



Citrix Installation Management Services 1.0b



Revision History

Revision	Change Description	Updated By	Date
1	Original Version	Syed Ali	1/29/01
1a	Updated Version with Peer Review	Orlando Melendez	2/6/01
1b	Office 2000 Installation Appendix	Jorge Gomez	2/9/01
1c	Network Diagram	Jorge Gomez	2/9/01
2	Modified to current template	Drew Robbins	3/7/01



Table of Contents

1	Overview	1
2	Best Practices	2
3	Citrix IMS 1.0b Overview	4
3.1	IMS Components	4
3.1.1	Citrix Packager	4
3.1.2	Application Publishing Enhancements	4
3.1.3	Citrix Installer	4
4	Using IMS	5
4.1	IMS Requirements	5
4.1.1	Packager System Requirements	5
4.1.2	File Server Requirements	5
4.1.3	Installer and Citrix Server Requirements	6
4.2	Packaging and Publishing Applications	6
4.2.1	Before Packaging Applications	6
4.2.2	After Packaging Applications	7
4.2.3	Publishing Applications	7
4.2.4	Maintenance	7
5	Windows File Script	8
5.1	Overview	8
5.2	Customizing Installation Scripts and Application Packages	8
5.2.1	Executing Installation Script TODO Items	8
5.2.2	Deploying Files and Registry Changes	9
5.2.3	Hiding Published Application Entries	9
5.2.4	Controlling How the Installer Copies Files	10
5.2.5	Adding Custom Commands	10
5.3	Customizing Packager Output	11
5.3.1	Packager.ini Syntax	11
5.3.2	Packager.ini Components	11
5.3.3	Adding Keys and Values	14
5.3.4	Packager.ini Supported Symbols	14
6	Appendix A – IMS Installation Instructions	16
6.1	Packager Installation	16
6.2	Installer Installation	16
7	Appendix B – Installation Script Symbols	17
8	Appendix C – Skeleton WFS file	18
9	Appendix D - Deploying MS Office 2000 using IMS 1.0b to Windows 2000 MetaFrame Servers	21
10	Appendix E – Network Diagram	23
11	Appendix F - References	24



1 Overview

This document pertains to the Citrix Installation Management Services 1.0b (IMS) usage and best practices. It begins by giving a general overview of IMS and its components, continues with IMS Requirements and usage, and finishes with Windows File Scripts and application deployment customization.



2 Best Practices

IMS allows for the installation of applications throughout the Citrix MetaFrame enterprise from a single installation point. This section is provided only as a reference for best practices. To learn more detail about the IMS process, refer to the remainder of this document.

Building the Packager

The installation of an application is recorded by the IMS Packager machine, which creates a package (all of the software necessary for the installation) and a script (a list of instructions on how the application is to be installed). The following steps are recommended when setting up the packager:

- ✓ Create a 4 GB partition with the same Operating System as the computers that will have the application installed to.
- ✓ Create a 1-2 GB partition of any Operating System after WinFrame 1.7 (Windows 2000 Professional is recommended).
- ✓ Create a 1-2 GB partition to hold the packages before copying to the file server.
- ✓ Label each partition with "Packager", "Backup", and, "Packages" so that the server may be easily browsed.
- ✓ Modify the desktop of the Packager and Backup partitions by changing the color and including the name of the Operating System. This will allow the user to immediately know which Operating System has been logged into.
- ✓ Do NOT add the server to a domain.
- ✓ Do NOT install MetaFrame on the packager partition – only Terminal Services.
- ✓ Do NOT set NTFS permissions.
- ✓ Back up the packager immediately after installing Operating System and IMS packager software. If all servers will be using Adobe Acrobat and/or an ODBC/DB access like Oracle client, then install these prior to backing up the partition.

Packaging the application

The application is packaged by installing the application onto the 4 GB packaging partition. Next, the packaging partition is restored to the original configuration allowing for another package to be created. This process is repeated for each package created. The following steps are recommended when creating a package:

- ✓ Track and plan the applications that will be packaged for the environment.
- ✓ Test to ensure that two applications can co-exist on a server properly.
- ✓ Document and organize user groups prior to publishing.
- ✓ Do NOT make any mouse or keystrokes during the packaging of an application. All changes to the Operating System are recorded and only modifications made by the installation are desired.
- ✓ You can package Adobe and Oracle. However, these applications need to be deployed first.
- ✓ Be patient when creating the packages. Each package takes a long time to create and the computer may appear to lock up at times.
- ✓ Test the application to ensure proper execution.
- ✓ Restore the packager partition immediately after packaging and testing the application.
- ✓ Even if unfamiliar with the syntax, review the packaging script.
- ✓ Edit the [RUN] section of the packaging script. Ensure that the proper executables are set along with the parameters supplied with those executables.



- ✓ If the script has trouble running the executable from the script, call the executable from a batch file and have the script call the batch file.
- ✓ Refer to the IMS Administrators Guide when publishing multiple applications to a suite.



3 Citrix IMS 1.0b Overview

Citrix IMS allows administrators single-point control for deploying applications. Using IMS, an application can be simultaneously installed on all Citrix servers in a farm from a single point regardless of physical locations, network connection types, or individual hardware setups, and all without manual intervention.

