
Liant Software Corporation

LiantInstall Program

User's Reference

First Edition

LIANT

This document contains information on the installation process for Liant products running on Microsoft Windows using Microsoft Windows Installer technology.

The information contained herein applies to systems running under Microsoft 32-bit Windows operating systems

The information in this document is subject to change without prior notice. Liant Software Corporation assumes no responsibility for any errors that may appear in this document. Liant reserves the right to make improvements and/or changes in the products and programs described in this guide at any time without notice. Companies, names, and data used in examples herein are fictitious unless otherwise noted.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopied, recorded, or otherwise, without prior written permission of Liant Software Corporation.

The software described in this document is furnished to the user under a license for a specific number of uses and may be copied (with inclusion of the copyright notice) only in accordance with the terms of such license.

Copyright © 2008 by Liant Software Corporation. All rights reserved. Printed in the United States of America.

Liant Software Corporation

5914 West Courtyard Dr., Suite 100
Austin, TX 78730-4911
U.S.A.

Phone (512) 343-1010

(800) 762-6265

Fax (512) 343-9487

Web site <http://www.liant.com>

RM, RM/COBOL, RM/COBOL-85, Relativity, Enterprise CodeBench, RM/InfoExpress, RM/Panels, VanGui Interface Builder, CodeWatch, CodeBridge, Cobol-WOW, WOW Extensions, InstantSQL, Xcentricity, XML Extensions, Liant, and the Liant logo are trademarks or registered trademarks of Liant Software Corporation.

Microsoft, MS, MS-DOS, Windows 98, Windows Me, Windows NT, Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, and Visual Basic are trademarks or registered trademarks of Microsoft Corporation in the USA and other countries.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Ltd.

All other products, brand, or trade names used in this publication are the trademarks or registered trademarks of their respective trademark holders, and are used only for explanation purposes.

Documentation Release History for the LiantInstall Program User's Reference:

Document Edition Number	Applies To Product Version	Publication Date
First	Version 12 and later	October 2008

Important These Liant manuals are distributed as Adobe Acrobat PDF files and require the Adobe Acrobat Reader, 6.0 or higher, in order to view them. The installation program for the Adobe Acrobat Reader is available from the Adobe web site at: www.adobe.com.

Contents

- Introduction to Liant Windows Installer Installations..... 1
- LiantInstall 2
 - Running LiantInstall 2
 - Configuring LiantInstall 2
 - Value Formats 3
 - Sections 3
- Windows Installer Packages for Liant Components 6
 - Installation Command Line Options 7
 - MSI Command Line Options..... 7
 - MSI Installation Properties 9
- Liant Component-Specific Installation Details 11
 - Relativity Data Client Installation Details 12
 - Relativity Data Manager Installation Details..... 14
 - Relativity Data Server Installation Details..... 16
 - Relativity DBA Installation Details 18
 - Relativity Designer Installation Details 20
 - Relativity Java Client Installation Details..... 21
 - RM/COBOL CodeWatch Integrated Development Environment Installation Details 22
 - RM/COBOL Compiler Installation Details 24
 - RM/COBOL Recover1 Utility Installation Details..... 26
 - RM/COBOL Runtime Installation Details..... 28
 - RM/InfoExpress Installation Details..... 30
 - WOW Extensions Designer Installation Details 32
 - WOW Thin Client Installation Details 33
 - Xcentrity Business Information Server (BIS) Installation Details 34
- Administrative Network Installations 36
 - Creating a Network Share..... 36
 - Placing the MSI Installation Packages into a Network Share..... 36
 - LiantInstall’s Administrative Installation 37
 - Copy LiantInstall and the Necessary MSI Installation Packages to the Network Share 38
 - Copy the Necessary MSI Installation Packages to the Network Share..... 39
 - Perform an Administrative Installation to the Network Share..... 40

Introduction to Liant Windows Installer Installations

The installation process for Liant products running on Windows uses Microsoft Windows Installer technology. The installations have been structured such that individual components (for example, the RM/COBOL Compiler and the RM/COBOL Runtime) of Liant products (for example, the Liant Development System) are installed by separate installations. These separate component installations, called “packages” by Microsoft Windows Installer, are self-contained, and you may invoke a package directly in order to install an individual Liant component. However, if you wish to install multiple components at one time, the invocations of the packages may also be coordinated by a program named **LiantInstall**. The LiantInstall program is driven by the contents of a license certificate file, **liant.lic**. Thus, while there is no Windows Installer package that installs a “Liant Development System” (that is, the RM/COBOL Compiler and the Runtime System together), LiantInstall, in conjunction with a “Development System” license certificate, can be used to achieve the same effect.

This document describes the Windows Installer packages, which install individual components, and the **LiantInstall** program. The purpose of this document is to describe LiantInstall and the Windows Installer packages in sufficient detail that they can be integrated into the installation of your COBOL application program. There are two ways to accomplish this integration:

1. **Invoke the LiantInstall program.** Using LiantInstall to install Liant components is useful if there are multiple components to be installed, or if the installation of the Liant components *is not* closely coupled with the installation of your COBOL application program. In the latter case, the end user will invoke LiantInstall directly in order to install the components. The LiantInstall program can be configured to reduce the set of dialog boxes that the user sees and to pass options directly to the installation of the Liant components.
2. **Invoke the installation of the Liant component directly.** Invoking the installation of a Liant component directly is useful if the installation of the component *is* to be closely integrated into the installation of your COBOL application program. In this case, the installation of the application program will initiate the installation of the component by executing the Windows Installer (MSI) package file for the component.

This document describes LiantInstall first, and then discusses the details of the Windows Installer package files for each Liant component, beginning in the section, [Windows Installer Packages for Liant Components](#) (see page 6). It also explains more advanced installation alternatives for system administrators in [Administrative Network Installations](#) (see page 36).

LiantInstall

The following topics discuss how to run and configure the **LiantInstall** program.

Running LiantInstall

The **LiantInstall** program may be started and run by any number of methods, including one of the following:

- Open **LiantInstall.exe** in a Windows Explorer window (as a standard Windows program).
- Use the Start | Run menu.
- Enter the name of the program, **LiantInstall**, in a command window.
- Create an autorun configuration file.
- Invoke LiantInstall from another program.

When **LiantInstall** begins running, the program searches for any Liant license certificate files (files with the extension **.lic**). LiantInstall starts the search in the current working directory and then looks in subdirectories for components to be installed. A series of self-explanatory dialog boxes are displayed: Welcome, License Agreement, Customer Information, License Certificates, Certificate Activation, Installation Type, Product Selection, Installation Folders, Ready To Begin Installations, and Installation Status.

To suppress the display of any of these dialog boxes, or to otherwise control the standard behavior of LiantInstall, see the following topic, “Configuring LiantInstall.”

Configuring LiantInstall

If there are multiple Liant product components to be installed, or if the installation of the Liant product components *is not* closely coupled with the installation of your COBOL application program, the preferred solution is to configure the behavior of **LiantInstall** rather than invoking the installation package directly. LiantInstall can be configured using a simple initialization file (**.ini**) named **LiantInstall.ini**, which is located in the current working directory.

The configuration file is a simple line-oriented text file. It contains one or more section headers, which are distinguished by lines that begin with an open square bracket ([) and that end with a close square bracket (]). The text between the brackets is the name of the section. Each section header is followed by one or more keyword value pairs. The keyword begins the line and is delimited by an equals sign (=). The value is the remainder of the line. Keywords themselves may not contain an equals sign. The white space around the equals sign is not ignored. Spaces preceding the equals sign will be treated as part of the keyword. Spaces after the equals sign are treated as part of the value.

The following is a sample configuration file:

```
[LiantInstall]
BypassWelcomeDialog=Y

[Trace Options]
TraceFile=C:\temp\trace.log
```

This example shows two sections, **LiantInstall** and **Trace Options**. The **LiantInstall** section has a single keyword, **BypassWelcomeDialog**, and it has the value Y. The **Trace Options** section also has a single keyword, **TraceFile**, and it has the value **C:\temp\trace.log**.

