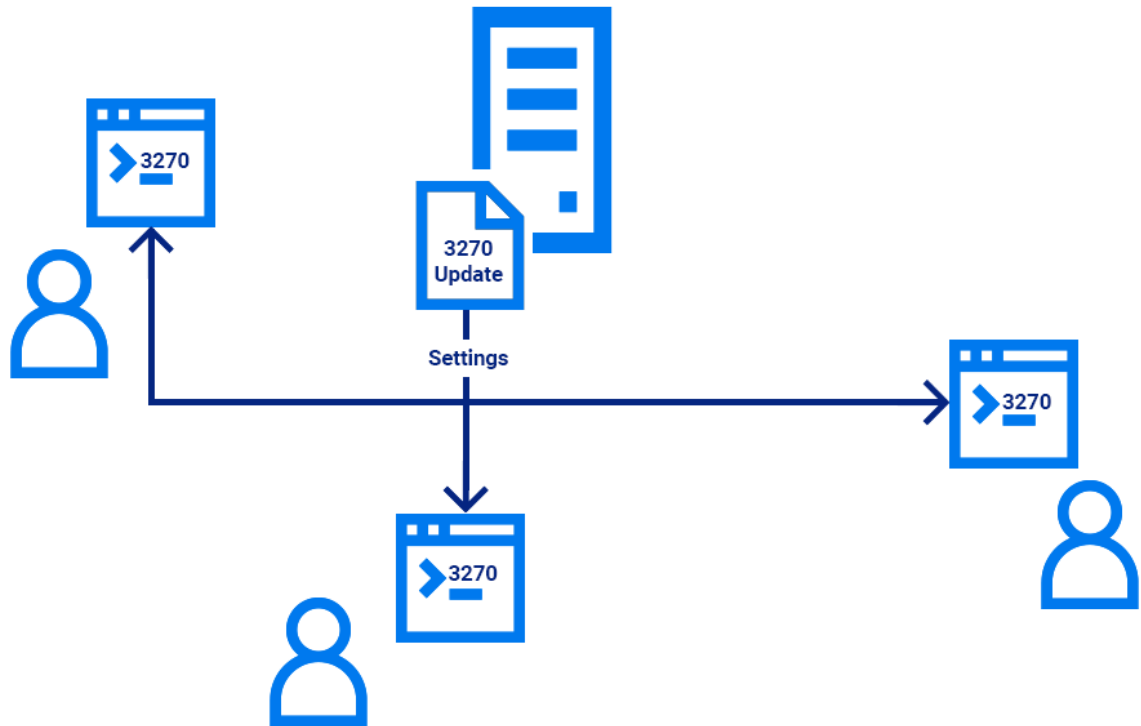
Referenced update files apply settings only to session document files that reference them. These files have no special naming requirements and have an `.xxxxu` extension. They are typically placed on a server but can also be placed in a local folder, such as a public documents folder on a user machine.

You might choose to use referenced update files when:

- ◆ You know you will need to periodically update settings in sessions that you have deployed.
- ◆ You want to update settings only for session documents that you deploy. (You don't need to update settings in session document files that users create.)
- ◆ You know that users often customize settings in sessions after you deploy them and you don't want to write over their session document settings, such as screen colors.

When Reflection opens a session document file that references an update file, it changes the session settings to match those specified in the update file.

For example, you could create a 3270 update file that has only the settings you need to modify and then place it on a server as shown below. Then add a reference to each session document file you want to update before deploying the files.



When users open one of these 3270 sessions, the settings in the update file are applied. None of the other settings in the users' session files are changed. Only the session files you reference are updated. You can periodically update session settings by changing the update file on the server.

Setting up referenced update files

Use the following process to set up a referenced update file and the session documents that reference it.



“Step One: Unlock access for saving update files” on page 67

“Step two: Create an update file and save it on a share” on page 67

“Step Three: Add the referenced update settings file location to the trusted locations list” on page 68

[“Step Four: Create a session document file that references the update file” on page 68](#)

[“Step Five: Deploy the trusted location and session document files” on page 69](#)

Step One: Unlock access for saving update files

Before you create a update file that you can apply to sessions, you’ll need to unlock the access settings that allow you to save it. You can unlock these settings with Permissions manager as follows or using Group Policy as shown in [“To unlock access for saving update files with Group Policy” on page 75](#).

To unlock access for saving update files with Permissions Manager

The following instructions show how to use Permissions Manager to unlock access for saving update files.

- 1 On a workstation on which you have installed Reflection, log on as administrator and in the Reflection Desktop [install folder](#), run `AccessConfig.exe`.
- 2 When prompted to create a new permission file, or edit an existing one, choose **Create new permission file**.
- 3 When prompted with a list of access file templates, select the `.access` option for the terminal for which you want to create the update file.
- 4 In the **Groups** panel, click **Next**.
- 5 In the **Additional Security options** panel, select **User can create session update files** and click **Next**.
- 6 Click **Finish** to save the file in the default folder.
- 7 Repeat these steps for each type of terminal that you want to create an update for.

Step two: Create an update file and save it on a share

You’ll need to create an update file and deploy it to a location that can be accessed by session document files. This is typically a server share but you can also use a public folder on a user workstation.

To create a update file that can be referenced from session document files

- 1 On the **File** menu, choose **New**.

NOTE: All the settings you configure that are different from the default settings or are not part of the default settings are included in the update. This includes connection settings such as the host address.

- 2 In the Create New Document window, select the terminal for the sessions you want to update and click **Create**.
- 3 In the Create New Terminal Document dialog box, select **Configure additional settings** and click **OK**. (Do not enter the host name unless you want to include it as part of the update.)
- 4 In the Settings window, click **OK**.
- 5 If prompted for a host name or IP Address, click **Cancel**.
- 6 Configure the settings you want to change on user desktops.

- 7 On the **File** menu, choose **Save As**, name the file and then, in the **Save as Type** list, select an **Encrypted Session Update** file extension and save the file.

NOTE: Unlike the special name of an update file that is deployed to the user data directory, you can use any file name for an update file that is referenced by session document files.

You can create update files only by saving the file from the Reflection Desktop interface. You cannot create an update file by renaming a session file.

To prevent file tampering, update files can be saved only as encrypted session update files.

- 8 Save the file on a network share or deploy it to a shared location on user workstations.

Step Three: Add the referenced update settings file location to the trusted locations list

Set the location of the referenced update settings file as a trusted location. If you are using a server, specify that network locations can be used.

To add the update settings file location to the trusted locations list


- 1 On the workspace File menu, select Reflection Workspace Settings.
- 2 In the Reflection Workspace Settings window, select **Specify Trusted Locations**.
- 3 In the Specify Trusted Locations dialog box, add the new location for the update file to the path list. (Be sure to use a UNC path to specify a location on a network share.)
- 4 If you are saving the update settings file on a network share, select **Allow trusted locations on my network**.

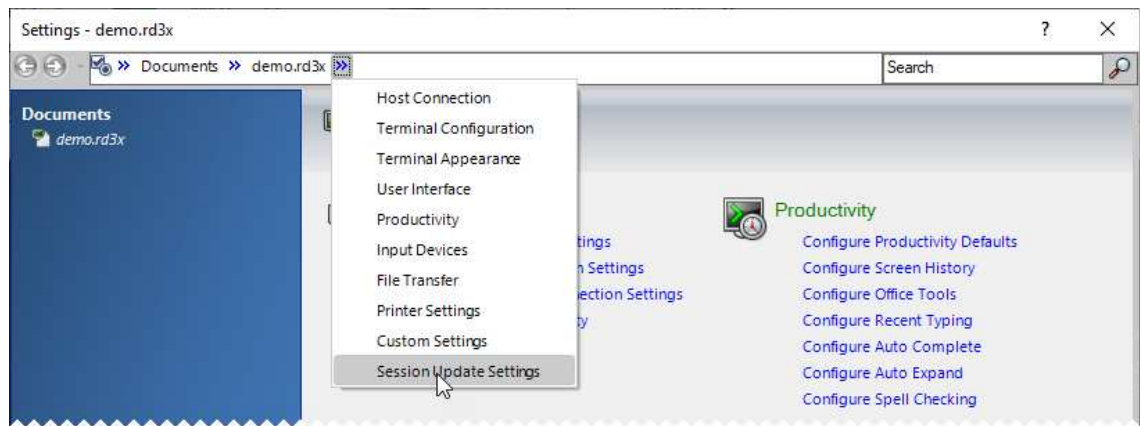
Step Four: Create a session document file that references the update file

Add a reference to the update file in each session document file you want to update.

To configure a session document file to reference an update file

NOTE: You'll need administrative access to perform this procedure.

- 1 Open the session document file you want to receive updates.
- 2 On the Quick Access toolbar, click the Document Settings button .
- 3 On the Settings window breadcrumbs, select **Session Update Settings**.



- 4 In the Session Update Settings window, click **Set Up Session Update Settings**.
- 5 Select **Change currently disabled settings** and then enter the path to the update file you placed on the server. Be sure to use a Universal Naming Convention (UNC) path to identify the server.
- 6 To automatically save the session file after updates are applied, select **Save changes after applying changes**.

Step Five: Deploy the trusted location and session document files

Deploy the workspace settings trusted location and the session document files you have configured to receive updates.

Deploy the trusted location settings and session document files to users

- 1 Create an MSI companion installer package that includes the session files you configured as shown in [“Walkthrough: Create a Package with the Installation Customization Tool” on page 117](#).
- 2 Add the file that contains the trusted locations you configured to the MSI companion installer package or create a new package to deploy these settings.

You can add the `Application.settings` file that contains the settings you configured earlier to the MSI companion file package. See [“Package Sessions and Custom Settings Files” on page 116](#) and [“Customized Files that Must be Deployed to Specific Locations” on page 119](#) to make sure you deploy this file to the right location.

or

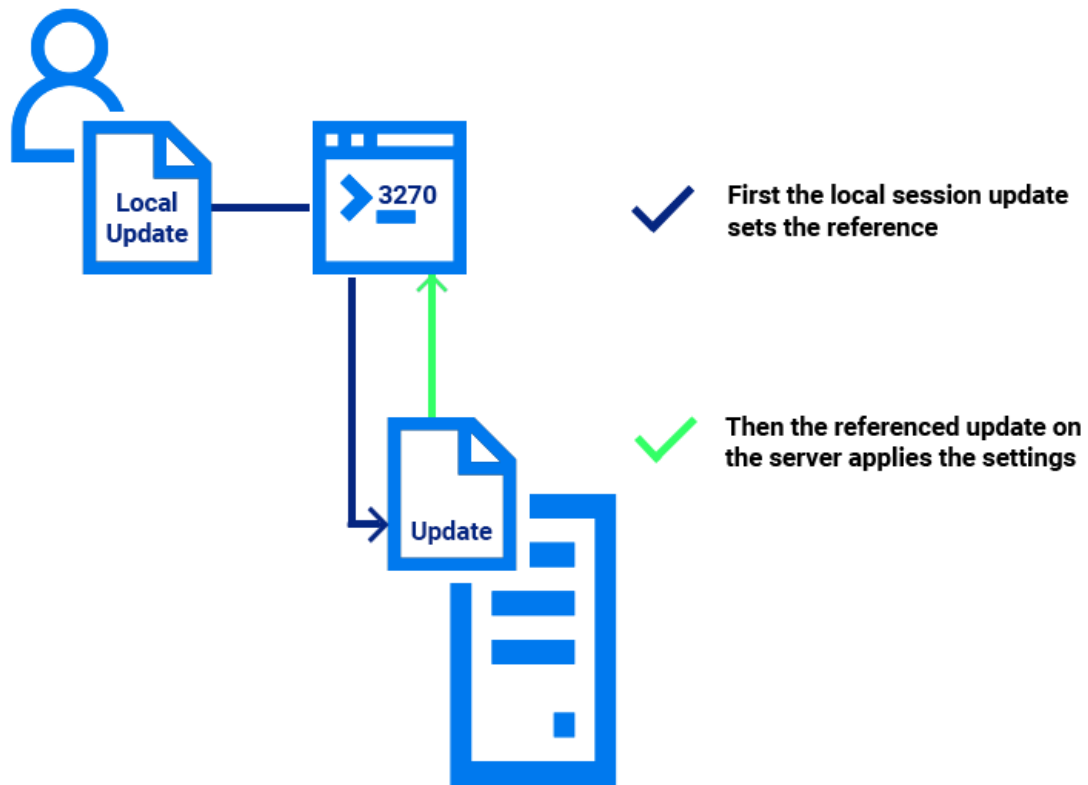
You can create a workspace `Update.settings` file that changes only these trusted location settings as shown in [“Update Only Specific Workspace Settings” on page 98](#).
- 3 Deploy the MSI package as shown in [“Deploy Companion MSI File from MSI Command Line” on page 144](#).

Set up Automatic Updates for All Session Document Files

You might choose to use a combination of referenced update files and local update files to meet the following requirements:

- You will need to periodically update settings in all sessions for a certain terminal type (including both sessions you deploy and existing or new sessions users create).
- You don't want to write over customizations (such as screen colors) that users have made to their session document settings.
- You want to simplify your updates so you can update settings from a central location, such as a server share.

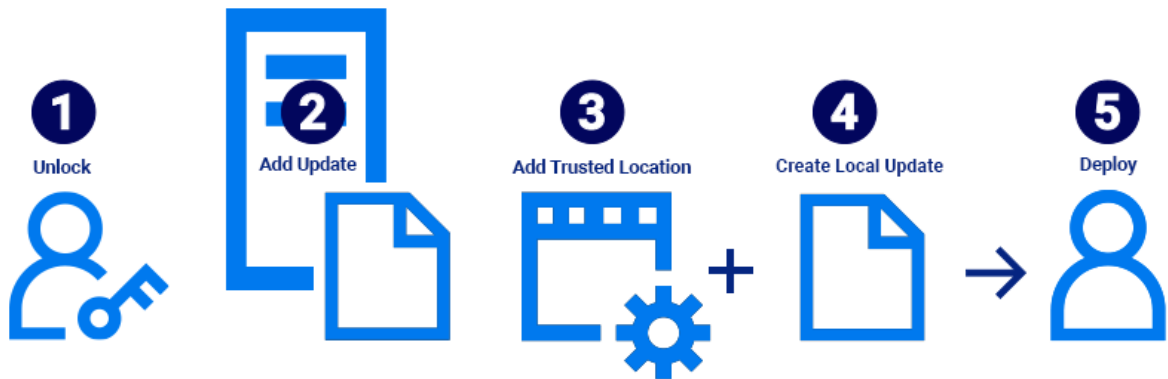
For example, you could create a 3270 update file that has the settings you want to change and then place it on a server as shown below. Then add a reference to this file location in a local 3270 update file and deploy that file to users. The first time a 3270 session is opened, the local update file configures the session to reference the update file on the server. The next time the session is opened, the local update file is ignored and the update file on the server applies its settings to the session.



When a user opens a 3270 session document, the local update file configures the session with a reference to the 3270 update file on the server. The next time the session is opened, the local update file is ignored and the referenced update file on the server applies its session settings.

Setting up a combination of local and referenced update files

Use the following process to use local update files to configure sessions with references to an update file on a server or another location.



“Step One: Unlock access for saving update files” on page 71

“Step Two: Create an update file that can be referenced from session document files” on page 72

“Step Three: Add the update file location to trusted locations” on page 72

“Step Four: Create a local update file that has a reference to the update file on the server” on page 73

“Step Five: Deploy the trusted location settings and local update file to users” on page 74

Step One: Unlock access for saving update files

Before you create an update file, you’ll need to unlock the access settings that allow you to save it. You can unlock these settings with Permissions manager as follows or using Group Policy as shown in “To unlock access for saving update files with Group Policy” on page 75.

To unlock access for saving update files with Permissions Manager

The following instructions show how to use Permissions Manager to unlock access for saving update files.

- 1 On a workstation on which you have installed Reflection, log on as administrator and in the Reflection Desktop [install folder](#), run `AccessConfig.exe`.
- 2 When prompted to create a new permission file, or edit an existing one, choose **Create new permission file**.
- 3 When prompted with a list of access file templates, select the `.access` option for the terminal for which you want to create the update file.
- 4 In the **Groups** panel, click **Next**.
- 5 In the **Additional Security options** panel, select **User can create session update files** and click **Next**.
- 6 Click **Finish** to save the file in the default folder.
- 7 Repeat these steps for each type of terminal that you want to create an update for.

Step Two: Create an update file that can be referenced from session document files

You'll need to create an update file and deploy it to a location that can be accessed by session document files. This is typically a server but you can also use a public folder on a user workstation.

To create an update file that can be referenced from session document files

- 1 On the **File** menu, choose **New**.

NOTE: All the settings you configure that are different from the default settings or are not part of the default settings are included in the update. This includes connection settings such as the host address.

- 2 In the Create New Document window, select the terminal for the sessions you want to update and click **Create**.
- 3 In the Create New Terminal Document dialog box, select **Configure additional settings** and click **OK**. (Do not enter the host name unless you want to include it as part of the update.)
- 4 In the Settings window, click **OK**.
- 5 When prompted for a host name or IP Address, click **Cancel**.
- 6 Configure the settings you want to change.
- 7 On the **File** menu, choose **Save As**, name the file and then, in the **Save as Type** list, select an **Encrypted Session Update** file extension and save the file.

NOTE: Unlike the special name of an update file that is deployed to a user data directory on a desktop, you can use any file name for a update file that is referenced by session document files.

You can create update files that apply to sessions only by saving the file from the Reflection Desktop interface. You cannot create an update file by renaming a session file.

To prevent file tampering, update files that apply to sessions can be saved only as encrypted update files.

- 8 Save the file on a network share or deploy it to a public folder on a user workstation.

Step Three: Add the update file location to trusted locations

Set the location of the update file as a trusted location. If you are using a server, specify that network locations can be used.

To add the update settings file location to the trusted locations list

- 1 On the workspace File menu, select Reflection Workspace Settings.
- 2 In the Reflection Workspace Settings window, select **Specify Trusted Locations**.
- 3 In the Specify Trusted Locations dialog box, add the location for the update file to the path list. (Be sure to use a UNC path to specify a location on a network share.)
- 4 If you are saving the update file on a network share, select **Allow trusted locations on my network**.


Step Four: Create a local update file that has a reference to the update file on the server

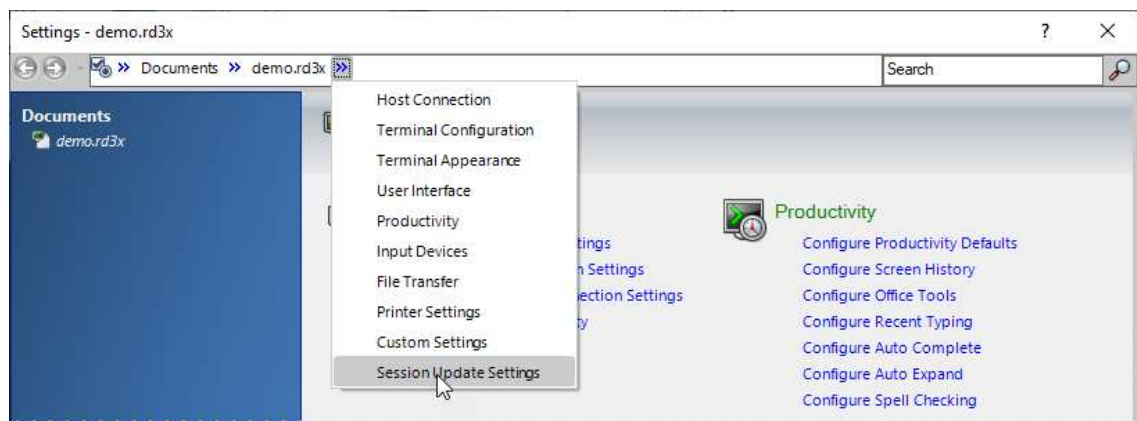
Create a local update file that has a reference to the update file on the server.

To create an update file that includes a reference to a session update file on a network share

- 1 On the **File** menu, choose **New**.

NOTE: All the settings you configure that are different from the default settings or are not part of the default settings are included in the update. This includes connection settings such as the host address.

- 2 In the Create New Document window, select the terminal you want apply the updates to and click **Create**.
- 3 In the Create New Terminal Document dialog box, select **Configure additional settings** and click **OK**. (Do not enter the host name unless you want to include it as part of the update.)
- 4 In the Settings window, click **OK**.
- 5 If prompted for a host name or IP Address, click **Cancel**.
- 6 On the Quick Access toolbar, click the Document Settings button .
- 7 On the Settings window breadcrumbs, select **Session Update Settings**.



- 8 In the Session Update Settings window, click **Set Up Session Update Settings**.
- 9 Select **Change currently disabled settings** and then enter the path to the update file you placed on the server. Be sure to use a Universal Naming Convention (UNC) path to identify the server.

NOTE: You can set a local update file to set a reference to an update file that does not exist yet. This can be useful for creating versions of updates.

- 10 To automatically save the session files after the update, select **Save changes after applying changes**.
- 11 On the **File** menu, choose **Save As**, name the file `update` and then, in the **Save as Type** list, select an **Encrypted Session Update** file extension. (For example, a 3270 update file should be named `update.rd3xu`.)

NOTE: Update files that apply automatically to all sessions must be named with the special `update` name. Unlike regular session files, update files contain only the settings you changed (the deltas).

You can create update files only by saving them from the Reflection Desktop interface. You cannot create an update file by renaming a session file.

Step Five: Deploy the trusted location settings and local update file to users

After you configure the trusted locations and the local update file, you'll need to deploy them to users.

To deploy the trusted location settings and local update file to users

- 1 Create an MSI companion installer package that includes the local update file you want to deploy as shown in [“Walkthrough: Create a Package with the Installation Customization Tool” on page 117](#).

NOTE: SettingsUpdate files are deployed to the same directory as regular session files.

- 2 Add the file that contains the trusted locations you configured to the MSI companion installer package or create a new package to deploy these settings.

You can add the `Application.settings` file that contains the settings you configured earlier to the MSI companion file package. See [“Package Sessions and Custom Settings Files” on page 116](#) and [“Customized Files that Must be Deployed to Specific Locations” on page 119](#) to make sure you deploy this file to the right location.

or

You can create a workspace `Update.settings` file that changes only these trusted location settings as shown in [“Update Only Specific Workspace Settings” on page 98](#).

- 3 Deploy the MSI package as shown in [“Deploy Companion MSI File from MSI Command Line” on page 144](#).

Create or Modify SettingsUpdate Files

Before you create or modify a SettingsUpdate file you want to apply to a session, you'll need to unlock the access required to save these special files. Access for saving update files is locked by default.

After you create an update file, you can change settings or add new settings to the file by simply opening it in the workspace, modifying it, and saving it. Then you'll need to deploy the file to replace the existing version of the file with your new version.

Step One: Unlock access for saving update files

By default, the access required to save update files for sessions is locked. Before you create, modify, and save update files, you'll need to change these default access settings, using the Permissions Manager tool or Group Policy.

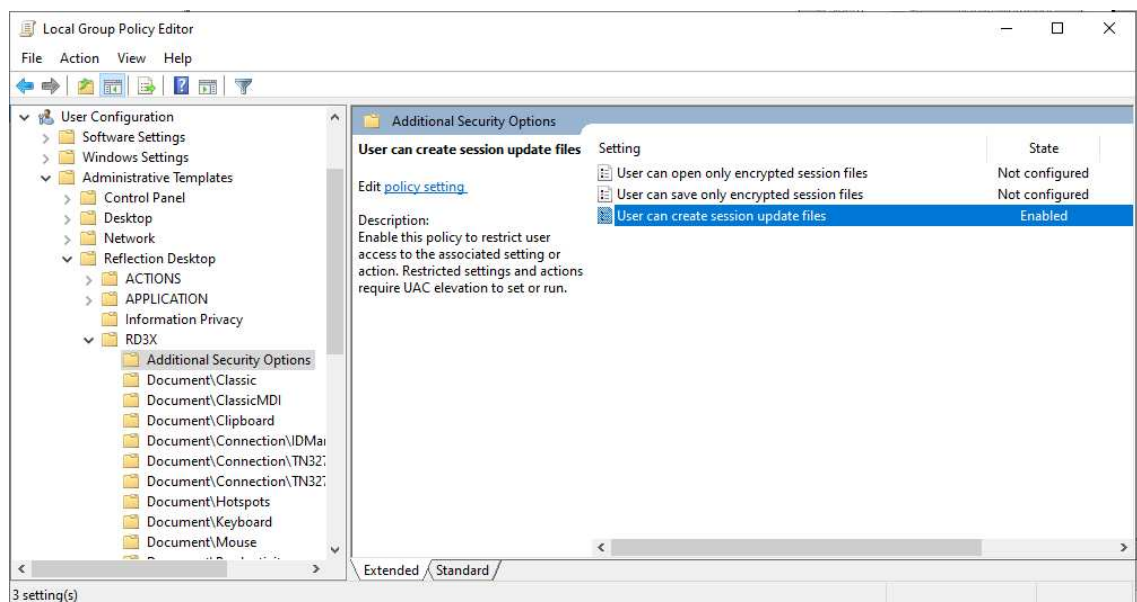
To unlock access for saving update files with Permissions Manager

The following instructions show how to use Permissions Manager to unlock access for saving update files.

- 1 On a workstation on which you have installed Reflection, log on as administrator and in the Reflection Desktop [install folder](#), run `AccessConfig.exe`.
- 2 When prompted to create a new permission file, or edit an existing one, choose **Create new permission file**.
- 3 When prompted with a list of access file templates, select the `.access` option for the terminal for which you want to create the update file.
- 4 In the **Groups** panel, click **Next**.
- 5 In the **Additional Security options** panel, select **User can create session update files** and click **Next**.
- 6 Click **Finish** to save the file in the default folder.
- 7 Repeat these steps for each type of terminal that you want to create an update for.

To unlock access for saving update files with Group Policy

- 1 Copy the `.admx` files from `...\install_dir\Configuration\GroupPolicy\ADMX` to the central store (`%systemroot%\PolicyDefinitions`)
- 2 Copy all required locale `.adml` files to: `%systemroot%\PolicyDefinitions\<locale>`
- 3 Open the Group Policy Object Editor (`gpedit.msc`)
- 4 Under either **Computer Configuration** or **User Configuration**, browse to **Administrative Templates | Reflection Desktop** and then the folder for the terminal you are configuring.
- 5 Under **Additional Security Options**, select **User can create session update files** and enable this setting, as shown below.



Step Two: Deploy modified update files

After you modify an update file, you'll need to deploy it as follows:

- ◆ If the file is a local update.xxxu file, you'll need to replace the file in the users' data directory. After you modify the file, you'll need to package the file and deploy it to users, as shown in [“Package Sessions and Custom Settings Files” on page 116](#). Update files are packaged and deployed in the same way as regular session files (see [“Walkthrough: Create a Package with the Installation Customization Tool” on page 117](#)).
- ◆ If the file is a referenced update file that resides on a server, you can just overwrite that file with the revised version.
- ◆ If the file is a referenced update file that resides in a public location on user desktops, you'll need to package and deploy this update file to that location as shown in [“Package Sessions and Custom Settings Files” on page 116](#).

5 Protect Data

Use the Trust Center to protect your working environment from information theft, and your data from potential damage caused by opening documents from non-trusted sources.

This article

[Add Trusted Locations \(page 77\)](#)

[“Configure API and Macro Security” on page 78](#)

Describes

How to define locations from which you can safely open (and store) documents.


How to control access to the Reflection API and control the execution of actions invoked by a macro or API call.

Add Trusted Locations

A trusted location is a directory that is designated as a secure source for opening files. By default, Reflection Desktop allows users to open documents only in directories specified as trusted locations in the Reflection settings. Reflection Desktop specifies three trusted locations in the workspace `Application.settings` file in the program directory.

When you add other locations, these locations are saved in a custom `Application.settings` file in the user data directory folder. If you add trusted locations, you will need to deploy this file.

To set up a trusted location

1. From the Reflection File menu or Reflection button  (if using the Office 2007 Look and Feel), select **Reflection Workspace Settings**.
2. Under Trust Center, click **Specify Trusted Locations**.
3. Click **Add new location** and then, under **Path**, browse to the location you want to add.

NOTE: You can use Windows environment variables to define the trusted location.

4. To trust all folders within the trusted location, click **Subfolders**.
5. To “lock down” these settings, see [“Control Access to “Lock Down” Settings and Controls” on page 103](#).
6. To package this file for deployment, see [Package Configuration Files](#).

NOTE:

These settings are saved in the `Application.settings` file. You can deploy this file to one of the following locations:


Location for a single user: `[AppDataFolder]\Micro Focus\Reflection\Desktop\v17.0`

Location for all users: `[CommonAppDataFolder]\Micro Focus\Reflection\Desktop\v17.0`

Configure API and Macro Security

You can enable the Reflection Desktop .NET API, and specify corresponding settings.

To set up API and macro and security

- 1 From the Reflection File menu or the Reflection button  (if using the Office 2007 Look and Feel), select **Reflection Workspace Settings**.
- 2 Under **Trust Center**, click **Set Up API and Macro Security**.
- 3 Configure the API settings as follows:

To	Select
Prevent custom applications from accessing this installation.	Disable .Net API
Determine if Reflection legacy macros are supported, and determine which legacy API has preference for the GetObject() method used to retrieve API COM objects. (Reflection supports multiple APIs, but can accept GetObject() calls for only one type of legacy API object at a time.)	Legacy API preference

- 4 Under **Action Permissions**, specify what you want to happen if an action that has been restricted through Group Policy or the Permissions Manager is initiated through a macro or API call.

To	Select
On a computer running Windows 10, Windows 8, or Windows 7, select to control restricted actions with User Account Control (UAC).	Require elevated rights
Select to run restricted actions that are initiated through a macro or API call as expected. The same actions won't run if they are initiated through the user interface.	Execute the action

- 5 To “lock down” these settings, see [“Control Access to “Lock Down” Settings and Controls” on page 103](#).
- 6 To package this file for deployment, see [Package Configuration Files](#).