Citrix IMS simultaneously installs an application from one location onto multiple load-balanced Citrix servers in a server farm. IMS Version 1.0b supports WinFrame 1.7/1.8, MetaFrame 1.0/1.8 for Windows NT Server 4.0 Terminal Services Edition (TSE) and MetaFrame 1.8a for Windows 2000 Servers. This version also includes latest application compatibility scripts. Since Windows 2000 uses different application compatibility scripts, IMS 1.0b detects the OS and displays relative scripts.

3.1 IMS Components

IMS divides the application deployment process into two separate operations: packaging an application installation and pushing that package to Citrix servers. To perform these operations, IMS is composed of three components; the Citrix Packager, the Application Publishing Enhancements, and the Citrix Installer.

3.1.1 Citrix Packager

The Packager monitors changes made by an application (such as the creation/deletion of registry keys, addition/modification of INI files, creation of desktop shortcuts and/or program groups) during its installation on a server and then records the changes as commands in an IMS script file. It then copies all files needed to run the application such as EXE files and DLLs for distribution to Citrix servers.

The process of recording an installation routine is called packaging an application. This process creates two objects: a package and an installation script. A package is defined as a container that holds all files needed to run an application. The packager assembles an application's installation directory in addition to files copied to directories within the system directory into a package directory on a separate file server. The installation script is an editable text file that is the instruction set for installing applications to Citrix servers. This script is discussed later in the section titled Windows File Script.

3.1.2 Application Publishing Enhancements

IMS 1.0b includes enhancements to the MetaFrame 1.0 and WinFrame 1.7 utility Application Configuration (called Published Application Manager (PAM) in MetaFrame 1.8 and WinFrame 1.8). The Application Publishing Enhancements give Application Configuration and PAM the ability to push the installation of applications to, uninstall applications from, and manage the application cache on Citrix servers. This component is used after packaging an application during application publishing (via Application Configuration or PAM). Instead of pointing to an application's EXE file, the command line for an application published with IMS points to the installation script created by the Packager. Therefore when an application is published, each Citrix server configured to receive the installation downloads the script and installs the application according to the script's instructions. IMS uses load-balanced application publishing as its method for pushing application installations to Citrix servers.

3.1.3 Citrix Installer

The Installer component downloads and executes the installation requests using the Packager's scripts published via Application Configuration or PAM and package. This component runs as an NT service on Citrix servers that performs all installation actions with no necessary intervention on the part of the administrator or ICA client user.



4 Using IMS

This section discusses hardware and software requirements of IMS as well as general guidelines and best practices for application packaging and publishing.

4.1 IMS Requirements

IMS requires the following components: a Packager machine, a File server for the storage of application packages and installation scripts, and Citrix servers to receive deployed applications through the Installer component.

4.1.1 Packager System Requirements

The Packager System needs to be as similar as possible to the environment of the servers to which the applications are being deployed. Do not make the Packager machine part of an NT domain.

To package an application deployment to: Primary Packager OS must be:

WinFrame 1.7	WinFrame 1.7 or WinFrame 1.8
WinFrame 1.8	WinFrame 1.7 or WinFrame 1.8
MetaFrame 1.0 on Windows TSE	Windows NT 4.0 TSE (No MetaFrame)
MetaFrame 1.8 on Windows TSE	Windows NT 4.0 TSE (No MetaFrame)
MetaFrame 1.8a on Windows 2000 Server	Windows 2000 Servers with TSE (No MetaFrame)

To package an application for deployment to a mixed farm, either have separate packager partitions for each OS or use multiple computers to create different packages.

Each computer running the Packager must meet the following requirements:

- ✓ A primary packager partition (called the Packaging partition) dedicated to packaging applications. This partition must be large enough to accommodate an installation of the Primary Packager OS and the largest application to be packaged. This should contain no files other than those required by IMS.
- ✓ A backup packager partition for use to restore the Primary Packager OS. One of the following Backup Packager Operating Systems must be installed on the backup partition: WinFrame (any version), Windows NT 4.0, Windows 2000. The backup partition must have sufficient free space for primary partition backup files
- ✓ IMS hotfix for WinFrame 1.7 or WinFrame 1.8 installed if needed.

4.1.2 File Server Requirements

The File server needs enough free disk space to hold all the applications that will be deployed via IMS. The disk space required per application is equal to the manufacturer's recommended space requirements for installation. In addition, the File server must support UNC share points, and be accessible to Citrix servers using IMS to install applications. The user who publishes applications for IMS deployment must have Write access to this File server. In the case that applications need to be deployed in multiple physical locations, use multiple File servers in each location to avoid repeated application downloads across a WAN.



4.1.3 Installer and Citrix Server Requirements

The Installer needs to be present in all Citrix servers participating in application deployment. This requires 30 MB of disk space and a valid Citrix license for each server. Applications can be deployed to Citrix servers that meet the following requirements:

- ✓ WinFrame 1.7 servers with Service Pack 5B and hotfix SE17B091 or its replacement installed
- ✓ WinFrame 1.8 with hotfix SE180001 or its replacement installed
- ✓ MetaFrame 1.0 for Windows NT Server 4.0 TSE with hotfix ME100037 or its replacement installed
- ✓ MetaFrame 1.8 for Windows NT Server 4.0 TSE with hotfix ME180001 or its replacement installed
- ✓ MetaFrame 1.8a for Windows 2000 Servers with Terminal Services in Application server mode with permissions compatible with Terminal Server 4.0 users

The hardware and software requirements for the above listed server types match those listed in their respective user manuals. In addition to the standard requirements for these Citrix server types, installation of Citrix Load Balancing Services is recommended on each server running IMS to facilitate application publishing on multiple Citrix servers. Without Load Balancing, commands must be issued to install an application individually on each server running IMS. With Load Balancing the command can be issued at a single time for all servers.