Value Formats

LiantInstall has three standard interpretations of keyword values:

- **String.** String values are straight text and have no further interpretation for formatting restrictions.
- **Integer.** Integer values are signed decimal numbers.
- **Boolean.** Boolean values can be signed decimal numbers, in which case the value 0 is interpreted as FALSE and any other value is interpreted as TRUE. If the value is not a valid signed decimal number, the first character of the value is examined. If the first character is **Y** or **y**, the value is interpreted as TRUE. If the first character is **N** or **n**, the value is interpreted as FALSE. Any other value results in an error.

Sections

The **LiantInstall.ini** initialization file has two standard sections with fixed names: **LiantInstall** and **Trace Options**. All other sections are treated as configuration sections for the specific component installations.

LiantInstall Section

The **LiantInstall** section is the main configuration area for the **LiantInstall** program, and it is the main section that you will most likely modify if you wish to install multiple Liant components without combining these installations with the installation of your COBOL application program. This section supports the following keywords:

- **BypassInstallFoldersDialog.** This Boolean keyword controls the display of the Install Folders dialog box. The default value of FALSE causes the dialog box to be displayed. If the Install Folders dialog box is bypassed, all product executables are installed in a product-specific directory in the **Liant** directory within the Program Files directory of the machine, usually **C:\Program Files**. Also, the products install their shortcuts in a **Liant** folder within the **Program Start** menu.
- **BypassInstallTypeDialog.** This Boolean keyword controls the display of the Install Type dialog box. The default value of FALSE causes the dialog box to be displayed. If the Install Type dialog box is bypassed, a Standard installation is performed.
- **BypassLicenseAgreementDialog.** This Boolean keyword controls the display of the License Agreement dialog box. The default value of FALSE causes the dialog box to be displayed. If the License Agreement dialog box is bypassed, a message indicating that

the configuration has accepted the license agreement is written to the log file of the application program.

- **BypassLicenseCertificatesDialog.** This Boolean keyword controls display of the License Certificates dialog box. The default value of FALSE causes the dialog box to be displayed. If the License Certificates dialog box is bypassed, an attempt is made to install all certificates for the current platform that LiantInstall locates by a search of the current directory and its subdirectories.
- **BypassNetworkFolderDialog.** This Boolean keyword controls the display of the Network Folder dialog box. The default value of FALSE causes the dialog box to be displayed. If the Network Folder dialog box is bypassed, the products are installed in the last Network Folder that LiantInstall was requested to use, or in the first shared network directory located on the machine, if this is the first invocation of LiantInstall to perform an Administrative Installation.
- **BypassProductSelectionDialog.** This Boolean keyword controls the display of the Product Selection dialog box. The default value of FALSE causes the dialog box to be displayed. If the Product Selection dialog box is bypassed, the products are installed based on installation type. If the installation type is Standard, all products are installed. If the installation type is Custom, no products are installed.
- **BypassReadyToBeginInstallationsDialog.** This Boolean keyword controls the display of the Ready To Begin Installations dialog box. The default value of FALSE causes the dialog box to be displayed. If the Ready To Begin Installations dialog box is bypassed, installations begin immediately.
- **BypassRetrievingActivationCodes.** This Boolean keyword instructs LiantInstall not to attempt to use the Internet to contact Liant for activation codes. If there are any stale license certificates present, the Stale Certificates dialog box is displayed (see the next item). The default value is FALSE.
- **BypassStaleCertificatesDialog.** This Boolean keyword controls display of the Stale Certificates dialog box. The default value of FALSE causes the dialog box to be displayed. If the Stale Certificates dialog box is bypassed, license certificates that require activation codes are not installed.
- **BypassUserInformationDialog.** This Boolean keyword controls the display of the User Information dialog box. The default value of FALSE causes the dialog box to be displayed.
 - If the User Information dialog box is bypassed, the default User Name and Organization values are used.
 - If the current user account has never run LiantInstall before, the defaults are the Registered Owner and Registered Organization recorded during the installation of Microsoft Windows.

After LiantInstall has been run once by the current user account, the defaults are the User Name and Organization values entered by the user during the previous execution of LiantInstall.

- **BypassWelcomeDialog.** This Boolean keyword controls the display of the Welcome dialog box. The default value of FALSE causes the dialog box to be displayed. If the Welcome dialog box is bypassed, the user is not able to press the **Browse CD** button to view the Liant product directory. Also, if the user attempts to “Back” into the bypassed Welcome dialog box, the program silently terminates.
- **InstallType.** This string keyword controls the type of installation initially presented in the Install Type dialog box. If the Install Type dialog is bypassed, **InstallType** indicates

the type of installation performed. The values for this keyword, which are case insensitive, include **Standard**, **Custom**, **Administrative**, and **Network**.

Trace Options Section

The Trace Options section is intended to direct tracing for the **LiantInstall** program, and deals mostly with the log file. This section supports the following keywords:

- **AppendLog.** This Boolean keyword indicates that the trace file should not be erased on each execution, but should be allowed to accumulate tracing. The default value is FALSE.
- **DeleteLog.** This Boolean keyword indicates the initial state of the Delete Log File on Exit flag. The default value is TRUE. Note, however, that in the event an installation fails, LiantInstall will reset this flag to FALSE.
- **TraceDirectory.** This string keyword contains the name of the directory in which to place the trace file. The default is the temporary directory of the current user. On Windows XP, the location of this temporary directory is C:\Documents and Settings*UserName*\Local Settings\Temp. On Windows Vista, the location of this temporary directory is C:\Users*UserName*\AppData\Local\Temp.
- **TraceFile.** This string keyword contains the name of the trace file. The default is **LiantInstall.txt**. If this is an absolute path, the contents of the **TraceDirectory** keyword are ignored.
- **WindowsInstallerOptions.** This string keyword contains the Windows Installer **log** options to pass on the command line of each MSI installation. Specifying the + (append to existing file) and ! (flush each line to the log) options are not necessary, as these are always supplied by LiantInstall. The default value is “iwemo”, which logs **I**nformative messages, **W**arning messages, **E**rror messages, out-of-**M**emory or fatal error messages, and **O**ut-of- disk space messages. Use “*v” for maximum tracing.

For more information, see [MSI Command Line Options](#) (on page 7).

Component Sections

Whenever installation is initiated for a component of a Liant product, **LiantInstall** searches its configuration file for a section with installation options for that component. You may then use the component sections to configure the behavior of installations by specifying options on the command lines of the component installation package.

If a section is found, all keyword value pairs in the section are appended to the command line for the invocation of the install. The installation will treat all uppercase keywords as options. LiantInstall will quote all of the keyword values are they are placed on the command line, and escape the quotes within the values.

The name of the configuration section for each Liant component is given below.

Component Name	Configuration Section Name
Relativity Data Client	[Relativity Data Client]
Relativity Data Manager	[Relativity Data Manager]
Relativity Data Server	[Relativity Data Server]
Relativity Database Administrator (DBA)	[Relativity DBA]
Relativity Designer	[Relativity Designer]
Relativity Java Client	[Relativity Java Client]
RM/COBOL CodeWatch	[RM/COBOL CodeWatch]
RM/COBOL Compiler	[RM/COBOL Compiler]
RM/COBOL Recover1 Utility	[RM/COBOL Recovery]
RM/COBOL Runtime	[RM/COBOL Runtime]
RM/InfoExpress	[RM/COBOL Server]
WOW Extensions Designer	[WOW Designer]
WOW Thin Client	[WOW Thin Client]
Xcentrisity Business Information Server (BIS)	[BIS Service Engine]

Windows Installer Packages for Liant Components

Each Liant product is composed of multiple components, which are installed by means of Windows Installer package files. These package files have the extension of **.msi** for historical reasons. If the installation of the Liant component is to be closely integrated into the installation of the COBOL application program, the preferable solution is to invoke the component installation directly rather than configuring the **LiantInstall** program. An installation may be invoked directly by double-clicking or opening an MSI file from Windows Explorer, or using the Start | Run menu from the command line. The installation will begin using a full user interface, which includes the display of a Welcome dialog box, prompting for the following: the acceptance of the license agreement, the location of the license certification files to install with the component, and the user name and organization, at a minimum. If multiple Liant components are to be installed, each installation will prompt for this information individually. LiantInstall can streamline the installation process by gathering information that is common to each installation and passing it to the component installations individually.