NOTE: These settings are saved in the `Application.settings` file. You can deploy this file to one of the following locations:

Location for a single user: `[AppDataFolder]\Micro Focus\Reflection\Desktop\v17.0`

Location for all users: `[CommonAppDataFolder]\Micro Focus\Reflection\Desktop\v17.0`

6 Set up Information Privacy

Reflection® Desktop products include support for several information privacy features that are designed to help you comply with industry and government regulations, like Payment Card Industry Data Security Standards (PCI DSS) that address data protection concerns. You can configure Reflection to protect sensitive data so that it is not displayed in Reflection productivity features like Screen History, and when sharing host data using Windows copy/paste for integration with other applications. For IBM hosts, you can mask sensitive data so that it is not displayed on host screens. You can also require secure connections for sessions that handle sensitive data.



Setting Up Information Privacy

This guide shows how to configure Reflection to support PCI DSS requirements.

- ♦ [“What is PCI DSS?” on page 80](#) describes PCI DSS and shows which requirements Reflection aids compliance with.
- ♦ [“Configure Information Privacy Settings” on page 80](#) is a high level summary of how to configure Reflection to protect information privacy.
- ♦ [“Setting up Redaction of Primary Account Numbers” on page 81](#) provides in-depth information about the three Reflection options for credit card PAN (Primary Account Number also referred to as “credit card number”) detection: Simple PAN Detection, Simple PAN Detection with Preceding Text, and Reflection PAN Detection. This section includes suggestions about when to use each option, the considerations of each, and examples of how to set them up.
- ♦ [“Configure Information Privacy with Group policy” on page 90](#) shows how to use Group Policy files to set up Information Privacy.
- ♦ [“Setting up Privacy Filters” on page 90](#) includes suggestions for using simple expressions to create privacy filters that redact personal data such as phone numbers or US Social Security numbers.
- ♦ [“More About Information Privacy” on page 92](#) provides references to general industry PCI DSS documentation and tutorials for creating regular expressions (used for PAN identification).

What is PCI DSS?

PCI DSS (Payment Card Industry Data Security Standard) is a proprietary information security standard comprising technology requirements and process requirements designed to prevent fraud when handling credit card information. All companies who handle credit cards are subject to this standard.

To be PCI DSS compliant, organizations must meet twelve PCI DSS requirements. Reflection aids compliance with requirements 3,4,6,7 and 10.

	PCI DSS Requirements	Support
1	Install and maintain a firewall configuration to protect cardholder data.	
2	Do not use vendor-supplied defaults for system passwords and other security parameters.	
3	Protect stored cardholder data.	Supported
4	Encrypt transmission of cardholder data across open, public networks.	Supported
5	Use and regularly update antivirus software.	
6	Develop and maintain security systems and applications.	Supported
7	Restrict access to cardholder data by business need-to-know.	Supported
8	Assign a unique ID to each person with computer access.	
9	Restrict physical access to cardholder data.	
10	Track and monitor all access to network resources and cardholder data.	Supported
11	Regularly test security systems and processes.	
12	Maintain a policy that addresses information security.	

Configure Information Privacy Settings

To set up Reflection to protect sensitive data, you'll need to configure the following options in the Information Privacy Dialog Box:

1. Under **Primary account (PAN) Number Redaction Rules**, choose **Enable Redaction** and then select options for displaying redacted data. If you want to prevent PAN data from being saved in an external file, end user messaging devices, or any component that saves screen data, such as the Reflection Recent Typing feature, select **Do not store typed PANs**.

NOTE: **Redact display data** and **Redact data while typing** options for both privacy filters and PAN detection currently apply only to IBM emulation sessions. They are not currently implemented for other terminal types.

2. Set up **Primary Account Number (PAN) Detection Rules**. You will need to choose a redaction method and configure it as shown in [Setting up Redaction of Primary Account Numbers](#).
3. If you want to use Privacy Filters, you will need to configure the **Privacy Filters Redaction Rules** and then create a simple or regular expression for the filter as shown in [Setting up Privacy Filters](#).
4. Under **PCI DSS Rules**, set up requirements for secure connections. You can require secure connections for all network connections or for wireless only. (See [Set Up Information Privacy Dialog Box](#) in the Reflection Desktop Help.)
5. If you want to log access to credit card data, select **Enable API events when PANs are viewed by the user**. Then follow the instructions in the NET API Guide or the VBA Guide for an example of how to handle an event to log access to cardholder data.
6. To “lock down” these settings, see [“Control Access to “Lock Down” Settings and Controls” on page 103](#).

Setting up Redaction of Primary Account Numbers

You can choose from three methods for redacting credit card PAN data: Simple Primary PAN Detection, Simple PAN Detection with Preceding Text, or Reflection PAN Detection.

Method	Use when	Considerations
Simple PAN Detection matches a credit card number sequence.	All of the credit card data in your host applications are displayed and entered in a “contiguous” fashion. You are only detecting PANs for the prepackaged major credit card issuers.	Easy to set up
Simple PAN Detection with Preceding Text matches preceding text (e.g., Account) followed by a credit card number sequence.	Same as above except credit card data in your host applications are always labeled in predictable ways.	Relatively easy to set up Avoids false positives

Method	Use when	Considerations
Reflection PAN Detection uses regular expressions to detect PANs.	<p>You need to define custom card issuer patterns to detect, such as oil company or department store cards.</p> <p>PANs appear in a non-contiguous format or are entered using non-standard digit group separators.</p> <p>You want PAN detection to be especially “aggressive” or “greedy” in that any digit grouping on any screen should be considered for redaction, and you need to be able to redact without regard to what other text or digit separators may appear between single or groups of digits in the PAN.</p>	<p>Allows the greatest degree of flexibility and customization for unique detection needs</p> <p>Computationally-intensive— can degrade performance on PCs with limited processing power or memory</p> <p>The likelihood of “false positive” redaction is much greater with this method than the other two, especially if your host screens are very digit-laden</p>

Setting up and using Simple PAN Detection

When Simple PAN Detection is selected, Reflection matches a credit card number sequence (a 13- 16 digit number).

NOTE: The credit card character sequences can also include whitespace and hyphen characters as digit grouping separators.

When to use Simple PAN Detection

Use Simple PAN Detection when your application meets all of the following conditions:

- ◆ All of the credit card account numbers in your host applications are displayed and entered in a “contiguous” fashion. In other words, the PANs always appear or are always entered as a single continuous string (e.g. 1111-1111-1111-1111, 2222 2222 2222 2222, 4444444444444444 etc.).
- ◆ All of the account numbers that need to be redacted are from one or more of the following issuers: Visa, MasterCard, American Express, Discover, Diner’s Club, Carte Blanche, Voyager, JCB, or enRoute.

NOTE: If data in your application are displayed or entered in a noncontiguous fashion or you need to detect other card issuers, use the Reflection PAN Detection option and enter additional custom patterns for those issuers in the Custom Detection Rules table, or use privacy filters to specify a custom pattern.

Advantages of Simple PAN Detection

This method requires no additional configuration and should be suitable for most situations.

Considerations for Simple PAN Detection


Although simple PAN Detection is easy to set up and use, there are a few items to consider when using this method:

- ♦ This method works only with the major credit card issuers noted above. (In some cases, privacy filters can be used in conjunction with this method to add additional issuers.)
- ♦ It is possible to get false positives in entry fields where large numbers of digits are entered consecutively and where there are no non-digit separator characters delimiting the overall sequence of digits.

How to set up Simple PAN Detection

Use the Information privacy dialog box to set up Simple PAN Redaction.

To set up Simple PAN redaction

- 1 From the Reflection File menu or the Reflection button  (if using the Office 2007 Look and Feel), select **Reflection Workspace Settings**.
- 2 Under Trust Center, click **Set Up Information Privacy**.
- 3 On the Information Privacy dialog box, select **Enable Redaction** and then select **Simple PAN Detection**.
- 4 To “lock down” these settings, see [“Control Access to “Lock Down” Settings and Controls” on page 103](#).
- 5 To package this file for deployment, see [Package Configuration Files](#).

NOTE: Privacy filter settings are saved in the `PrivacyFilters.xml` file. All other Information for Privacy settings is saved in the `PCIDSS.settings` file. You can deploy these files to one of the following locations:

Location for a single user: `[AppDataFolder]\Micro Focus\Reflection\Desktop\v17.0`

Location for all users: `[CommonAppDataFolder]\Micro Focus\Reflection\Desktop\v17.0`

Setting up and Using Simple PAN Detection with Preceding Text

When Simple PAN Detection is selected, Reflection matches preceding text (e.g., keywords like “Account”) followed by a credit card number sequence (a 13-16 digit number).

NOTE: The credit card character sequences can also include whitespace and hyphen characters as digit grouping separators.

When to use Simple PAN Detection with Preceding Text

Use Simple PAN Detection with Preceding Text when your application meets all of the following conditions:

- ♦ All of the credit card account numbers in your host applications are displayed and entered in a “contiguous” fashion. In other words, the PANs always appear or are always entered as a single continuous string (e.g. 1111-1111-1111-1111, 2222 2222 2222 2222, 4444444444444444 etc.).
- ♦ All of the account numbers that need to be redacted are from one or more of the following issuers: Visa, MasterCard, American Express, Discover, Diner’s Club, Carte Blanche, Voyager, JCB, or enRoute.
- ♦ Your host application screens that contain credit cards are very well defined, and credit card information is always “tagged” or prefixed in predictable ways. For instance, your host application has only a handful of screens that contain (or potentially can contain) credit card numbers, and those numbers on the screen are always preceded by a label such as “Account Number: “or “Credit Card.”

Advantages of Simple PAN Detection with Preceding Text

Simple PAN Detection with Preceding Text has the following advantages:


- ♦ This method further restricts the data subject to potential redaction and can serve to virtually eliminate “false positives” in other areas of the screen that do not contain credit card data.
- ♦ Any potential card numbers, even valid ones, are not considered unless they immediately follow one of the defined strings and the digits do not contain any other data but digits, whitespace, and hyphen separators. This is appropriate for host applications that contain a lot of other numeric data that should not be considered for redaction.
- ♦ If your host applications have a large numbers of “digit intensive” screens, especially ones that contain lengthy digit data such as part/SKU numbers, ISBN numbers, etc., use of this option greatly reduces the chance of accidental “false positives” in data that could mistakenly be detected as a credit card number.

Considerations for Simple PAN Detection with Preceding Text

Simple PAN Detection with Preceding Text has a few items to consider when using this method:

- ♦ Before you deploy Reflection, you will need to define the text strings that precede card numbers. This means examining your host applications and noting the strings that precede areas where credit cards are either displayed (protected) or entered (unprotected).
- ♦ Redaction occurs only after defined text strings. The entire PAN must appear immediately after one of the defined strings, without any additional non-digit, non-whitespace or hyphen separator characters appearing.

How to set up Simple PAN Detection with Preceding Text

- 1 In your host application screens, identify all of the keywords that precede credit card numbers.
- 2 From the Reflection File menu or the Reflection button  (if using the Office 2007 Look and Feel), select **Reflection Workspace Settings**.

- 3 Under Trust Center, click **Set Up Information Privacy**.
- 4 On the Information Privacy dialog box, select **Enable Redaction** and then select **Simple PAN Detection**.
- 5 In the Information Privacy dialog box, select **Detect PANs based on preceding text** and add the keywords to the **Text Items** table.
- 6 To “lock down” these settings, see [“Control Access to “Lock Down” Settings and Controls” on page 103](#).
- 7 To package this file for deployment, see [Package Configuration Files](#).

NOTE: Privacy filter settings are saved in the `PrivacyFilters.xml` file. All other Information for Privacy settings is saved in the `PCIDSS.settings` file. You can deploy these files to one of the following locations:

Location for a single user: `[AppDataFolder]\Micro Focus\Reflection\Desktop\v17.0`

Location for all users: `[CommonAppDataFolder]\Micro Focus\Reflection\Desktop\v17.0`

Setting up and Using Reflection PAN Detection

When Reflection PAN Detection is selected, Reflection uses the following process to detect credit card PAN data:

1. Read a host screen’s data as input.
2. Mask out “exclusion patterns” of digit data that are defined as “not credit card data.”
3. Remove all non-digit data from the input host screen data (leaving just a continuous ordered string of digits).
4. Apply recognition methods to the remaining digit-only data using stock credit card patterns (provided with Reflection) and specified custom credit card patterns of 13 to 16 digits, from left to right.
5. Redact recognized PANs within this data.

NOTE: Only matches that pass checksum calculation according to the Luhn Algorithm are redacted. (The Luhn Algorithm, also known as the “modulus 10” or “mod 10” Algorithm, is a checksum formula used to identify identification numbers (see http://en.wikipedia.org/wiki/Luhn_algorithm).

6. Merge any redactions back into the original host screen data.
7. Restore the data that was masked by exclusion patterns.

You can specify custom credit card patterns that you want Reflection to recognize. To avoid “false positive” redaction, you will also likely need to define additional exclusion patterns (or literal strings containing digits such as application ids, screen ids, or copyright notices).

When to use Reflection PAN Detection

Use Reflection PAN Detection for any of the following applications:

- ◆ You need to detect non-standard credit card patterns/issuers (for instance, oil company or department store cards).
- ◆ Your host application has specialized screens where credit cards can be entered or are displayed in non-standard ways (e.g., non-contiguous sets of digits such as multiple input fields of data arranged in a vertical table or contiguous sets of digits using nonstandard digit group separators).
- ◆ You want PAN detection to be especially “aggressive” or “greedy” in that any digit grouping on any screen should be considered for redaction, and you need to be able to redact without regard to what other text or digit separators may appear between single or groups of digits in the PAN.

Advantages of Reflection PAN Detection

Reflection PAN Detection allows the greatest degree of flexibility and customization for unique detection needs.

- ◆ This method can be configured to detect non-standard credit card issuer patterns of 13-16 digits.
- ◆ If you have host application screens with other numeric data such as part or SKU numbers that look very similar to credit cards, you can exclude those custom patterns from redaction.
- ◆ This method can detect credit cards that use separator text or characters (other than whitespace or hyphens) that are mixed in with the full account number (e.g. “1111 / 3333 / 4444 / 5555”, “first: 1111 second: 2222 third: 3333 fourth: 4444”).
- ◆ This method is suitable for applications that have host screens where credit cards are entered in multiple fields, especially if the screens are laid out in a vertical “table” format.
- ◆ This method can be configured to detect non-standard credit card issuer patterns of 13-16 digits. It also allows detection and redaction of PANs that have nonstandard digit group separation or other random string data in between the digits.

For example, the following input data can be detected as a PAN with this method, assuming 1111222233334444 is a potential valid credit card number:

First: 1111 Second: 2222 Third: 3333 Fourth: 4444

Or

1111#2222#3333#4444

The other methods could not be used to detect a credit card number that appears in such a way.


Considerations for Reflection PAN Detection

Reflection PAN Detection has a few items to consider when using this method:

- ◆ This is the most complex method to set up. In order to configure exclusion patterns, you will need to be familiar with regular expressions and their syntax.

- ◆ This method is the most computationally-intensive. It can result in performance degradation and increase response time on PCs with limited processing power and/or memory, especially when the **Redact data while typing** option is selected.
- ◆ The likelihood of “false positive” redaction is much greater with this method than the other two, especially if your host screens are very digit-laden.
- ◆ It is likely that you will need to go through a review process with all of your host applications to eliminate false positives, by identifying and defining exclusion patterns that are not supported “out of the box”. Some examples of these are custom part or SKU numbers for inventory applications, screen or application numeric identifiers, copyrights, international phone number formats, and the like.

How to Set up Reflection PAN Detection

- 1 From the Reflection File menu or the Reflection button  (if using the Office 2007 Look and Feel), select **Reflection Workspace Settings**.
- 2 Under Trust Center, click **Set Up Information Privacy**.
- 3 On the Information Privacy dialog box, select **Enable Redaction** and then select **Reflection PAN Redaction**.
- 4 Under **Primary Account Number (PAN) Detection Rules**, configure **Custom Detection Rules** and **Custom Exception Expressions** as shown in “How to Set up Detection Rules” on page 87 and “How to Set up Custom Exclusions to avoid false positives” on page 88.
- 5 To “lock down” these settings, see “Control Access to “Lock Down” Settings and Controls” on page 103.
- 6 To package this file for deployment, see [Package Configuration Files](#).

NOTE: Privacy filter settings are saved in the `PrivacyFilters.xml` file. All other Information for Privacy settings is saved in the `PCIDSS.settings` file. You can deploy these files to one of the following locations:

Location for a single user: `[AppDataFolder]\Micro Focus\Reflection\Desktop\v17.0`

Location for all users: `[CommonAppDataFolder]\Micro Focus\Reflection\Desktop\v17.0`

How to Set up Detection Rules

To set up detection rules, you specify the custom credit card sequences (patterns) that you need Reflection to recognize.

To enter a custom credit card number sequence, add a regular expression that specifies the pattern the sequence must follow to the Custom Detection Rules table. Because the Reflection PAN Detection method detects digit-only data, do not enter digit grouping separator characters such as hyphens or whitespace in these custom expressions.

NOTE: These patterns are applied on “remaining digit” data only, i.e. a string of all digits on the screen that have not been excluded by stock or custom exclusion patterns. Therefore, the custom PAN pattern should not include non-numeric data such as separators, nor can it contain specifications of preceding or trailing characters, whitespace, or word boundaries.

To match fixed prefixes, you can combine literal text with a regular expression. Typically card issuers have a prefix of 1 or more digits that are fixed, followed by the remainder of digits that can vary. If the prefix is always “static” this can be expressed with literal text in the regular expression. Do not specify something like `\d{16}`, which would match any 16 digit number. An expression like this is very likely to result in unintended false positives!

Examples of regular expressions that detect credit card PANS

Example 1: Using a regular expression

To detect cards issued by a fictitious Acme Corporation, which are always 15 digits starting with “7200”, you might add the following regular expression:

```
7200\d{11}
```

This means match all consecutive instances starting with “7200” followed by any 11 additional digits.

Example 2: Using literal text and a regular expression

Let’s say we need to detect cards issued by a fictitious “National Bank” that are 16 digits starting with “88” with the next digit ranging from 0 to 5 (e.g. 880, 881, 882, 883, 884, or 885 are the valid prefixes). One could add the following regular expression for this case:

```
88[0-5]\d{13}
```

This is read as “match the literal text 88, followed by a digit in the range 0-5 inclusive, and followed by 13 additional digits.”

How to Set up Custom Exclusions to avoid false positives

Because Reflection PAN Detection disregards the context of non-digit text when detecting PANs, all digits appearing on the screen could potentially be aggregated together to form a potential PAN. It is likely that your host screen data contains digit data that is not to be considered for PAN redaction.

To exclude this data from PAN redaction, Reflection uses a set of regular expressions. These exception expressions are listed in the Custom Exception Expressions table, in the Information Privacy dialog box. Reflection provides exception expressions that exclude some common digit patterns, such as North American phone numbers, currencies, short date/time formats, US social security numbers, and others.

However, you can also exclude digit formats that are proprietary to your applications, such as custom screen identifiers or inventory part/SKU numbers. To exclude proprietary formats from the redaction process, you will need to add one or more regular expressions to the Custom Exception Expressions table. Literal strings (such as screen ids, copyright notices/dates, etc.) can also be specified here.

NOTE: Unlike the expressions that detect credit card PANs, the expressions for exclusions are applied to the input screen data before removal of non-digit data. These expressions should be specified “as they would appear” on the original host screen.

Examples of Custom Exclusions

Example 1: Using a regular expression

In the USA, postal (or ZIP) codes can follow two formats – 5 consecutive digits, or 5 consecutive digits followed by a hyphen and four additional digits. Typically, on a host application information screen, these codes are preceded and trailed by at least one space character. If we had a screen like this we could add the following expression to the Custom Exception Expression table to eliminate one potential source of “false positives” by excluding ZIP codes from redaction:

```
\s\d{5}([\-]\d{4})?\s
```

When we read this expression from left to right, it says “match a leading whitespace character, followed by 5 consecutive digits, and then match zero or one instance of a character group consisting of one hyphen followed by 4 additional digits, followed by a whitespace character.”

NOTE: Regular expression syntax sometimes requires the “escaping” of certain reserved characters. In this case, the hyphen must be escaped since it is specified within a character group (the text within the parentheses enclosed in square brackets above).

This expression would match strings like “ 88888 ”, “ 88888-7777 ”. Because leading and trailing whitespace is required, strings like “88888” and “88888-7777” would NOT be matched.

Omitting the \s in the expression above would result in matches where the digit patterns are embedded inside other longer strings.

For instance if the expression were modified to:

```
\d{5}([\-]\d{4})?
```

Unlike when we included the \s in the expression, the following text strings would result in a match:

```
Tst88888Str
```

```
Embedded88888-7777Text
```

```
Prefix88888
```

```
88888-7777Suffix
```

Example 2: Using an exact literal string:

Some of the host screens in an application have a copyright date string such as “Copyright © Acme Corporation 1990, 1992, 2004” that could cause a false positive. In this example, we will exclude the digits in that string from PAN redaction processing.

This can be solved by entering the exact literal string that is desired for exclusion. In this case enter this string in the exclusion expression table:

```
Copyright © Acme Corporation 1990, 1992, 2004
```

IMPORTANT: When developing custom regular expressions, it is highly recommended to use a regular expression development and test tool to ensure that the expression truly behaves as you intended. In other words, it matches ONLY what you intend to match and does not match text that you don’t want. One such freeware tool is the Espresso regular expression development tool that can currently be found at <http://www.ultrapico.com/Espresso.htm>. This tool requires registration to

use beyond an introductory trial period, but is currently free to use upon completion of registration. Also, there are many common patterns that have regular expression implementations “published” in the public domain. An internet search for “regular expression library” will turn up several sites that can be searched for pre-constructed regular expressions. One popular site is <http://www.regexlib.com>. These can be used as a starting point for your own expressions. Make sure that you thoroughly test expressions before they are deployed for Reflection.

Configure Information Privacy with Group policy

You can configure Information Privacy with the Installation Customization Tool or with Group Policy.

To set up Information Privacy with Group Policy

1. Copy the following files to the central store as follows:

Copy these files	To
ReflectionPCIDSS.admx and ReflectionWorkspace.admx in: ...\install_dir\Configuration\GroupPolicy\ ADMX	%systemroot%\PolicyDefinitions
ReflectionPCIDSS.adml and ReflectionWorkspace.adml in: ...\install_dir\Configuration\GroupPolicy\ ADMX\en-us	%systemroot%\PolicyDefinitions\ <locale>

2. Open the Group Policy Object Editor (`gpedit.msc`).
3. Under either the Computer Configuration or User Configuration branch, browse to **Administrative Templates | Reflection Workspace | Information Privacy**.
4. In the **Information Privacy** panel, select and edit the policy settings.

NOTE: If you want to include the default regular expressions used for Custom Detection Rules and Custom Exception Expressions, you must add these expressions through the Group Policy editor. For detailed instructions, see [Knowledge Base Article 7021396: Adding Regular Expressions for Custom Detection Rules and Custom Exception Expressions to Group Policy](https://support.microfocus.com/kb/doc.php?id=7021396) (<https://support.microfocus.com/kb/doc.php?id=7021396>).

Setting up Privacy Filters

You can set up privacy filters to protect personal information that is not credit card data. You can also use these filters together with the Information Privacy redaction to enhance protection of credit card data.

When to Use Privacy Filters

Privacy filters are useful when you need to meet one or more of the following requirements:

- ◆ You have certain patterns of data outside the realm of credit card formats that you would like to redact. For instance, you may need to redact US Social Security numbers, proprietary sensitive account numbers, motor vehicle registration or license identifiers, and the like.
- ◆ You need to specify credit card patterns that fall outside the range of 13-16 digit lengths. (You could use this approach along with any of the PAN detection methods.)
- ◆ You need to specify 13-16 digit custom formats when using a PAN detection method other than “Reflection PAN Detection.”

NOTE: If used in conjunction with the simple detection method based on preceding text, privacy filters do not honor the preceding text requirement.

When privacy filters detect a match, ALL of the non-whitespace characters are redacted. This differs slightly from PAN redaction, where only certain portions of the PAN are redacted according to the configuration setting **Portion of PAN to redact**.

Privacy filters are flexible and are easy to set up: The patterns can be any combination of alphanumeric “placeholders”, wildcard specifiers (meaning they match anything) or literal text and can be set up for any length of text. The patterns also follow a simpler syntax than regular expressions without the potentially confusing syntax and rules. The filter format should be familiar to Extra! X-Treme customers, and is supported in Reflection 2007 and later.


NOTE: Privacy Filters and PAN detection work together to keep sensitive information private. Privacy filters are applied after PAN detection, so keep in mind that portions of the host screen data may already be redacted by the time privacy filters are applied. This can prevent a configured filter from fully redacting data that would have “matched” had PAN redaction not been active.

To improve performance, do not duplicate existing PAN patterns in privacy filters.

How to set up Privacy Filters

Use the Information Privacy dialog box to set up privacy filters.

To set up Information Privacy Filters

- 1 From the Reflection File menu or the Reflection button  (if using the Office 2007 Look and Feel), select **Reflection Workspace Settings**.
- 2 Under Trust Center, click **Set Up Information Privacy**.
- 3 In the Information Privacy dialog box, specify the **Privacy Filters Redaction Rules** and **Privacy Filters** you want to use.
- 4 To “lock down” these settings, see [“Control Access to “Lock Down” Settings and Controls” on page 103](#).
- 5 To package this file for deployment, see [Package Configuration Files](#).

NOTE: Privacy filter settings are saved in the `PrivacyFilters.xml` file. All other Information for Privacy settings is saved in the `PCIDSS.settings` file. You can deploy these files to one of the following locations:

Location for a single user: `[AppDataFolder]\Micro Focus\Reflection\Desktop\v17.0`

Location for all users: `[CommonAppDataFolder]\Micro Focus\Reflection\Desktop\v17.0`

More About Information Privacy

PCI DSS Web Site

<https://www.pcisecuritystandards.org/> -- Download the current PCI DSS specification here. The Reflection implementation is based on version 2.0 of the specification, issued in October of 2010.

Tools for regular expressions

The Espresso regular expression development tool can be found at: <http://www.ultrapico.com/Espresso.htm>.

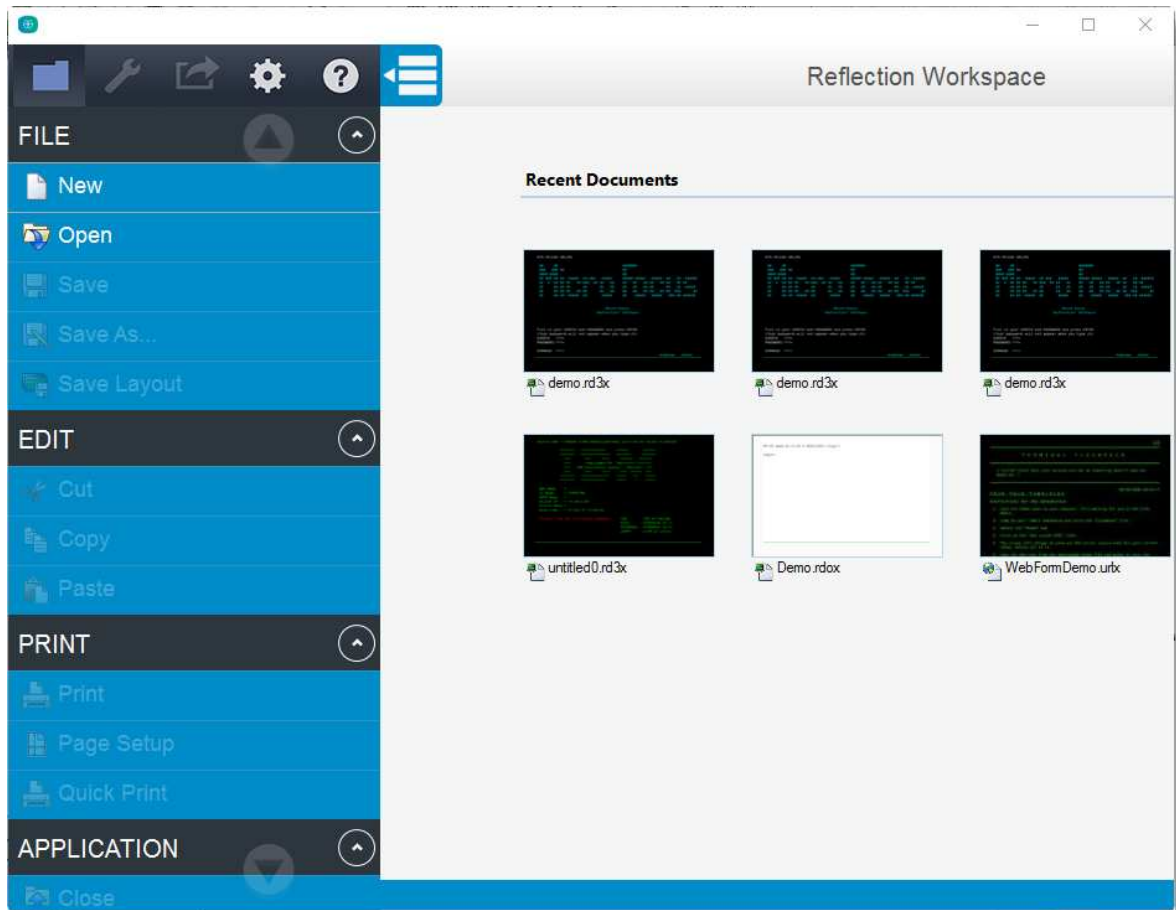
This site provides common patterns for regular expressions: <http://www.regexlib.com>.

Micro Focus expressly disclaims any responsibility for the availability or performance of any of the tools or sources suggested in this document.