4.2 Packaging and Publishing Applications

Packaging an application involves using the Citrix Packager to monitor an application's installation routine. For each application, the package and installation script are used in conjunction with the Installer to deliver the application to Citrix servers.

4.2.1 Before Packaging Applications

Follow these guidelines before packaging an application:

- ✓ Set up the packaging environment for the correct target platform.
- ✓ Backup Primary Packager OS
- ✓ Confirm that the Primary Packager OS is clean. If the packaging environment was just created and no applications have been packaged, it is clean. If an application has been packaged, restore the Primary Packager OS. The only exception with respect to a clean environment is Adobe Acrobat and/or ODBC access like the Oracle client. If these will be on the base installation of each system, then make sure these are on the server.
- ✓ Do not set NTFS permissions on files or directories.
- ✓ Create a directory on the File server to store the package and installation script. It is not recommended to store the package and installation script on the packaging machine. Never store these items in the Program Files directory on Windows Terminal Services machines.
- ✓ When packaging do not perform any activities on the Packager such as perusing the Internet, copying, moving, deleting files or directories, because all activity will be recorded and will become part of the installation script.
- ✓ Follow the Packaging Wizard to completely package stand-alone applications. Application suites, applications requiring Service Packs, and application upgrades involve different packaging procedures; refer to Chapter 4 in the IMS 1.0b Administrator's Guide.
- ✓ The user who publishes applications for IMS deployment must have Write access to the File server.



- ✓ Packaging applications can take a long time. The bigger applications may appear to have locked the system, but this has yet to be seen.

4.2.2 After Packaging Applications

Follow these guidelines after packaging an application:

- ✓ Review the WFS file and understand it. Edit the [RUN] section of the file to modify the EXE that is set to execute once the application is published. If publishing a multi application suite like Office, modify the singular EXE. Also in the [RUN] section, update the parameters variable to pass in required fields to the EXE. Refer to the IMS 1.0b Administrator's Guide for publishing multiple applications in a suite. Appendix E contains a step-by-step procedure on deploying Office 2000 on Windows 2000 Servers.
- ✓ Merge any included application compatibility scripts for the application. These scripts can be found at <http://www.citrix.com/support>. Some of these require special instructions so review the Application Compatibility Wizard before merging scripts.
- ✓ Document and organize user groups before publishing. Develop lists of applications that each user will get and maintain this document.
- ✓ Restore the packaging environment

4.2.3 Publishing Applications

Use the Application Configuration or PAM to publish the application's installation script. In doing this, the application will install on selected Citrix servers. Documentation should be kept and maintained to track what applications go on the servers.

Using Load Balancing is recommended. Application Configuration or PAM, when used with Citrix add-on Load Management Services allow the creation of a single published application that points to multiple Citrix servers. For publishing applications in a mixed server farm, refer to Chapter 5 in the IMS 1.0b Administrator's Guide.

Test the application after publishing, and document all items placed on the Citrix server by the install. If problems occur with some applications that do not start from a WFS file, add a batch file with the execution lines and then add that to the WFS file.

4.2.4 Maintenance

To monitor the application installation status, either the Event Viewer, MetaFrame Administration, or the Citrix Server Administration can be used. If the Citrix servers on the network are all WinFrame 1.7, Event Viewer must be used to monitor status. Otherwise, the MetaFrame Administration in MetaFrame 1.0, and the Citrix Server Administration in WinFrame 1.8 and MetaFrame 1.8 can be used to view the application installation status.

There are several tasks that must be done to maintain published applications. These include, but are not limited to, installing the application on additional Citrix servers, changing user access to applications and uninstalling applications from all or specific Citrix servers. Refer to Chapter 5 in the IMS 1.0b Administrator's Guide for details.



5 Windows File Script

This section discusses provides an overview, customization techniques and options to the installation scripts created by the Packager.

5.1 Overview

The Packager records all application installation activities in an installation script using a script language that Citrix servers in the farm can interpret and execute to reproduce the installation. The installation script is an editable text file that is the instruction set for installing applications to Citrix servers. It is composed of sections that contain items, refer to Appendix C for a basic WFS file skeleton of sections and items (parameters). The file extension of the installation script is WFS short for Windows File Script. Use a text editor such as Notepad or Wordpad to edit the script file.

5.2 Customizing Installation Scripts and Application Packages

There are several ways to customize installation scripts. Some of these methods are discussed in the following sections.

5.2.1 Executing Installation Script TODO Items

Installation Scripts include a section labeled [TODO_WINNTX], where X is 351 for WinFrame, 4 for Terminal Server, or 5 for Windows 2000 packaging environments. This section has information on operations the Packager either could not perform or performed but that require verification before using the script to install the application. If no items are included in this section, no additional actions must be taken when installing the application via the script.

Items in this section are listed sequentially and set to a numeric value that represents to the Packager one of 15 conditions. These items contain a text string that describes the condition and aids in further investigation of potential problems.

