Easy Manage ® Reference Guide

Version 4.4

The System Management and
Configuration Tool
for your Complete ICT infrastructure.

MulCom B.V.
http://www.ezmanage.nl

Restricted Rights Legend

About This Guide

The purpose of the Easy Manage Reference Guide is to inform the Easy Manage specialist using procedures and standards to increase efficiency in PC-Management. The Guide will explain the control files and procedures in detail of the Management and Configuration tool.

Other Easy Manage Manuals are:

- Easy Manage Configuration Guide
- Easy Manage Admin Program Guide
- Easy Manage Helpdesk Program Guide

Table of contents

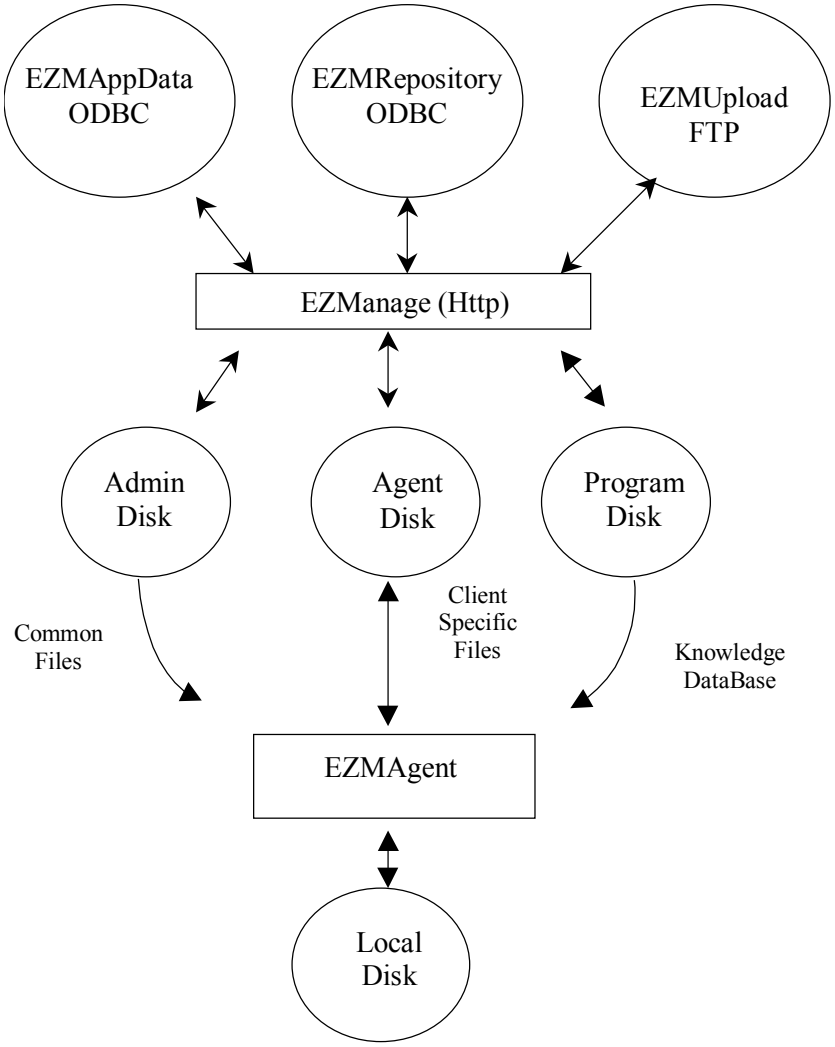
Easy Manage Directories

The directories that will be used by Easy Manage are:

Easy Manage Disks	Directory	Name
Web Server		
Apache Tomcat MainDir	\webapps\ezmanage	EZManageDir
ODBC	EZMAppData	EZMAppData
ODBC	EZMRepository	EZMRepository
File Server		
AdminDisk (N:)	\ADMIN	AdminDir
AdminDisk (N:)	\ADMIN\SITES	SitesDir
ProgramDisk (P:)	\	ProgramDir
ProgramDisk (P:)	\EZMBASE\EZM	MasterDir
ProgramDisk(P:)	\EZMAPPS	ApplDir
AgentDisk (R:)	\	InstallDir
AgentDisk (R:)	\	ReportDir
FTP Server		
EZMAdmin	\ADMIN	AdminDir
EZMAdmin	\ADMIN\SITES	SitesDir
EZMProgs	\	ProgramDir
EZMProgs	\EZMBASE\EZM	MasterDir
EZMProgs	\EZMAPPS	ApplDir
EZMAgent	\	InstallDir
EZMAgent	\	ReportDir
EZMUpload	\	UploadDir
Workstation		
Workstation (C:)	C:\EZM	LocalDir
Workstation (C:)	C:\EZM\PARAM	ParamDir
Workstation (C:)	C:\EZM\WINDOWS	EZMWinDir
Workstation (C:)	C:\EZM\USERS	UserDir
Workstation (C:)	C:\STARTUP	StartupDir
Workstation (C:)	C:\TEMP	TempDir