The following topics describe how to use the **msiexec** executable program, which is part of the Windows Installer package, to invoke the installation of a Liant component directly and to pass command line options in order to configure the installation. The individual Liant components and the specifics of their configuration are discussed in [Liant Component-Specific Installation Details](#) (see page 11).

Installation Command Line Options

The Microsoft Windows Installer executable program that interprets packages and installs product components is **msiexec.exe**. The **msiexec.exe** program employs two types of options on the command line:

- Those options that are preceded by a forward slash (/) are interpreted by the **msiexec** program directly. For the purposes of this document, these will be called MSI command line options. For more information, see the following topic, MSI Command Line Options.
- Those options that are *not* preceded by a forward slash and that are formatted as a *NAME=Value* pair control the actual installation. For the purposes of this document, these options will be called MSI installation properties. For more information, see [MSI Installation Properties](#) (on page 9).

MSI Command Line Options

To run the **msiexec.exe** program, you must specify at least one option on the command line. The syntax for command line options is as follows:

```
msiexec {/i | /f[RepairOptions] | /a | /x | /p} package.msi [/l[LogOptions] logfile] [/q[UserInterfaceOptions]] [/help] [/norestart | /forcerestart | /promptrestart] [Property ...]
```

where:

- **/i** indicates that an installation is to be performed.
- **/f** indicates that an installation is to be repaired. The following **RepairOptions** are supported:
 - **p** reinstalls only if file is missing.
 - **o** reinstalls only if file is missing or an older version is installed.
 - **e** reinstalls only if file is missing or an equal or older version is installed.
 - **d** reinstalls only if file is missing or a different version is installed.
 - **c** reinstalls only if file is missing or calculated checksum is different.
 - **a** forces all files to be reinstalled.
 - **u** rewrites all user-specific registry entries.
 - **m** rewrites all computer specific registry entries.
 - **s** overwrites all existing shortcuts.
 - **v** runs from source (media) and re-caches the local package.

Note Do not use this option with the initial installation.
- **/a** indicates that an Administrative Installation is to be performed (that is, install a product component on the network). For more information, see [Perform an Administrative Installation to the Network Share](#) (on page 40).
- **/x** indicates that a product is to be uninstalled.
- **/p** indicates that a product is to be patched.

- **/l** indicates that logging is to be performed. The following **LogOptions** are supported:
 - **i** logs status messages.
 - **w** logs nonfatal warnings.
 - **e** logs all error messages.
 - **a** logs start up of actions.
 - **r** logs action specific records.
 - **u** logs user requests.
 - **c** logs initial user interface parameters.
 - **m** logs out-of-memory or fatal exit information.
 - **o** logs out-of-space messages.
 - **p** logs MSI properties at termination.
 - **v** logs verbose output.
 - **x** logs extra information. (This option is available on Windows Server 2003 only.)
 - ***** indicates all logging options except **v** and **x**.
- **/q** indicates the desired level of the user interface. The following **UserInterfaceOptions** are supported:
 - **f** indicates the full user interface. This is the default if **/q** is not specified.
 - **n** indicates no user interface. This is the default if **/q** is specified without any **UserInterfaceOptions**.
 - **n+** indicates no user interface, except for a modal dialog box displayed at the end.
 - **b** indicates the basic user interface.
 - **b+** indicates the basic user interface with a modal dialog box displayed at the end, except that the dialog box is not displayed if the installation is cancelled.
 - **b!** indicates the basic user interface but with the Cancel button hidden.
 - **b+!** indicates the basic user interface but with the Cancel button hidden that displays a modal dialog box at the end.
 - **b-** indicates the basic user interface without modal dialog boxes.
 - **b-!** indicates the basic user interface without modal dialog boxes and the Cancel button hidden.
 - **r** indicates a reduced user interface with no modal dialog box displayed at the end.
- **/help** indicates that a usage message is to be displayed.
- **/norestart** indicates that the installer should never restart the computer after installation.
- **/forcerestart** indicates that the installer should restart the computer after every installation.
- **/promptrestart** indicates that the installer should ask the user if it may restart the system.
- **Property** indicates a property variable within the installation that is to be overridden. Properties are described in [MSI Installation Properties](#) (on page 9).

MSI Installation Properties

The Microsoft Windows Installer installation is controlled internally by using the values of variables called properties, as defined in a specific MSI installation package. The user may alter the installation by specifying a property and its value on the command line.

There are numerous Microsoft-defined properties, but only the most pertinent ones are detailed here. (To view the entire list of Microsoft-defined properties, consult the Microsoft web site.) Additionally, other Liant-defined properties are supplied to an MSI installation by means of the command line. Some of these Liant-defined properties are common to nearly all installations of the Liant components while others apply only to particular Liant components. The properties that are common to most Liant product component installations are included in the list below. For more information on the specific installation properties, see [Liant Component-Specific Installation Details](#) (on page 11).

All of the properties are supplied to the MSI installation by means of the command line option, [Property](#) (see page 8). On the command line, **Property** is in the form *NAME=value*.

```
msiexec /command-line-option... [NAME=value ]...
```

where:

Name must be all capital letters.

value must be quoted if it contains spaces. If *value* contains double quotes, they are escaped by supplying a pair of double quotes.

The MSI installation properties common to most Liant components include the following:

- **ACTCODE.** This Liant-defined property contains four, four-letter words separated by spaces that are used to activate the base license certificate contained in the **LICENSEFILES** property.
- **ADDLOCAL.** This Microsoft-defined property contains a comma-separated list of features to install. This list is case sensitive. See the specific component installation details for available features, beginning on 11. Use the value of **ALL** to install all features in the installation package.
- **ALLUSERS.** This Microsoft-defined property indicates that the component is to be installed for all users.
 - The value **1** indicates installation for all users. If the current user does not have sufficient permissions to perform an installation, an error is generated and the installation is cancelled.
 - The value **2** indicates installation for all users, assuming the current user has sufficient permissions; otherwise, the component is installed for the current user only and no error message is generated.

The default is to install the component for the current user.

- **COMPANYNAME.** This Microsoft-defined property contains the default company name for the installation. The default is the string stored at:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\Current Version\RegisteredOrganization.
- **INSTALLDIR.** This Microsoft-defined property contains the absolute path of the directory into which the component is to be installed. The presence of this property overrides the value contained in the **LIANTDIR** property.

- **INSTALLMENU.** This Liant-defined property contains the complete name of the Start Menu folder where the installation will store shortcuts. The presence of this property overrides the value contained in the **LIANTMENU** property.
- **LDI.** This Liant-defined property indicates that an installation is being invoked by the LiantInstall program. In the User Interface sequence, this property will cause the installation to bypass the Welcome, License Agreement, User Information, and Setup Type dialog boxes and display the Custom Setup dialog box directly. Furthermore, in an Administrative Installation, the LDI option causes the **NETWORKDIR** option to be taken as the complete name of the target directory into which the installation is to take place.
- **LIANTDIR.** This Liant-defined property contains the name of a directory where the installation will create its own directory for storing the executables. The default value is **C:\Program Files\Liant**.
- **LIANTMENU.** This Liant-defined property contains the name of a Start Menu folder where the installation will create its own folder for storing shortcuts. The default value is **Liant**.
- **LICENSEFILES.** This Liant-defined property contains a semicolon-separated list of license certificate files to be used in installing a component.
- **NETWORKDIR.** This Liant-defined property contains the name of a directory where the installation will create its own directory for storing an Administrative Installation. This property is valid only during an Administrative Installation. The default value is the root of the first (alphabetically) network drive letter defined on the current machine.
- **NOCOMPANYNAME.** This Microsoft-defined property indicates whether the installation should determine the company name from information in the Windows registry.
 - The value **1** indicates suppressing use of the registry to initialize the **COMPANYNAME** option.
 - Any other value will enable the use of the registry to initialize the **COMPANYNAME** option.
- **NOUSERNAME.** This Microsoft-defined property indicates whether the installation should determine the user name from information in the Windows registry.
 - The value **1** indicates suppressing use of the registry to initialize the **USERNAME** option.
 - Any other value will enable the use of the registry to initialize the **USERNAME** option.
- **REMOVE.** This Microsoft-defined property contains a comma-separated list of features to uninstall. This list is case sensitive. For a list of component-specific features, see the discussions of the actual component installations in [Liant Component-Specific Installation Details](#) (on page 11).
- **REBOOT.** This Microsoft-defined property indicates how the installation should handle rebooting. Note that only the first character of the value is necessary.
 - The value **Force** indicates that the installation should always reboot following a successful installation. If the user interface is running, the installation prompts for permission to reboot. If the user interface is not running, the installation reboots the system automatically.