Frequently, this section contains a single item followed by the message: "Please verify that the 'Newer' flag is appropriate for all 'CopyFile' commands." This is a reminder that by default during the installation of applications to Citrix servers, the Installer replaces a file that already exists on the servers only if the file in the package is a newer version. This can be modified. In all but a few special conditions, this default behavior is correct and nothing manual must be done when using the script that contains only the CopyFile message in this section.

If any items are listed with either or both of the following messages, the script may have to be manually fixed:

- ✓ "This path points to a CD-ROM and may not be available on a client machine." This message appears when a registry value in the script references a path that points to a file or directory located on a CD-ROM drive.
- ✓ "This path points to a network drive that may not be available on a client machine." This message appears when a registry value in the script references a path that points to a file or directory located on a network drive.

For step by step instructions on how to fix the script for the items above, see Appendix A in the IMS 1.0b Administrator's Guide.

Make all script modifications prior to publishing the application. If the script is published already, increment the value of the Version parameter after modifications to allow the Citrix server Installer to download the updated script.



5.2.2 Deploying Files and Registry Changes

In addition to application installations, the Packager can package system changes such as registry modifications or file replacements. This is useful when deploying a registry update to servers or copying a file onto servers in a farm. This functionality can be applied to package any operation or change that does not have an installation program. The following procedure describes how to create a simple batch file that contains commands that need to be performed on servers. Use the Packager to package this "installation" and then publish the resulting installation script to put the system modifications into effect.

- ✓ Create a batch file with commands that need to be executed.
- ✓ The command line syntax for creating a registry key via a batch file is:

```
Regedit /s regfile.reg
```

regfile.reg is the name of a REG file containing the name and value of the key(s) that need to be created
- ✓ Save the batch file on the Packager machine. If deploying a registry change, save the Reg file in the same directory
- ✓ Package the batch file by specifying it instead of an application executable
- ✓ Publish the installation script
- ✓ To keep the script from appearing as a published application, see the next section

5.2.3 Hiding Published Application Entries

To hide a published application entry:

- ✓ Open the installation script on the File server
- ✓ Locate section titled [Remove] and comment out all lines beginning with "ConfigActions" by placing a semicolon at the beginning of each line
- ✓ Save changes to the script
- ✓ Publish the script and complete the installation to Citrix servers
- ✓ Unpublish the application by deleting its entry in Application Configuration or PAM. Unpublishing an application that references an installation script containing commented-out [Remove] section entries removes all references to the published application but does not remove file or registry changes created by installation.
- ✓ To remove the registry changes, remove the semicolons from the [Remove] section in the script, republish and then unpublish the application.

To hide an already-published application entry:

- ✓ Open the installation script on the File server
- ✓ Locate section title [Remove] and comment out the following three lines by placing a semicolon at the beginning of each line:

```
;ConfigActions=%!if(%_NTx%)&RemoveRegistryConfigActions_WINNTX  
;ConfigActions=%!if(%_NTx%)_RemoveFileConfigActions_WINNTX  
;ConfigActions=%!if(%_NTx%)_RemoveFileDirectories_WINNTX
```
- ✓ Replace the Version parameter's value with one greater than the current value.
- ✓ Save changes to the script



- ✓ The Citrix servers will notice the version incremented script and update their locally cached versions. Verify the update by confirming the version number through this registry key on each Citrix server: HKLM\Software\Citrix\AppCloning\Gateway\Applications\X\Version, where X is the file and path for the application.
- ✓ Unpublish the application by deleting its entry in Application Configuration or PAM.
- ✓ To uninstall the application on servers, remove the semicolons from the [Remove] section in the script, republish and then unpublish the application.

5.2.4 Controlling How the Installer Copies Files

During packaging, the Packager writes in the installation script section [&SetupConfigActions_WINNTx] a statement for each file the application copies to the packaging machine. This statement is the CopyFile parameter. During installation of the application to Citrix servers, the Installer uses these CopyFile items to determine where and how to copy files to the Citrix servers. The following syntax applies when writing CopyFile items:

CopyFile=Source-file, Destination-file, Overwrite-flag, Version, Date, Compression-flag, MD5-hash

- ✓ Source-file is the path and name of the file as it exists in the package.
- ✓ Destination-file is the full path and name of the file as it exists on the Citrix servers.
- ✓ Overwrite-flag determines the Installer's behavior if the destination file already exists. The possible values are:
 - Never. The destination file is never overwritten.
 - Write. The destination file is overwritten if it is not a system, hidden, or read-only file.
 - Always. The destination file is always overwritten, even if it is a system, hidden or read-only file.
 - Newer. The destination file is overwritten if the version and/or date of the source file are newer than the version and/or date of the destination file. If this field is null (represented by a comma, not a version number) the date alone is used. If the date field is null, the version field alone is used. If both are present, both fields must be newer for the file to be overwritten. The version field can only be used for EXE or DLL files that have a version resource. If the source file is newer, the destination file is overwritten even if it has Read-only attributes. When copying a file over an existing destination file, the Installer preserves the file attributes of the destination file.
- ✓ Version is an optional field that specifies the resource file version of the source file. File must be an EXE or DLL. This field is used along with the overwrite-flag. Format is n.n.n.n where each n is a number.
- ✓ Date is an optional field that specifies the date of the source file. Format is mm:dd:yyyy:hh:mm:ss.
- ✓ Compression-flag should be represented by a null value (a comma).
- ✓ MD5-hash should be represented by a null value (a comma).