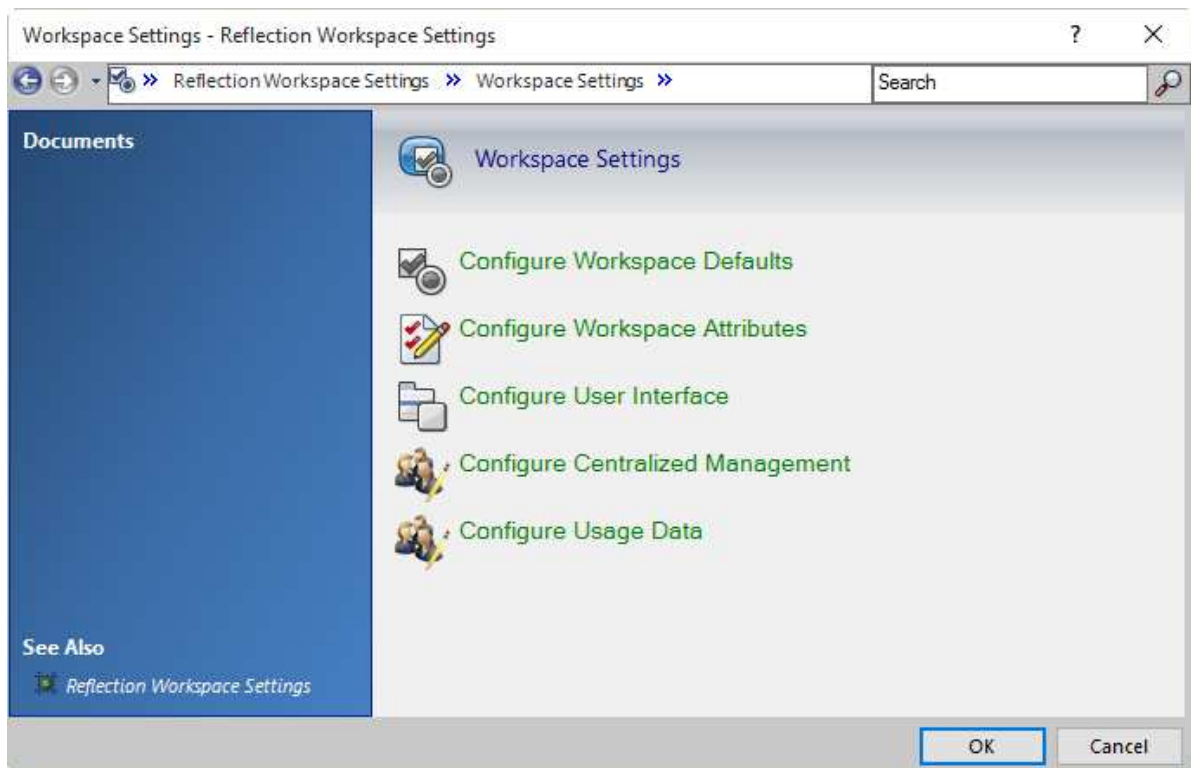
7 Customize the Reflection Workspace

You can customize the Reflection workspace to control its appearance and behavior.

You can change the basic functionality of the main Reflection window in a variety of ways. For example, you can specify startup macros and actions. You can also change the appearance of the window (for example, by setting Reflection to open in the TouchUx interface mode as shown below).



Workspace settings include all of the settings you can access from the Workspace Settings dialog box.



You can customize workspace settings in several different ways:


- ◆ [“Customize Workspace Settings Directly With Reflection” on page 94](#) to configure the settings in Reflection. Then save the settings in the application.access file. It’s easier to verify your settings with this approach but you’ll need to package the setting as shown in [“Package Sessions and Custom Settings Files” on page 116](#).
- ◆ [“Configure and Automatically Package Workspace Settings” on page 95](#), using the Installation Customization Tool to configure the settings. When you use this approach, the settings are automatically packaged in an MSI file.
- ◆ [“Update Only Specific Workspace Settings” on page 98](#) to change a specific workspace setting without changing the other settings on a user desktop.

[Walkthrough: Customize Reflection Appearance and Behavior](#) is an example that shows how to use the Installation Customization Tool to configure Reflection so that multiple sessions are displayed in separate windows instead of in a single window with tabs.

Customize Workspace Settings Directly With Reflection

To customize workspace settings from Reflection

- 1 Open the Workspace Settings dialog box.
The steps depend on your user interface.

User Interface Mode	Steps
Ribbon	On the File menu or the Reflection button  (if using the Office 2007 Look and Feel), choose Reflection Workspace Settings .
Reflection Browser	On the Reflection menu, choose Settings and then Reflection Workspace Settings .
Mobile UI	Tap the Gear icon and then select Reflection Workspace Settings .

2 Under Workspace Settings, select the type of workspace setting you want to configure:

Select this option	To
Configure Workspace Defaults	Configure the actions to perform when the Reflection workspace opens or closes and preferences for automatically saving session document files.
Configure Workspace Attributes	Configure options for logging, running remote sessions, and displaying Help. You can also specify the user data directory, in which session documents and other related files are saved.
Configure User Interface	Configure which type of user interface to use (Reflection provides four interfaces), its look and feel, and other user interface options.
Configure Centralized Management	Set up the workspace to access sessions that are centrally managed on a Micro Focus Management and Security Server.
Configure Usage Data	Choose whether to participate in the Product Experience Improvement program.

3 Change the settings as needed and then save them.

4 Close and reopen Reflection and then verify the settings.

5 To “lock down” these settings, see [“Control Access to “Lock Down” Settings and Controls” on page 103](#).

6 To package this file for deployment, see [Package Configuration Files](#).

NOTE: These settings are saved in the `Application.settings` file. You can deploy this file to one of the following locations:

Location for a single user: `[AppDataFolder]\Micro Focus\Reflection\Desktop\v17.0`

Location for all users: `[CommonAppDataFolder]\Micro Focus\Reflection\Desktop\v17.0`

To find out more about configuring workspace settings, see [How Do I?](#) in the Micro Focus Reflection Desktop Help.

Configure and Automatically Package Workspace Settings

When you use the Installation Customization Tool to configure workspace settings, the custom configuration files that include the settings are automatically added to an MSI.

CAUTION: Do not use the Installation Customization tool to configure Trust Center settings. This approach should be used only to configure settings in the Configure Workspace Defaults, Configure Workspace Attributes, Configure User Interface, Configure Centralized management, or Configure Usage Data dialog boxes.

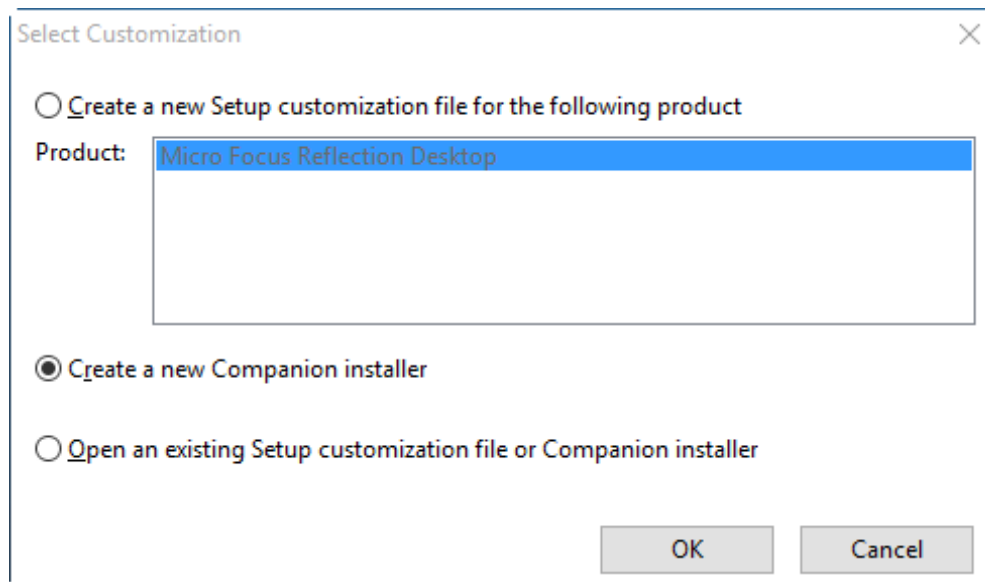
You'll probably want to configure these settings on your workstation first to make sure you get the results you want. Then reconfigure these settings as shown below.

To create a custom workspace with the Installation Customization Tool

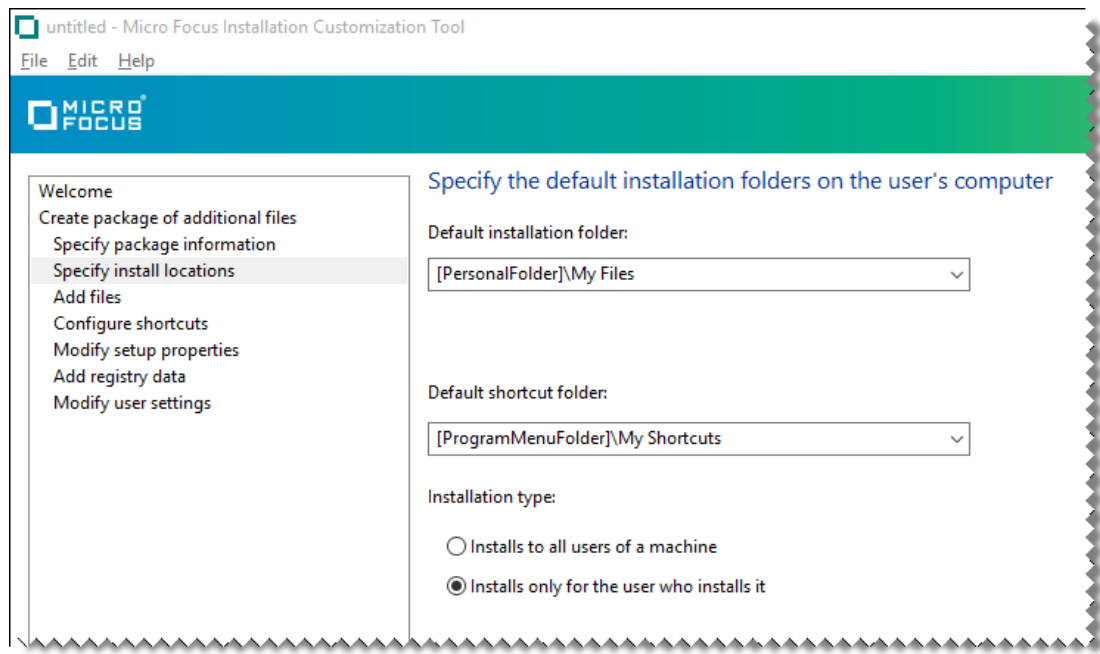
- 1 From your administrative installation point, open the Installation Customization Tool from a [shortcut \(page 38\)](#) or by typing the following command line:

```
<path_to_setup> \setup.exe /admin
```

- 2 From the **Select Customization** dialog box, select **Create a new Companion installer**, and then click **OK**.



- 3 From the navigation pane, click **Specify install locations** and choose whether to install the workspace settings for all users of a machine or only for specific users.



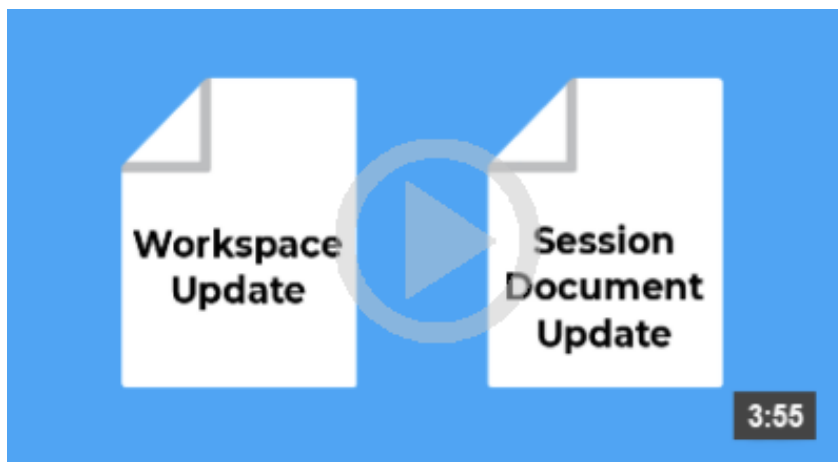
- 4 On the navigation pane, click **Modify user settings**.
- 5 From the list of **Application - Settings**, select **Reflection Desktop - Workspace Settings** and then click **Define**.
The **Reflection Workspace Settings** dialog box opens in a separate window.
- 6 Under **Workspace Settings**, click the type of setting to configure (for example, **Configure Workspace Defaults**).
- 7 Configure the settings just as you would if you opened the workspace settings dialog box from the product.
- 8 To lock down these settings, follow the instructions in [“Specify Access Using Permissions Manager with the Installation Customization Tool” on page 105](#).
- 9 Save the package (`.msi`) file and close the Installation Customization Tool. You can deploy the package file as it is, or you can edit it to add additional files.

NOTE: These settings are automatically saved in the `Application.settings` file. The package `.msi` file is automatically configured to deploy this file to one of the following locations, depending on which option you selected when you specified install locations.

- ♦ If you selected **Installs only for the user who installs it**, the file is deployed to:
`[AppDataFolder]\Micro Focus\Reflection\Desktop\v17.0`
 - ♦ If you selected **Installs to all users of a machine**, the file is deployed to:
`[CommonAppDataFolder]\Micro Focus\Reflection\Desktop\v17.0`
-

Update Only Specific Workspace Settings

You can change a specific workspace setting without changing the other settings on a user desktop. This allows you to change a setting after you deploy Reflection without the risk of overwriting other settings that users have configured.



Deploying Specific Settings

To deploy a specific setting, you'll use the Installation Customization Tool (ICT) to create and package an Update.settings file in an MSI that you can deploy to users. When Reflection starts, it applies the settings in this file to the workspace.

If you want to modify a setting that you have updated in this way or change it back to the default setting, you'll use the ICT to modify the Update.settings file in the MSI you deployed and then deploy the new version of the MSI to users.

NOTE: You can update only workspace settings that are set in the Application.Settings file. You cannot update Information Privacy or Access file settings.

To create and package a workspace settings update

- 1 On a workstation on which you have installed Reflection, open the Installation Customization Tool from a desktop shortcut.
- 2 From the Select Customization dialog box, select **Create a new Companion installer**.
- 3 From the navigation pane, click **Specify install locations**.
- 4 Under **Installation type**, select how to install the configuration file (the Update.settings file) that specifies your workspace setting.
 - ♦ **Installs to all users of a machine** installs the file in a common folder so that the settings apply to all users of the machine.
 - ♦ **Installs only for the user who installs it** installs the file in a personal folder so that the settings apply only to a single user.
- 5 On the navigation pane, click **Modify user settings**.
- 6 In the Make changes to user settings... panel, under **Application – Settings**, select **Reflection Desktop - Update Workspace Settings** and then click **Define**.

- 7 Configure the settings you want to change and then close the Workspace Settings dialog box.
- 8 Save the companion file and close the Installation Customization Tool.

NOTE: When you deploy the new version of the MSI to users, the new Update.settings file is installed with the setting you modified.

Walkthrough: Customize Reflection Appearance and Behavior

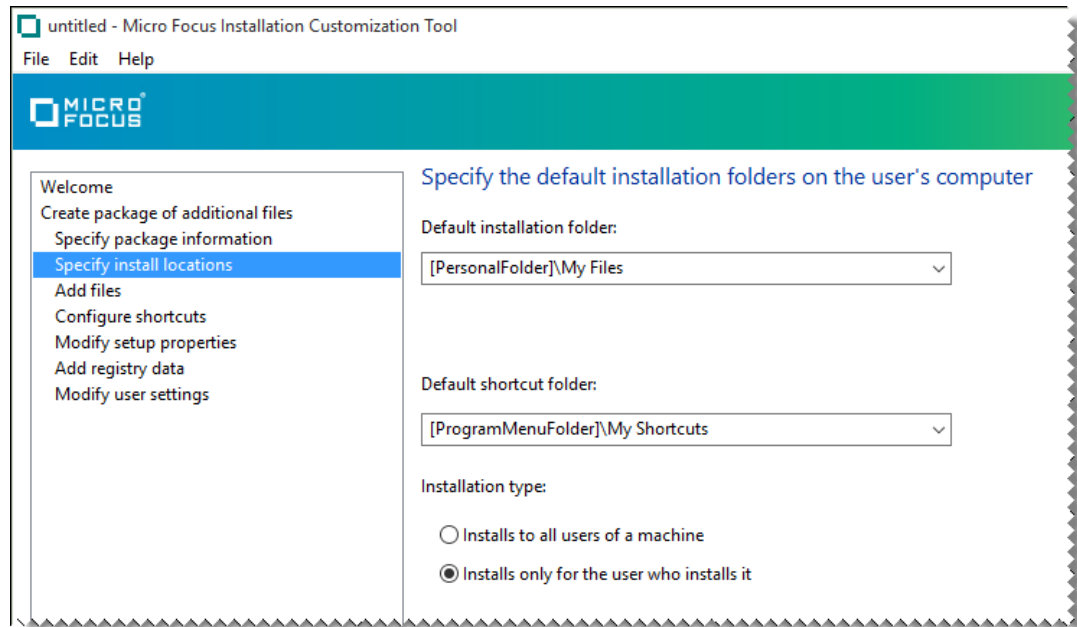
You can change the appearance and basic functionality of the main Reflection window in a variety of ways. You can also specify startup macros, startup actions, whether to open Reflection with the Ribbon open or closed, and other options. You will use the Installation Customization Tool (ICT) to configure most of the custom workspace settings.

The following example shows how to configure a workspace so that multiple sessions are displayed in separate windows instead of in a single window with tabs (the default). It also shows how to control access to (“lock down”) this setting.

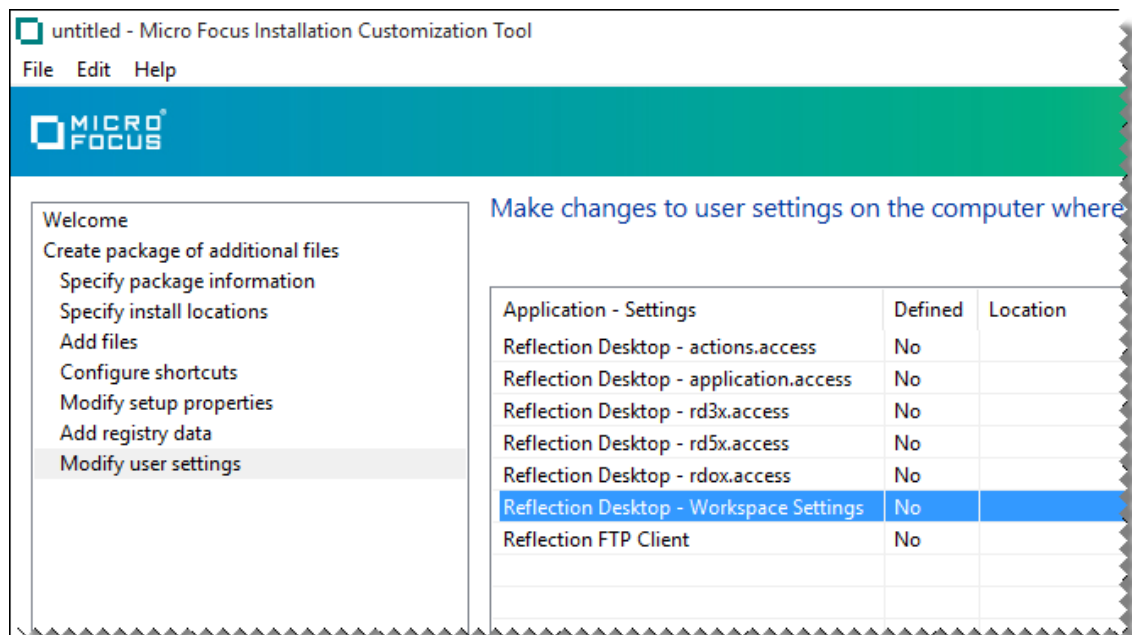
To create a custom workspace

- 1 On a workstation on which you have installed Reflection, open the Installation Customization Tool from a desktop shortcut.
- 2 From the Select Customization dialog box, select **Create a new Companion installer**.
- 3 From the navigation pane, click **Specify install locations**.
- 4 Under **Installation type**, select how to install the configuration file (the `Application.settings` file) that specifies your workspace settings.
 - ♦ **Installs to all users of a machine** installs the file in a common folder so that the settings apply to all users of the machine.

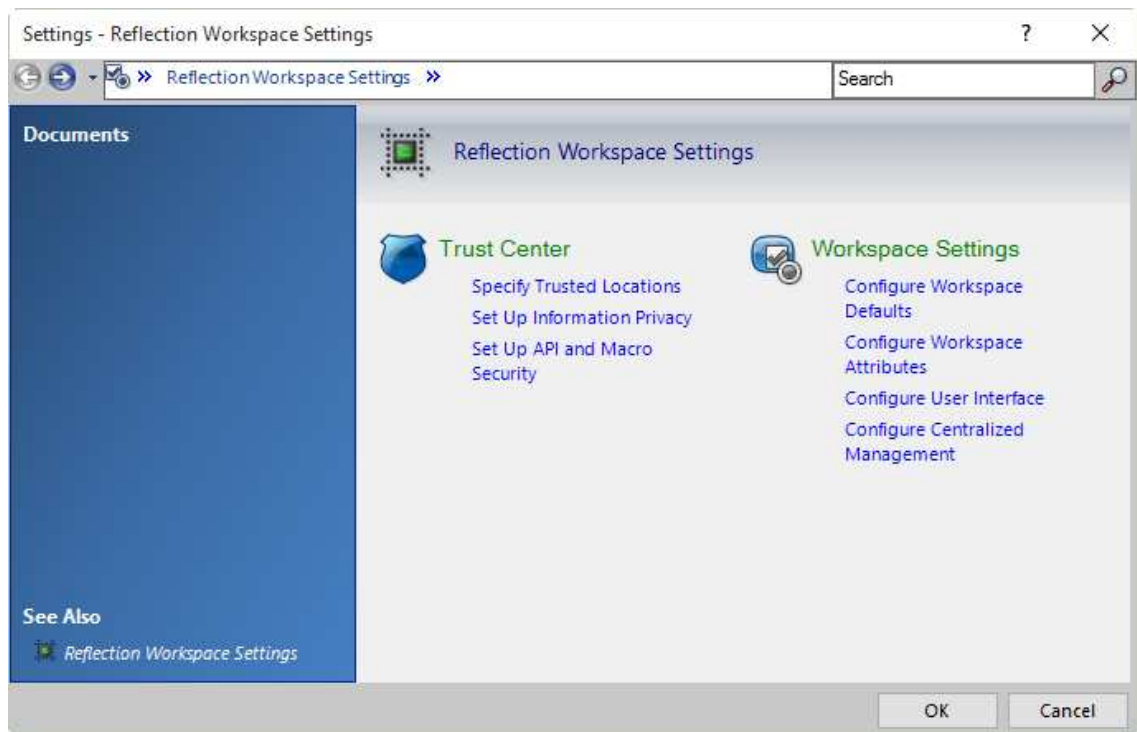
- ◆ **Installs only for the user who installs it** installs the file in a personal folder so that the settings apply only to a single user.



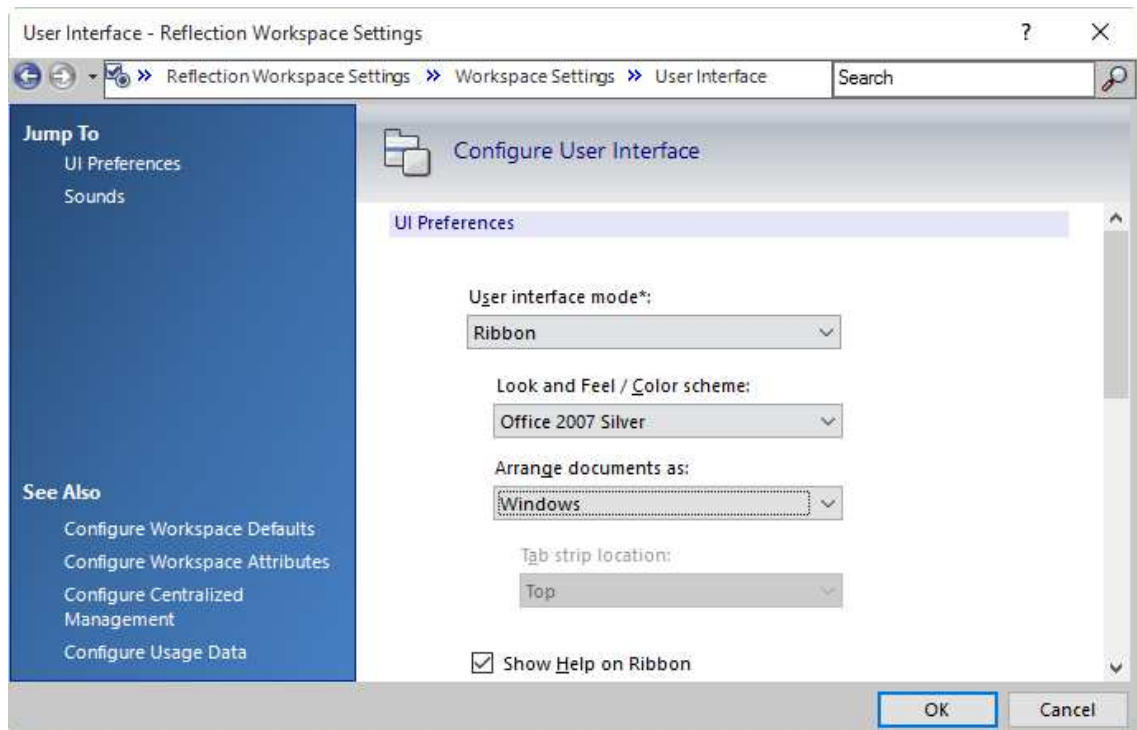
- 5 On the navigation pane, click **Modify user settings**.
- 6 In the Make changes to user settings... panel, under **Application – Settings**, select **Reflection Desktop -Workspace Settings** and then click **Define**.



- 7 In the Reflection Workspace Settings dialog box, under **Workspace Settings**, click **Configure User Interface**.



- 8 In the Configure Workspace Settings dialog box, under UI Preferences, in the Arrange documents as list, select **Windows**.



- 9 Close the Reflection workspace.
- 10 In the Make changes to user settings... panel, under **Application – Settings**, select **Reflection Desktop -application.access** and then click **Define**.

- 11 In the Permissions Manage tool, In the **Groups** list, select **UX Configuration** and then set the **DocumentLayout** item to **Restricted**.
- 12 Save the companion file and close the Installation Customization Tool.

NOTE: These settings are automatically saved in the `Application.settings` file. The package `.msi` file is automatically configured to deploy this file to one of the following locations, depending on which option you selected when you specified install locations.

- ◆ If you selected **Installs only for the user who installs it**, the file is deployed to:
`[AppDataFolder]\Micro Focus\Reflection\Desktop\v17.0`
 - ◆ If you selected **Installs to all users of a machine**, the file is deployed to:
`[CommonAppDataFolder]\Micro Focus\Reflection\Desktop\v17.0`
-

8

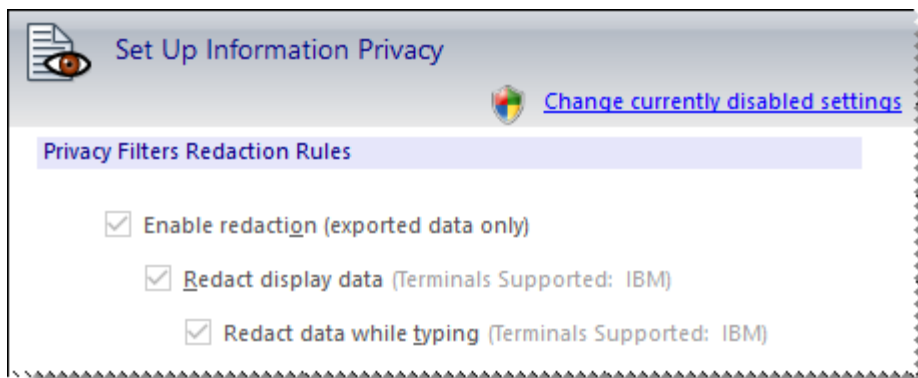
Control Access to “Lock Down” Settings and Controls


You can restrict access to almost any of the Reflection settings or controls. For example, you can prevent users from changing the host address that a session connects to, or from running a macro. This allows you to simplify support requirements and resolve security concerns.

NOTE: You can restrict access to Reflection controls only on systems on which the Windows User Access Control (UAC) is enabled. If the UAC is disabled, the configuration settings that restrict access do not apply.

Restricting Access to Settings

As shown below, the restricted settings are grayed out. The security shield and the **Change currently disabled settings** link indicate administrative access is required to change them. Users cannot change these options unless they elevate their access level to administrator.



The security shield  and the **Change currently disabled settings** link indicate that settings are disabled and administrative access is required to enable them.

NOTE: The customized files that include these settings can be bundled in a companion installer package (.msi) and deployed with the installation or later.

Restricting Access to Controls

You can prevent the user from using a particular control by restricting access to the control. In the following example, the security shield over the Auto Complete icon indicates that the "action" associated with this option is "restricted".



Tools for Restricting Access

You can restrict access to Reflection controls and settings with the Permissions Manager tool or through Microsoft Windows Group Policy settings.

- [“Control Access to Settings and Controls with Reflection Administrative Tools” on page 104](#). The Permissions Manager tool is a Reflection administrative tool that is automatically installed on your administrative installation point when you create an administrative install image. It can be accessed through the Installation Customization Tool or independently. With this tool, you can create and deploy .access files that restrict access to controls or settings.
- [“Control Access to Settings and Controls with Microsoft Group Policy” on page 111](#). Microsoft Group Policy provides another method for restricting access. If you are using Group Policy, you can import the Reflection ADMX files into your environment and modify the settings in the Group Policy Editor.

[“Walkthrough: Restrict Access to Settings and Controls” on page 109](#) is an example that shows how to use the Permissions Manager administrative tool to restrict access to the Auto Complete control.

Control Access to Settings and Controls with Reflection Administrative Tools

To prevent a user from changing a setting, you set the permission level for that setting or control to “Restricted.” When a setting is restricted, administrative access is required to change the setting. For example, you could restrict the user’s ability to modify security settings.

NOTE: You can restrict access to Reflection controls only on systems on which the Windows User Access Control (UAC) is enabled. If the UAC is disabled, the configuration settings that restrict access do not apply.