- The value **Suppress** indicates that the installation should suppress reboots, even if the installation encountered a file that is in use.
- The value **ReallySuppress** indicates that the installation should never reboot.
- **REBOOTPROMPT.** This Microsoft-defined property indicates how the installation should handle the prompting for reboot. Note that only the first character of the value is necessary.
 - The value **Suppress** indicates that the installation should perform any necessary reboots without prompting.
 - Any other value indicates that the installation should perform any necessary reboots with prompts.
- **USERNAME.** This Microsoft-defined property contains the default user name for the installation. The default is the string stored at:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\Current
Version\RegisteredOwner.

Liant Component-Specific Installation Details

In addition to the installation properties common to most Liant product components (as described in the previous topic), each Liant component may have additional properties supplied to the MSI installation by means of the command line option, [Property](#) (see page 8). For specific installation properties and further details, see the following:

- [Relativity Data Client](#) (see page 12)
- [Relativity Data Manager](#) (see page 14)
- [Relativity Data Server](#) (see page 16)
- [Relativity DBA](#) (see page 18)
- [Relativity Designer](#) (see page 20)
- [Relativity Java Client](#) (see page 21)
- [RM/COBOL CodeWatch Integrated Development Environment](#) (see page 22)
- [RM/COBOL Compiler](#) (see page 24)
- [RM/COBOL Recover1 Utility](#) (see page 26)
- [RM/COBOL Runtime](#) (see page 28)
- [RM/InfoExpress](#) (see page 30)
- [WOW Extensions Designer](#) (see page 32)
- [WOW Thin Client](#) (see page 33)
- [Xcentrisity Business Information Server \(BIS\)](#) (see page 34)

Relativity Data Client Installation Details

This section explains how to invoke the Relativity Data Client installation package directly. It also describes the name of the package file, and the Data Client-specific properties and features that are supplied to the MSI installation by means of the command line option, [Property](#) (see page 8).

Filename

The name of the Relativity Data Client's package file is **Dclient.msi**. This file is located in the Liant product directory **Dclient\046**.

Properties

The Relativity Data Client package supports additional, component-specific installation properties:

- **DRIVERNAME**. This property indicates the ODBC driver name to be used for the Relativity Data Client. The default driver name is **Relativity Client**.
- **SERVER**. This property indicates the server name and port number to be installed as the default server for the Data Client. The format is **[tcp:]NetworkName.ServicePort**. The default server name is the contents of:

HKEY_LOCAL_MACHINE\Software\Liant Software
 Corporation\Relativity\RELEDBMS.INI\Set Up\ServerName

Note The following table lists the original command line options previously used to automate the installation of the Relativity Data Client. The second column lists the equivalent command line options (or properties) provided for the component in the MSI Windows Installer program. For more information, see [Installation Command Line Options](#) (on page 7).

Original Command Line Option	Equivalent MSI Command Line Option
-_AUTO	/q
-_DELETE	Not supported
-_DRIVER	DRIVERNAME
-_FOLDER	INSTALLMENU
-_MFFH	Not supported
-_MOVE=Y	Standard behavior
-_ORG	COMPANYNAME
-_SERVER	SERVER
-SMS	Not required
-_TARGETDIR	INSTALLDIR
-_TYPE=DBA	ADDLOCAL=ServerAdministrator
-_USERNAME	USERNAME

Features

The Relativity Data Client package includes the following installation features:

- **DataClient** installs the Relativity Data Client ODBC Driver.
- **ServerAdministrator** installs the Relativity Server Administrator.
- **DriverHelpFiles** installs the help files for the Relativity Data Client.
- **InstallKit** installs the Relativity Installation Kit library.
- **QueryPlanViewer** installs the Query Plan Viewer utility.
- **ReadMes** installs the README files for the Relativity Data Client.
- **SupportTools** installs the Liant support tools.

The [ADDLOCAL property](#) (see page 9) is a comma-separated list of features that are to be installed.

To install all of the features, enter the following:

```
msiexec /i DClient.msi ADDLOCAL=ALL
```

To install all of the features, except the Relativity Server Administrator, enter the following:

```
msiexec /i DClient.msi ADDLOCAL="ReadMes,  
DataClient,DriverHelpFiles,InstallKit,  
QueryPlanViewer,SupportTools"
```

Relativity Data Manager Installation Details

This section explains how to invoke the Relativity Data Manager installation package directly. It also describes the name of the package file, and the Data Manager-specific properties and features that are supplied to the MSI installation by means of the command line option, [Property](#) (see page 8).

Filename

The name of the Relativity Data Manager's package file is **DManager.msi**. This file is located in the Liant product directory **DManager\046**.

Properties

The Relativity Data Manager package supports additional, component-specific installation properties:

- **DRIVERNAME**. This property indicates the ODBC driver name to be used for the Relativity Data Manager. The default driver name is **Relativity (*.rcg)**.
- **EXTFH**. This property indicates the Micro Focus configuration file to use when accessing Micro Focus COBOL files. The default is the value of the EXTFH environment variable.
- **FILEMGR**. This property indicates the types of COBOL data files for which to install the support. The value is a comma-separated list of COBOL data files. The default is **RM**, unless the Micro Focus COBOL file manager is present or the **MFFM** command line option is present, in which case, the default is **RM,MF**.
 - The value **RM** indicates RM/COBOL data file support.
 - The value **MF** indicates Micro Focus COBOL data file support.
- **MFFM**. This property indicates the complete filename to use for the Micro Focus File Handler DLL. The default is the location of the **mffh.dll** file installed by Micro Focus COBOL on the machine. Because it is impossible to determine the location of the **mffh.dll** file automatically on Microsoft Vista, it will be necessary to communicate its location to the Data Manager's installation through this property.

Note The following table lists the original command line options previously used to automate the installation of the Relativity Data Manager. The second column lists the equivalent command line options (or properties) provided for the component in the MSI Windows Installer program. For more information, see [Installation Command Line Options](#) (on page 7).

Original Command Line Option	Equivalent MSI Command Line Option
-_AUTO	/q
-_BTREIVE	ADDLOCAL=Btrieve
-_DELETE	Not supported
-_DRIVER	DRIVERNAME
-_FILEMGR	FILEMGR
-_FOLDER	INSTALLMENU

Original Command Line Option	Equivalent MSI Command Line Option
-_INFOEXPRESS	ADDLOCAL=InfoExpress
-_LICENSE	LICENSEFILES
-_MFFH	MFFM
-_MOVE=Y	Standard behavior
-_ORG	COMPANYNAME
-_SAMPLES	ADDLOCAL=Samples
-SMS	Not required
-_STARTMTR	Standard behavior
-_TARGETDIR	INSTALLDIR
-_USERNAME	USERNAME

Features

The Relativity Data Manager package includes the following features:

- **Btrieve** installs the RM/COBOL-to-Btrieve Adapter program.
- **InfoExpress** installs the RM/InfoExpress Client.
- **DataManager** installs the Relativity Data Manager ODBC Driver.
- **DriverHelpFiles** installs the help files for the Relativity Data Manager.
- **QueryPlanViewer** installs the Query Plan Viewer utility.
- **ReadMes** installs the README files for the Relativity Data Manager.
- **Samples** installs the Verify sample files.
- **SupportTools** installs the Liant support tools.

The [ADDLOCAL property](#) (see page 9) is a comma-separated list of features to be installed.

To install all of the features, enter the following:

```
msiexec /i DManager.msi ADDLOCAL=ALL
```

To install all of the features, except the Btrieve and RM/InfoExpress clients, enter the following:

```
msiexec /i DManager.msi ADDLOCAL="ReadMes,  
DataManager,DriverHelpFiles,QueryPlanViewer,  
Samples,SupportTools"
```

Relativity Data Server Installation Details

This section explains how to invoke the Relativity Data Server installation package directly. It also describes the name of the package file, and the Data Server-specific properties and features that are supplied to the MSI installation by means of the command line option, [Property](#) (see page 8).