5.2.5 Adding Custom Commands

In addition to the installation actions recorded by the Packager, installation scripts can include custom installation actions called ShellExecute commands. A ShellExecute command is defined as a command line inserted in an installation script. The Installer processes these command lines as part of the application's installation routine. The ShellExecute commands can be used to execute batch files or external applications. Add the ShellExecute command to the [&SetupConfigActions_WINNTX] section. The following syntax applies when writing ShellExecute commands:



ShellExecute=File,Wait-flag,Operation,Default-directory,Arguments,Show

- ✓ File is the full path and name of the want to be executed. This path must exist on each Citrix server. See the symbols in Appendix B.
- ✓ Wait-flag indicates whether to wait for the file to exit before continuing with processing the installation script. Specify this as a Boolean ("T" or "F").
- ✓ Operation indicates what action to use when executing the file. Values include:
 - Open. Opens the specified file. This is the default value.
 - Print. Prints the specified file. This value is valid only when File specifies a document file.
- ✓ Default-directory is the full path of the working folder for the file. This path must exist on each Citrix server.
- ✓ Arguments are passed on the command line when File is an executable file and otherwise ignored. These parameters must be separated by spaces.
- ✓ Shot specifies how the executed file is displayed. Values include:
 - Min. Minimized window
 - Max. Maximized window
 - Normal. Normal window. This value is the default value.

5.3 Customizing Packager Output

The Packager contains a support file called Packager.ini that allows for the adjustment of how the Packager records application installations. This file has various settings that control filtering of file information captured during packaging and this affects what the Packager writes in the installation script for an application. Before the completion of packaging, these filters are applied to remove events that need not be performed on the target Citrix servers. The Packager.ini is located in the same directory as the Packager executable.

5.3.1 Packager.ini Syntax

The Packager.ini is an editable text file that follows INI conventions. It is composed of six sections; each section contains keys and associated values. Section names are in brackets [], and each text string in a section represents a key and value, separated by an equals sign. With the exception of keys in a single section, key names are written as numbers. Each key represents a single filtering operation. Filters are listed sequentially in each section. Key names are arbitrary.

Each value for a key is composed of a file name and a path information or registry key name and value. These values can also contain the standard wildcards (? and *).

5.3.2 Packager.ini Components

This section describes each of the Packager.ini file's sections. Each description includes the section's purpose and a list of the keys and values included by default.

5.3.2.1 [Replaced_Filters]

Do not edit the keys and values in this section.



5.3.2.2 [Shared_Files]

This section indicates files or directories that should include “shared file usage count” logic for use during Unpublishing or the application. The Installer indicates usage count information in registry entries for each file specified. During unpublishing, the Installer checks the usage count for each file. If the usage count is 1 (meaning that no application other than the current application uses the file), the file can be deleted.

5.3.2.3 [System_Files]

This section shows files that are used by the Citrix servers' OS. These files do not require filtering by the Packager, should not be deleted during unpublishing, or should never be overwritten on the Citrix server. Each item listed in this section must include one or more of the following arguments:

- ✓ /nd. The no delete argument means that no references to the specified file should appear in the [Remove] section of the installation script.
- ✓ /nf. The no filtering argument indicates that the specified file should never be filtered.
- ✓ /now. The no overwrite argument indicates that the specified file should have the “Never” flag associated with its CopyFile command.

This section contains the following keys and values:

- ✓ 1=*\\win.ini /nd /nf /now
- ✓ 2=*\\system32* /nd /nf

5.3.2.4 [Policies]

This section allows for the entry of either default information for certain commands referenced in the installation script or to enable or disable certain features of the Packager. This section contains the following keys and values:

- ✓ UNINSTALL=F
- ✓ DELETEFILES=0
- ✓ DEBUG=-2.0
- ✓ WFSFORMAT=VER2.3
- ✓ REMOVE_SECTION=T
- ✓ SETUPREQUIRESADMIN=1
- ✓ ADFRegInfo=Software\\ADF\\%ADFAPPID%
- ✓ FileNumThreshold=250
- ✓ LOCAL_DIR=%MAXFREEDISK%\\
- ✓ SHOWCMD=%srvShowCmd%
- ✓ WORKINGDIRECTORY=%srvWorkDir%
- ✓ NETPATH=%srvNetPath%
- ✓ CanRunOffline=1
- ✓ CDROMTODOITEMS=T
- ✓ UNCPATHTODOITEMS=F
- ✓ SetCLocale=1



A description of each key follows:

- ✓ UNINSTALL. Do not edit this key.
- ✓ DELETEFILES. Do not edit this key.
- ✓ DEBUG. Do not edit this key.
- ✓ WFSFORMAT. Do not edit this key.
- ✓ REMOVE_SECTION. The value of this key determines whether or not the Packager creates a [Remove] section in installation scripts. The Installer follows the commands in the [Remove] section when an application is unpublished from Citrix servers. This key can have a value of "T" or "F." If the value is set to "T," the Packager generates a Remove section in the ADF file.
- ✓ SETUPREQUIRESADMIN. Do not edit this key.
- ✓ ADFRegInfo The ADFREGINFO key is a path that points to a key in the registry. During packaging, the Packager may need to create logic in the installation script to inform the Installer as to where to store certain files on the Citrix servers. This information is saved by the Installer in the registry using the path referenced by the ADFREGINFO value. If a value is not present, the Packager uses the default "Software\ADF\%ADFAPPID%".
- ✓ FileNumThreshold. Do not edit this key.
- ✓ LOCAL_DIR. Do not edit this key.
- ✓ SHOWCMD. Do not edit this key.