The following access file templates are distributed with Reflection Desktop:

This File	Controls access to...
actions.access	Actions (for example, Auto Complete)
application.access	Reflection workspace settings
rd3x.access	3270 terminal settings
rd5x.access	5250 terminal settings
rdox.access	VT terminal settings
security.access	TLS and PKI settings

Individual permissions are merged in the following order (from highest to lowest):

- ♦ Group Policy – user
- ♦ Group Policy – machine
- ♦ Local permissions file (.access)

Deploying local permissions (.access) files

Use the Reflection Permissions Manager tool to set local permissions and save them in .access files that you can deploy.

You can deploy user-specific access settings for any type of .access files. To deploy user-specific files, install the .access files to [AppDataFolder]\Micro Focus\Reflection\Desktop\v17.0.

You can install some types of access configuration (actions.access, application.access, and security.access files) for all users of the system. To deploy these settings for all users, install these files in [CommonAppDataFolder]\Micro Focus\Reflection\Desktop\v17.0.

NOTE: Settings files in the [CommonAppDataFolder] location are copied to the [AppDataFolder] location when the user opens the Workspace.

You can set permissions and create .access files by using the Permissions Manager with or without the Installation Customization Tool. When you use Permissions Manager with this tool, the tool automatically determines the correct location in which to install the required files. When you open Permissions manager and create .access files outside of this tool, you'll need to be sure the files are installed in the correct directory.

Specify Access Using Permissions Manager with the Installation Customization Tool

You can open Permissions Manager from the Installation Customization Tool to lock down access. When you use this approach, the resulting .access files are automatically added to the correct directory in the package (MSI file).

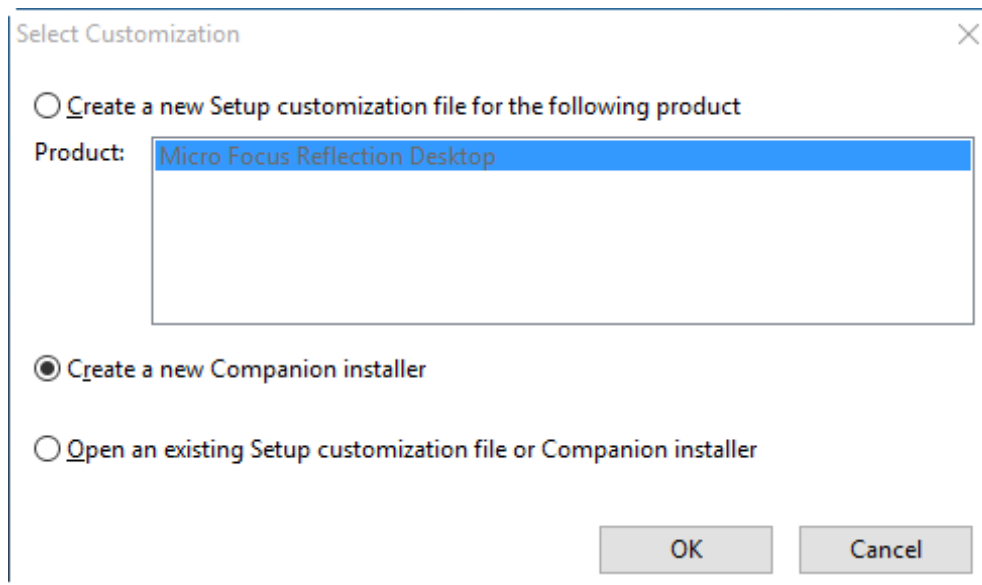
NOTE: These files are not saved to your local machine. They are saved only in your MSI database. To make changes to these files, you will need to use the same approach to edit them as you used to create them. You'll need to open the MSI file in the Installation Customization Tool and then open Permissions Manager from the tool.

To set user and group access with the Installation Customization Tool

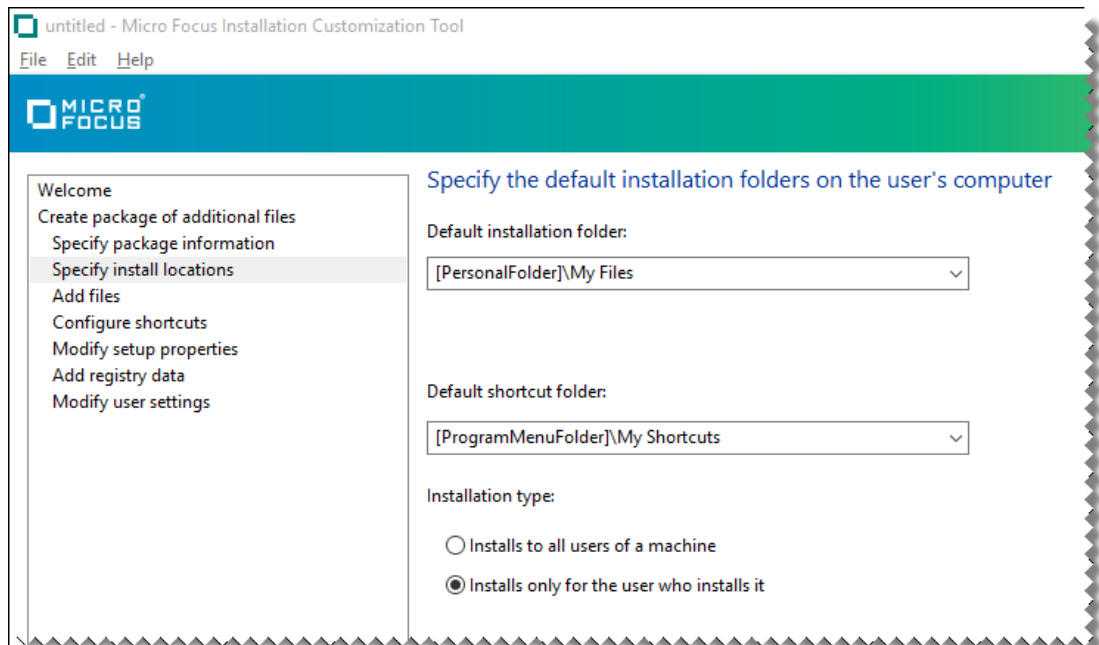
- 1 From your administrative installation point, open the Installation Customization Tool from a [shortcut \(page 38\)](#) or by typing the following command line:

```
<path_to_setup> \setup.exe /admin
```

- 2 In the **Select Customization** dialog box, select **Create a new Companion installer**.



- 3 On the left pane, select **Specify install locations**.



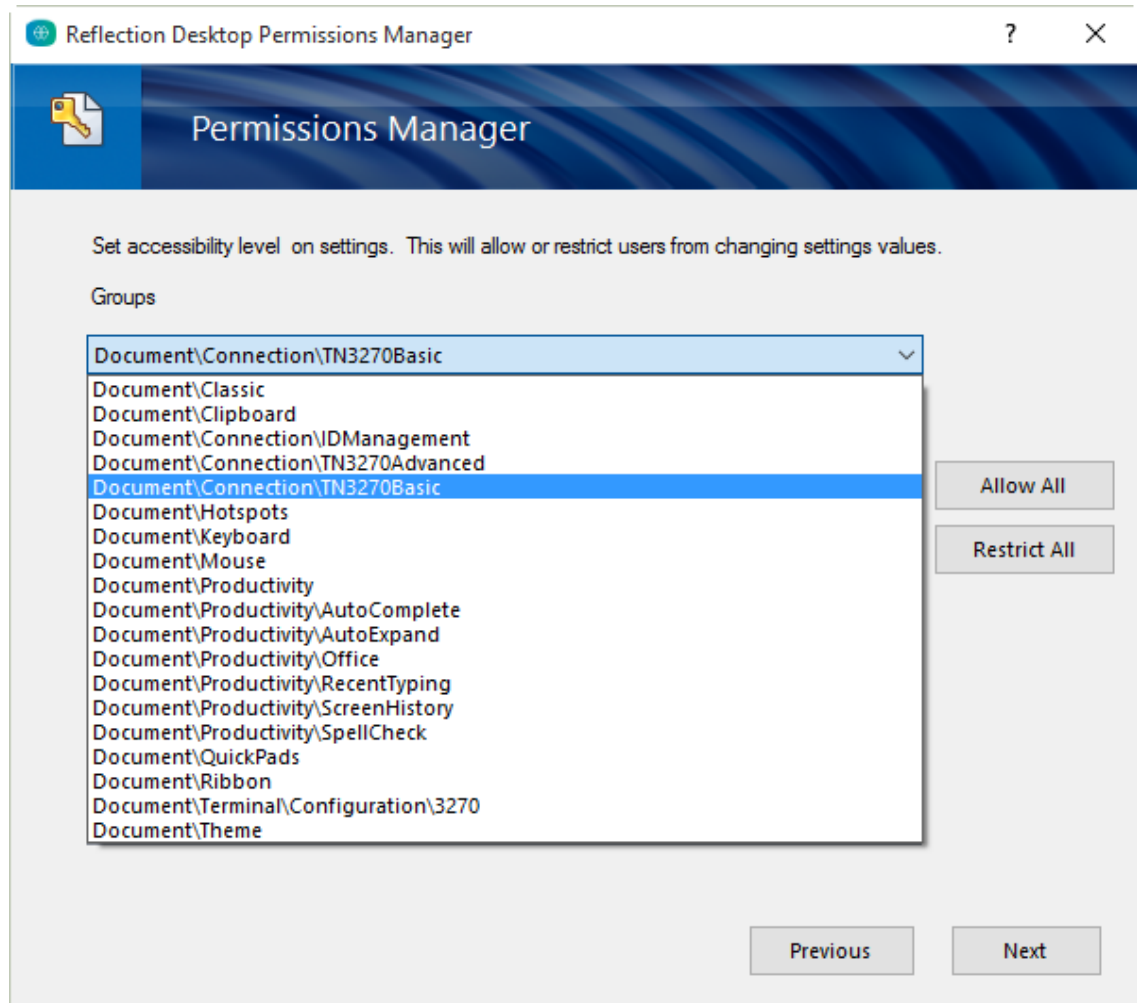
- 4 Under **Installation type**, select whether to install the settings to all users of a machine or only for the user who installs it.

NOTE: Only actions.access, application.access, and security.access files can be deployed to all users.

- 5 In the left pane, select **Modify user settings**.

NOTE: Under **Application - Settings**, the Permissions Manager displays groups of configurable items. These items are listed by their internal names, which may not exactly match the user interface item. The item's **Accessibility** indicates whether the user can configure the item (**Full**) or if administrator assistance is required to configure the item (**Restricted**).

- 6 In the **Make changes to user settings** pane, select one of the .access options and click **Define**.
- 7 In Permissions Manager, under **Groups**, select the group of settings you want to control access to (for example, Document\Connection\TN3270Basic).



- 8 In the **Items** box, in the **Accessibility** column for the item (or items) you want to restrict, click **Full** and then select **Restricted** from the drop down menu.

NOTE: The Accessibility drop down menu includes three items:

- ◆ **Full:** All users can configure the item.
 - ◆ **Restricted:** Only administrators of the system can configure the item. These items have the Windows access shield added to their icons.
 - ◆ **Read-only:** No users of the system can configure the item. These items are grayed out.
-

- 9 Under **Additional security options**, select how to control session file encryption:

To do this

Configure all sessions so that users can open only encrypted display session files.

Configure all sessions so that users can save a display session only if it is encrypted.

Select

User can open only encrypted session files

User can save only encrypted session files

10 From the **File** menu, choose **Save As** and save the companion installer package.

If you selected **Installs only for the user who installs it** when you specified install locations, the companion installer package automatically specifies to deploy this `.access` file to `[AppDataFolder]\Micro Focus\Reflection\Desktop\v17.0`.

If you selected **Installs to all users of a machine**, it specifies to deploy the `.access` file to `[CommonAppDataFolder]\Micro Focus\Reflection\Desktop\v17.0`.

NOTE

- ◆ Make sure to set file access rights on `.access` files to prevent users from deleting, replacing, or editing them.
- ◆ To deploy files to the `[AppDataFolder]\Micro Focus\Reflection\Desktop\v17.0` folder, you will need to use a deployment tool that allows you to install the companion installer package as the user.
- ◆ When accessing a setting via an API, such as executing a macro, a setting with restricted access cannot be modified. (When attempting to set a restricted setting via an API, an error is logged.)
- ◆ Setting session encryption options in an `*.access` file affects only the associated session type. For example, limiting users to opening only encrypted session files in `rd3x.access` affects only 3270 terminal session files, and not 5250 session files.

Specify Access Using Permissions Manager

To prevent a user from changing a setting, you set the permission level for that setting or control to “Restricted.” When a setting is restricted, administrative access is required to change the setting. For example, you could restrict the user’s ability to modify security settings.

You can lock down access by running Permissions Manager (without using the Installation Customization Tool) to edit `.access` files. If you use this approach, be sure to deploy the customized `.access` files to the correct directory.

NOTE: Important: Be sure to set file access rights on `.access` files that you deploy to prevent users from deleting, replacing, or editing them.

To set access with Permissions Manager

- 1 On a workstation on which you have installed Reflection, log on as administrator and in the Reflection Desktop **install folder**, run `AccessConfig.exe`.
- 2 When prompted to create a new permission file, or edit an existing one, choose **Create new permission file**.

- 3 When prompted with a list of access file templates, choose the type of permission file to create.
- 4 Under **Groups**, select the type of setting to control access to (for example, the **Document\Connection\TN3270Basic** group).
- 5 In the **Items** box, in the **Accessibility** field for the item (or items) you want to restrict, click **Full** and then select **Restricted** from the drop down menu.
- 6 If you are configuring `rd3x.access`, `rd5x.access`, or `rdox.access` files, under **Additional security options**, select how to control session file encryption:

To do this

Select

Configure all sessions so that users can open only encrypted display session files.

User can open only encrypted session files

Configure all sessions so that users can save a display session only if it is encrypted.

User can save only encrypted session files

- 7 Be sure to deploy the `.access` files to the correct directory as shown in “[Package Sessions and Custom Settings Files](#)” on page 116:

To deploy settings that are user-specific, deploy the `.access` files to `[AppDataFolder]\Micro Focus\Reflection\Desktop\v17.0`.

To deploy settings for all users of a machine, deploy the `.access` files to `[CommonAppDataFolder]\Micro Focus\Reflection\Desktop\v17.0`.

NOTE: Only `actions.access`, `application.access`, and `security.access` files can be deployed to all users.

IMPORTANT

- ♦ To deploy files to the `[AppDataFolder]` folder, your deployment tool must allow you to install the companion installer package as the user.
 - ♦ Setting session encryption options in an `.access` file affects only the associated session type. For example, limiting users to opening only encrypted session files in `rd3x.access` affects only 3270 terminal session files, and not 5250 session files.
 - ♦ When accessing a setting via an API, such as executing a macro, a setting with restricted access cannot be modified. (When attempting to set a restricted setting via an API, an error is logged.)
-

Walkthrough: Restrict Access to Settings and Controls

The following example shows how to use the Permissions Manager tool to restrict access to the Auto Complete control.

To use Permissions Manager to restrict access to Auto Complete

- 1 On a workstation on which you have installed Reflection, open the Installation Customization Tool from a desktop shortcut or from a command line as follows:

path_to_setup\setup.exe /admin

- 2 In the Select Customization dialog box, choose to either create a new companion installer or edit an existing one.

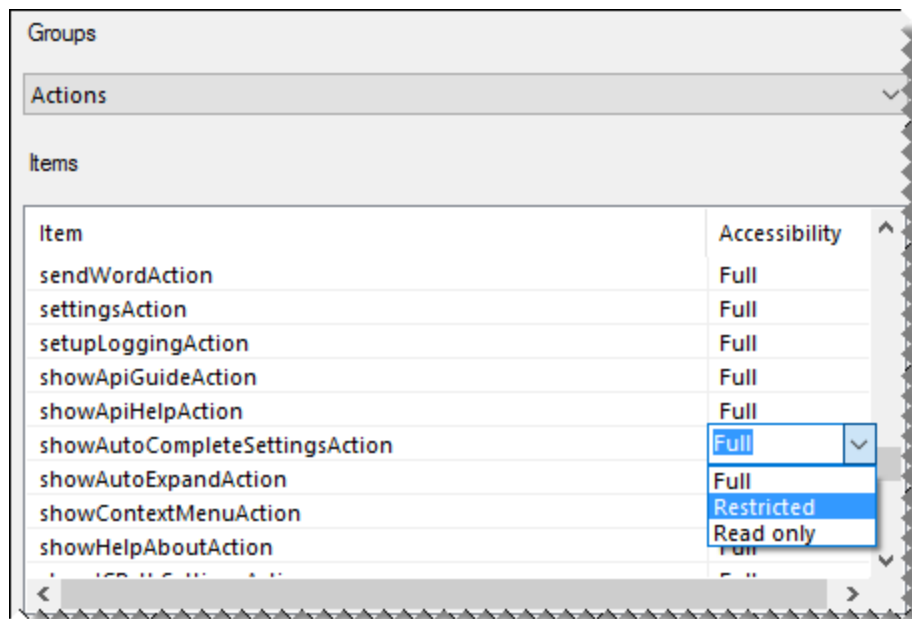
The Installation Customization Tool opens in the mode used to create or edit companion installer packages (MSI files). This mode is also used to open the Permissions Manager tool that is used to restrict access.

- 3 From the navigation pane, click **Specify install locations**.
- 4 Under **Installation type**, select **Installs only for the user who installs it**.
- 5 From the navigation pane, click **Modify user settings**.
- 6 Under **Application – Settings**, select **Reflection Desktop – actions.access** and then click **Define**.

NOTE: The access file templates in the **Application-Settings** list are grouped by function. Features such as Auto Complete are in the **actions.access** template. Workspace settings are in the **application.access** template. Session Settings are in the **rd3x.access**, **rd5x.access**, and **rdox.access** terminal templates.

The Permissions Manager Tool opens in a separate window. This tool displays all of the controls (actions) and settings that you can restrict.

- 7 In the items list, scroll down to **showAutoCompleteSettingsAction**. Then, under **Accessibility**, on the drop-down menu, choose **Restricted**.



- 8 Click **Next**, accept the default values, and then click **Finish**.

The settings are automatically saved in the **actions.access** file and the companion installer package is automatically configured to deploy this file to **[AppDataFolder]\Micro Focus\Reflection\Desktop\v17.0**

- 9 From the ICT File menu, choose **Save As** and save the companion installer package file on the administrative installation point.

Control Access to Settings and Controls with Microsoft Group Policy

As an administrator, you can limit users' ability to modify their workspace or session documents by setting permissions from the Microsoft Group Policy Management Console using group policy templates.

NOTE: You can restrict access to Reflection controls only on systems on which the Windows User Access Control (UAC) is enabled. If the UAC is disabled, the configuration settings that restrict access do not apply.

Reflection installs a set of group policy templates (ADMX files) to the following directory:

```
\Program Files\Micro Focus\Reflection\Configuration\GroupPolicy
```

NOTE: To use this feature, you must be running Windows 7, Windows 8.1, or later on an administrative machine. For more information about managing group policy, see [Managing Group Policy ADMX Files Step-by-Step Guide \(https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-vista/cc709647\(v=ws.10\)\)](https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-vista/cc709647(v=ws.10)).

ADMX files

ADMX files are divided into language-neutral files (.admx) and language-specific resource files (.adml), available to all Group Policy administrators. These factors allow Group Policy tools to adjust their UI according to the administrator's configured language.

Reflection `setup.exe` installs ADMX files to:

```
... \install_dir\Configuration\GroupPolicy\ADMX
```

It installs ADML files to the following directory:

```
... \install_dir \Configuration\GroupPolicy\ADMX\en-us
```

Reflection provides the following ADMX Group Policy files. Each of these files has a corresponding ADML language file.

This file	Controls access to
ACTIONS.admx	Actions
APPLICATION.admx	Reflection Workspace
RD3X.admx	3270 terminal settings
RD5X.admx	5250 terminal settings
RDOX.admx	VT terminal settings
ReflectionWorkspace.admx	Root-level ADMX file
ReflectionSecurity.admx	TLS and PKI settings

NOTE: This directory also includes the `ReflectionPCIDSS.admx` file. This file is used to configure information privacy through Group Policy and is not used to control access.

Install Group Policy Templates

Before you deploy group policy definitions, set and test them on a local test machine.

To deploy ADMX files on a local test machine

- 1 Copy the `.adm` files from `... \install_dir \Configuration \GroupPolicy \ADMX` to the central store (`%systemroot%\PolicyDefinitions`)
- 2 Copy all required locale `.adml` files to: `%systemroot%\PolicyDefinitions\<locale>`
- 3 Open the Group Policy Object Editor (`gpedit.msc`)
- 4 Under either **Computer Configuration** or **User Configuration**, browse to **Administrative Templates** | **Reflection Desktop** or **Administrative Templates** | **Reflection Security**.
- 5 In the Group Policy Management Editor, navigate to the setting or feature you want to configure.
- 6 Enable the Group Policy settings you want to restrict access to.

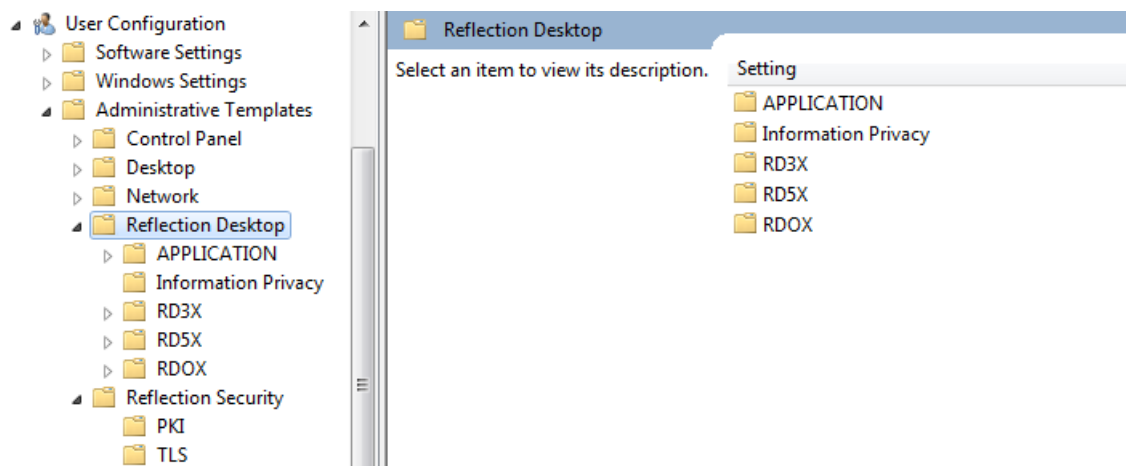
NOTE: For more about using ADMX files to set group policy, see [Managing Group Policy ADMX Files Step-by-Step Guide](http://technet.microsoft.com/en-us/library/cc709647(v=ws.10).aspx) ([http://technet.microsoft.com/en-us/library/cc709647\(v=ws.10\).aspx](http://technet.microsoft.com/en-us/library/cc709647(v=ws.10).aspx)).

Set Access with Group Policy

To set access with Group Policy Object Editor

- 1 In the Group Policy Management Editor, navigate to the setting or feature you want to configure.

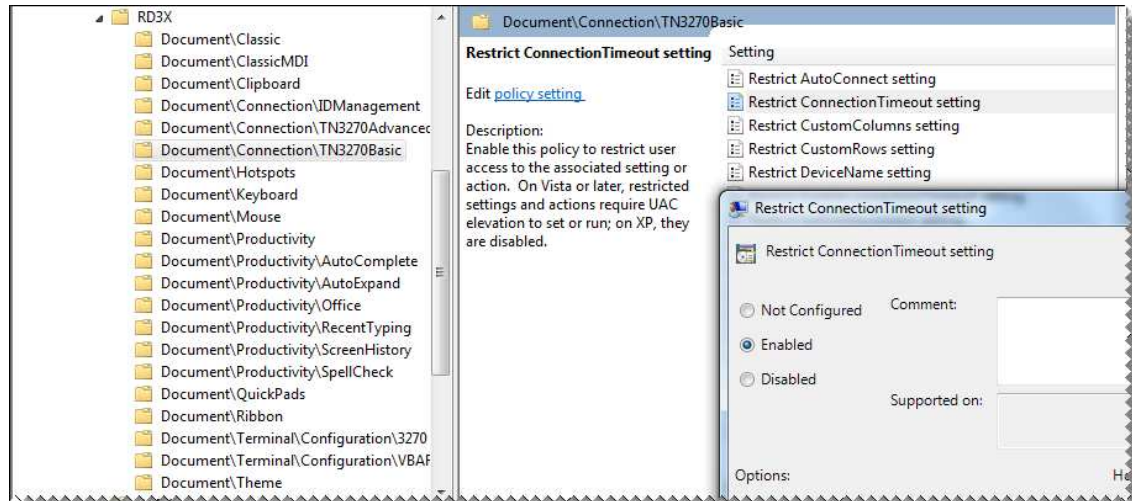
The following example shows all shipping ADMX files loaded into the GPO Editor under User Configuration. Group Policies can be set at the machine (Computer Configuration) or user (User Configuration) levels.



- 2 Enable the Group Policy settings you want to use.

The following example shows the following:

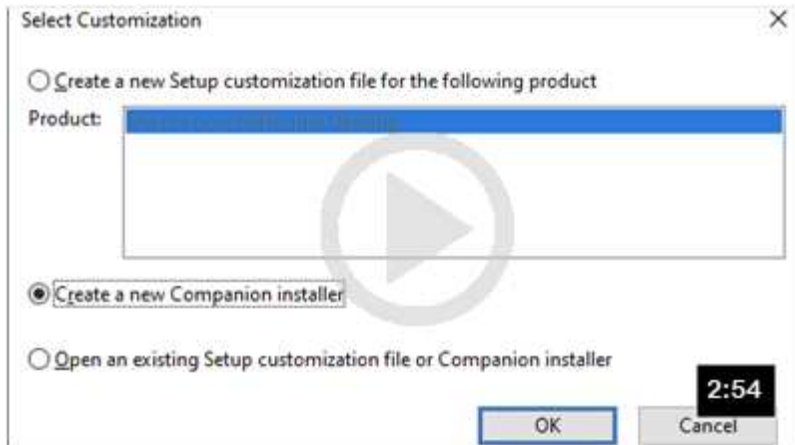
- ◆ The current node is the RD3X Document\Connection\TN3270Basic group.
- ◆ All the settings for this group are listed in the right-hand panel.
- ◆ The Restrict ConnectionTimeout setting policy is **Enabled**. This setting for 3270 display sessions is restricted.



Registry keys are added when policy settings are **Enabled**. These keys remain in the registry when policy settings are **Disabled**. No key is present when policy settings are **Not Configured**.

9 Package Configuration Files

Create a companion installer package (also called a "companion database") to install any custom files you created when you customized Reflection. A companion installer package is a standalone MSI file that is independent of any Reflection installer package.



Packaging Configuration Files

You can deploy companion installer packages separately or you can add them to a customized installation, so that the companion installer packages are automatically installed when the Reflection installer package completes.

You can also create and install packages at any time after the initial installation. Because companion installer packages are installed independently of Reflection, you can upgrade the product without removing these support files and you can deploy additional support files without re-installing the product.

If you support several business units that require their own customized configuration files, you can create a companion installer package for each business unit.

Companion installer packages have no built-in user interface except a standard progress bar. They are displayed as independent entries in the Windows **Add\Remove Programs** list and can be installed or uninstalled independently of Reflection.

If you use centralized management, you can also use [“Use Central Management to Deploy MSI Packages”](#) on page 153.

This article	Describes
Package Sessions and Custom Settings Files	How to bundle configuration files into Microsoft MSI files that you can deploy using standard Windows deployment tools.

This article	Describes
Create and Automatically Package Workspace and Access file Settings	How you can use the Installation Customization Tool to create and automatically package workspace settings that customize the Reflection workspace and .access files that lock down Reflection.
Walkthrough: Create a Package with the Installation Customization Tool	An example that shows how to create and deploy a companion installer package to install a keyboard map file and a session document file.
Customized Files that Must be Deployed to Specific Locations	Directory locations for custom configuration files that you create when you customize Reflection sessions or other settings. Reflection cannot find these files unless they are in these specific locations.
Other Customized Files and Supported Files from Other Products	Customized files that Reflection doesn't look for in the user data directory, other supported files that you would most likely not deploy, and files that can be imported from Extra!, legacy Reflection, and other products.

Package Sessions and Custom Settings Files

After you customize Reflection, bundle the customized session or settings files into a deployable companion installer package (.msi).

To create a companion installer package

- 1 Note the locations of custom configuration files you've created.
- 2 From your administrative installation point, open the Installation Customization Tool from a shortcut or by typing the following command line:


```
< path_to_setup > \setup.exe /admin
```
- 3 In the **Select Customization** dialog box, open the companion installer view:
 - ◆ If you created a companion installer package earlier, select **Open an existing Setup file or Companion installer**, click **OK**, and then browse to select the file.
 - ◆ If you have not created a companion installer package, select **Create a new Companion installer**.
- 4 From the navigation pane, select **Specify package information**.
- 5 In the **Add/Remove name** box and the **Organization name** box, type a name for the installation and the publisher that you want to be displayed in the Microsoft Windows **Uninstall or change a program** (**Add or remove programs**) list.
- 6 From the navigation pane, click **Specify install locations**, and specify the **Installation type**. Select whether to install the files for all users or for only one user:
 - ◆ **Installs to all users of a machine** makes files available for every user who logs onto the computer. Use this option for settings files, macros, and other configuration files that you want to be available to all users.
 - ◆ **Installs only for the user who installs it** makes files available only for the user who installs them.

- 7 In the **Default installation folder** list, select the folder in which to install the files. (Files are deployed to this folder unless you specify another folder when you add a file.)

NOTE: The folders available in this list depend on which Installation type you chose. Installation Type folder options specify the access for the files (after the installation). These options affect only the folders to which you can install — you cannot change these options after you add a file.