Filename

The name of the Relativity Data Server's package file is **DServer.msi**. This file is located in the Liant product directory **DServer\046**.

Properties

The Relativity Data Server package supports additional, component-specific installation properties:

- **ACCOUNT**. This property indicates the Windows account under which the Relativity Data Server is to run. It is in the format: `\\Machine Name\Domain Name\User Name`. (If present, the machine name is ignored.) The default is the system account.
- **DRIVERNAME**. This property indicates the ODBC driver name to be used for local data sources. The default driver name is **Relativity Server**.
- **EXTFH**. This property indicates the Micro Focus configuration file to use when accessing Micro Focus COBOL files. The default is the value of the EXTFH environment variable.
- **FILEMGR**. This property indicates the types of COBOL data files for which to install the support. The value is a comma-separated list of COBOL data files. The default is **RM**, unless the Micro Focus COBOL file manager is present or the **MFFM** command line option is present, in which case, the default is **RM,MF**.
 - The value **RM** indicates RM/COBOL data file support.
 - The value **MF** indicates Micro Focus COBOL data file support.
- **LOGMASK**. This property indicates the messages that are logged to the Windows Event log. The mask is the summation of the following values:
 - The value **1** indicates that error events are to be logged.
 - The value **2** indicates that warning events are to be logged.
 - The value **4** indicates that informational events are to be logged.
- **MFFM**. This property indicates the complete filename to use for the Micro Focus File Handler DLL. The default is the location of the **mffh.dll** file installed by Micro Focus COBOL on the machine. Because it is impossible to determine the location of the **mffh.dll** file automatically on Microsoft Vista, it will be necessary to communicate its location to the Data Server's installation through this property.
- **PASSWORD**. This property indicates the password for the Windows account under which the Relativity Data Server is to run.
- **PORTNUMBER**. This property indicates the service port on which the Relativity Data Server will listen for connections. The default service port is **1583**.

Note The following table lists the original command line options previously used to automate the installation of the Relativity Data Server. The second column lists the equivalent command line options (or properties) provided for the component in the MSI Windows Installer program. For more information, see [Installation Command Line Options](#) (see page 7).

Original Command Line Option	Equivalent MSI Command Line Option
-_ACCOUNT	ACCOUNT
-_AUTO	/q
-_FILEMGR	FILEMGR
-_FOLDER	INSTALLMENU
-_LICENSE	LICENSEFILES
-_ORG	COMPANYNAME
-_PASSWORD	PASSWORD
-_PORTNUMBER	PORTNUMBER
-SMS	Not required
-_TARGETDIR	INSTALLDIR
-_USERNAME	USERNAME

Features

The Relativity Data Server package includes the following features:

- **Btrieve** installs the RM/COBOL-to-Btrieve Adapter program.
- **InfoExpress** installs the RM/InfoExpress Client.
- **DataServer** installs the Relativity Data Server.
- **DriverHelpFiles** installs the help files for the Relativity Data Server.
- **QueryPlanViewer** installs the Query Plan Viewer utility.
- **ReadMes** installs the README files for the Relativity Data Server.
- **Samples** installs the Verify sample files.
- **SupportTools** installs the Liant support tools.

The [ADDLOCAL property](#) (see page 9) is a comma-separated list of features to be installed.

To install all of the features, enter the following:

```
msiexec /i DServer.msi ADDLOCAL=ALL
```

To install all of the features, except the Btrieve and RM/InfoExpress clients, enter the following:

```
msiexec /i DServer.msi ADDLOCAL="ReadMes,  
DataServer,DriverHelpFiles,QueryPlanViewer,  
Samples,SupportTools"
```

Relativity DBA Installation Details

This section explains how to invoke the Relativity Database Administrator (DBA) installation package directly. It also describes the name of the package file, and the DBA-specific features that are supplied to the MSI installation by means of the command line option, [Property](#) (see page 8).

Filename

The name of the Relativity DBA's package file is **DBA.msi**. The file is located in the Liant product directory **DBA\046**.

Properties

The Relativity DBA package does not support any additional, component-specific installation properties.

Note The following table lists the original command line options previously used to automate the installation of the Relativity DBA. The second column lists the equivalent command line options (or properties) provided for the component in the MSI Windows Installer program. For more information, see [Installation Command Line Options](#) (see page 7).

Original Command Line Option	MSI Command Line Option
-_AUTO	/q
-_FOLDER	INSTALLMENU
-_LICENSE	LICENSEFILES
-_ORG	COMPANYNAME
-SMS	Not required
-_TARGETDIR	INSTALLDIR
-_USERNAME	USERNAME

Features

The Relativity DBA package includes the following features:

- **DBA** installs the Relativity Database Administrator.
- **DBAHelpFiles** installs the help files for the Relativity Database Administrator.
- **ReadMes** installs the README files for the Relativity Database Administrator.
- **ReportScripts** installs the List and Label report files.
- **SupportTools** installs the Liant support tools.

The [ADDLOCAL property](#) (see page 9) is a comma-separated list of features to be installed.

To install all of the features, enter the following:

```
msiexec /i DBA.msi ADDLOCAL=ALL
```

To install all of the features, except the Report Scripts files, enter the following:

```
msiexec /i DBA.msi ADDLOCAL="ReadMes,  
DBA,DBAHelpFiles,SupportTools"
```

Relativity Designer Installation Details

This section explains how to invoke the Relativity Designer installation package directly. It also describes the name of the package file, and the Designer-specific features that are supplied to the MSI installation by means of the command line option, [Property](#) (see page 8).

Filename

The name of the Relativity Designer's package file is **RelDesigner.msi**. This file is located in the Liant product directory **Designer\046**.

Properties

The Relativity Designer package does not support any additional, component-specific installation properties.

Features

The Relativity Designer package includes the following features:

- **Designer** installs the Relativity Designer.
- **DesignerHelpFiles** installs the help files for the Relativity Designer.
- **ReadMes** installs the README files for the Relativity Designer.
- **ReportScripts** installs the List and Label report files.
- **Samples** installs the Stored Procedures sample files.
- **SamplesMF** installs the Micro Focus COBOL Shirt sample files.
- **SamplesRM** installs the RM/COBOL Shirt sample files.
- **SupportTools** installs the Liant support tools.

The [ADDLOCAL property](#) (see page 9) is a comma-separated list of features to be installed.

To install all of the features, enter the following:

```
msiexec /i RelDesigner.msi ADDLOCAL=ALL
```

To install all of the features, except the Micro Focus COBOL Shirt sample files, enter the following:

```
msiexec /i RelDesigner.msi ADDLOCAL="ReadMes,  
Designer,DesignerHelpFiles,ReportScripts,  
Samples,SamplesRM,SupportTools"
```

Relativity Java Client Installation Details

This section explains how to invoke the Relativity Java Client installation package directly. It also describes the name of the package file, and the Java Client-specific features that are supplied to the MSI installation by means of the command line option, [Property](#) (see page 8).

Filename

The name of the Relativity Java Client's package file is **JClient.msi**. This file is located in the Liant product directory **JDBC**.

Properties

The Relativity Java Client package does not support any additional, component-specific installation properties.

Features

The Relativity Java Client package includes the following features:

- **JavaClient** installs the Relativity Java Client JDBC Driver.
- **HelpFiles** installs the help files for the Relativity Java Client.
- **ReadMes** installs the README files for the Relativity Java Client.
- **Samples** installs the sample files for the Relativity Java Client.

The [ADDLOCAL property](#) (see page 9) is a comma-separated list of features to be installed.

To install all of the features, enter the following:

```
msiexec /i JClient.msi ADDLOCAL=ALL
```

To install all of the features, except the sample files, enter the following:

```
msiexec /i JClient.msi ADDLOCAL="ReadMes,  
JavaClient,HelpFiles"
```

RM/COBOL CodeWatch Integrated Development Environment Installation Details

The section explains how to invoke the RM/COBOL CodeWatch integrated development environment installation package directly. It also describes the name of the package file, and the CodeWatch-specific properties and features that are supplied to the MSI installation by means of the command line option, [Property](#) (see page 8).