5.3.2.5 [File_Filters]

This section lists all of the files or registry keys to be ignored by the Packager when packaging applications for distribution to MetaFrame servers. This section contains the following keys and values:

- ✓ 1=*install.log
- ✓ 2=*netscape.reg
- ✓ 3=*?.tmp
- ✓ 4=*uninst*
- ✓ 5=*\temp*
- ✓ 6=*\~?*
- ✓ 7=*.~*
- ✓ 8=*\mscreate.dir
- ✓ 9=*\system.ini
- ✓ 10=*currentcontrolset\enum*
- ✓ 11=*controlset???\enum*
- ✓ 12=*?filenameoperations*
- ✓ 13=*currentcontrolset\services*
- ✓ 14=software\microsoft\windows nt\currentversion\terminal server\install*
- ✓ 15=environment*
- ✓ 16=system\currentcontrolset\control\session manager\environment*
- ✓ 17=*autoexec.*



- ✓ 18=sam\sam\domains*
- ✓ 19=*\ls\lsuser.lis
- ✓ 20=software\citrix\mslicensing*
- ✓ 21=security\policy\secrets*

5.3.2.6 [File_Filters_WinFrame]

This section lists all of the files or registry keys to be ignored by the Packager when packaging applications for WinFrame servers. This section contains the following keys and values:

- ✓ 1=*install.log
- ✓ 2=*netscape.reg
- ✓ 3=*?.tmp
- ✓ 4=*uninst*
- ✓ 5=*\temp*
- ✓ 6=*\~?*
- ✓ 7=*.~*
- ✓ 8=*\mscreate.dir
- ✓ 9=*\system.ini
- ✓ 10=*currentcontrolset\enum*
- ✓ 11=*controlset???\enum*
- ✓ 12=*currentcontrolset\services*
- ✓ 13=*?filenameoperations*
- ✓ 14=software\citrix\install*
- ✓ 15=*software\program groups*
- ✓ 16=*unicode program groups*
- ✓ 17=sam\sam\domains*
- ✓ 18=*\ls\lsuser.lis

5.3.3 Adding Keys and Values

To create a new filter, follow this procedure:

- ✓ Open Packager.ini
- ✓ Add MetaFrame filters to the [FILE_FILTERS] section and WinFrame filters to the [FILE_FILTERS_WINFRAME] section. In mixed mode, edit both sections.
- ✓ After the last, filter add an incremented key number, add the equals sign and if needed, specify wildcards to reference multiple files or registry keys.
- ✓ Be sure to use only lowercase letters in values

5.3.4 Packager.ini Supported Symbols

Packager.ini supports the inclusion of symbols in filters. The following symbols are supported: %WINDISK%, %WINDIR% and %WINSYSDIR%. See Appendix A for the meaning of these symbols.





6 Appendix A – IMS Installation Instructions

This appendix provides a general procedure for the installation of the IMS Packager and Installer software.

6.1 Packager Installation

1. Confirm that the Packager machine meets the requirements listed in the Packager System Requirements section.
2. To prepare, remove all data from the packaging partition.
3. If installing WinFrame 1.7 or WinFrame 1.8 as a Primary Packager Operating System on a machine with Windows NT 4.0 installed, backup Ntdetect.com and Ntldr, and restore these boot files after the Primary Packager Operating System installation below. For restoring boot files instructions, see Chapter 3 in the IMS 1.0b Administrator's Guide.
4. Install a fresh copy of the Primary Packager Operating System and designate as a Stand-Alone Server in a Workgroup.
5. Boot into the Primary Packager Operating System and install the Packager
6. Backup the Primary Packager Operating System via the Packager backup wizard.

6.2 Installer Installation

1. Confirm that the proper hotfix and service pack is present. See the Installer and Citrix Server Requirements section for specifics.
2. Install the Installer on a Citrix Server, and be sure to configure the Application Compatibility Drive correctly.
3. Register the Installer Service by specifying a valid username, password, and domain name for an administrator of the Citrix server that also has network access to the File Server. Make sure this account information does not change.
4. Install an IMS license.
5. If installing the Installer on a MetaFrame system, run the Chkroot.cmd script after completing Installer setup and reenter the drive letter specified during the Application Compatibility Drive setup in Step 2 above. If installing Installer on a WinFrame Server, no further configuration is necessary.



7 Appendix B – Installation Script Symbols

Symbol	Meaning
%WINDISK%	The drive that contains the operating system directory. For example: C:
%WINDIR%	The drive and directory where the operating system is installed. For example: C:\WTSRV.
%WINSYSDIR%	The drive and directory of the Windows system directory. For example: C:\WTSRV\System32.
%PROGRAMFILES%	The directory that contains the Program Files. For example, C:\WTSRV\Program Files.
%CDROM%	The drive letter of the first CD-ROM drive found. For example: E:
%DESKTOPDIR%	The path to the current desktop folder. For example: C:\WTSRV\Profiles\user\Desktop.
%PROGRAMSDIR%	The path to the current programs folder. For example: C:\WTSRV\Profiles\user\Start Menu\Programs.
%STARTMENUDIR%	The path to the current start menu folder. For example: C:\WTSRV\Profiles\user\Start Menu.
%STARTUPDIR%	The path to the current startup folder. For example: C:\WTSRV\Profiles\user\Start Menu\Programs\Startup.
%COMMONDESKTOP%	The path to common desktop folder. For example: C:\WTSRV\Profiles\All Users\Desktop.
%COMMONPROGRAMS%	The path to common programs folder. For example: C:\WTSRV\Profiles\All Users\Start Menu\Programs.
%COMMONSTARTMENU%	The path to common start menu folder. For example: C:\WTSRV\Profiles\All Users\Start Menu.
%COMMONSTARTUP%	The path to common startup folder. For example: C:\WTSRV\Profiles\All Users\Start Menu\Programs\Startup.
%TEMPDIR%	The directory for temporary files. For example: C:\Temp.
%APPINSTALLDISK%	The disk on which you install applications if different from disk containing Citrix server operating system.