- 8 In the **Default shortcut folder** list, select the folder in which to install program shortcuts. (Shortcuts are deployed to this folder unless you specify another folder when you add a file.)
 - 9 From the navigation pane, select **Add files** and then click **Add** and browse to the files you want to include.
 - 10 Specify the destination directory for each file as follows:
 - 10a In the table of files that you added, select the file.
 - 10b In the **Add files to** list (at the bottom of the panel), enter the destination directory. (You can choose a location from the list or edit the path by typing. Use the [“Customized Files that Must be Deployed to Specific Locations” on page 119](#) table as a guide for where to install the custom files you have created.)
-
- NOTE:** The folder location you add must already exist on all target computers.
-
- 10c Click **Update**. (The destination directory for the file is displayed in the table’s Location column.)
- 11 If you want to create a shortcut for the file, select **Include shortcut** and then click **Update**.
 - 12 From the **Filemenu**, save the `.msi` file on the administrative installation point.

Create and Automatically Package Workspace and Access file Settings

You can use the Installation Customization Tool to create and automatically package workspace configuration files and access security files that lock down Reflection. When you use this approach to customize and deploy these files, they are automatically added to the correct directory in the package (MSI file). For more about automatically packaging these files, see:

[“Configure and Automatically Package Workspace Settings” on page 95](#)

[“Specify Access Using Permissions Manager with the Installation Customization Tool” on page 105](#)

Walkthrough: Create a Package with the Installation Customization Tool

The following example shows how to create and deploy a companion installer package to install a keyboard map file and a session document file.

To create a companion installer package MSI file that includes configuration files

- 1 Make sure you know where the files you want to add are located. For this example, we are packaging the files in the following locations:

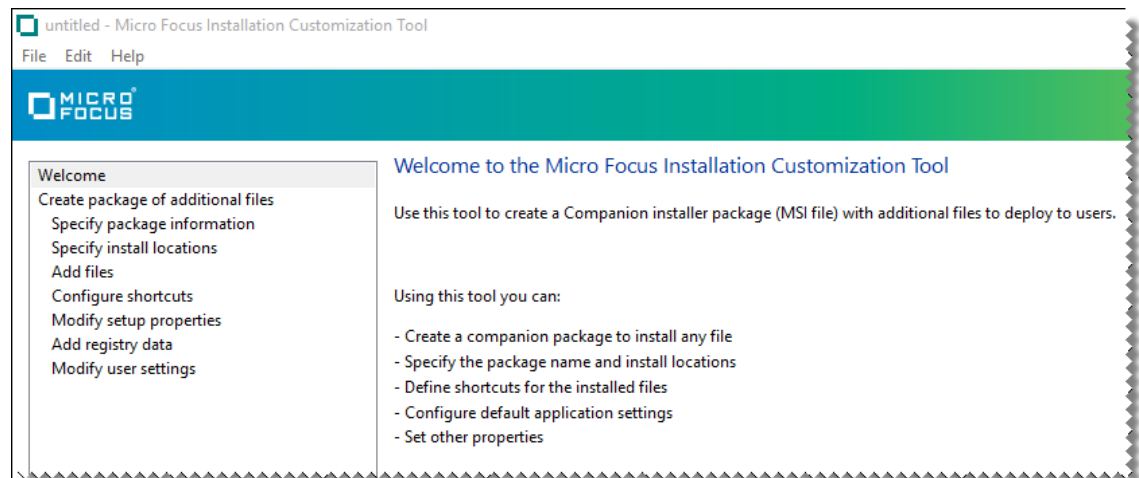
Add this file	In this directory
mySession.rd3x	C:\Users\yourUserName\Documents\Micro Focus\Reflection
myKeyboardMap.xkb	C:\Users\yourUserName\Documents\Micro Focus\Reflection\Keyboard Maps

- On your administrative workstation, open the Installation Customization Tool from a desktop shortcut or from a command line as follows:

```
path_to_setup\setup.exe /admin
```

- In the Select Customization dialog box, choose **Create a new Companion installer**.

The Installation Customization Tool opens in the mode used to create companion installer packages.



- From the ICT navigation pane, click Specify install locations. Then, under Installation type, select **Installs only for the user who installs it**.
- In the Navigation pane, select **Add Files**.
- in the **Add files to** list, select [PersonalFolder]. Then type in:

```
\Micro Focus\Reflection\Keyboard Maps
```

When you are done, the list entry is:

```
[PersonalFolder]\Micro Focus\Reflection\Keyboard Maps
```
- Click **Add**. Then browse to the configuration file (myKeyboardMap .xkb) and click **Open**.
- Repeat steps 6 and 7 to add the mySession .rd3x file to the [PersonalFolder]\Micro Focus\Reflection location.

NOTE: Be sure to specify the correct location for each file you add. Reflection looks for configuration files in specific locations. To find the correct file locations for any type of file see [“Customized Files that Must be Deployed to Specific Locations”](#) on page 119.

When you are finished adding files, the panel should look similar to the following:

Deploy these files	To these folders
Customized information privacy settings PrivacyFilters.xml (includes all Privacy Filter settings) PCIDSS.settings (includes all other Information Privacy settings)	For all users: [CommonAppDataFolder] \Micro Focus\Reflection\Desktop\v17.0 For only the user who installs: [AppDataFolder] \Micro Focus\Reflection\Desktop\v17.0
Customized (or minimized) ribbon frame.settings	For all users: [CommonAppDataFolder] \Micro Focus\Reflection\Desktop\v17.0 For only the user who installs: [AppDataFolder] \Micro Focus\Reflection\Desktop\v17.0
Customized Quick Access toolbar Application.settings frame.settings Reflection2007.Application.Ribbon.xml	For all users: [CommonAppDataFolder] \Micro Focus\Reflection\Desktop\v17.0 For only the user who installs: [AppDataFolder] \Micro Focus\Reflection\Desktop\v17.0
Restricted feature settings .access	For all users: [CommonAppDataFolder] \Micro Focus\Reflection\Desktop\v17.0 For only the user who installs: [AppDataFolder] \Micro Focus\Reflection\Desktop\v17.0
User templates .rsft	For only the user who installs: [AppDataFolder] \Micro Focus\Reflection\Desktop\v17.0\templates

Shared Security Settings

File Type	Deployment location
Secure Shell settings <i>Global</i> ssh_config ssh_known_hosts	<p>Note that the global filenames are different than the user-specific filenames, although the content may be the same.</p> <p>For all users</p> <p>[CommonAppDataFolder]\Micro Focus\Reflection</p>
<i>User-specific</i> config known_hosts	<p>.</p> <p>For the user who installs</p> <p>[PersonalFolder]\Micro Focus\Reflection\.ssh</p>
Reflection Trusted Certificate Authorities trust_store.p12	<p>For all users</p> <p>[CommonAppDataFolder]\Micro Focus\Reflection\.pki</p> <p>For the user who installs</p> <p>[PersonalFolder]\Micro Focus\Reflection\.pki</p>
Reflection Certificate Manager settings pki_config	<p>For all users</p> <p>[CommonAppDataFolder]\Micro Focus\Reflection\.pki</p> <p>For the user who installs</p> <p>[PersonalFolder]\Micro Focus\Reflection\.pki</p>
Kerberos settings rsckrb5.xml	<p>Settings in this location are migrated to the Windows registry for each Windows user the first time the user runs Kerberos Manager or any Reflection client configured to use Reflection Kerberos.</p> <p>For all users</p> <p>[CommonAppDataFolder]\Micro Focus\Reflection</p> <p>For the user who installs</p> <p>[AppDataFolder]\Micro Focus\Reflection</p>

Reflection FTP Client Settings

File Type	Deployment location
rftp.xml	<p>Settings in the .xml settings file are migrated to a settings.rfw file for each user the first time the user runs the FTP Client.</p> <p>Note: You can also use Modify User Settings to deploy FTP Client settings.</p> <p>For all users</p> <p>[CommonAppDataFolder]\Micro Focus\Reflection</p> <p>For the user who installs</p> <p>[AppDataFolder]\Micro Focus\Reflection</p>
settings.rfw	<p>For the user who installs</p> <p>The default user folder is:</p> <p>[PersonalFolder]\Micro Focus\Reflection</p>

Regular Session Document Files, Update Files, and Session Settings

Deploy these files	To these folders
Session document files (.rdox, .rd3x, .rd5x, .urlx, , .rd6x)	<p>Regular session document files and session settings files are installed under the location that is defined as the user data directory. The following locations are based on the default location of this directory:</p> <p>[PersonalFolder]\Micro Focus\Reflection</p> <p>For only the user who installs:</p> <p>[PersonalFolder]\Micro Focus\Reflection</p>
Session Update files (.rdoxu, .rd3xu, .rd5xu, .rd6xu)	<p>Reflection looks for Session Update files with the special "Update" name in the user data directory. The following locations are based on the default location of this directory:</p> <p>[PersonalFolder]\Micro Focus\Reflection</p> <p>For only the user who installs:</p> <p>[PersonalFolder]\Micro Focus\Reflection</p>
Layout file (.rwsp)	[PersonalFolder]\Micro Focus\Reflection
keyboard map (.xkb)	[PersonalFolder]\Micro Focus\Reflection\Keyboard Maps
mouse map (.xmm)	[PersonalFolder]\Micro Focus\Reflection\Mouse Maps
Hotspots (.xhs)	[PersonalFolder]\Micro Focus\Reflection\Hotspots Maps

Deploy these files	To these folders
Custom Ribbon Interfaces (.xuml)	[PersonalFolder]\Micro Focus\Reflection\CustomUI
Themes (.themex)	[PersonalFolder]\Micro Focus\Reflection\Themes
Quickpads (.rqpz)	[PersonalFolder]\Micro Focus\Reflection\QuickPads

Custom Office Productivity Templates

Deploy these files	To these folders
Word and PowerPoint templates (.dotx or. ppt)	For only the user who installs: [PersonalFolder]\Micro Focus\Reflection

Plus Archives

Deploy these files	To these folders
Plus archive (.rdar)	For only the user who installs: [PersonalFolder]\Micro Focus\Reflection\Plus\Rules

VBA SharedMacros Files and the VBA Common Project

Deploy these files	To these folders
Shared macros files	For only the user who installs:
sharedmacros.rd3x	The location that is defined as the user data directory. The default location is: [PersonalFolder]\Micro Focus\Reflection
sharedmacros.rd5x	
sharedmacros.rdox	
VBA Macros in the Common project	For all users:
vbaProject.bin	[CommonAppDataFolder]\Micro Focus\Reflection\Desktop\v17.0
	For only the user who installs:
	[AppDataFolder]\Micro Focus\Reflection\Desktop\v17.0

NOTE: The default values for the [AppDataFolder], [PersonalFolder], and [CommonAppDataFolder] Reflection properties are:

[AppDataFolder]: C:\Users\username\AppData\Roaming

[PersonalFolder]: C:\Users\username\Documents

[CommonAppDataFolder]: C:\ProgramData

Other Customized Files and Supported Files from Other Products

In addition to the custom files that must be deployed to specific locations, Reflection also supports the following files.

Customized files that you can deploy

Unlike other customized session files, Reflection doesn't look for these files in the user data directory, so you don't need to deploy them to specific locations.

- ◆ [“Other Customized Files” on page 124](#)

Other supported files

The best practice for handling legacy files or files from other products is to convert them to the Reflection Desktop file format and then deploy the new files.

- ◆ [“Other Supported Files” on page 125](#)
- ◆ [“Supported Extra! Files” on page 125](#)
- ◆ [“Supported Legacy Reflection Files” on page 126](#)
- ◆ [“Supported Files from Other Products” on page 127](#)

For a list of Reflection Desktop files that are deployed to specific locations, see [“Customized Files that Must be Deployed to Specific Locations” on page 119](#).

Other Customized Files

In addition to the files listed in [“Customized Files that Must be Deployed to Specific Locations” on page 119](#), you can deploy the following session files. Unlike other customized session files, Reflection doesn't look for these files in the user data directory.

Description	File Extensions
HP session	.r1w
3270 and 5250 printer	.rsf
HP session Update file	.r1u

Other Supported Files

The following Reflection file types are also supported. You most likely would not deploy these files.

Description	File Extensions
IBM host trace	.hst
IBM internal trace	.rit
Host simulation	.sim
Plus Rules	.rsdo
Rumba Screen Designer Project	.rsdp
Screen History	.rshx
VT Trace file	.rev
Express Macro Language	.js
Mainframe Transfer Request	.mto
AS/400 file Transfer Request	.xto

Supported Extra! Files

The following Extra! file types also continue to be supported.

Description	File Extensions
Extra! session	edp
Extra! Layout File macro	elf
HotSpot scheme	ehs
Legacy hotspots are partially supported. Text hotspots mappings are imported. Region hotspots mappings are not imported.	
Keyboard map §	ekm
QuickPad §	eqp
Toolbar §	etb
Extra! Basic macro	ebm
Extra! Transfer List	.eil
Extra! Transfer Scheme	.eis
Extra! Basic Header	.ebh
Extra! 3270 and 5250 File Transfer List	.etl

Description	File Extensions
Extra! 5250 File Transfer Scheme	.ftb
Extra! Basic Dialog	.ebd

Supported Legacy Reflection Files

The following legacy Reflection file types also continue to be supported.

Description	File Extensions
Reflection for UNIX and OpenVMS Settings File	.r2w
Reflection for Secure IT Settings File	.r3w
Reflection for REGIS Graphics Settings File	.r4w
Reflection for IBM Settings	.rsf
Macro	.rvx
FTP Settings File	.rfw
Mainframe Transfer Request File	.mto
AS/400 Transfer Request File	.xto
Reflection Command Language Scripting File	.rcl
Reflection Basic Scripting Files	.rbs
Reflection Macro File	.rma
Reflection FTP Script File	RFS
VT Trace File	.rev
Reflection for IBM partial toolbar settings	.btp
Reflection for UNIX and OpenVMS partial toolbar settings	.rtb
NonStop 6530 session	.rd6

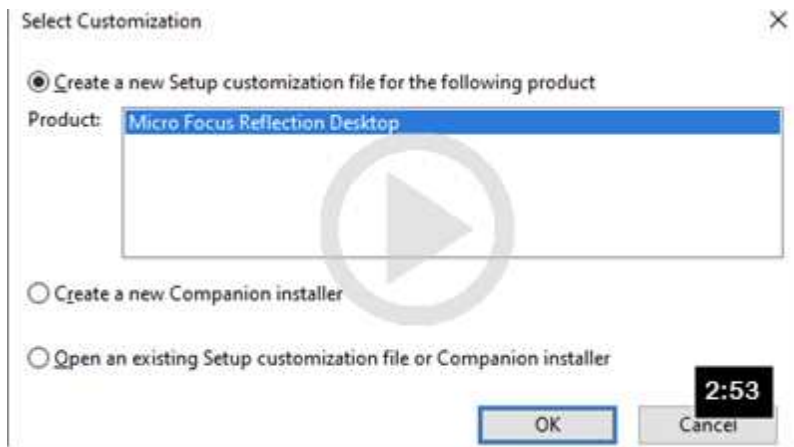
Supported Files from Other Products

The following legacy Reflection file types also continue to be supported.

Description	File Extensions
Micro Focus Rumba sessions	.wdm, .wda, .rsdm, .rsda, .wdu, .rsdu
Micro Focus Rumba macros	.rmc
IBM PComm Keyboard Map	.kmp
IBM PComm sessions	.ws
IBM PComm Macro	.mac
IBM PComm VBScript	.vbs
KEA! sessions	.ktc
OpenText HostExplorer macros	.ebs
Brandon Systems\Jolly Giant QWS3270 macros	.jgs
Ipswitch WS_FTP	.wsftp_options.ini

10 Modify the Installation

Create and deploy a transform to customize how Reflection is installed on user workstations. As defined by Microsoft, "a transform is a collection of changes applied to an installation. By applying a transform (*.mst file) to a base installation package, the installer can add or replace data in the installation database."



Creating Transforms

By deploying the transform with the Reflection base installation package, you can specify the installation directory, change the user data location, change the Remove or Add commands from the Windows Uninstall or change a program list, and change other default settings.

This article

[Create or Modify a Transform](#)

Create a Microsoft transform (.mst file) that you can deploy with your product MSI file to change how the product is installed.

[Walkthrough: Create a Transform](#)

Example that shows how to add a desktop shortcut and disable the Change button on the Uninstall or change a program list.

[Apply a Transform to Your Installation](#)

How to add a transform to installs started with setup.exe

Create or Modify a Transform

You can use the Installation Customization Tool to create a transform (.mst) file.

- ♦ ["Change the Installation Directory" on page 130](#)
- ♦ ["Modify Setup Properties" on page 131](#)
- ♦ ["Add/Modify Registry Data" on page 131](#)

- ♦ “Select Features, Components, and Languages” on page 132
- ♦ “Add (Chain) Installations and Run Programs” on page 133
- ♦ “Install the Reflection Help” on page 134
- ♦ “Predefined System Folders” on page 135
- ♦ “Configure Shortcuts” on page 136

To create a transform with the Installation Customization Tool

- 1 From your administrative installation point, open the Installation Customization Tool from a [shortcut \(page 38\)](#) or by typing the following command line:

```
<path_to_setup> \setup.exe /admin
```

- 2 In the Select Customization dialog box, select **Create a new Setup customization file for the following product** or **Open an existing Setup customization file**.
- 3 Select items from the list in the left panel to open configuration panels on the right, and then make your customizations.
- 4 Save your transform as an .mst file.

NOTE: Micro Focus recommends that you save transform files in the same folder as the Reflection base installation package .msi file.

Change the Installation Directory

You can change the Reflection installation directory.

To change the installation directory

- 1 From your administrative installation point, open the Installation Customization Tool from a [shortcut \(page 38\)](#) or by typing the following command line:

```
<path_to_setup> \setup.exe /admin
```

- 2 From the **Select Customization** dialog box, do one of the following:
 - ♦ Select **Create a new Setup customization file for the following product**.
 - ♦ Select **Open an existing Setup customization file or Companion installer** and, in the **Open** dialog box, select an .mst file.
- 3 On the left panel of the Installation Customization Tool, click **Install location and organization name**.
- 4 In the **Default installation folder** box, specify where to install the product files.

NOTE: List items that refer to folders (for example, [ProgramMenuFolder]) are pre-defined folder keywords. You can create customized directories by adding new folder names with typical directory syntax (such as, [ProgramFilesFolder]\My Folder). Alternatively, you can enter a fully qualified path (for example, C:\Program Files\My Folder), as long as that location is known to exist on the target machine.

Modify Setup Properties

In some cases you may want to customize your installation using Windows Installer (MSI) properties that support features not configurable on other Installation Customization Tool panels. For example, the installer property ARPHELPLINK, sets the URL used by the support link in the Programs and Features list.

NOTE: If you want to create a custom (non-default) user data location that will be used by all installed Reflection products, specify the product-specific property WRQ_USERDIR. This property affects only those applications whose default user data directory is [PersonalFolder]\Micro Focus\Reflection\. Reflection applications that use this default data location include: Reflection Workspace, Reflection FTP Client, Reflection Kerberos Manager, Reflection Key Agent, Reflection IBM Printer, and Reflection X (legacy). Reflection X Advantage does not use this user data location.

To set the location where Reflection X Advantage artifact files are saved, specify the file path in the RXA_ARTIFACTS_DIR property.

To modify installation properties

- 1 From your administrative installation point, open the Installation Customization Tool from a [shortcut \(page 38\)](#) or by typing the following command line:

```
<path_to_setup> \setup.exe /admin
```
- 2 From the **Select Customization** dialog box, do one of the following:
 - ♦ Select **Create a new Setup customization file for the following product**.
 - ♦ Select **Open an existing Setup customization file or Companion installer** and, in the **Open** dialog box, select an .mst file.
- 3 From the Installation Customization Tool navigation pane, select **Modify setup properties**.
- 4 Click **Add** to open the **Add/Modify Property Value** dialog box.
- 5 In the **Name** box, use the drop-down list to select commonly-used public properties that are standard to the Windows Installer. Select an item in the list to see a brief description of the selected property. For additional information about Windows Installer properties, see the [Microsoft Windows Installer Guide \(http://msdn.microsoft.com/en-us/library/windows/desktop/aa372845\(v=vs.85\).aspx\)](http://msdn.microsoft.com/en-us/library/windows/desktop/aa372845(v=vs.85).aspx). Some Micro Focus products support additional properties that do not appear in the drop-down list. You can configure these properties by manually entering the property name.
- 6 From the **File** menu, select **Save As** and save the file to the same folder as the installer package file for Reflection (setup.exe).

Add/Modify Registry Data

To modify registry data

- 1 From your administrative installation point, open the Installation Customization Tool from a shortcut or by typing the following command line:

```
<path_to_setup> \setup.exe /admin
```

- 2 From the **Select Customization** dialog box, do one of the following:
 - ◆ Select **Create a new Setup customization file for the following product**.
 - ◆ Select **Open an existing Setup customization file or Companion installer** and, in the **Open** dialog box, select an `.mst` file.
- 3 From the navigation pane, click **Add registry data**.
- 4 To add a new registry value, click **Add**.

-or-

To modify a registry value in the table, select the value, and then click **Modify**.

- 5 Specify registry keys and values to add or modify during the installation process. By modifying registry values, you can change the way the application operates. For example, for certain Micro Focus applications, you can add a value that specifies to never save settings on exit.

In this field	Enter or select
Key	The complete Registry path from the root, for example: HKEY_LOCAL_MACHINE\SOFTWARE\Reflection\Rwin\Reflection
Name	The registry value name, for example: SaveChanges If the Name box is blank, the data entered into the Value box are written to the Default registry key.
Type	The data type of the value. For example: DWORD Types include strings, integers (DWORD), or binary values.
Value	The value. For example: 0x00000000 (0)

Select Features, Components, and Languages

You can select which features, components, and languages to install for your end users. In addition, you can make features available to users for a later installation or hide them from view.

To select features, components, and languages to install

- 1 From your administrative installation point, open the Installation Customization Tool from a shortcut or by typing the following command line:

`<path_to_setup> \setup.exe /admin`
- 2 From the **Select Customization** dialog box, do one of the following:
 - ◆ Select **Create a new Setup customization file for the following product**.
 - ◆ Select **Open an existing Setup customization file or Companion installer** and select an `.mst` file.

- 3 From the Installation Customization Tool navigation pane, select **Set feature installation states**, and for each feature, choose from the following states:

Choose



Feature will be installed on local hard drive



Feature will be installed when required



Feature will be unavailable

To do this

Add a feature to the installation.

NOTE: Some features listed under a selected feature may not be included when you select to install the higher-level feature. The features that are included are the recommended defaults. If you select the higher level feature a second time, all sub-features will be included.

Advertise a feature.

Leave a feature uninstalled. End users will still be able to select and install the item from the Windows **Program and Features** or the **Add or Remove Programs** control panel.

Related Topics

- ♦ [“Apply a Transform to Your Installation” on page 138](#)

Add (Chain) Installations and Run Programs

Reflection makes it easy to "chain" installs. You can set up an install to run companion install packages automatically before or after the primary installation. You can also specify to run other scripts or programs. For example, the Reflection product Help is available as a separate program (.msi) that you can add to the base product installation. See [“Install the Reflection Help” on page 134](#).

NOTE: This method of chaining installations applies only to installs performed with `setup.exe`. It does not apply to installs that use the MSI command-line method.

To chain installations and programs

- 1 From your administrative installation point, open the Installation Customization Tool from a [shortcut \(page 38\)](#) or by typing the following command line:

```
<path_to_setup> \setup.exe /admin
```
- 2 From the **Select Customization** dialog box, do one of the following:
 - ♦ Select **Create a new Setup customization file for the following product**.
 - ♦ Select **Open an existing Setup customization file or Companion installer** and select an .mst file.
- 3 From the navigation pane, select **User Interface** and then select **Use this customization with interactive installs using setup.exe**.
- 4 From the navigation pane, select **Add installations and run programs**.

5 Click **Add**.

The **Add/Modify Program Entry** dialog box opens.

6 In the **Target** list, enter or select the folder where the program `.exe` file or the `.msi` file resides, and then enter the executable to run. For example:

```
msiexec.exe
```

7 Under **Arguments**, enter the command-line arguments to execute. For example:

```
/i my_installation.msi
```

8 To specify when to run the program, select one of the following:

- ♦ **Run this program after base product has been installed**
- ♦ **Run this program before the base product has been installed.**

NOTE: For most cases, select **Run this program after the base product has been installed**. If you select **Run this program before the base product has been installed** and the program fails, Reflection is not installed.

9 Repeat these steps to add other programs or `.msi` files.

10 To change the execution sequence, use the arrows next to **Move** in the lower-left area of the pane. To remove a program from the list, select it in the list and click **Remove**.


Install the Reflection Help

By default, the Reflection Help browser opens the Help from the Micro Focus website. To allow users to open the Help from the local hard drive you must install the product Help after installing Reflection and change the [Reflection Workspace settings](#) to use the installed help system.

To configure the Reflection Workspace to open Help locally

1 Open the **Reflection Workspace Settings > dialog box**.

The steps depend on your user interface mode.

User Interface Mode	Steps
Ribbon (Office 2007)	On the Reflection button  (if using the Office 2007 Look and Feel), choose Reflection Workspace Settings .
Ribbon (Office 2010)	On the File menu, choose Reflection Workspace Settings .
Reflection Browser	On the Reflection menu, choose Settings and then Reflection Workspace Settings .
Mobile UI	Tap the Gear icon and then select Reflection Workspace Settings .

2 Under **Workspace Settings**, select **Configure Workspace Attributes**.

3 Under **Help System**, select **Use installed help system**.

If Reflection can't find the Help files on the local drive, it will start your default browser and open the help files from Micro Focus website.

To install the Help locally

Do one of the following:

- ♦ On a workstation where Reflection is already installed, from the installation image, open the HelpInstaller folder and double-click `setup.exe`.
- or -
- ♦ Chain the Help installer to the main installation, selecting **Run this program after base product has been installed**. For more information, see [“Add \(Chain\) Installations and Run Programs” on page 133](#).

To set up Help on a Windows server

NOTE: If you install Reflection on a Windows Server and Enhanced Security (IE ESC) is enabled, viewing the Help will result in security prompts each time you open a page. To resolve this problem, do the following:

- 1 Turn off **Internet Explorer Enhanced Security for users and administrators**.
- 2 Turn Off **Internet Explorer Enhanced Security** on Windows 2008 Server.
- 3 Enable scripting for Internet Explorer. See [Knowledge Base Article 7021458 \(https://support.microfocus.com/kb/doc.php?id=7021458\)](https://support.microfocus.com/kb/doc.php?id=7021458).

Predefined System Folders

When you configure destination locations using the Installation Customization Tool, your options include Reflection properties that correspond to Windows [system folder properties \(http://msdn.microsoft.com/en-us/library/aa372057.aspx?ppud=4\)](http://msdn.microsoft.com/en-us/library/aa372057.aspx?ppud=4). During installation, the Windows installer expands these to show the appropriate location for your operating system.

The list of available folders for adding files to a companion installer depends on whether you are installing for all users (the default) or for individual users.

All-user installations

Reflection Property name	Default Windows location	Default path using Windows variables
[CommonAppDataFolder]	C:\ProgramData	%Programdata%
[CommonDocumentsFolder]	C:\Users\Public\Documents	%Public%\Documents
[CommonFilesFolder]	C:\Program Files\Common Files	%ProgramFiles%\Common Files
[ProgramFilesFolder]	C:\Program Files	%ProgramFiles%
[RootDrive]	C:\	\
[WindowsFolder]	C:\Windows	%windir%

Individual user installations

Reflection Property name	Default Windows location	Path using Windows variables
[AppDataFolder]	C:\Users\ <user>\appdata\roaming\< td=""><td>%appdata%</td></user>\appdata\roaming\<>	%appdata%
[LocalAppDataFolder]	C:\Users\ <user>\appdata\local\< td=""><td>%localappdata%</td></user>\appdata\local\<>	%localappdata%
[PersonalFolder]	C:\Users\ <user>\documents\< td=""><td>%userprofile%\documents</td></user>\documents\<>	%userprofile%\documents
[RootDrive]	C:\	\
[%UserProfile]	C:\Users\ <user>< td=""><td>%userprofile%</td></user><>	%userprofile%

Configure Shortcuts

You can change the attributes associated with the pre-defined Reflection shortcuts. Also, you can configure shortcuts associated with files you've added to a custom install package.

NOTE: This method of chaining installations applies only to installs performed with `setup.exe`. It does not apply to installs that use the MSI command-line method.

To configure shortcuts

- 1 On a workstation on which you have installed Reflection, open the Installation Customization Tool from a desktop shortcut (if you set up a shortcut as shown on page 11) or from a command line as follows:

```
path_to_setup\setup.exe /admin
```
- 2 From the **Select Customization** dialog box, do one of the following:
 - ♦ Select **Create a new Setup customization file for the following product**.
 - ♦ Select **Open an existing Setup customization file or Companion installer** and select an .mst file.
- 3 From the Installation Customization Tool navigation pane, choose **Configure shortcuts**.
- 4 Select the shortcut that you want to configure, and then click **Modify**.
- 5 In the **Modify Shortcut** dialog box, in the **Location** list, enter or select the folder where you want the shortcut to reside.

NOTE: List items that refer to folders (for example, [ProgramMenuFolder]) are pre-defined folder keywords. You can create customized directories by adding new folder names with typical directory syntax (such as, [ProgramFilesFolder]\My Folder). Alternatively, you can enter a fully qualified path (for example, C:\Program Files\My Folder), as long as that location is known to exist on the target machine.

- 6 Enter a descriptive name and tooltip for the shortcut.
- 7 Enter the command-line arguments for the shortcut in the **Arguments** window.