Filename

The name of the RM/COBOL CodeWatch integrated development environment's package file is **CodeWatch.msi**. This file is located in the Liant product directory **046**.

Properties

The RM/COBOL CodeWatch integrated development environment package supports an additional, component-specific installation property:

- **NETWORKINSTALL**. This property indicates that CodeWatch should be installed to run from the network share directory. CodeWatch must first be installed to a directory using an Administrative Installation, usually to a shared network drive. Some installation files cannot run from the network and are copied to the local machine. Any files that can be run from the network, however, are not copied to the local machine during the installation.
 - The value **Y** or **1** indicates that the component should be installed to run from the network, where possible.

Features

The RM/COBOL CodeWatch integrated development environment package includes the following features:

- **BISSamples** installs sample files that demonstrate using the RM/COBOL CodeWatch integrated development environment with Xcentrisity Business Information Server service programs.
- **CodeWatch** installs the RM/COBOL CodeWatch integrated development environment.
- **CodeWatchHelp** installs the help file for the RM/COBOL CodeWatch integrated development environment.
- **ReadMes** installs the README files for the RM/COBOL CodeWatch integrated development environment.
- **Documentation** installs the documentation for the RM/COBOL CodeWatch integrated development environment.
- **SupportTools** installs the Liant support tools.

The [ADDLOCAL property](#) (see page 9) is a comma-separated list of features to be installed.

To install all of the features, enter the following:

```
msiexec /i CodeWatch.msi ADDLOCAL=ALL
```

To install all of the features, except the BIS sample files, enter the following:

```
msiexec /i CodeWatch.msi ADDLOCAL="ReadMes,  
CodeWatch,CodeWatchHelp,  
Documentation,SupportTools"
```

RM/COBOL Compiler Installation Details

This section explains how to invoke the RM/COBOL Compiler installation package directly. It also describes the name of package file, and the RM/COBOL Compiler-specific properties and features that are supplied to the MSI installation by means of the command line option, [Property](#) (see page 8).

Filename

The name of the RM/COBOL Compiler's package file is **Compiler.msi**. This file is located in the Liant product directory **046**.

Properties

The RM/COBOL Compiler package supports additional, component-specific installation properties:

- **COMPILETYPE**. This property indicates the compiler type installed for the **rmcobol** executable. Note that the value for this parameter is case sensitive.
 - The value **console** indicates that the installed **rmcobol** will execute in console mode. This is the default value.
 - The value **GUI** indicates that the installed **rmcobol** will execute in GUI mode.
- **NETWORKINSTALL**. This property indicates that the RM/COBOL Compiler should be installed to run from the network share directory. The compiler must first be installed to a directory using an Administrative Installation, usually to a shared network drive. Some installation files cannot run from the network and are copied to the local machine. Any files that can be run from the network, however, are not copied to the local machine during the installation.
 - The value **Y** or **1** indicates that the component should be installed to run from the network, where possible.

Features

The RM/COBOL Compiler package includes the following features:

- **CodeBridge** installs the CodeBridge components.
- **CodeBridgeSamples** installs the CodeBridge samples.
- **Compiler** installs the RM/COBOL Compiler.
- **CompilerHelp** installs the help file for the RM/COBOL Compiler.
- **Documentation** installs the documentation for the RM/COBOL Development System.
- **ReadMes** installs the README files for the RM/COBOL Compiler.
- **Samples** installs the sample files.
- **SupportTools** installs the Liant support tools.
- **ToolBarEditor** installs the Tool Bar Editor.
- **Utilities** installs the source to the **ANALYSIS.CBL** utility.
- **Verify** installs the source to the verification suite.

- **NonCOBOLDevTools** installs the source for DLL and Shared Object development.
- **XMLExtensions** installs XML Extensions.
- **XMLExamples** installs the example files for XML Extensions.

The [ADDLOCAL property](#) (see page 9) is a comma-separated list of features to be installed.

To install all of the features, enter the following:

```
msiexec /i Compiler.msi ADDLOCAL=ALL
```

To install all of the features, except the verification suite source, enter the following:

```
msiexec /i Compiler.msi ADDLOCAL="ReadMes,  
CodeBridge,CodeBridgeSamples,Compiler,  
CompilerHelp,Documentation,Samples,SupportTools,  
ToolBarEditor,Utilities,NonCOBOLDevTools,  
XMLExtensions,XMLExamples"
```

RM/COBOL Recover1 Utility Installation Details

This section explains how to invoke the RM/COBOL Recover1 Utility installation package directly. It also describes the name of the package file, and the RM/COBOL Recover1 Utility-specific properties and features that are supplied to the MSI installation by means of the command line option, [Property](#) (see page 8).

Filename

The name of the RM/COBOL Recover1 Utility's package file is **Recover1.msi**. This file is located in the Liant product directory **046**.

Properties

The RM/COBOL Recover1 Utility package supports an additional, component-specific installation property:

- **DISABLEOPTS**. This property indicates that the NT File System optimizations, which can corrupt RM/COBOL indexed files, should be disabled by the installation. The default is to disable the optimizations.
 - The values **1**, **Y**, or **y** indicate that the optimizations will be disabled. If any optimizations are disabled, the installation will place a file named **RegistryChanges.log** into the installation directory. This file will contain a list of the registry changes made to disable the optimizations and will include the original value of the registry entry. If this file contains *only* **dword:** as the original value for an entry, then the indicated registry entry did not exist before.
 - Any other value indicates that the current state of optimizations will not be changed.
- **NETWORKINSTALL**. This property indicates that the RM/COBOL Recover1 Utility should be installed to run from the network share directory. The utility must first be installed to a directory using an Administrative Installation, usually to a shared network drive. Some installation files cannot run from the network and are copied to the local machine. Any files that can be run from the network, however, are not copied to the local machine during the installation.
 - The value **Y** or **1** indicates that the component should be installed to run from the network, where possible.

Features

The RM/COBOL Recover1 Utility package includes the following features:

- **ReadMes** installs the README files for the RM/COBOL Recover1 Utility.
- **Recover1** installs the RM/COBOL Recover1 Utility.
- **SupportTools** installs the Liant support tools.

The [ADDLOCAL property](#) (see page 9) is a comma-separated list of features to be installed.

To install all of the features, enter the following:

```
msiexec /i Recover1.msi ADDLOCAL=ALL
```

To install all of the features, except the support tools, enter the following:

```
msiexec /i Recover1.msi ADDLOCAL="ReadMes,Recover1"
```

RM/COBOL Runtime Installation Details

This section explains how to invoke the RM/COBOL Runtime installation package directly. It also describes the name of the package file, and the RM/COBOL Runtime-specific properties and features that are supplied to the MSI installation by means of the command line option, [Property](#) (see page 8).

Filename

The name of the RM/COBOL Runtime's package file is **Runtime.msi**. This file is located in the Liant product directory **046**.

Properties

The RM/COBOL Runtime package supports additional, component-specific installation properties:

- **DISABLEOPTS**. This property indicates that the NT File System optimizations, which can corrupt RM/COBOL indexed files, should be disabled by the installation. The default is to disable the optimizations.
 - The values **1**, **Y**, or **y** indicate that the optimizations will be disabled. If any optimizations are disabled, the installation will place a file named **RegistryChanges.log** into the installation directory. This file will contain a list of the registry changes made to disable the optimizations and will include the original value of the registry entry. If this file contains *only* **dword:** as the original value for an entry, then the indicated registry entry did not exist before.
 - Any other value indicates that the current state of optimizations will not be changed.
- **NETWORKINSTALL**. This property indicates that the RM/COBOL Runtime should be installed to run from the network share directory. The runtime must first be installed to a directory using an Administrative Installation, usually to a shared network drive. Some installation files cannot run from the network and are copied to the local machine. Any files that can be run from the network, however, are not copied to the local machine during the installation.
 - The value **Y** or **1** indicates that the component should be installed to run from the network, where possible.