NB: Do not change the default Easy Manage directories, unless you have good reasons to do.

Schematic view of Easy Manage:



EZManageDir

The EZManageDir contains the complete Easy Manage Web Application. The Easy Manage application is configured as "ezmanage"in the root of the Tomcat wepapps directory.

\
This is the root of the Easy Manage Application. It contains the control logic, started with the index.html file. It contains all .html and the EZMFiles.jar (Compiled application files). The filenames are case sensitive, so DO NOT CHANGE the filenames.

\conf
EasyManage.cfg Easy Manage configuration file

\data_mig
Temporary directory which will be used to migrate from 3.9 to 4.1

\docs
This is the documentation directory
Ezmadmin.pdf Administrator Program Guide
EzmHelpdesk.pdf Helpdesk Program Guide
EzmConf.pdf Configuration Guide
EzmRef.pdf Reference Guide(This file)

\downloads
This is the directory with files to download. It contains the ezmcopy program, which is used for the BOOT DISK in an FTP environment and the Java Runtime Environment programs for the different platforms. After startup, Easy Manage checks if the runtime environment is installed. If not, it will try to install the Java Runtime Environment depending on the platform. The filename is always j2re.exe
\win2kxp Windows 2000, Windows 3000, Windows XP
\winnt98 Windows NT and Windows 9x

\images
This is the directory with all the images.

\log
This is the directory, which will be used to create the loggings

\repository
This is the directory, which will be used to maintain the repository and the CMDB.

\WEB-INF
This is the directory with the compiled servlets and beans

EZMRepository

EZMRepository is an ODBC link defined on the Web Server to the Repository Database. The Repository contains all the definitions of the tables defined in the AppData database.

EZMAppData

EZMAppData is an ODBC link defined on the Web Server to the AppData Database. The AppData contains the tables of the Configuration Management Database.

AdminDir

The AdminDir contains common information and programs.
AdminDir=<AdminDisk>\ADMIN

EZWSINS.EXE Easy Manage Single Installer Program
EZMSYNC.EXE Easy Manage Sync Program

The following files will only be created at initial setup.
ALWAYS.EIF Easy Manage Custom Information File
LICENSE.DAT License File
Servers.INI Servers Control file

The following files will be managed by Easy Manage when something changes in the Database.

MACINVNR.DAT	Mac Address mapping file
UAGENT.UAF	Easy Manage User and Application File
UAGENT.AIF	Easy Manage Application Information File

The following files could be managed per site and will also be in the Site directory

INSTALL.DAT	Install file with encrypted passwords
PARAMS.DAT	Path definitions for the workstation
FIRST.UAP	First Universal Agent Profile
LAST.UAP	Last Universal Agent Profile
\Sites\<SiteName>	Site directory

DatabaseDir

This is the directory where the database of version 3.9 was located.

DatabaseDir=<AdminDisk>\EZMDB

EZM_PAR.DAT	Path definitions for the Admin PC
EZDRVINS.EXE	Easy Manage Driver Installer Wizard
EZW95REP.EXE	Easy Manage Report program
EZMAccessor.exe	Easy Manage Assessor
EZMScan.exe	Easy Manage Network Scanner
PCIDEVS.TXT	PCI devices definition file
PNPID.TXT	PNP devices definition file

ProgramDir

The ProgramDir or Easy Manage **Knowledge Database**, is a directory tree containing all the operating system distributions, the scripts and drivers for applications, adapters and devices for each platform and Application Workspaces that have been created by the Easy Manage Application Workspace Wizard.