Walkthrough: Create a Transform

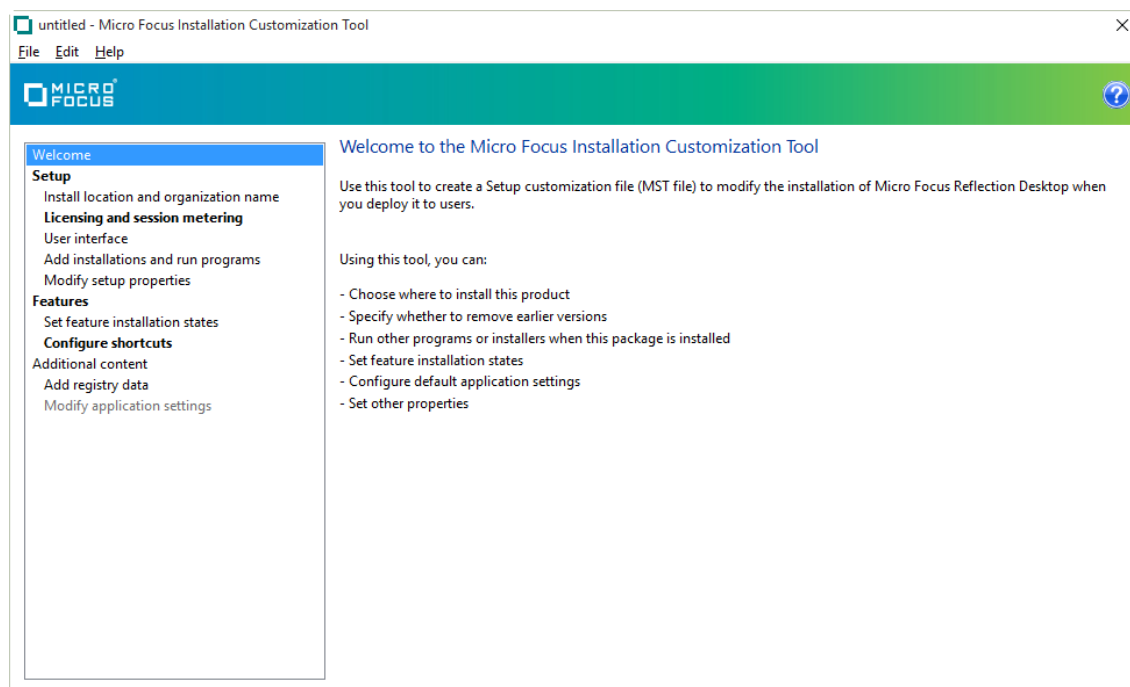
This example shows how to set up a transform that can be applied to an install on a command line. To automatically deploy the transform in an installation that uses the Reflection Setup program, see [“Apply a Transform to Your Installation” on page 138](#).

To create a transform that adds a desktop shortcut and disables the Change button

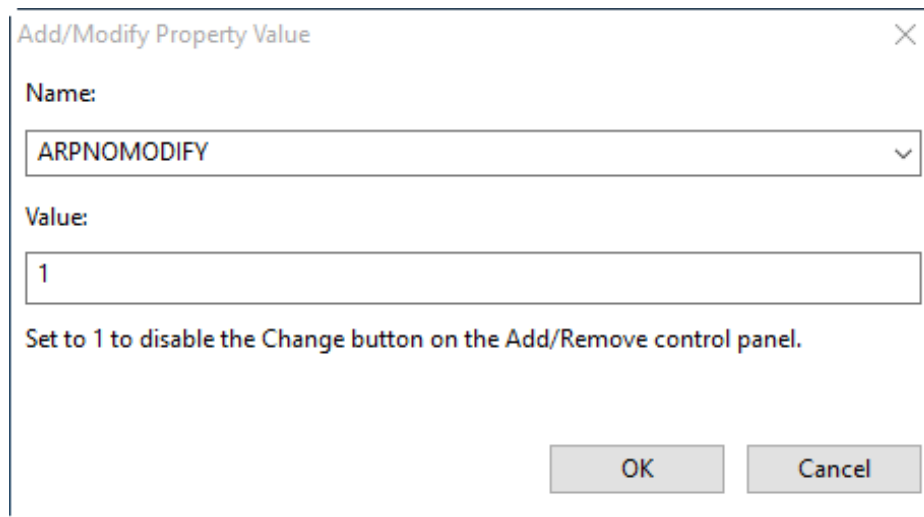
- 1 From your administrative installation point, open the Installation Customization Tool from a [shortcut \(page 38\)](#) or by typing the following command line:

```
<path_to_setup> \setup.exe /admin
```

- 2 Select **Create a new setup customization file for the following product**, and then click **OK**.



- 3 From the navigation pane, click **Licensing and session metering** and then select **I accept the terms of the Software License Agreement**.
- 4 The following optional steps describe how to create a Reflection Workspace shortcut to the user's desktop and remove the **Change** button from the Windows **Uninstall or change a program** list.
 - 4a From the navigation pane, click **Modify setup properties**.
 - 4b In the lower-right corner, click **Add**.
 - 4c In the **Add/Modify Property Value** dialog box, for **Name** select **ARNOMODIFY** and for **Value** enter **1**. This step removes the **Change** button.



- 4d From the navigation pane, click **Configure shortcuts**.
- 4e Under **Modify shortcuts for this product**, select **Reflection workspace** and click **Modify**.
- 4f In the **Modify shortcut** dialog box, in the **Location** field, select **[DesktopFolder]** and click **OK**. This step creates a desktop shortcut.
- 5 From the **File** menu, save the transform on the administrative installation point as `myCustomInstall.mst`.

Apply a Transform to Your Installation

If you have created a transform to customize how Reflection is installed, you need to deploy the transform with the primary installation. (This is in contrast to companion installation packages, which can be chained with the primary installation or installed separately.)

You can include a transform by modifying the `setup.ini` file or by adding it to the `setup.exe` command line. Any install started with `setup.exe` or with command-line installs can include a transform.

To apply a transform by modifying the `setup.exe` program

- 1 From your administrative installation point, open the Installation Customization Tool from a [shortcut \(page 38\)](#) or by typing the following command line:

```
<path_to_setup> \setup.exe /admin
```
- 2 From the **Select Customization** dialog box, do one of the following:
 - ◆ Select **Create a new Setup customization file for the following product**.
 - ◆ Select **Open an existing Setup customization file or Companion installer** and select an `.mst` file.
- 3 Go to the **User interface** panel.
- 4 Select **Use this customization with interactive installs using setup.exe**.

When you save your transform with this option selected, the Installation Customization Tool automatically updates the `setup.ini` file to apply the transform to the Reflection installation by adding the following line to the `[Setup]` section:

```
CustomTransform=<your_transform.mst>
```

This modification to `setup.ini` means that any install using `setup.exe` (using either the interactive user interface or using `setup.exe` on a command line) will automatically apply this transform.

- 5 Save the transform to the default location (the folder that includes `setup.exe`).

The transform can now be deployed to end users via the `setup.exe` file. (Users can run `setup.exe` directly or `setup.exe` can be called from a script or initiated from a command line.)

NOTE: If the `setup.ini` file already specifies the transform file you want to install (as noted in step 3 of the previous procedure) do not specify the transform on the command line.

To add the transform to a `setup.exe` command-line install

- ♦ From the command line, use the following syntax:

```
<path_to_setup>\setup.exe /install TRANSFORMS=<transform.mst>
```


11 Deploy Reflection

This chapter provides instructions for deploying Reflection, session document files, and other configuration files.

This article	Describes
Deploy with the Reflection Setup program	How to deploy using the setup.exe command line.
Deploy with MSI	How to deploy using the MSI command line and handle prerequisites that must be installed before you deploy with MSI directly.
Publish with Active Directory	Requirements for assigning and publishing your product installation using Microsoft Active Directory.
Deploy with System Center Configuration Manager	How to deploy Reflection with Microsoft Systems Center Configuration Manager (or Microsoft Systems Management Server).
Apply Software Updates (Service Packs or Patches)	How to install software updates with the Micro Focus Patch utility included with your distribution.
Remove an Installation	How to remove Reflection or a package of configuration files.
Repair an Installation	How to use the Repair option, which automatically searches for and replaces missing or corrupted files.

Deploy with the Reflection Setup program

The Reflection Setup program (`setup.exe`) is the recommended tool for installing and deploying Reflection.

`Setup.exe` has a command-line interface that you can run from a command line, a batch file, or a deployment tool. You can type command-line options to set installation parameters and limit user interaction as Reflection is installing, or use command-line options to prepare Reflection for installation by users. Command-line installations may have additional [“System Requirements” on page 30](#).

To see a list of available command-line parameters, type:

```
setup.exe /?
```

To install a companion installer package with the Setup program, add it to a transform and then apply that transform to `setup.exe`. This "chains" the companion installer package to the main installation. You can set the package to deploy before or after the main installation.

The following procedures show command-line options commonly used for deployment, including switches for silent installations.

To deploy Reflection “out-of-the-box”

Use this command to deploy Reflection with default settings.

- ♦ At a command prompt on a test workstation, enter:

```
path_to_administrative_installation_point \setup.exe /install
```

To perform a silent installation

- ♦ At a command prompt, change to the directory in which the `setup.exe` file resides and do one of the following:

To perform	Type
A silent install that displays a progress bar and disables the Cancel button	<code>setup.exe /install /passive</code>
A silent install with no display	<code>setup.exe /install /quiet</code>

To deploy Reflection and a transform

- ♦ At a command prompt, enter:

```
path_to_administrative_installation_point \setup.exe /install TRANSFORMS=  
myCustomInstall .mst
```

NOTE: You can also set up the Reflection Setup program to deploy a transform automatically. See “[Add \(Chain\) Installations and Run Programs](#)” on page 133.

Deploy with MSI

You can deploy Reflection directly from the MSI command line. You can also deploy companion .msi files that you have created to contain your custom configuration files.

Deploy Reflection from MSI Command Line

Use these procedures to install Reflection from a command line with MSI.

To customize your installation, specify Windows Installer properties on the command line or pass them in a transform file. For a list of public properties that are standard to the Microsoft Windows Installer, see the “[Modify Setup Properties](#)” on page 131 pane in the Installation Customization Tool, or refer to the [Microsoft Windows Installer Guide \(http://msdn.microsoft.com/en-us/library/windows/desktop/aa372845\(v=vs.85\).aspx\)](http://msdn.microsoft.com/en-us/library/windows/desktop/aa372845(v=vs.85).aspx).

If you want to create a custom (non-default) user data location that will be used by all installed Reflection products, specify the product-specific property `WRQ_USERDIR`. This property affects only those applications whose default user data directory is `[PersonalFolder]\Microsoft\Focus\Reflection\`. Reflection applications that use this default data location include: Reflection Workspace, Reflection FTP Client, Reflection Kerberos Manager, Reflection Key Agent, Reflection IBM Printer, and Reflection X (legacy). Reflection X Advantage does not use this user data location.

Handling prerequisites in command line installs

If you use setup.exe for your install, the setup program checks for any prerequisites required by the features you have selected and installs them automatically.

If you use msixec.exe for your install, prerequisites are not installed automatically. You need to install them separately if they are not already on your users' workstations. You can find installers for the required prerequisites in the Prerequisites folder in the distribution media, or in your administrative installation. The prerequisites you need to install depend on which programs and features you are installing:

- ♦ All Reflection Workspace features require Microsoft .NET Framework 4.7.1. If you attempt an install using msixec.exe and this prerequisite is not found, a message displays and the installer stops. To install the .NET Framework, run the executable file in Prerequisites\DotNet471.
- ♦ The Visual Basic for Applications feature requires Microsoft VBA 7.1. Use the core and language-specific *.msi packages in the Prerequisites\VB71 folder.

Handling upgrades in command line installs

If you are upgrading a previous version of Reflection, the prior version must be uninstalled before you can install the current version.

If you use setup.exe for your install, the setup program checks for a previous versions of Reflection and uninstalls it automatically if one is found.

If you use msixec.exe for your install, you must first manually uninstall any earlier versions. If you upgrade these products by deploying the .msi file directly and have not removed the earlier version, a message displays telling you to uninstall the older software first.

To deploy Reflection “out-of-the-box” directly with MSI

- ♦ At a command prompt on a test workstation, change to the directory in which the .msi file resides and enter:

```
msiexec /i path_to_administrative_installation_point \ yourVersion .msi
```

where *yourVersion* .msi is the specific version of the reflection MSI that you downloaded (for example, ReflectionV17.0.msi).

To deploy Reflection and a transform directly with MSI

- ♦ At a command prompt, enter:

```
msiexec /i path_to_administrative_installation_point \ yourVersion .msi TRANSFORMS=  
yourCustomInstall .mst
```

where *yourVersion* .msi is the specific version of the Reflection MSI that you downloaded (for example, ReflectionV17.0.msi).

Deploy Companion MSI File from MSI Command Line

You can deploy configuration files that are “packaged” in a companion installer package separately from Reflection. This allows you to deploy and maintain these files between Reflection software updates without removing Reflection.

If you use the Host Access Management and Security Server, you can upload MSI files to the Package Manager and silently deploy them to users' workstations. See [Use Central Management to Deploy MSI Packages](#).

To deploy a companion installer package directly with MSI

- ◆ At a command prompt, enter:

```
msiexec /i path_to_administrative_installation_point \ your_companion_file .msi
```

To remove a companion installer package directly with MSI

- ◆ At a command prompt, change to the directory in which the companion installer package file resides and enter:

```
msiexec /x your_companion_file .msi
```

Publish with Active Directory

To assign and publish your product installation using Microsoft Active Directory, you must meet the following requirements:

- ◆ Windows Administrative Tools are installed on your workstation.
- ◆ You are a member of **Domain Admins** and **Group Policy Creators and Owners**. (This is required to publish software.)

For more information, see "Active Directory groups" in the Microsoft Management Console help.

To install with Active Directory

- 1 From the **Active Directory User and Computers Console**, advertise your product installation to members of any organizational units in your Active Directory using appropriate transform modifications.
- 2 If multiple transforms are specified, make sure that the listed order of the transforms is correct, and click **OK**. (If you need to change the order for any reason after you click **OK**, you will have to start over again.)

NOTE: For more information about assigning and publishing, see "assigning applications" and "publishing applications" in the Microsoft Management Console help.

Deploy with System Center Configuration Manager

You can deploy Reflection with Microsoft Systems Center Configuration Manager (or Microsoft Systems Management Server).

To deploy with System Center Configuration Manager

- 1 Create an administrative install image on your site server.
This serves as the administrative installation point for deployment.
- 2 Use the product Package Definition File (.sms) to create the product installation package.

NOTE: The Package Definition File (.sms) is created during the administrative installation and can be found at the root of the administrative installation point. Alternatively, you can reference the .msi file directly — consult the Microsoft SMS documentation for more information.

- 3 Advertise the installation packages to your users.

Apply Software Updates (Service Packs or Patches)

Software updates are distributed as executable files that contain Microsoft .msp files. You can deploy these updates in a number of ways:

- ♦ “Apply a Patch to a Workstation” on page 145 using the patch utility user interface.
- ♦ “Apply a Patch to an Administrative Installation” on page 146 and then silently redeploy the Reflection MSI to users.
- ♦ “Directly Install a Patch on Workstations” on page 147 by extracting the patch from the downloaded executable and silently deploying it to users.
- ♦ “Perform a new Installation with a Patch” on page 147 to a “clean” machine on which Reflection is not installed.

NOTE: Patch log files are saved in the user's Windows temporary folder(%tmp%) with a generated name. The file naming convention for these log files is atmpatchxxxxxx.log.

Apply a Patch to a Workstation

When you open the downloaded software executable file, it unzips the files required to install the patch and opens a patch utility that you can use to apply the patch to a workstation.

Apply a patch to a workstation using the patch utility interface

- 1 Before you start, check the following requirements:
 - ♦ You must have administrative rights for the workstation.
 - ♦ The Reflection version that the patch applies to must be installed on your workstation.
 - ♦ You may require access to the original installation files – either the distribution media or administrative installation point, depending on the original installation type.
- 2 Download the executable file for the update.
- 3 Close all running applications.
- 4 Using Windows Explorer, locate the downloaded executable file and double-click the file. After the files needed to install the patch are extracted, a patch utility dialog box opens.
- 5 Accept the license.

- 6 In the Apply, Remove, or Save this update window, select **Apply to the Local installation on this workstation** and then click **Finish**.

If the patch is successfully installed, a message is displayed indicating "The update was applied successfully."

- 7 If problems occur, open the `atmpatchxxxxxx.log` file, created in each user's temporary folder.

Apply a Patch to an Administrative Installation

If you deploy Reflection from an administrative installation point, you can apply a patch to your administrative installation and then redeploy the Reflection MSI to users.

CAUTION: Do not patch the administrative installation while users are running Reflection "from the network."

Apply a patch to an administrative installation and silently install it on workstations

- 1 Before you start, check the following requirements:
 - ◆ You must have administrative rights for your administrative installation point.
 - ◆ The Reflection version that the patch applies to must be installed on the administrative installation point (as an administrative installation).
- 2 Download the executable file for the update.
- 3 Close all running applications so that system files can be updated if required.
- 4 Using Windows Explorer, locate the downloaded update file and double-click the file. After the files needed to install the patch are extracted, the patch utility dialog box opens.
- 5 Accept the license.
- 6 In the Apply, Remove, or Save this update window, select **Apply to the Administrative installation on a server** and then click **Finish**. Then browse to the administrative installation location and select the Reflection MSI Installer Package.

If the patch is successfully installed, a message is displayed indicating "The update was applied successfully."

- 7 If problems occur, open the `atmpatchxxxxxx.log` file, created in the user's temporary folder.
- 8 Deploy the MSI that includes the patch to users from your administrative installation point.

The following commands silently install the patched MSI on workstations from the administrative installation point:

```
msiexec /i <path_to_admin.msi> REINSTALLMODE=vomus REINSTALL=ALL /qn
```

For example:

```
msiexec /i "\\myserver\shared\AdminPoint\ReflectionV17.0.msi"  
REINSTALLMODE=vomus REINSTALL=ALL /qn
```

NOTE: If you prefer to use a command line to silently install the patch on your administrative installation, you can extract the .msp file as shown in ["To extract an msp file from the downloaded executable"](#) on page 147 and then apply the .msp file to your administrative installation as follows:

```
msiexec /p <path_to_patch.msp> /a <path_to_admin.msi> /qn
```

Directly Install a Patch on Workstations

If you deploy Reflection using an imaging technique or deployment tools such as Microsoft Systems Management Server (SMS) (rather than from an administrative installation point), you may want to either deploy the patch so that individual users can update their workstations, or deploy the patch automatically or silently.

In these cases, the you can extract the *.msp file from the Reflection patch executable and use the MSIEXEC command to apply the patch.

To extract an msp file from the downloaded executable

- 1 From the distribution image, double-click the self-extracting executable update file.
After the files needed to install the patch are extracted, the Patch Utility dialog box opens.
- 2 Accept the license.
- 3 In the Apply, Remove, or Save this update window, select **Save to a file (*.msp) for later application** and then click **Finish**.
- 4 Save the .msp file.

To deploy the patch to workstations

- 1 Extract the patch from the downloaded executable as shown above.
- 2 Use your standard deployment tools along with MSIEXEC and the MSIEXEC command line switches to apply and deploy the service pack:

```
msiexec /p <file name>.msp REINSTALL=ALL REINSTALLMODE=omus <switches>
```

The following example uses the /qn switch to install the service pack with no user interface (a silent install).

```
msiexec /p "E:\RTM Image\t170442.msp" REINSTALL=ALL REINSTALLMODE=omus /qn
```

Perform a new Installation with a Patch

You can install Reflection with a patch on a “clean” workstation (a workstation on which Reflection is not installed).

For example, you could use this approach to install Reflection with a service pack.

Use the following command to silently install Reflection to a “clean” workstation and apply a patch:

```
msiexec /i <path_to_original.msi> PATCH=<path_to_patch.msp> /qn
```

Remove an Installation

To remove Reflection, you can use the Windows Control Panel, the Reflection Setup program user interface, or a command line. To remove a companion installation, you can use the Windows Control Panel or a command line.

NOTE: You must log on with administrator privileges to remove Reflection.

To remove an installation using the Windows control panel

- 1 To open the **Programs and Features** control panel go to **Start > Control Panel > Programs and Features**. (On older Windows systems, this Control Panel is called Add or Remove Programs.)
- 2 Select the name of the installation that you want to remove.
- 3 Click **Uninstall** (or **Remove**).

To remove an installation with the Reflection Setup program user interface

- 1 From an administrative installation image, click the `setup.exe` file.
- 2 From the tab, select **Remove**, and then click **Continue**.

To remove an installation from the Reflection Setup program command line

CAUTION: If you use the following instructions to find the product code in the registry, make sure you do not change any registry values. Changing these values can damage an installation. If you prefer not to use the registry, you can get the product code by contacting Technical Support.

- 1 Open the registry editor (`regedit.exe`) and find this key:

For 32-bit platforms

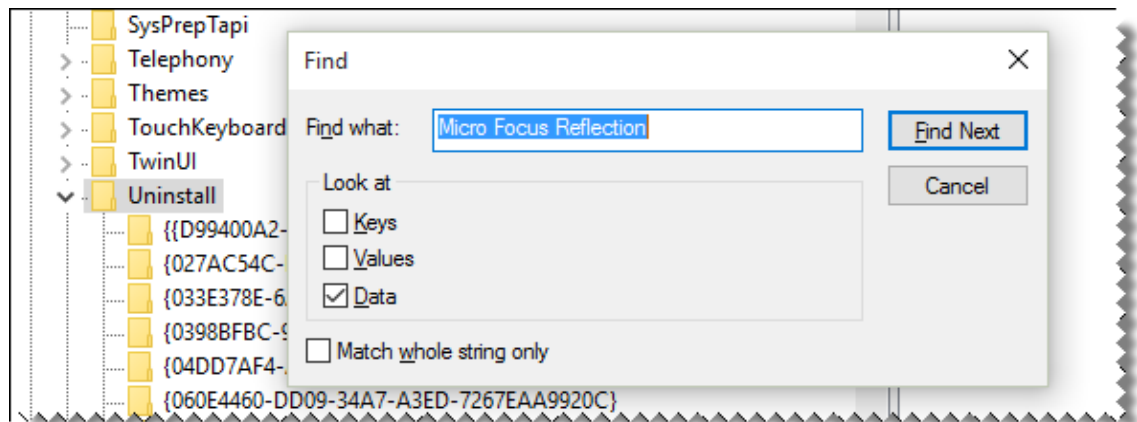
```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall
```

For 64-bit platforms

```
HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Uninstall
```

Each key under the `Uninstall` key is the product code or Globally Unique Identifier (GUID) for a product installed on the computer.

- 2 In the `Uninstall` key, search for `Micro Focus Reflection` to locate the GUID associated with the product.



- 3 At the command prompt or the Start menu Run command, change to the directory in which the `setup.exe` file resides and enter:

```
setup.exe /uninstall ProductCode
```

where *ProductCode* is the Globally Unique Identifier (GUID) that is the principal identifier for the product.

To remove a companion installer package with MSI directly

- ♦ At a command prompt, change to the directory in which the companion installer package file resides and enter:

```
msiexec /x your_companion_file.msi
```

Repair an Installation

If you are experiencing problems with your installation, you can use the **Repair** option, which automatically searches for and replaces missing or corrupted files.

To repair an installation with the Reflection Setup program user interface

- 1 From an administrative installation image, click the `setup.exe` file.
- 2 Click **Repair**, and then follow the installer instructions.

To repair an installation using Windows Add/Remove

- 1 From the Windows **Programs and Features** (or the **Add or Remove Programs**) control panel, select the name of the installation that you want to repair, and then click **Change**.
- 2 From the Reflection Setup program, select **Repair**, and then click **Continue**.

12 Use a Centralized Management Server

You can centrally manage, secure, and monitor users' access to host connections with the Micro Focus Host Access Management and Security Server (MSS), a separately available product that is designed to provide centralized management for Reflection sessions.

Using this centralized management server, you can grant or deny access based on group or role, quickly apply security updates and configuration changes to align with changing regulatory or business needs, and make post-install adjustments on the fly. MSS allows you to configure and lock down large numbers of desktops with ease.

MSS Servers and Add-Ons

MSS includes two servers that you can use to configure and monitor your sessions:

- ◆ **The Administrative Server**

Using the MSS Administrative Console, you can define terminal emulation sessions, configure and save session settings, and then manage and configure secure settings for those sessions. You can also use directory services, such as Active Directory, to authorize access to host applications—without changing your LDAP schema or data. Sessions that you create in this way are saved to the server and can be made available to users from the server and modified at any time. See [“Create and Deploy Sessions and Settings with the MSS Administrative Console” on page 153](#).

- ◆ **The Metering Server**

Use the Metering Server to track Reflection sessions and determine how many client workstations use the product. See [“Enable Usage Metering” on page 154](#).

You can enhance your ability to manage sessions and reinforce security with MSS Add-Ons. You'll gain additional critical functionality when you pair MSS with these products:

- ◆ **Security Proxy Add-On**

The Security Proxy acts as a proxy for terminal sessions and provides token-based access control, routing encrypted network traffic to and from user workstations. The Security Proxy Server can be installed on the same server as the Administrative Server or on another system.

To set up the Security Proxy for client authorization, pass through, end-to-end TLS, and end-to-end SSH security connections, see [“Connect to Hosts using the Security Proxy Add-On” on page 156](#).

- ◆ **Terminal ID Manager Add-On**

You can use the Terminal ID Management Add-On to monitor a pool of resource IDs that a client can use to establish a host session, thereby eliminating the need to create configurations for every client. The Terminal ID Manager enables you to pool terminal IDs, track ID usage, and manage inactivity timeout values for specific users, thus conserving terminal ID resources and significantly reducing operating expenses. The Terminal ID Manager can be installed on the same server as the Administrative Server or on another system. See [“Set Up Terminal ID Management for Reflection Desktop Sessions” on page 163](#).

- ◆ **Automated Sign-On for Mainframe Add-On**

The Automated Sign-On for Mainframe Add-On enables users to authenticate to a front-end system using a modern form of authentication (such as a smart card, certificate, LDAP password, Kerberos, etc.) and then be automatically logged on to a z/OS mainframe application. To add Automated Sign-On for Mainframe, you need to install the activation file and configure settings using the Administrative Console. Some configuration is also needed on the mainframe. See *“Set up Automated Sign-On for Mainframe Sessions” on page 166.*

Requirements and Best Practices for Using Centralized Management

Requirements for Centralized Management

To use centralized management, your environment must meet the following requirements.

- ◆ Reflection Desktop must be installed on the Administrative workstation and on user computers.
- ◆ The users' Reflection Desktop clients must be configured to use Centralized Management as shown in *Set up Reflection to Access a Centralized Management Server* in the *Reflection Help & HowTo Guide*. If user workstations are not configured with this setting, you will need to configure and save this workspace setting and then deploy it to users (see Package Sessions and Custom Settings Files).
- ◆ Java must be enabled in the browser you use to run the Administrative Console.
- ◆ Management and Security Server (MSS) must be installed on one or more servers available over the network. You will need to know the administrative credentials to log onto the Management Server.

Best Practices for using Centralized Management

- ◆ If your session uses associated settings (such as a theme or keyboard map file), be sure to select Send as compound session to include these settings. When you save the session as a compound file, all of the QuickPads, keyboard maps, themes, mouse maps, hotspots, and ribbons that apply to that session are saved in the session file. Compound files simplify the deployment process because you do not have to deploy these settings in separate files.

NOTE: Custom workspace settings are not saved in compound session document files. These settings are saved in other files that you will need to package and deploy separately.

- ◆ Be sure to choose to store all settings files in a trusted location.

Create and Deploy Sessions and Settings with the MSS Administrative Console

Using the MSS Administrative Console, you can define terminal emulation sessions, configure and save session settings, and then manage and configure secure settings for those sessions.

Create or Modify a Centrally Managed Session

You can create and manage Reflection sessions from the MSS Administrative Console. Sessions that you create in this way are saved to the server and can be made available to users from the server and modified at any time.

For instructions that show how to add or modify sessions, see the *Management and Security Server Administrator Guide*, in the [Host Access Management and Security Server Documentation](#).

Make Centrally Managed Sessions Available to Users

After you have saved sessions to the centralized management server, you can make these sessions available to users who have Reflection installed on their workstation. Users can launch these sessions in the same way they launch locally created sessions. When Centralized Management is configured, each time a user opens the workspace:

- ◆ Reflection contacts the server and prompts for user credentials (if required by the server).
- ◆ Sessions that are available to the user are downloaded to the user data directory.

For instructions that show how to make sessions available to users, see the *Management and Security Server Administrator Guide*, in the [Host Access Management and Security Server Documentation](#).

Use Central Management to Deploy MSI Packages

Use the Administrative Console to upload companion install packages (.msi) to the Administrative Server for deployment to specified users. Companion install packages can be created in the Installation Customization Tool or other MSI creation tools, and may include toolbars, macros, keyboard maps, and settings files.

NOTE: You can simplify deployment of MSI packages by saving your customized sessions as compound session document files.

These packages are automatically deployed to a user's desktop when the user logs on to the Management and Security Server or starts a Reflection Workspace session with Centralized Management enabled.

For instructions that show how to configure, upload, or update a package, see the *Management and Security Server Administrator Guide*, in the [Host Access Management and Security Server Documentation](#).

Enable Certificate Management for IBM Terminals

You can import certificates to the centralized management server and then make them available to Reflection users on their workstations. When configured to use certificate management, Reflection uses these managed certificates instead of the Reflection or Windows Certificate store.

To Upload a Certificate

- 1 Log on to the Management and Security Server Administrative Console as an administrator.
- 2 From the **Configure Settings** sidebar menu, select **Trusted Certificates**.
- 3 Select **Import** and upload your certificate to the Management and Security Server.
- 4 To configure Reflection Desktop to use certificate management, select the **Enable Certificate Management** setting on the Configure Centralized Management dialog box and then deploy this setting to users.

When launching the workspace with this setting enabled, Reflection uses the managed certificates instead of the Reflection or Windows Certificate store.

Enable Usage Metering

Usage metering allows you to track Reflection sessions and determine how many client workstations use the product.

When a user starts the Windows-based client session, the session notifies the Metering server, which begins logging the product usage. The session then sends updates to the Metering server at regular intervals until the user shuts down the client.

How can you use metering?

You can set up metering to monitor host sessions using the following features.

- ♦ **License Pool Configuration** A license pool comprises the licenses for a given product, type (production, evaluation, pre-release), and VPA number. When the server receives a metering message from a new product/type/VPA, a license pool is automatically created in the server's list of license pools.
- ♦ **Usage Logging** Connection activity is recorded in daily log files. You can specify the number of days the log files are stored before they are automatically deleted. The centralized management server uses the log data to generate reports summarizing usage information, such as connection activity, hourly usage levels, and connections from a specific client or to a specific host. You can use the Metering Reports page to create a variety of reports that use the logged data.
- ♦ **Connection Monitoring** When the client and server are configured for metering, you can monitor the number of user workstations connecting to host computers. You can also choose to receive e-mail notification when the concurrent number of workstations using metered clients increases beyond a specified limit. This notification feature also allows you to monitor concurrent product usage without enforcing a license limit.
- ♦ **Concurrent License Enforcement** The concurrent license enforcement feature allows you to control the total number of user workstations using Reflection to make host connections at the same time. When you configure metering, you can enter the maximum number of computers allowed to concurrently run Reflection; all attempted license use beyond this number can be blocked.