8 Appendix C – Skeleton WFS file

[\$version 2\$]

[TODO_WINNT5]

Item_0=6, Please verify that the 'Newer' flag is appropriate for all 'CopyFile' commands.

[Main]

ProductID=

PackagerVersion=

ID=

ProductDirectory=

Description=

ProductDescription=

Creator=Packager Version 1.0

Version=

Require=""%ClientVersion%" = "" | (!Split(ClientVersion,cv)% %cv1%>2 | (%cv1%=2 & %cv2%>=3))', Citrix Installation Management Services version 1.0 or greater required.

Require=""%OS%"="" | (!Split(OSVER,v)% "%OS%"="WinNT" & %v1%=5)', This application is only Packaged to run on the Windows 2000 platform with MetaFrame.

Require=""%CPU%"="" | "%CPU%"="", CPU required.

CanRunOffLine=

AppConn= , WFS.AppConn

DefineSymbols= , OSPlatform

DefineSymbols= ,GlobalSymbols_WINNT5

FileInfo= ,FileInfo_WINNT5

SetupRequiresAdmin=

[GlobalSymbols_WINNT5]

_CHECKOS=

DISPLAYNAME=

[OSPlatform]

_NT351=!Split(OSVER,v)%!eval("""%OS%"="WinNT" & %v1%=3 & %v2%=51)%)

_NT4=!eval("""%OS%"="WinNT" & %v1%=4)%)

_NT5=!eval("""%OS%"="WinNT" & %v1%=5)%)

_WIN95=!eval("""%OS%"="Win95")%)

[WFS.AppConn]



NetPath=%srvNetPath%

ConnProtocol=

[FileInfo_WINNT5]

Dir=

[Setup]

ConfigActions=%!if(%_CHECKOS%)%,CheckOS

ConfigActions=%!if(%_NT5%)%,&FetchLocalDirs

ConfigActions=%!if(%_NT5%)%,&SetupConfigActions_WINNT5

ConfigActions=%!if(%_NT5%)%,_SetupLinks_WINNT5

DestFileInUseAction=reboot

FileErrorAction=fail

[CheckOS]

ExitIf=%!if(!%_NT5%)%, This ADF file was not Enabled for the current OS.

[&SetupConfigActions_WINNT5]

CopyFile=

AddIni=

DeleteReg=

AddReg=

CreateRegKey=

[&FetchLocalDirs]

ReadReg=

[_SetupLinks_WINNT5]

CreateDir=

CreateShortCut=

[Run]

DestFileInUseAction=

FileErrorAction=

RunApplication= , RunApp

[RunApp]



ApplicationName=

CommandLine=

WorkingDirectory=

ShowCmd=

[Exit]

ConfigActions=%!if(%_NT5%)%,&FetchLocalDirs

[Remove]

ConfigActions=%!if(%_NT5%)%,&FetchLocalDirs

ConfigActions=%!if(%_NT5%)%,&RemoveRegistryConfigActions_WINNT5

ConfigActions=%!if(%_NT5%)%,_RemoveAddIniConfigActions_WINNT5

ConfigActions=%!if(%_NT5%)%,_RemoveFileConfigActions_WINNT5

ConfigActions= ,_RemoveFileDirectories_WINNT5

DestFileInUseAction=

FileErrorAction=

[_RemoveFileConfigActions_WINNT5]

DeleteFile=

[&RemoveRegistryConfigActions_WINNT5]

DeleteReg=

DeleteEmptyRegKey=

[_RemoveAddIniConfigActions_WINNT5]

DeleteIni=

[_RemoveFileDirectories_WINNT5]

DeleteEmptyDir=



9 Appendix D - Deploying MS Office 2000 using IMS 1.0b to Windows 2000 MetaFrame Servers

This appendix describes the process of deploying MS Office 2000 to Windows 2000 MetaFrame Servers using IMS 1.0b. It draws on material present in the IMS 1.0b software, the Citrix Solution Knowledge Base and actual lab experience.

This appendix contains step-by-step instructions on getting started with the "packaging" of this application suite and comments on the instructions given by the IMS software for the deployment of MS Office 2000. Appendix E contains a network diagram of the configuration used in our lab for a sample deployment.