The following rules apply for the directory tree:

[<PLATFORM>][\<CLASS>][\<MODEL>]\Script + Files

- **<PLATFORM>** is the tree for the Operating System dependent software. The following operating systems are defined:

\WIN98	Windows 98 Setup Information Files and drivers
\WIN2000	Windows 2000 Setup Information Files Scripts and drivers.
\WINXP	Windows XP Setup Information Files Scripts and drivers.
\WINNT	Windows NT Setup Information Files and drivers
\WINXP	Windows XP Setup Information Files and drivers

- **<CLASS>** is used for the program type. This can be an application, adapter drivers, network or Operating System distribution software. An Operating System distribution has **OS** as CLASS. Class **EZM** is reserved and will be used for the platform templates (**Platform Template Directory)**.

- **<MODEL>** will inform about the version number of a program or the model type of an adapter.

- The script for the installation and the directory is the last entry. It has the name of the CLASS and the extension ".INF".

An application or adapter software for Windows 98 could look like:

```
PLATFORM =WINXP
CLASS=OS
MODEL=UKPRO
        <ProgramDir>\WINXP\OS\UKPRO

PLATFORM =WINXP
CLASS=MSTCPIP
        <ProgramDir>\WINXP\MSTCP\MSTCP.INF
```

PLATFORM =WINXP
CLASS=ETHERCRD
MODEL=3C509
<ProgramDir>\WINXP\ETHERCRD\3C509\ETHERCRD.INF

Platform Template Directory

<PLATFORM>\EZM

Each platform directory has its own templates to create the Initial Workspace. The following files will be distributed by Easy Manage and should not be changed.

DEFAULT.INF	Default Windows setup answer file
DEFAULT.TPL	Default System template (Local install)
NETWORK.TPL	Default System template (Network install)
GHOST.TPL	Default System template (Ghost install)
EZMDISTR.DIS	Easy Manage Distribution Template
EZM.ROF	Easy Manage RunOnce template
POSTINST.BAT	Easy Manage Post Installation template

EZMBaseDir

\EZMBASE

This directory contains the Easy Manage Preparator and the sub directories with the tools. The contents of this directory will be copied to the ramdisk, when a boot is executed with the BOOT DISK.

EZDOSPRP.EXE	Easy Manage Preparation Program
EZDOSPRP.DAT	Easy Manage Preparation Control File
TweakUI.cab	Windows 98 TweakUI environment

ImageDir

\EZMBASE\IMAGE

This directory contains the Partition Dump/Restore program and the tag file if the program could be used

Save.bat	Tag file to execute the Partition Dump program
Restore.bat	Tag file to execute the Partition Restore program

SetupDir

\EZMBASE\Setup

This directory contains the setup distribution

GhostDir

\EZMBASE\Ghost

This directory contains the Ghost tag procedures

MasterDir

\EZMBASE\EZM

This directory contains the Easy Manage Distribution for each workstation and is Operating System independent. The software will be distributed to the client.

The Easy Manage agent will always check for updates that have to be done.

Bootdisk

\BOOTDISK

This directory contains the BOOT DISK distribution software, to generate a BOOT DISK.

\Cabs		All the Cab files with the tools needed by the BOOT DISK procedure, which will be extracted at boottime
\Dos		The current language dependent DOS programs (UK is default).
	\NL	The Dutch version of DOS.
	\UK	The English version of DOS.
\Linux		Bootimage for a floppy to boot a Linux system.
\Floppy		Programs that will be copied to the Boot Disk.
	\Bin	Extended programs needed on the floppy.
	\Root	Startup files needed on the root of the floppy.
\Net		Microsoft MSNET network protocol software.
\Tools		Easy Manage Boot Disk tools.
\CDRomSup		CDRom drivers.
\Drivers		MSNET NDIS driver Database.
\CD		CD-writer software
\UNIDRV		Undi Drivers for PXE

ApplDir

\EZMAPPS
This directory contains the Application Workspaces, generated by the Easy Manage Application Workspace Wizard and is "Operating System independent".
Each application has its own directory tree identified by its unique Application ID. This complete directory tree is an Application Workspace. The sub directories in this directory are transformed to short directory names, so you can even use a server who doesn't support long file names.

InstallDir/ReportDir

This is a directory tree with subdirectories, which are used to transfer data to and from the workstations.

MachineDir
The machine depended data will be written in the MachineDir.
Each machine has its own subdirectory identified by its name <CI-Number>.