Setting up Metering

To enable metering, you must configure client workstations to report to the Metering server. You can do this by configuring metering with a transform as part of the initial installation or by setting group policy after Reflection is installed.

Requirements

- ◆ Micro Focus Management and Security Server (MSS) with a Metering server must be installed and configured for Reflection as shown in the *Management and Security Server Installation Guide*, available in the [Host Access Management and Security Server Documentation](#).
- ◆ You must know the complete URL for the Metering server. The syntax is: `http://<servername:port>/meter/AdminStart.html`. (If you used the default port, you can omit the colon and port number.)

For example:

```
http://<servername:port>/meter/AdminStart.html
```

To enable metering with a transform

- 1 From your administrative installation point, open the Installation Customization Tool from a shortcut or by typing the following command line:

```
<path_to_setup> \setup.exe /admin
```

- 2 In the Select Customization dialog box, select **Create a new Setup customization file for the following product**.
- 3 On the left panel, select **Licensing and session metering**.
- 4 Select to accept the terms of the license agreement on behalf of users.
- 5 In the **Meter As** list, select the product you want to meter.
- 6 In the **Metering URL** box, enter the URL of your metering server.

Syntax:

```
http://<servername:port>/meter/AdminStart.html
```

For example:

```
http://Myserver.com:80/meter/AdminStart.html
```

NOTE: If you use the default port, you can omit the colon and port number.

- 7 If you want to prevent users from launching Reflection when the metering server is not available, Select **Require metering**. (Enabling this setting can be useful when you are creating a trial installation and want to test to see if the metering server is running and available.)
- 8 If you are going to deploy using the `setup.exe` file in your administrative installation point, click **User Interface** in the left pane, and then select **Use this customization with interactive installs using setup.exe**.
- 9 Save your transform as an `.mst` file.

When you deploy this transform with your installation, the Reflection client is configured to use metering.

NOTE: Micro Focus recommends that you save transform files in the same folder as the Reflection base installation package .msi file.

To enable metering via group policy

- 1 Download and install the [administrative template file for Reflection \(https://download2.microfocus.com/fileinfo.asp?filename=ReflectionPolicy.zip\)](https://download2.microfocus.com/fileinfo.asp?filename=ReflectionPolicy.zip).
- 2 Open the Windows Group Policy Editor (`gpedit.msc`).
- 3 Under **Computer Configuration**, right-click on **Administrative Templates** and select **Add/Remove Templates**.
- 4 Click **Add**, select the Reflection.adm file you need to add, and then click **Open**. The added ADM file is listed in the Add/Remove Templates dialog box, in the **Current Policy Templates** list.
- 5 Open **Computer Configuration > Administrative Templates > Classic Administrative Templates (ADM) > Reflection Settings > Client Metering**.
- 6 Select the **Configure Client Metering** policy setting and then select to edit this setting.
- 7 In the Configure Client Metering dialog box, select **Enabled**.
- 8 In the **Metering web server** box, enter the URL of your metering server.

Syntax:

```
http://<servername:port>/meter/AdminStart.html
```

For example:

```
http://Myserver.com:80/meter/AdminStart.html
```

NOTE: If you use the default port, you can omit the colon and port number.

- 9 Select **Require connection to metering server** only if you want to prevent users from launching Reflection when the metering server is not available. (Enabling this setting can be useful when you are creating a trial installation and want to test to see if the metering server is running and available.)

Connect to Hosts using the Security Proxy Add-On

The Security Proxy Add-On acts as a proxy for terminal sessions and provides token-based access control, routing encrypted network traffic to and from user workstations.

NOTE: The Security Proxy Add-On requires the base installation of Host Access Management and Security Server. It is not included with the Management and Security Server license. To activate this product, you must purchase a separate license.

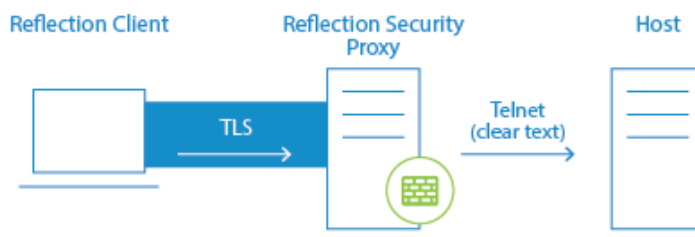
Using the Security Proxy Add-On, you can set up the following types of centrally managed secure connections.

Connect using... **Description**

Client Authorization When using the default configuration for the Security Proxy, users are authorized using security tokens. Transmitted data between the client and the Security Proxy is encrypted; transmitted data between the Security Proxy and the host is not. The Security Proxy server should be installed behind a corporate firewall when used in this mode.

Before you create a connection, review the [“Requirements for connections through the Security Proxy”](#) on page 159 and [“Getting to the Reflection Desktop Security Settings”](#) on page 160.

Then see the instructions for creating and assigning secure sessions in the *Management and Security Server Administrator Guide*, in the [Host Access Management and Security Server Documentation](#).



Pass Through When configured as a Pass Through Proxy, the Security Proxy passes data to the destination host without regard to content (that is, it ignores any TLS handshaking data) and does not provide client/server authentication or encryption. If TLS is used in this mode, the TLS session is created between the client and destination host and encrypted data simply passes through the Security Proxy.

You can secure data traffic using TLS between the client and the destination host by enabling TLS [user authentication](#) on the destination host. When using a Pass Through proxy, client authorization is not an option.

Before you create a connection, review the [“Requirements for connections through the Security Proxy”](#) on page 159 and [“Getting to the Reflection Desktop Security Settings”](#) on page 160.

For instructions that show how to configure the Security Proxy, see the *Management and Security Server Administrator Guide*, in the [Host Access Management and Security Server Documentation](#).

NOTE: If you want to establish an TLS-secured connection between Reflection and the destination host using the Security Proxy in Pass Through mode, you may need to unselect **Host name must match certificate** or, preferably, add the Security Proxy as the Subject Alternate name in the host server certificate.

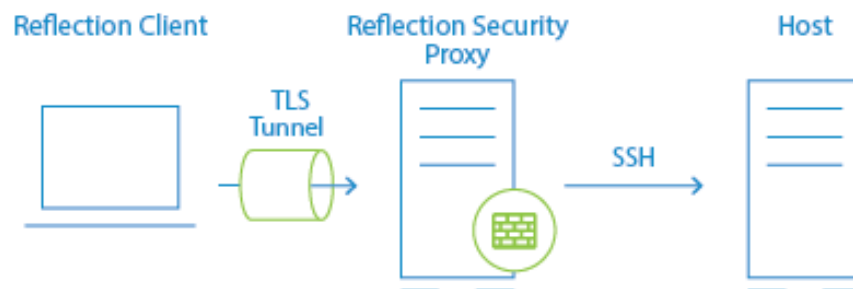
To create a Reflection Desktop session that connects to the Security Proxy, see the instructions for creating secure sessions in the *Management and Security Server Administrator Guide*.

Connect using...**Description****End-to-End TLS Security**

This option, available for 3270 sessions only, combines user authorization with TLS security for the entire connection. Single sign-on capability using the [IBM Express Logon Feature \(page 181\)](#) is also supported, provided the host supports TLS. See [Connect using End-to-End Security in 3270 Sessions](#).

**End-to-End SSH Security**

In a standard configuration for a secure Reflection session, the connection between the client and security proxy server is encrypted using TLS, but the connection between the security proxy server and the host uses unencrypted Telnet. By sending an SSH-encrypted connection through the security proxy tunnel, you can configure a secure Reflection session so that the entire communication path is encrypted from the client, through the proxy server, and on to the host. See [“Connect using End-to-End Encryption in VT SSH Sessions” on page 161](#)



Requirements and Other Information for Using the Security Proxy

Requirements for connections through the Security Proxy

Requirements for connections through the Security Proxy include:

- ◆ TLS must be enabled on the hosts to which you want to establish TLS connections. See the documentation included with the hosts for instructions.

- ◆ A centralized management server with the Security Proxy Add-On installed must be available. The Security Proxy must be configured to require client authorization. (It can optionally be configured to require client authentication. For client authentication, you can use a single certificate or two separate client certificates on each server (Security Proxy and destination host).
- ◆ You may need multiple digital certificates to successfully establish the TLS sessions between the client and the Security Proxy, and the client and the destination host.

Getting to the Reflection Desktop Security Settings

This section includes references to the *Management and Security Server Administrator Guide* in the [Host Access Management and Security Server Documentation](#). The procedures referenced in this guide does not include product-specific information for setting up TLS connections. When you create a session in Reflection Desktop, use the following steps to go to these security settings:

- 1 Enter the **Host name** and **Port**.
- 2 Select **Configure additional settings** to open the Settings window and then choose **Set Up Connection Security**.
- 3 In the **Configure Advanced Connection Settings** dialog box, click **Security Settings**.

About Certificates

Server Certificates

Destination TLS hosts and Security Proxy servers typically have server certificates already installed. Each of these server certificates must be trusted by the client. The client will trust a server certificate if:

- ◆ It is signed by the certificate authority that is trusted by the client, or
- ◆ It is self-signed and imported into the trusted root certificate store where Reflection can find it.

To use a single server certificate for both the destination host and the Security Proxy, do one of the following:

- ◆ In the Reflection session, unselect the **Verify Server Identity** check box on the **Connection Editor** dialog box.
- ◆ (Recommended) Create a certificate that uses the destination host address for the **Subject Common Name** and the Security Proxy address for the **Subject Alternative Name**.

Client certificates

Certificates used for client authentication must be signed by a certificate authority that is trusted by both the Security Proxy and the destination host's TLS server.

Express Logon also requires that the client certificate used to authenticate on the TN3270 server be registered with RACF. (For details, see the documentation that came with the 3270 server.)

For more details on configuring TLS and creating certificates on the host, see Technical Note 1760 (<https://support.microfocus.com/kb/doc.php?id=7022182>)

Connect using End-to-End Encryption in 3270 TLS Sessions

End-to-end encryption tunnels a direct TLS connection to the host, while still connecting through the Security Proxy Server. These connections require two certificates and TLS handshakes — one for the client/proxy server connection and another for the client/host connection. Without end-to-end encryption, only data between the client and proxy server is encrypted.

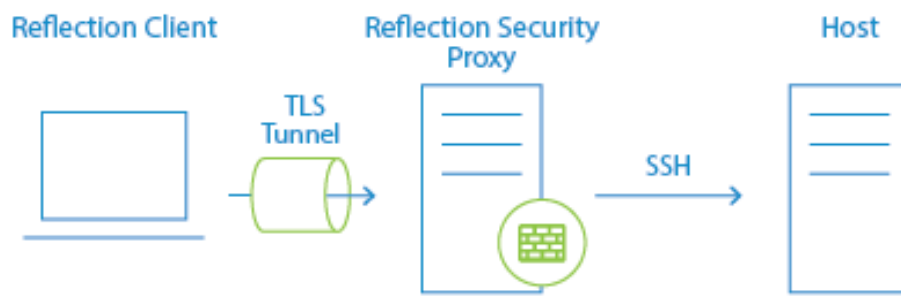
Before you create a connection, review the [“Requirements for connections through the Security Proxy”](#) on page 159 and [“Getting to the Reflection Desktop Security Settings”](#) on page 160.

To configure a session with end-to-end encryption, see the instructions for creating secure sessions in the *Management and Security Server Administrator Guide* in the [Host Access Management and Security Server Documentation](#).

NOTE: You can modify the level of security by adjusting the TLS protocol version and encryption key-strength setting. Click **PKI Manager** to add the Certificate Revocation List (CRL) and Online Certificate Status protocols (OCSP) to certificate validation.

Connect using End-to-End Encryption in VT SSH Sessions

You can configure a Reflection Desktop session to send an SSH-encrypted connection through the Security Proxy Server.



In a standard Administrative Console configuration for a secure Reflection session, the connection between the client and security proxy server is encrypted using TLS, but the connection between the security proxy and the host uses unencrypted Telnet. By sending an SSH-encrypted connection through the security proxy tunnel, you can configure a secure Reflection session so that the entire communication path is encrypted from the client, through the proxy server, and on to the host. This feature has the following advantages:

- ◆ Encryption is used for the entire connection.
- ◆ The IP addresses and names of your secure hosts are not exposed outside of the internal network.
- ◆ Only clients with a valid authorization token can launch a secure session.

- ◆ The authorization token contains connection information. This enables the security proxy to send all secure host connections through a single port, eliminating the need to open multiple firewall ports.
- ◆ All settings required for a connection (such as the trusted certificate, the personal certificate, user keys, and host keys) reside on the Administrative Console and are downloaded to users' workstations when they start sessions.

You can set up this configuration using the Reflection VT Terminal type (used for UNIX and OpenVMS sessions).

Prerequisites

To make these SSH connections through the Security Proxy, you must have the following:

- ◆ The host must have an SSH server installed.
- ◆ Reflection Desktop v16 or Reflection 2014 R1 SP1 must be installed locally on your workstation. Note: Earlier versions and other products (such as Reflection 2014 R1, Reflection 2011, Reflection 14.x, or Extra!) are not supported.
- ◆ You must have access to the centralized management server Security Proxy and Administrative Console.
- ◆ Host Sessions must be opened from the Administrative Console or the Links List page.
- ◆ Sessions must be configured from the Administrative Console in the centralized management server.
- ◆ The Security Proxy must be running with Client authorization enabled.

NOTE: Management and Security Server is not licensed for connections from Reflection Desktop clients. You must have a Management and Security Server activation file installed to configure Reflection Desktop sessions.

To connect your VT session SSH connection through the Security Proxy Add-On

- 1 Follow the instructions for creating a new session in the *Management and Security Server Administrator Guide* in the [Host Access Management and Security Server Documentation](#) to create a new VT session.
- 2 When you configure the session, select **Secure Shell** for the connection type.
- 3 Enter the host name and user name (optional; users are otherwise prompted when they connect). Then select **Configure additional settings** and click **OK** to open the Settings dialog box.
- 4 Under **Host Connection**, click **Set up Connection Security**.
- 5 In the Reflection Secure Shell Settings dialog box, on the Reflection Security Proxy tab, select **Use Security Proxy**, and then choose a Security proxy and a Proxy port.

NOTE: The Destination host values you entered in step 6 should be entered automatically here. If you don't see them, select the Security proxy name from the drop-down list to populate these fields.

- 6 Configure SSH connection settings such as the trusted certificate, the personal certificate, user keys, and host keys as required for your connection. For more information about configuring your SSH-specific settings, refer to the Reflection Help topic [Secure Shell Settings Dialog Box](#).

- 7 Click **OK** to close the open dialog boxes and initiate the connection. Select **Always** to import the host key for these sessions.

NOTE: If you do not want to include the user name in the configuration, cancel the connection. If you cancel, you will be unable to import the host key for the session.

- 8 Save the session. When prompted, choose to send the settings for this session to the Administrative Console, and then exit the Reflection workspace.

All the files required for your configuration are uploaded to the Administrative Console. When a user launches the session, these files are downloaded to their workstation so that Reflection has access to all configuration data required to establish a connection.

NOTE: All non-default SSH settings required to establish a connection are saved in three files:

- ♦ The `sessionname.rssh` file contains the public key (if public key authorization is used), the host key (if a host key is accepted while in administrative mode), and the settings normally stored in both the `pki_config` file and the `config` file. It also includes all TLS settings such as the TLS version, cipher suites, and applicable proxy data.
- ♦ The `sessionname.ps` file stores any personal certificates included for the connection.
- ♦ The `sessionname.ts` file includes any trust certificates.

When you send settings for the session to the Administrative Console, these files are uploaded along with the session document file.

Set Up Terminal ID Management for Reflection Desktop Sessions

The Terminal ID Management Add-On configures and monitors a pool of resource IDs that a client can use to establish a host session, thereby eliminating the need for administrators to create configurations for every client. Depending on the type of terminal or printer, these resource IDs may represent addresses or identifiers as required. You can use ID Manager with the following types of Reflection terminals and printers:

- ♦ IBM 3270 Terminal
- ♦ IBM 3270 Printer
- ♦ IBM 5250 Terminal
- ♦ IBM 5250 Printer

NOTE: The Terminal ID Management Add-On requires the base installation of Host Access Management and Security Server. It is not included with the Management and Security Server license. To activate this product, you must purchase a separate license.

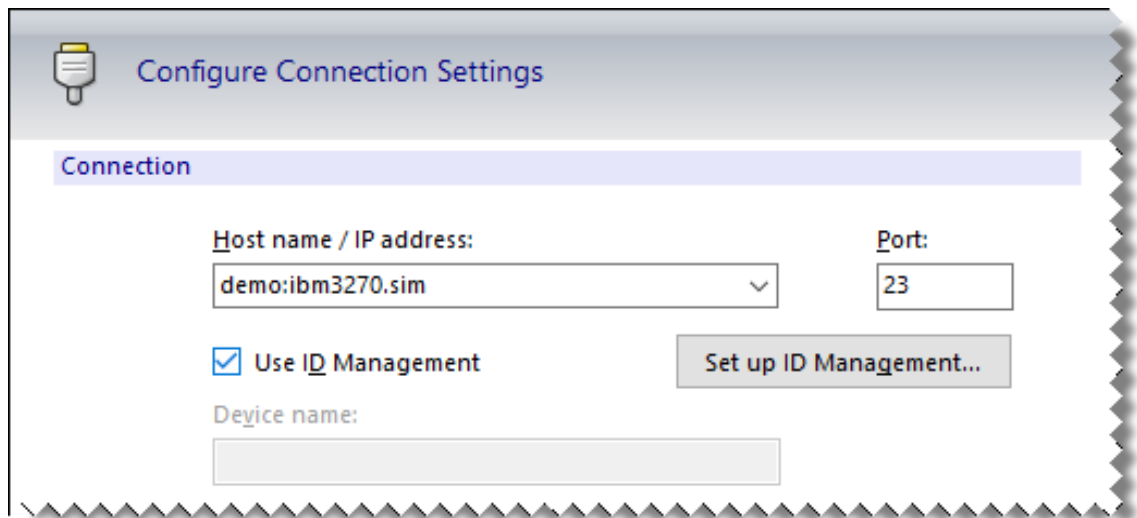
Prerequisites

To use ID Manager, you need access to a Management and Security Server with the Terminal ID Management Add-On configured.

For instructions that show how to configure ID Manager, see the *Terminal ID Manager Guide* in the [Host Access Management and Security Server Documentation](#):

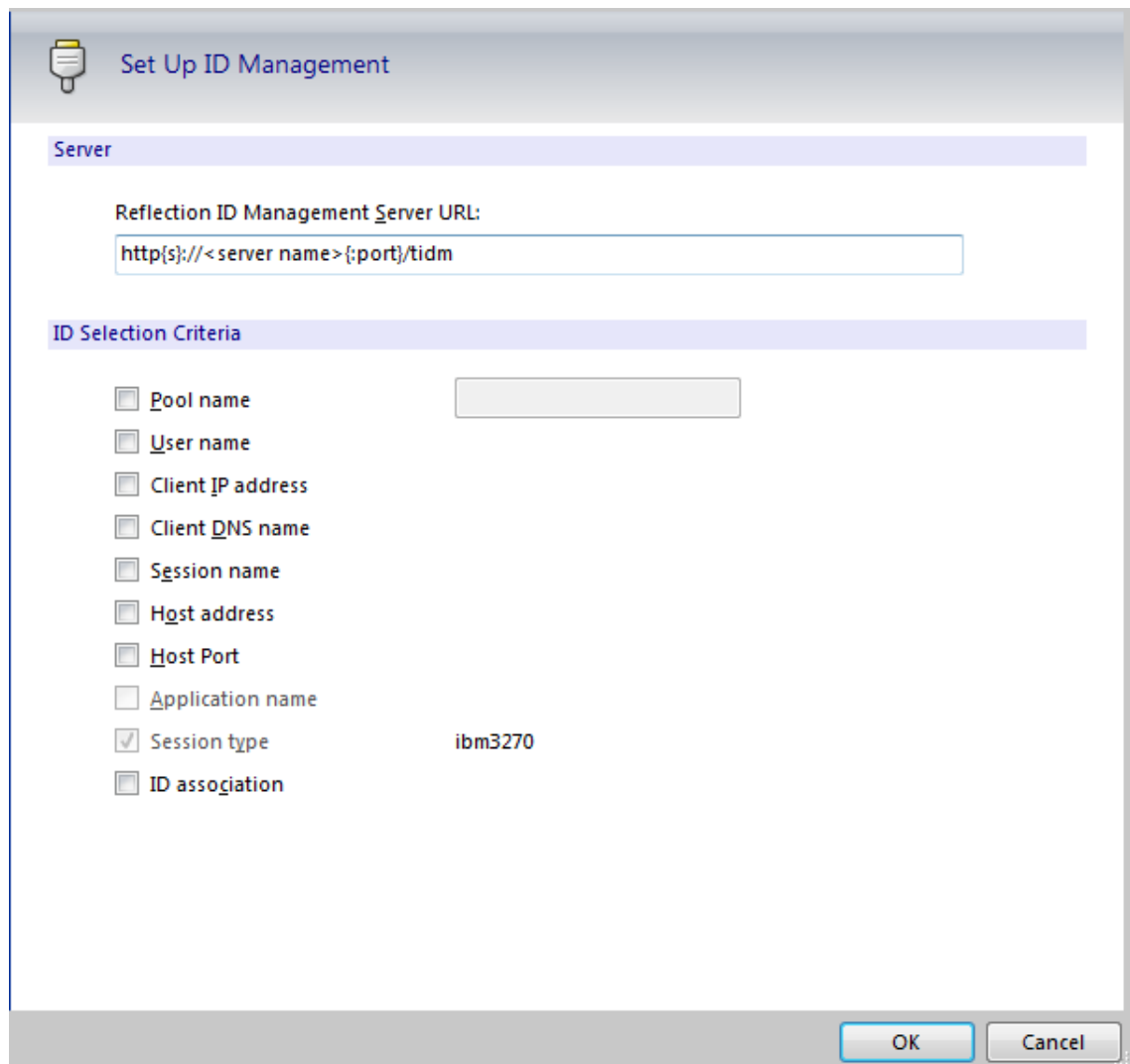
To create a terminal session that accesses IDs from the Terminal ID Management Server every time it connects

- 1 Make sure that you have the following information for the Terminal ID Management Server:
 - ◆ The complete URL. For example, `http://server.name/tidm`, where `tidm` is typically case sensitive, but `server.name` is not.
 - ◆ The parameters required by the Terminal ID Management Server to allocate an ID, such as a pool name).
- 2 Open the Document Settings dialog box by going to **File > Settings > Document Settings**. Under **Host Connection**, select **Configure Connection Settings**. Under **Connection**, select **Use ID Management** and then click **Set up ID Management**.



The screenshot shows the 'Configure Connection Settings' dialog box with the 'Connection' tab selected. The 'Host name / IP address' dropdown menu is set to 'demo:ibm3270.sim'. The 'Port' field is set to '23'. The 'Use ID Management' checkbox is checked, and the 'Set up ID Management...' button is visible. The 'Device name' field is empty.

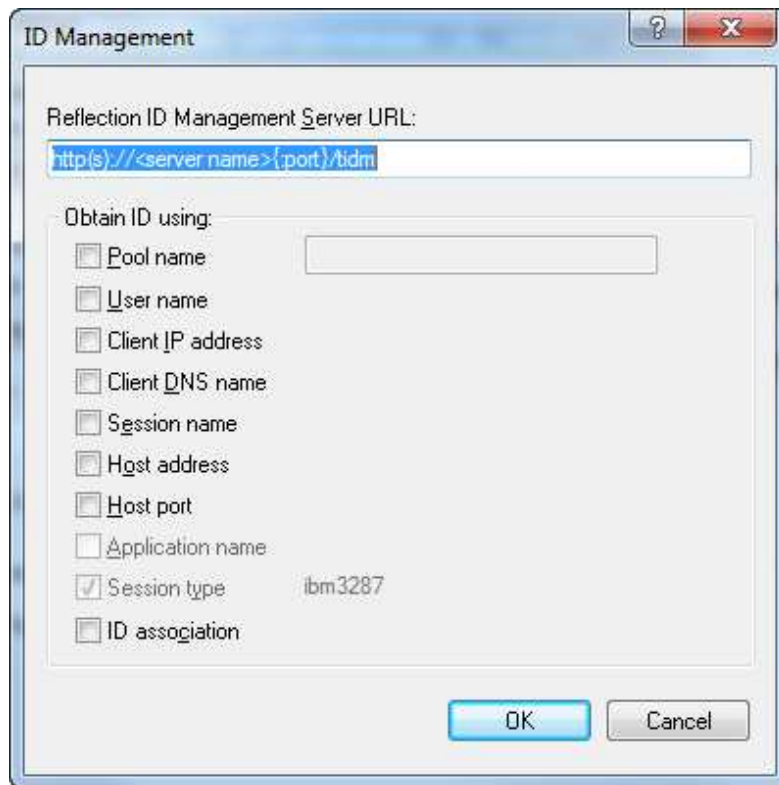
- 3 Under **Server**, enter the **Reflection ID Management Server URL**.



- 4 Select the **ID Selection Criteria**, such as **Pool name**.

To configure a printer session document to access IDs from the Terminal ID Management Server every time it connects

- 1 Make sure that you have the following information for the Terminal ID Management Server:
 - ♦ The complete URL. For example, `http://server.name/tidm`, where `tidm` is typically case sensitive, but `server.name` is not.
 - ♦ The parameters required by the Terminal ID Management Server to allocate an ID, such as a pool name).
- 2 Open a Reflection Printer Session Document File.
- 3 From the **Connection** menu, choose **Session Setup**.
- 4 Under **Transport**, select **Use ID Management** and then click **Set up ID Management**.



- 5 In the Set up ID Management dialog box, enter the **Reflection ID Management Server URL**.
- 6 Select the **ID Selection Criteria**, such as **Pool name**.

Set up Automated Sign-On for Mainframe Sessions

Using the Automated Sign-On for Mainframe Add-On, you can enable a user to authenticate to a front-end system using a modern form of authentication (such as a smart card, certificate, LDAP password, Kerberos, etc.) and then be automatically logged on to a z/OS mainframe application.

NOTE: The Automated Sign-On for Mainframe Add-on requires the base installation of the Management and Security Server (MSS), which provides the Administrative Server. This add-on is not included with the Management and Security Server license. To activate this product, you must purchase a separate license.

Automated Sign-On solves problems associated with credentials typically required for mainframe applications. Mainframe applications prompt for traditional credentials (a user name and password) and are typically hard-coded to accept a maximum of 8 characters for these credentials. Changing the password to match the user enterprise password is often not practical because of the mainframe limits on password character length and coordination of password changes. Because of this limitation, logging on to mainframe applications requires an identify that is separate from the user enterprise identity.

This add-on solves this problem by providing middleware that maps the user enterprise identity to the user mainframe identity. When using a Reflection session configured to use Automated Sign-On, the user authenticates to the front-end system using a modern authentication method. After authentication through the front-end system, the user is automatically logged into the host application.

Implementing Automated Sign-On for Mainframe

To implement Automated Sign On, you'll need to configure the Administrative Server, the Reflection emulation client session, and the z/OS mainframe as shown in the [Automated Sign-On for Mainframe Administrator Guide](#).

The process for setting up automated sign-on depends on your environment

Process for Centrally Managed Environments

If your environment is set up to centrally manage and control all session document files on an MSS server and you do not allow users to create their own session documents, use the following instructions in the [Automated Sign-On for Mainframe Administrator Guide](#).

- ◆ *Reflection or InfoConnect Desktop - Managed Sessions* in the *Enable your emulator for automated sign-on* section

Process for Unmanaged Environments

If your environment is not set up to centrally manage session document files with MSS and you want to use MSS only for automated sign-on, use the following instructions in the [Automated Sign-On for Mainframe Administrator Guide](#).

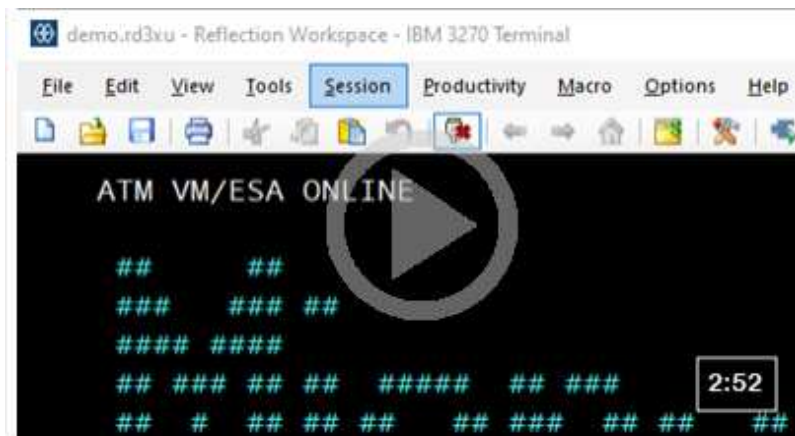
- ◆ *Reflection or InfoConnect Desktop - Workspace Automated Sign-on* in the *Enable your emulator for automated sign-on* section

13 Move from Extra! to Reflection

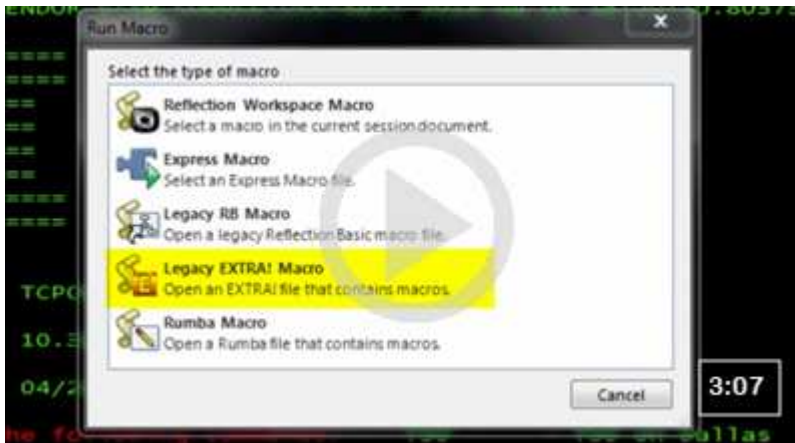
This guide includes key issues for system administrators to consider when you are preparing to deploy Reflection Desktop to an Extra! install base.