Necessary Environment and Getting Started

1. Verify that Citrix MetaFrame is installed with appropriate licenses. Verify that specific hotfixes are not required for IMS to work under your server configuration. The Citrix Server used to put together this document was a Windows 2000 MetaFrame Server with FR1.
 - a. Install IMS Installer (Refer to Appendix A)
 - b. Apply Installer license. License different from MF
2. Set up Packager machine (Refer to Appendix A)
3. Once Packager is installed, create a share on a file server which will be used for packages and installation scripts.
 - a. Make share available and assign appropriate permissions
4. Commence standard publishing process by opening Citrix Packager on your Packager server (on the Primary Packager Operating System).
 - a. Under "Package", select New.
 - b. A prompt stating that the PPOS has not been restored will come up if that is the case.
 - c. Click next.
 - d. Citrix Packager gives special instructions for packaging certain applications. Office 2000 is one of them.

Packager Instructions

The following instructions given by Packager are not altogether clear for packaging Office 2000 successfully (suggestions in bold follow each section):

"The packaging of Office 2000 does NOT require installation of Office 2000. Instead, you must set up an unattended install of Office 2000. This will still give you complete control over the installation.

1. Package the Office 2000 Resource Kit. It comes on its own CD or can be accessed from Microsoft's web site (<http://www.microsoft.com/office/ork/2000/appndx/toolbox.htm>). Before installing Microsoft Office 2000 on Windows NT 4.0, Terminal Server Edition, please familiarize yourself with the Microsoft Office Resource Kit Journal Article titled "Installing Office in a Windows Terminal Server Environment". This article can be found at the Microsoft Web site (<http://www.microsoft.com/office/ork/2000/Two/30t3.htm>). Remember where the Resource Kit is installed."

This instruction poses the first problem. The Resource Kit does not have a main executable. Citrix Solution Knowledge Base Article ID – CTX728797 states:

"The Office 2000 Resource Kit does not have a main executable. Therefore, when the packager asks for the main executable for the application, point it to path\orktools.exe or something similar."

The article leads the reader to believe that orktools.exe exists. If it cannot be found Support says that any executable can be used. Langver.exe worked fine in this case.



"2. Go to the Installation Script you just created and comment out (make the first character a semicolon) the line with "CreateShortCut=Microsoft Office 2000 Resource Kit" in the "_SetupLinks_WINNT5" section."

This line was not found after reviewing the script several times.

"3. Publish this package to the servers where you will want Office 2000 to be installed. Wait until the installations are complete."

The status of the installation can be checked in Citrix Server Administration by clicking the Published Applications tab in the left pane.

Select the application you published and scroll right in the right pane.

There will be an installation status.

"4. On your package file server, create a directory in the directory where you store Installation Scripts called "Office2k". In that directory, create a subdirectory called "CitrixCF.adf".

Example: \\APPSERVER\APPDIRECTORY\Office2k\CitrixCF.adf

5. Make a share point to a CD-ROM drive that will be available to all your servers. Alternately, copy the entire contents of the Office 2000 CD to your file server and create a share point.

6. Edit the office2k.wfs file found in the Packager's "AppCompat\win2k" directory. Change the TERMSRVR_MSTFILE environment variable if you installed the Resource Kit in a directory other than Program Files. Change the OFFICE2K_SHAREPOINT variable to the share point you created in the previous step.

7. Copy the edited office2k.wfs file to \\APPSERVER\APPDIRECTORY.

8. Publish an application that uses the office2k.wfs Installation Script to the servers. Once you see the success status, you may immediately unpublish the Installation Script.

9. Logged in users will have to log out and log back in before they can use Office 2000.

10. You may now publish any applications to the programs contained in Office 2000 (Word, Excel, IE 5, etc.)."

The Knowledge Base Article also mentions modifying the script to state where to place a log file:

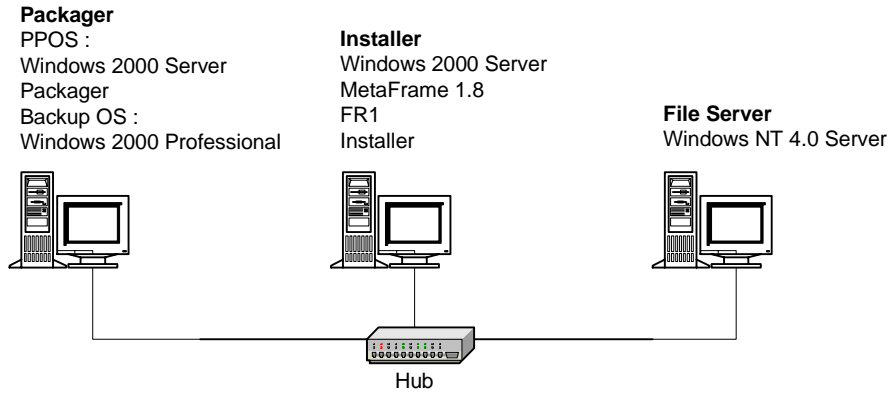
"Change the O2Klog.txt path at the end of the second ShellExecute line to someplace appropriate."

After going through the script a number of times, that line of code was not found. Support's explanation was that it is part of the script when installing Office 2000 on NT 4. It is taken out when installing Office 2000 on Windows 2000. The log is very important in the case of finding out what stage the installation is in according to the Knowledge Base Article.



10 Appendix E – Network Diagram

Displayed is the network diagram, which contains the components, used in the lab and to successfully deploy an Office 2000 package.





11 Appendix F - References

Stagray, K., Jenkins, E., Gosson, M., Edwards, S. *Administrator's Guide Citrix Installation Management Services 1.0b*. Citrix Systems, Inc., 2000.

Advanced Citrix Server Implementation Version 1.1. Citrix, Systems, Inc., 2000