UAGENT.ANS	Machine Answer File
UAGENT.ICF	Initial Control File
MACHINE.UAP	Machine Profile
NEWPART.TXT	Partition description
RESTORE.DAT	Restore command file
CHKAPPL.TCH	Reinstall Application command file

LoggingDir
The logging of each workstation will be written to the LoggingDir
LoggingDir=<ReportDir>\Logging

<CI-Number>.LOG	Log File

SysInfoDir
The audit files of each workstation will be written to the SysInfoDir
SysInfoDir=<ReportDir>\SysInfo

<CI-Number>.ESI	Easy Manage System Information File

NetUserDir
The user data will be stored on the NetUserDir.
Each user has its own subdirectory identified by its name \USERS\<UserName>.
NetUserDir=< InstallDir>\Users\<UserName>

<NetUserDir>\USER.ANS	User Answer File
<NetUserDir>\USER.UAP	User Profile

ParamDir

The ParamDir is the directory on the workstation where the install parameters should be stored for that PC.

PCIDENT.DAT	PC identification
<Appl>.EIF	Installed Application Easy Manage Information File

LocalDir

The LocalDir is the directory on the workstation, where the Easy Manage programs files reside and contains data that is generated by the Easy Manage programs.

Easy Manage Main Control Files

This chapter describes the Main Control Files that will be used by Easy Manage.

Parameter file PARAMS.DAT

The Parameter File contains the parameters for the directories that will be used by all Easy Manage programs. This file is located in the AdminDir and will be copied to the LocalDir of the client PC. This is the view from the client PC.

Syntax: <keyword>=<directory_spec>
! Remote directories
InstallDir= (Default: none)
ProgramDir= (Default: none)
AdminDir= (Default: none)
MasterDir= (Default: 'ProgramDir\EZMBASE\EZM')
ApplDir= (Default: 'ProgramDir\EZMAPPS')

If no keyword or a directory is entered, the stated default is taken. The parameter could contain MS-DOS environment variables.
E.g.
DOS:
SET PCIDENT=Node1
SET NETDRIVE=Z:
PARAMS.DAT:
ParamDir=%NetDrive%\%Pcident%
EZMANAGE:
ParamDir will be after the substitution: Z:\Node1

You may use UNC notation for the directory_spec like: \\<Server>\<Share>[\path]

Parameter file EZM_PAR.DAT

The Parameter File contains the parameters for the directories that will be used by the Easy Manage Admin Program. This file is located in the DatabaseDir. This is the view from the Admin PC.

Syntax: <keyword>=<directory_spec>
AdminDir= (Default: none)
InstallDir= (Default: none)
ReportDir= (Default: none)
ProgramDir= (Default: none)

You may use UNC notation for the directory_spec like: \\<Server>\<Share>[\path]

Configuration file EasyManage.cfg

The Configuration File contains the parameters for the directories that will be used by the Easy Manage Admin Program. This file is located in the EasyManageDir\conf.

```
[Database]
DSNAPPData=EZMAppData
DSNRepository=EZMRepository
DBType=MSAccess 2000

[Debug]
LogFilePath=/log
LogPlace=1
LogLevel=0
QMLogLevel=1

 [Serverpaths]
AdminDir=D:\ezmdrives\EZMADMIN\ADMIN
ProgramDir=D:\ezmdrives\EZMPROGS
InstallDir=D:\ezmdrives\EZMAGENT
ReportDir=D:\ezmdrives\EZMAGENT
```

LinuxIn=D:\ezmdrives\EZMADMIN\ADMIN\LINUXIN
LinuxOut=D:\ezmdrives\EZMADMIN\ADMIN\LinuxOut
LoadDir=D:\ezmdrives\EZMADMIN\ezmdb
FTPPort=21
FTPUser=installer
FTPPassword=EADF4E
FTPHost=pluto
FTPUploadDir=ezmupload
FTPEZMProgs=ezmprogs

[Clientpaths]
AdminDir=\\pluto\ezmadmin\ADMIN
ProgramDir=\\pluto\ezmprogs
InstallDir=\\pluto\ezmagent
ReportDir=\\pluto\ezmagent
LinuxIn=\\pluto\ezmadmin\ADMIN\LinuxIn
LinuxOut=\\pluto\ezmadmin\ADMIN\LinuxOut
LoadDir=\\pluto\ezmadmin\ezmdb

[]

PCIdent file PCIDENT.DAT

This file contains the identification of the PC and will be created in the ParamDir the first time the program is executed on that PC.