Features

The RM/COBOL Runtime package includes the following features:

- **ExternalAccessMethod** installs external access support for Btrieve and RM/InfoExpress.
- **Panels** installs the RM/Panels support.
- **Runtime** installs the RM/COBOL Runtime.
- **ReadMes** installs the README files for the RM/COBOL Runtime.
- **SupportTools** installs the Liant support tools.
- **Utilities** installs the object to the following utility programs: Analysis, DspDialog, PDevice, PDialog, PGetFont, PrInfo, PTextMet, Recover2, Recovery, RMDefInx, RMInfo, RMMMapInx, RMMMapPgm, RMPgmCom, and SysInfo.

Note The Recover1 Utility is not in this list.

- **Verify** installs the objects to the verification suite.
- **WOWRuntime** installs the WOW Extensions Runtime components.
- **WOWThinClient** installs the WOW Thin Client Server components.
- **NonCOBOLDevTools** installs the source for DLL and Shared Object development.

The [ADDLOCAL property](#) (see page 9) is a comma-separated list of features to be installed.

To install all of the features, enter the following:

```
msiexec /i Runtime.msi ADDLOCAL=ALL
```

To install all of the features, except the non-COBOL development tools, enter the following:

```
msiexec /i Runtime.msi ADDLOCAL="ExternalAccessMethod,  
Panels,Runtime,ReadMes,SupportTools,Utilities,  
Verify,WOWRuntime,WOWThinClient"
```

RM/InfoExpress Installation Details

This section explains how to invoke the RM/InfoExpress installation package directly. It also describes the name of the package file, and the RM/InfoExpress-specific properties and features that are supplied to the MSI installation by means of the command line option, [Property](#) (see page 8).

Filename

The name of the RM/InfoExpress' package file is **InfoExpress.msi**. This file is located in the Liant product directory **046**.

Properties

The RM/InfoExpress package supports an additional, component-specific installation property:

- **DISABLEOPTS**. This property indicates that the NT File System optimizations, which can corrupt RM/COBOL indexed files, should be disabled by the installation. The default is to disable the optimizations.
 - The values **1**, **Y**, or **y** indicate that the optimizations will be disabled. If any optimizations are disabled, the installation will place a file named **RegistryChanges.log** into the installation directory. This file will contain a list of the registry changes made to disable the optimizations and will include the original value of the registry entry. If this file contains *only* **dword:** as the original value for an entry, then the indicated registry entry did not exist before.
 - Any other value indicates that the current state of optimizations will not be changed.

Features

The RM/InfoExpress package includes the following features:

- **ClnIniFile** installs the sample RM/InfoExpress Client initialization (**.ini**) file.
- **Documentation** installs the RM/InfoExpress documentation
- **InfoExpress** installs RM/InfoExpress Server.
- **IXViewer** installs the RM/InfoExpress Viewer program. Note that this program is different from the RM/InfoExpress server display program, which is installed with the **InfoExpress** feature.
- **ReadMes** installs the README files for RM/InfoExpress.
- **SupportTools** installs the Liant support tools.
- **Verify** installs the RM/InfoExpress verification program.

The [ADDLOCAL property](#) (see page 9) is a comma-separated list of features to be installed.

To install all of the features, enter the following:

```
msiexec /i InfoExpress.msi ADDLOCAL=ALL
```

To install all of the features, except the RM/InfoExpress Viewer program, enter the following:

```
msiexec /i InfoExpress.msi ADDLOCAL="ClnIniFile,  
Documentation,InfoExpress,ReadMes,  
SupportTools,Verify"
```

WOW Extensions Designer Installation Details

This section explains how to invoke the WOW Extensions Designer installation package directly. It also describes the name of the package file, and the WOW Designer-specific features that are supplied to the MSI installation by means of the command line option, [Property](#) (see page 8).

Filename

The name of the WOW Extensions Designer's package file is **WowDesigner.msi**. This file is located in the Liant product directory **046**.

Properties

The WOW Extensions Designer package does not support any additional, component-specific installation properties.

Features

The WOW Extensions Designer package includes the following features:

- **Designer** installs the WOW Extensions Designer.
- **DesignerHelpFiles** installs the WOW Extensions Designer help files.
- **ReadMes** installs the README files for the WOW Extensions Designer.
- **SupportTools** installs the Liant support tools.
- **Samples** installs the WOW Extensions sample files.
- **PanelsSamples** installs the RM/Panels sample files.

The [ADDLOCAL property](#) (see page 9) is a comma-separated list of features that are to be installed.

To install all of the features, enter the following:

```
msiexec /i WowDesigner.msi ADDLOCAL=ALL
```

To install all of the features, except the RM/Panels sample files, enter the following:

```
msiexec /i WowDesigner.msi ADDLOCAL="Designer,  
DesignerHelpFiles,ReadMes,SupportTools,Samples"
```

WOW Thin Client Installation Details

This section explains how to invoke the WOW Thin Client installation package directly. It also describes the name of the package file, and the WOW Thin Client-specific features that are supplied to the MSI installation by means of the command line option, [Property](#) (see page 8).

Filename

The name of the WOW Thin Client's package file is **TClient.msi**. This file is located in the Liant product directory **046**.

Properties

The WOW Thin Client package does not support any additional, component-specific installation properties.

Features

The WOW Thin Client package includes the following features:

- **WOWThinClient** installs the WOW Thin Client.
- **ReadMes** installs the README files for the WOW Thin Client.

The [ADDLOCAL property](#) (see page 9) is a comma-separated list of features to be installed.

To install all of the features, enter the following:

```
msiexec /i TClient.msi ADDLOCAL=ALL
```

To install all of the features, except the README files, enter the following:

```
msiexec /i TClient.msi ADDLOCAL="WOWThinClient"
```

Xcentrivity Business Information Server (BIS) Installation Details

This section explains how to invoke the Xcentrivity Business Information Server (BIS) installation package directly. It also describes the name of the package file, and the BIS-specific properties and features that are supplied to the MSI installation by means of the command line option, [Property](#) (see page 8).

Filename

The name of the Xcentrivity Business Information Server's package file is **BISServiceEngine.msi**. This file is located in the Liant product directory **046**.

Properties

The Xcentrivity Business Information Server (BIS) supports additional, component-specific properties.

- **BISUSERNAME** contains the user account name under which the Service Engine should be installed and run. This value should be in the format **DomainName\Account**.
- **BISPASSWORD** contains the password for the account **BISUSERNAME**.
- **DISABLEOPTS** indicates that the NT File System optimizations, which can corrupt RM/COBOL indexed files, should be disabled by the installation. The default is to disable the optimizations.
 - **1, Y, or y** indicates that the optimizations will be disabled. If any optimizations are disabled, the installation will place a file named **RegistryChanges.log** into the installation directory. This file will contain a list of the registry changes made to disable the optimizations and will include the original value of the registry entry. If this file contains *only* **dword:** as the original value for an entry, then the indicated registry entry did not exist before.
 - Any other value indicates that the current state of optimizations will not be changed.

Features

The Xcentrivity Business Information Server (BIS) package includes the following features:

- **BusinessInformationServer** installs the BIS Request Handler and Service Engine.
- **Documentation** installs the Xcentrivity Business Information Server documentation.
- **ExternalAccessMethod** installs the Btrieve and RM/InfoExpress external access support.
- **ReadMes** installs the README files for the Xcentrivity Business Information Server.
- **Samples** installs the common files for the samples.
- **Sample1** installs the Program-driven sample application.
- **Sample2** installs the Transform-driven sample application.
- **Sample3** installs the Calculator Web Service sample application.
- **Sample4** installs the AccountInfo Web Service sample application.

- **SampleClient** installs a sample web client that invokes the calculator server in SAMPLE3 using SOAP.
- **Sudoku** installs the Transform-driven Sudoku sample application.
- **SupportTools** installs the Liant support tools.
- **Verify** installs the installation verification sample.
- **Web2COBOLSamples** installs a sample Visual Basic program that parses a web page and emits a COBOL data division.
- **WOWSamples** installs samples that demonstrate BIS+WOW.

The [ADDLOCAL property](#) (see page 9) is a comma-separated list of features to be installed.