- ♦ [What's Similar?](#) describes similar features and capabilities of Reflection and Extra!. The process for customizing sessions is similar and many of the custom files you created in Extra! can be imported into Reflection. You can also run and edit Extra! macros.
- ♦ [What's Changed](#) compares the new Reflection tools and processes with those of Extra! This includes packaging and deploying with the Reflection Installation and Customization Tool, locking down and changing the interface with Reflection's Permissions Manager tool, and using compound session document files that include many of your settings in a single file.
- ♦ [What's New?](#) shows new capabilities of Reflection including flexibility in setting application (workspace) settings, new security features, and installer improvements. It also highlights other features, including the ability to decorate the screen with controls, map actions to events, and create Visual Basic for Applications (VBA) and Express Macro Language (EML) macros.
- ♦ ["Best Practices for Moving from Extra!"](#) on page 178 lists some things to check to make sure you avoid common problems and mistakes when you deploy Reflection.

For other information about upgrading from Extra! to Reflection Desktop, see the following videos:



Setting up Reflection Desktop with a User Interface like Extra!'s



Running Extra! Macros in Reflection Desktop

What's Similar?

Micro Focus has put forth a lot of effort to make it easier for you to upgrade to Extra!

- ◆ Reflection uses many of the same types of settings as Extra! and you'll find that many of the settings you configured for Extra! can be easily imported into Reflection, including session files, Hotspots, QuickPads, and keyboard maps.
- ◆ You can heavily customize Reflection, just as you can with Extra!. Although the terminology for settings is a little different, you can configure most of the same types of custom settings you set in the Extra! Settings window in the Reflection Settings window.
- ◆ You can use Reflection group policy templates to lock down Reflection, similar to how you use templates to lock down Extra!.
- ◆ You can run and edit Extra! macros, although more modern languages and APIs are available, including: the Reflection Microsoft Visual Basic for Applications API, a .NET API, and the Express Macro Language.
- ◆ You can deploy Reflection using group policy, MSI commands or deployment tools.
- ◆ Although Reflection is configured by default to use a modern ribbon interface, you can configure it to use an alternate interface that is similar to Extra! to ease adoption for your users.
- ◆ Some of the settings in the Extra! Global Preferences dialog box are available in the Reflection Workspace settings.

What's Changed?

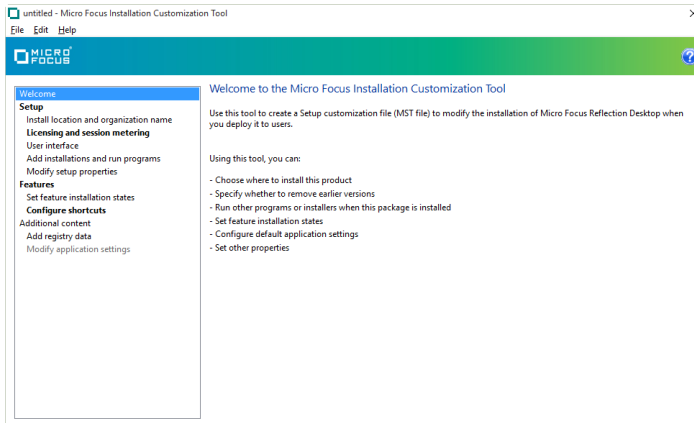
Reflection provides comparable tools with additional and enhanced capabilities:

- ◆ [“The Installation Customization Tool” on page 171](#) is used to modify the install and package files. This tool replaces the Custom Installation Wizard (CIW).
- ◆ [“Permissions Manager” on page 171](#) is used to lock down access to settings and controls. This tool provides many benefits over the Extra! Security Editor.
- ◆ [“Compound Session Documents” on page 173](#) allow you to save many of your custom settings (such as keyboard maps) in a Reflection session file.

The Installation Customization Tool

Instead of the Installation Customization Wizard, Reflection provides the Installation Customization Tool that you can use to package your customized files in companion MSI files or to create transforms that you deploy with the Reflection MSI.

For your production deployment, you'll need to install this tool on an administrative installation point on a server and then run it from your workstation.



For more information about setting up and using the Installation Customization Tool, see:

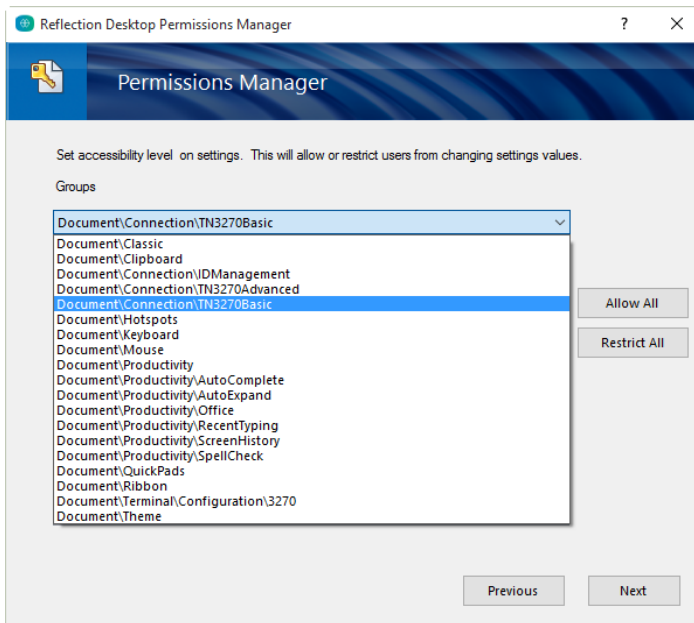
[Setting up the Reflection Administrative Tools](#)

[Package Sessions and Custom Settings Files](#)

[Modify the Installation](#)

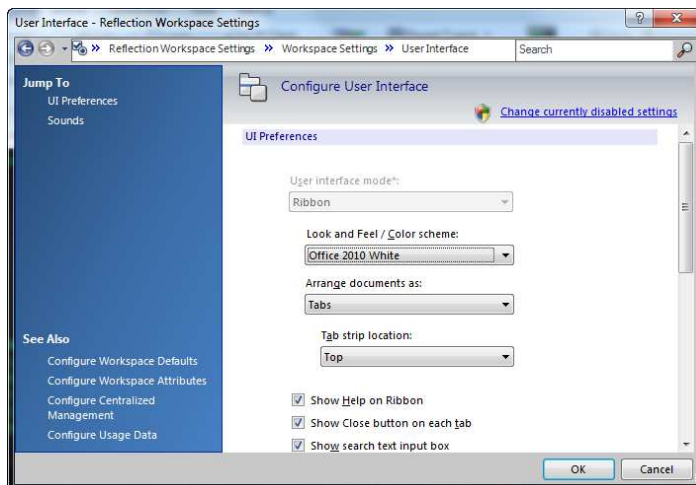
Permissions Manager

Instead of using the Security Editor to remove controls or settings you don't want to be available to users, you'll use the Reflection Permissions Manager administrative tool or Microsoft Group Policy to restrict access to these controls. These methods allow Reflection to leverage the Microsoft User Access Control (UAC) to restrict access to the interface.



Restricting access through the UAC offers significant advantages over simply removing the controls that improve your ability to manage the product after it is deployed. If you need to change a setting for a user, you can make the change without deploying (or redeploying) any files.

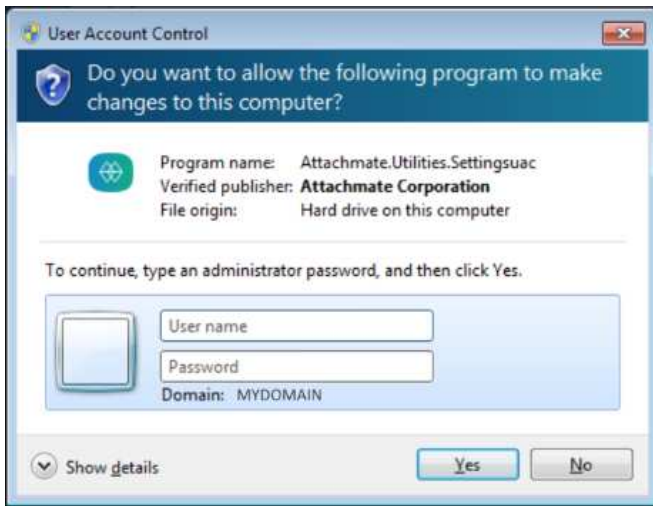
For example, in the settings dialog box shown below, the **User interface mode** setting is restricted.



A shield is displayed, indicating this dialog box contains restricted controls and the restricted control is grayed out.

NOTE: Restricted controls are grayed out or disabled for all Reflection interfaces. In the Reflection ribbon interface and on Reflection dialog boxes, a shield indicates controls are restricted.

If a user selects **Change the currently disabled settings**, the User Account Control dialog box is displayed.



The user must contact an administrator (such as a Help desk administrator) and ask them to change the setting. The administrator can then enter their credentials and change the setting without logging on to the machine. After the setting is changed, the control remains restricted.

If you want to completely remove controls from the interface, use the UI Designer to modify the Toolbars or menus as shown in [Customize Classic Menus and Toolbars](#) (for Classic interface mode) or [Customize the Ribbon](#).

Permissions Manager (`accessconfig.exe`) and Reflection Group Policy templates are installed in the Reflection installation directory. Permissions Manager settings are saved as `.access` files and can be packaged and deployed to users in MSI files created with the Installation Customization Tool.

For more information, see:

[Chapter 8, “Control Access to “Lock Down” Settings and Controls,” on page 103](#)

Compound Session Documents

You can set up and customize session files that use custom keyboard maps, controls, themes, and other settings. To customize these settings, see [Chapter 3, “Create and Customize Sessions,” on page 39](#).

You can save these session settings as compound session files or standard session document files (the default).



Compound Session Document files include all of the customized settings for Quick-pads, keyboard maps, themes, mouse maps, Hotspots, and ribbons. These files allow you to deploy your sessions without having to deploy dependent files for these settings.

Session Document files save these settings in separate files. If you save sessions using this default, you'll need to make sure that you deploy the custom files to the correct locations as shown in [Customized Files that Must be Deployed to Specific Locations](#).

Compound files are easier to deploy because you don't have to deploy the supporting files. This can be especially useful when deploying to a centralized environment, such as a Citrix server. Regular session files offer more flexibility for sharing common configurations. For example, you can share one keyboard map file in several regular session files. With compound session documents, you have to recreate the custom settings in each file.

What's New?

You'll find many new features in Reflection Desktop. These features combine the best emulation features of Reflection and Extra! into a single solution, optimized for Microsoft Windows 10 and Microsoft Office.

- ◆ [“Installation Improvements” on page 174](#) include the advantages of using the Reflection Setup program and the Reflection Installation Customization tool.
- ◆ [“Workspace Settings” on page 175](#) allow you to set many global application settings that are not available in Extra!.
- ◆ [“The Trust Center” on page 175](#) is used to protect sensitive data (such as credit cards), specify secure locations for opening files, and control programmatic access to Reflection.
- ◆ [“User Interface Options” on page 177](#) include a number of alternate interfaces. The most commonly used interfaces can be heavily customized.
- ◆ [“Other Features” on page 178](#) include features for customization and automation that are not available in Extra!.

Installation Improvements

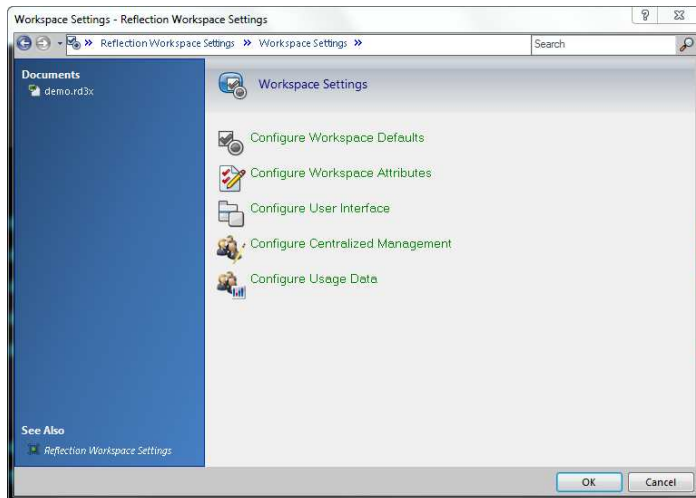
The Reflection Setup program (setup.exe) is the recommended tool for installing and deploying Reflection. This tool uses the primary Reflection MSI file to install Reflection but it also installs prerequisite software (if needed) and has several other features that provide a smoother deployment than installing directly with the primary Reflection MSI file. When the Setup Program installs Reflection, it determines whether each workstation has the required .NET Framework and Microsoft Windows Installer version and automatically installs them if necessary. It also automatically uses the correct language for the installation and removes previous versions of Reflection. (This is required to install the new version.) If the Visual Basic feature is selected, it also installs the Visual Basic core MSI, along with the appropriate Visual Basic language MSI.

The Reflection Installation Customization Tool allows you to customize settings that apply to the application and automatically sets the correct location for these files in the MSI.

Workspace Settings

Global settings that apply to the Reflection application are configured in Reflection Workspace Settings. Most of these Workspace settings are not available in Extra!.

Workspace configuration settings affect all terminal session and Web page documents opened in Reflection. These settings include security, user interface options, file locations, and other settings related to Reflection.



Configure Workspace Defaults

Configure the actions to perform when the Reflection workspace opens (such as startup macros) or closes and preferences for automatically saving session document files.

Configure Workspace Attributes

Configure options for logging, running remote sessions, and displaying Help. You can also specify the user data directory, in which session documents and other related files are saved.

Configure User Interface

Configure which type of user interface to use (Reflection provides a number of alternate interfaces), its look and feel, and other user interface options.

Configure Centralized Management

Set up the workspace to access sessions from a centralized management server.

Configure Usage Data

Choose whether to participate in the Product Experience Improvement program.

The Trust Center

Use the Trust Center to protect your working environment from information theft, and your data from potential damage caused by opening documents from non-trusted sources. The trust center is a Reflection Desktop feature that is not available in Extra!.

The Trust Center includes three types of global security settings.

Specify Trusted Locations	Set up trusted locations (directories that are designated as secure sources for opening files). By default, Reflection allows you to open documents only in directories that are specified as trusted locations in the Reflection settings.
Set Up Information Privacy	Configure Information Privacy features to protect sensitive data so that it is not displayed on the screen or in productivity features, such as Screen History.
Set Up API and Macro Security	Enable the Reflection .NET API, and specify corresponding settings.

Trusted Locations

A trusted location is a directory that is designated as a secure source for opening files. By default, Reflection Desktop allows users to open documents only in directories specified as trusted locations in the Reflection settings.

NOTE: Reflection Desktop specifies three trusted locations in the workspace `Application.settings` file in the program directory. If you want users to be able to open or save documents in a location other than one of these default directories, you'll need to configure and deploy the trusted location settings.

When you add other locations, these locations are saved in a custom `Application.settings` file in the user data directory folder. If you add trusted locations, you will need to deploy this file.

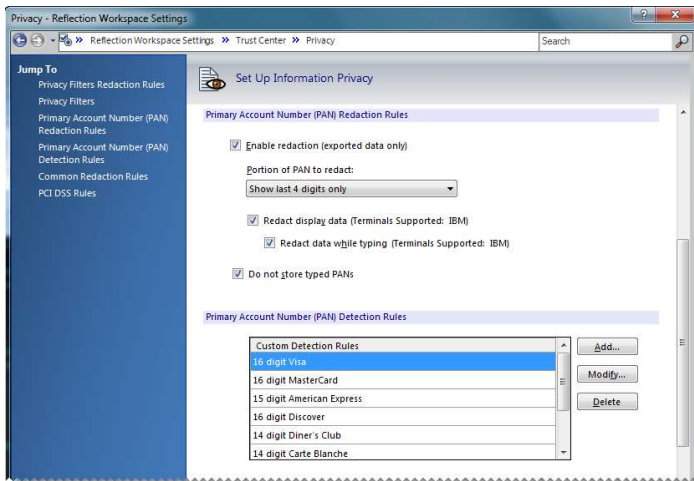
For more information, see [Add Trusted Locations](#)

Information Privacy

In Extra!, you can set privacy filters to mask patterns of data, such as Social Security numbers or phone numbers.

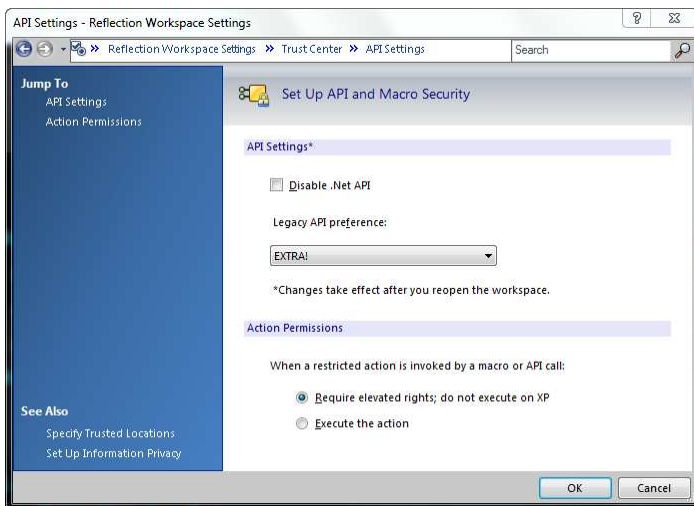
Reflection expands this capability to automatically recognize and redact credit card numbers as they are entered on the screen and allows you to use regular expressions for advanced pattern recognition.

Reflection Information Privacy settings are global and are configured in the Information Privacy dialog box.



API and Macro Security

If you plan to use Extra! macros, make sure that Extra! is selected in the Set up API and Macro Security dialog box.



See [Configure API and Macro Security](#)

User Interface Options

The Ribbon is the default interface for Reflection but you can configure Reflection to use the Classic interface, which is very similar to Extra!.

NOTE: Reflection provides a number of alternate interfaces: the Ribbon is the modern interface used for many Windows products, Classic is similar to Extra!, and Classic MDI is similar to Classic with a multi-document interface. The Browser interface has a look and feel of a Web browser, and TouchUx provides a mobile look and feel.

Reflection's UI Designer allows you to heavily customize the interfaces used for the Ribbon, Classic, and Classic MDI modes.

Other Features

You'll find many new features in Reflection Desktop that combine the best emulation features of Reflection and Extra! into a single solution, optimized for Microsoft Windows 10 and Microsoft Office.

You can customize sessions to set up custom controls for your program screens, configure Reflection to initiate actions when events occur in a session, choose from several user interfaces to address different preferences and needs, and configure many other new settings to improve the user experience.

Using the Reflection object-oriented API model, you can add functionality to terminal sessions and Web pages using .NET, Visual Basic for Applications (VBA), or HLLAPI programming. You can also use the Express Macro Language to develop macros that can be used in multiple Micro Focus products.

Best Practices for Moving from Extra!

Use these guidelines to avoid problems when you are upgrading to Extra!.

✓ **Set up Trusted Locations**

The Trusted Locations feature provides a way to differentiate safe files from potentially harmful files. When a file is in a trusted location, it's files are assumed to be safe. Reflection enforces trusted locations by default, so if you want to save sessions in directories that are not default trusted locations, you'll have to define these directories or disable trusted locations. See [Add Trusted Locations](#) in the Reflection Deployment Guide.

✓ **Set up API and Macro Security for Extra!**

If you plan to run Extra! macros, set up API and Macro Security to use Extra! for the Legacy API preference.

✓ **Install Extra! compatibility**

Make sure that the Extra! compatibility feature is selected in the installer.

✓ **Test setting with a user account**

If you deploy to custom user data or machine data folders, make sure that you test your settings with a user account instead of just your administrator account.

✓ **Determine which Reflection interface works the best for your users**

Determine whether to set up Reflection with a Classic user interface that is very similar to Extra! or with a more modern Ribbon interface (the default). (Reflection provides optional interfaces designed to meet different needs and preferences.)

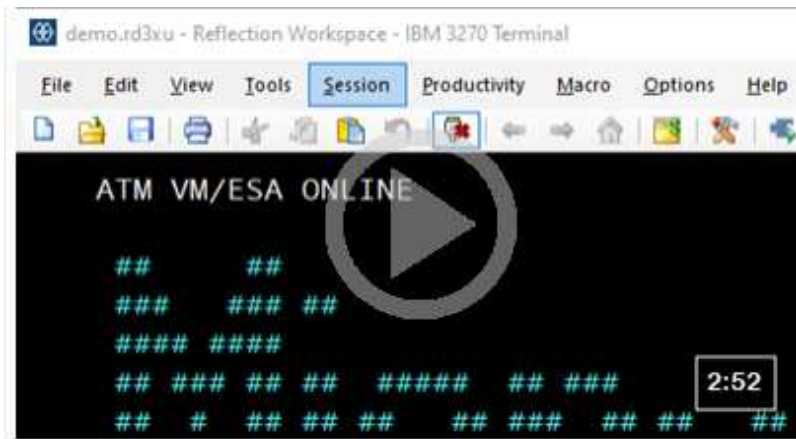
✓ **Consider using compound session documents, especially for centralized environments**

If you are deploying to Citrix or another centralized environment, consider saving sessions and related files in compound session document files.

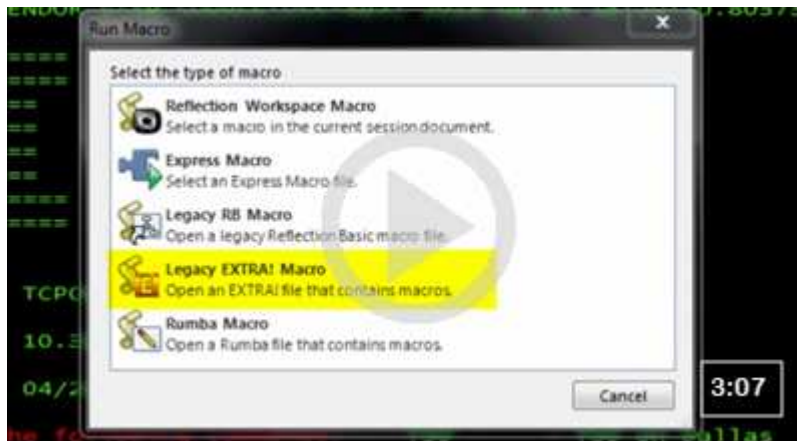
- ◆ Follow [“Best Practices for Planning a Reflection Deployment”](#) on page 19.

Videos for Moving from Extra!

For more information about migrating from Extra! to Reflection Desktop, see the following documents and videos:



Setting up Reflection Desktop with a User Interface like Extra!'s



Running Extra! Macros in Reflection Desktop

Glossary

Reflection database. The Reflection database (`ic32.cfg`) contains connection settings information for ALC, T27, and UTS terminal sessions. The database contains information about all the Reflection packages, path templates and libraries that have been installed, as well the paths that have been created. The Reflection packages, path templates and libraries are included based on which product features (emulations and transports) are installed.

Reflection global application data folder. Settings here apply to all users of the system. The location is version-specific: `\ProgramData\Micro Focus\Reflection\Desktop\v17.0`.

Reflection global ssh data folder. Reflection stores global Secure Shell information in the Windows common application data folder. The default is `\ProgramData\Micro Focus\Reflection`.

Reflection program folder. The default on English language systems is `C:\Program Files (x86)\Micro Focus\Reflection` on 64-bit systems and `C:\Program Files\Micro Focus\Reflection` on 32-bit systems.

Reflection user application data folder. The default is `\Users\username\AppData\Roaming\Micro Focus\Reflection\Desktop\v17.0`.

Reflection user data folder. This folder location is configurable using the **Data Location** tab during installation. The default is `C:\Users\username\Documents\Micro Focus\Reflection`.

Reflection user ssh folder. Reflection stores Secure Shell information for individual users in the following location in the Windows personal documents folder. The default is `\Users\username\Documents\Micro Focus\Reflection\.ssh`.

AppDataFolder property. The full path of the Roaming folder for the current user. The default is `C:\Users\username\AppData\Roaming\`.

authentication. The process of reliably determining the identity of a communicating party. Identity can be proven by something you know (such as a password), something you have (such as a private key or token), or something intrinsic about you (such as a fingerprint).

Auto Expand. Use the Auto Expand feature to add acronyms or shortcuts for long words, phrases, or complex repeat commands. The shortcut, when typed and followed by the Spacebar, automatically expands to the full word or phrase.

cipher. A cipher is an encryption algorithm. The cipher you select determines which mathematical algorithm is used to obscure the data being sent after a successful Secure Shell connection has been established.

Client authentication. Client authentication (also referred to as *user authentication*) requires users to prove their identity using digital certificates (the default setting for the Reflection Security Proxy).

Client authentication is typically required when an SSL session is first established. It will also be required by a TN 3270 server if the user is using the Express Logon Feature provided by some mainframe systems.

Client authorization. Used in connections secured by the Reflection Security Gateway to ensure that access to host systems is approved before the connection can proceed.

When a user logs into the Reflection Security Gateway, he or she only has access to terminal session files and other features for which he has been explicitly authorized to use.

CommonAppDataFolder property. The full path to application data for all users. The default is `C:\ProgramData`.

CRL (Certificate Revocation List). A digitally signed list of certificates that have been revoked by the Certification Authority. Certificates identified in a CRL are no longer valid.

digital certificate. An integral part of a PKI (Public Key Infrastructure). Digital certificates (also called X.509 certificates) are issued by a certificate authority (CA), which ensures the validity of the information in the certificate. Each certificate contains identifying information about the certificate owner, a copy of the certificate owner's public key (used for encrypting and decrypting messages and digital signatures), and a digital signature (generated by the CA based on the certificate contents). The digital signature is used by a recipient to verify that the certificate has not been tampered with and can be trusted.

digital signature. Used to confirm the authenticity and integrity of a transmitted message. Typically, the sender holds the private key of a public/private key pair and the recipient holds the public key. To create the signature, the sender computes a hash from the message, and then encrypts this value with its private key. The recipient decrypts the signature using the sender's public key, and independently computes the hash of the received message. If the decrypted and calculated values match, the recipient trusts that the sender holds the private key, and that the message has not been altered in transit.

Express Logon Feature (ELF). Also referred to as *single sign-on (SSO)*, express logon is an IBM mainframe feature that lets users log on and connect to the host without entering a user ID and password each time. Express Logon authenticates the user on the mainframe by using her SSL client certificate in lieu of entering a user ID and password.

FCC. Field Control Character. A UTS terminal field attribute.

hash. Also called a message digest, a hash or hash value is a fixed-length number generated from variable-length digital data. The hash is substantially smaller than the original data, and is generated by a formula in such a way that it is statistically unlikely that some other data will produce the same hash value.

KDC (Key Distribution Center). The security server that maintains the database of principal information, uses the information in the database to authenticate users, and controls access to kerberized services in a realm.

keyboard map. A keyboard map is a configuration file that allows you to use your PC keyboard as a host terminal keyboard. Keyboard maps also include definitions for keyboard shortcuts.

OCSP (Online Certificate Status Protocol). A protocol (using the HTTP transport) that can be used as an alternative to CRL checking to confirm whether a certificate is valid. An OCSP responder responds to certificate status requests with one of three digitally signed responses: "good", "revoked", and "unknown". Using OCSP removes the need for servers and/or clients to retrieve and sort through large CRLs.

passphrase. A passphrase is similar to a password, except it can be a phrase with a series of words, punctuation, numbers, white space, or any string of characters. Passphrases improve security by limiting access to secure objects, such as private keys and/or a key agent.

PCI DSS. PCI DSS (Payment Card Industry Data Security Standard) is a worldwide standard comprising technology requirements and process requirements designed to prevent fraud and is published by PCI Security Standards Council, LLC. All companies who handle credit cards are likely to be subject to this standard.

PersonalFolder property. The full path to the Documents folder for the current user. The default is `C:\Users\username\Documents`.

port forwarding. A way to redirect unsecured traffic through a secure SSH tunnel. Two types of port forwarding are available: local and remote. Local (also called outgoing) port forwarding sends outgoing data sent from a specified local port through the secure channel to a specified remote port. You can configure a client application to exchange data securely with a server by configuring the client to connect to the redirected port instead of directly to the computer running the associated server. Remote (also called incoming) port forwarding sends incoming data from a specified remote port through the secure channel to a specified local port.

product installation folder. The default on English language systems is `C:\Program Files (x86)\Micro Focus\Reflection` on 64-bit systems and `C:\Program Files\Micro Focus\Reflection` on 32-bit systems.

public key/private key. Public keys and private keys are pairs of cryptographic keys that are used to encrypt or decrypt data. Data encrypted with the public key can only be decrypted with the private key; and data encrypted with the private key can only be decrypted with the public key.

Screen History. Screen History creates recordings of host screens as you navigate to them. (VT screens are not recorded automatically; they can be recorded using manual capture.) You can view and/or verify the information from those screens, and send multiple host screens to Microsoft Word, PowerPoint, and Outlook (Email Message and Note only), if they are installed on your computer.


socket. The combination of a host name (IP address or DNS name) and a port number. This creates a unique identifier that a client application uses as an end point of communications.

trusted host. A trusted host is one for which you hold the public key.

trusted locations. A trusted location is a directory that's designated as a secure source for opening files. By default, Reflection allows you to open documents only in directories specified as trusted locations using the Specify Trusted Locations dialog box.

Windows common application data folder. The application data folder is hidden by default. The default is `\ProgramData\`.

Windows personal documents folder. The default on English systems is `\Users\ username \Documents\`.

Workspace Menu. The Workspace menu contains layout options, application and document settings, and a list of recent documents. It is accessed by clicking the **File** menu or Reflection button  (when using the Ribbon user interface).