To install all of the features, enter the following:

```
msiexec /i BISServiceEngine.msi ADDLOCAL=ALL
```

To install all of the features, except all of the samples, enter the following:

```
msiexec /i BISServiceEngine.msi ADDLOCAL="BusinessInformationServer,  
Documentation,ExternalAccessMethod,ReadMes,SupportTools"
```

Administrative Network Installations

When administering many machines, you may wish to place the installation for a product component on a network share of a Windows server so that clients may install the component themselves, instead of having the administrator visit every machine with physical media.

This section explains how to do the following:

- Create a network share on the Windows server.
- Place the MSI installation packages into a network share.

Creating a Network Share

The creation of a network share is simply a process of creating a folder (directory) on the Windows server and then designating that the folder be visible to other machines on the network. To create a network share folder:

1. Create a folder on the Windows server in a location where the Windows client workstations can access it. This new folder can be in a folder that is already shared, or it may be a folder that can be shared after creation.
2. Share the folder by doing the following:
 - a. In Windows Explorer, right-click the folder you want to be shared.
 - b. On the shortcut menu, click **Sharing**. A Properties dialog box is displayed for the selected folder.
 - c. If necessary, select the **Sharing** tab, and then click **Shared As**. (The defaults for these fields are usually sufficient; however, modify them as necessary to suit your needs.)
 - d. In the **Share Name** text box, type the name of the shared folder for use with the Windows client workstation installations.
 - e. Click **OK** to apply the changes you have made and close the dialog box.

Placing the MSI Installation Packages into a Network Share

While there are a number of ways that the MSI installation package files may be placed into a network share directory, only three techniques are discussed here:

- Copy LiantInstall and the necessary MSI installation packages to the network share.
- Copy the necessary MSI installation packages to the network share.
- Perform an MSI Administrative Installation to the network share.

The method you choose depends on your requirements.

LiantInstall's Administrative Installation

This technique is the most straight-forward approach, as **LiantInstall** will be responsible for placing the installation files, as well as a copy of the LiantInstall program itself and the license certificates into the network share.

1. Invoke **LiantInstall** normally, specifying the license certificate files of the products to be installed.
2. In the Installation Type dialog box, select the Administrative Installation option.
3. In the Product Selection dialog box, select the product components to install.
4. In the Network Folder dialog box, enter the name of the network share folder into which the files are to be placed.
5. After LiantInstall has completed installing the components to the network share folder, it will copy the license certificate files for the products installed to the network share directory and finally, a copy of the program itself. If the name of a license certificate file conflicts with a name of a file already in the network share folder, the name of the license certificate file will be copied with a new name that does not conflict. If LiantInstall is already present in the network share, you will be asked if it can be replaced.

As an optional step, you may also create an **autorun.inf** file in the root directory of the network share with the following contents:

```
[AutoRun]
icon=LiantInstall.exe
open=LiantInstall
shell\Install\command=LiantInstall.exe
shell\Install=Liant Install
```

In this case:

- If the Windows client machine has autorun enabled, LiantInstall will start automatically when the network share is opened on the client machine.
- If an **autorun.inf** file is not placed in the root directory of the network share, the end users must open LiantInstall to initiate an installation.

After LiantInstall is invoked, the installation will proceed as normal. When the Installation Type dialog box is displayed, the users can select Network Installation. If the installation of the product component supports that installation type, the installation will run from the network share. To determine whether an installation component supports this capability, see the NETWORKINSTALL property in the specific installation detail sections for the following:

- [RM/COBOL CodeWatch Integrated Development Environment](#) (see page 22)
- [RM/COBOL Compiler](#) (see page 24)
- [RM/COBOL Recover1 Utility](#) (see page 26)
- [RM/COBOL Runtime](#) (see page 28)

Copy LiantInstall and the Necessary MSI Installation Packages to the Network Share

This technique is most useful when you need to automate the installation, including supplying answers to the installation prompts for the user.

1. Copy the LiantInstall program executable into the root directory of the network share.
2. Copy the directory containing the MSI installation package to be installed into the root directory of the network share. For most components, the directory containing the MSI installation will have the following structure:

```
ComponentName\Version\PlatformKey\ComponentName.msi
```

The value in the *PlatformKey* directory is a number that Liant uses to distinguish between various hardware and software platforms. In the case of components that run on only one platform, there will be no *PlatformKey* directory present in the *Version* directory. The *PlatformKey* value for Windows is **046**.

Note In this technique, it is necessary for the LiantInstall program to preserve the directory structure. Since LiantInstall installs only Windows components, you need only copy the *Version* directory for the components that you wish end users to install and the **046** value of the *PlatformKey* directory, if present, for those components. Other *PlatformKey* directories will contain the components for UNIX platforms and are not necessary to copy to the workstation.

3. Copy the Liant license certificate files for the components to be installed into the network share.

If you place multiple license certificates into the root directory of the network share, you will need to give them unique names. As long as these files have the extension of **.lic**, the LiantInstall program will be able to locate them. You may also place these files into the directories, *ComponentName*, *Version*, or *PlatformKey*, in which case, they may continue to have the name **liant.lic**.

4. Following the instructions on how to [configure LiantInstall](#) (see page 2). Create a **LiantInstall.ini** file into the root directory of the network share to control the dialog boxes that end users will see.

As an optional step, you may also create an **autorun.inf** file in the root directory of the network share with the following contents:

```
[AutoRun]
icon=LiantInstall.exe
open=LiantInstall
shell\Install\command=LiantInstall.exe
shell\Install=Liant Install
```

In this case:

- If the Windows client machine has autorun enabled, LiantInstall will start automatically when the network share is opened on the client machine.
- If an **autorun.inf** file is not placed in the root directory of the network share, the end users must open LiantInstall to initiate an installation.

Copy the Necessary MSI Installation Packages to the Network Share

This technique is the simplest. It involves copying the *ComponentName.msi* file of the desired component from the *Version* directory or the **046** *PlatformKey* directory into the root directory of the network share. This method should be used only when the end users are skilled Windows users.

The disadvantage to this technique is that the network share may contain only one MSI installation in the root directory, as the Liant certificate file must have the name **liant.lic**. If necessary, subdirectories may be created in the network share for different component installations and the MSI installation file placed in the desired directory.

1. Copy the directory containing the MSI installation package to be installed into the root directory of the network share, or create a subdirectory and copy the MSI installation directory into it. For most components, the directory containing the MSI installation will have the following format:

```
ComponentName\Version\PlatformKey\ComponentName.msi
```

The value in the *PlatformKey* directory is a number that Liant uses to distinguish between various hardware and software platforms. In the case of components that run on only one platform, there will no *PlatformKey* directory present in the *Version* directory. The *PlatformKey* value for Windows is **046**.

Note In this technique, it is *not* necessary for the LiantInstall program to preserve the directory structure. It is only necessary to copy the MSI installation file from the **046** *PlatformKey* directory, if present, for the components you wish end users to install. It is also not necessary to preserve the name of the MSI installation file. You may rename it to any name that is meaningful to you and your end users.

2. Copy the Liant license certificate files for the component to be installed into the root directory of the network share. The license certificate must have the name **liant.lic**.

To initiate an installation, end users must locate the MSI installation file for the desired component and open it. You may also create a shortcut or command script to encapsulate the invocation of the installation and its command line options.

Perform an Administrative Installation to the Network Share

This technique involves using an option of the **msiexec** program to unpack a component to the network share for installation to the end users machines later. (For more information, see [MSI Command Line Options](#) on page 7.) This method should be used only when the end users are skilled Windows users.

The disadvantage to this technique is that the network share may contain only one installation in the root directory, as the Liant license certificate file must have the name **liant.lic**. If necessary, subdirectories may be created in the network share for different product component installations and the MSI installation package file unpacked in the desired directory.

The advantage of this technique is that the new MSI installation package file will be smaller than the original MSI installation package file, and, if needed, this will improve network performance.

1. Change directories to the directory containing the Liant component to be installed.
2. Execute the following command:

```
msiexec /a ComponentName.msi
```

3. When the installation prompts for a **Network location**, enter either the root directory of the network share or the desired name of a subdirectory within the network share where the installation should be unpacked.
4. Place the Liant license certificate file for the component into the directory specified for the Network location.

To initiate an installation, the end user must locate the MSI installation file for the desired component and open it. You may also create a shortcut or command script to encapsulate the invocation of the installation and its command line options.