Easy Manage Package Information file EZMANAGE.EIF

This file contains the commands to copy the Easy Manage programs to the client PC, when programs are updated in the MasterDir.

Easy Manage Custom Information file ALWAYS.EIF

This file will always be executed as last procedure of the Install and Desktop Agent. You can put here the commands to copy or delete your own files to or from the client PC. Only copy and delete commands in this EIF File will be executed.

User Administration file UAGENT.UAF

The User Administration File **UAGENT.UAF** will be generated and stored in the AdminDir by the Easy Manage Admin Program in case one of the keys are changed by an operator action. It describes the registered Easy Manage users with their allowed applications, the configured workstations and the applications to be installed on them. If this file is changed, then the Easy Manage Install Agent will copy this file to the LocalDir of the workstation.

[DeskTop Users]
This section describes all registered Easy Manage users and the Desktop settings for Windows for the users. It also describes the authorization of the users.
There is a record for each user with the following format:

<User>,<Reserved>,<Reserved>,<Reserved>,<Reserved>,<Reserved>,<UserType>

Where :

<User>	Unique User Name Max. 8 characters.
<UserType>	0=Admin, 1=Installer, 2=User
	The Admin is allowed to delete, install and use Application Workspaces.
	The Installer is allowed to install and use Application Workspaces.
	The User is only allowed to use Application Workspaces.

[MachineAuth]
This section describes all configured workstations and their authorization. There is a record for each workstation with the following format:

<CINumber>,<NonRegUsers>,<SwitchUser>,<AskForUpdate>,<CheckAppl>,<AutoLogon>,<Secure>

Where:

< CINumber>	CI Number of the machine max. 8 characters.
<NonRegUsers>	Allow Nonregistered Users 0 = No, 1= Yes.
<SwitchUser>	Allow User Desktop Switching 0=No, 1=Yes.

<AskForUpdate> Show the screen to ask for update 0=No, 1=Yes.
<CheckAppl> Reserved.
<AutoLogon> Allow AutoLogon 0=No, 1=Yes.
<Secure> Secure Workstation 0=No, 1=Yes.

[MachineAppls]
This section describes all Configured Workstations and their applications to install. There is a record for each workstation with the following format:

<CINumber>[,ApplicationIndex[:ScheduleInfo]]...

Where :

< CINumber>	CI Number of the machine max. 8 characters
<ApplicationIndex>	Index to the Application defined in the Easy Manage Application Information File.
<ScheduleInfo>	<ScheduleCommand>:<Scheduletime>
<ScheduleCommand>	'I' = Install 'R' = Remove
<ScheduleTime>	yyyymmddhhnnss

[UserAppls]
This section describes all users with their allowed application(s). There is a record for each user with the following format:

<User>[,ApplicationIndex]...

Where:

<User>	Unique user name (max. 15 characters)
<ApplicationIndex>	Index to the Application defined in the Easy Manage Application Information File.

Site file EZMSites.DAT

The Easy Manage Site File **EZMSites.DAT** will be generated and stored in the AdminDir by the Easy Manage Admin Program in case an application is added to or removed from a workstation. It describes the applications and workstations per site. It could be used as interface for the synchronization process.

<SiteCode>	Code of a defined site
<Count>	Sequence number
<InventoryNumber>	Unique CI Number
<ApplicationId>	Application Workspace Id

There is a section for each site, with the Site Code as sectionname.
[<**SiteCode**>]

The applications and workstations used by a site are defined within a section like:
WKS<Count>=<InventoryNumber>
AWS<Count>=<ApplicationId>

Application Installation file UAGENT.AIF

This Application Installation File **UAGENT.AIF** is the interface between the Easy Manage Admin Program and the Easy Manage Agent. The Easy Manage Admin Program updates the file when a new application is added or one of the concerning keys are changed. The Easy Manage Application Workspace Wizard also updates the file, when an Application Workspace is installed or removed. The file is located in the AdminDir. The Admin Program reads this file, before it is updated. If this file is changed, then the Easy Manage Install Agent will copy it to the LocalDir of the Workstation.

The file consists of a record for each application. Each record is one line with the following format:
<State>,<ApplicationIndex>,<PLATFORM>,<Common>,<Description>,<EIFFile>,<DiskUsage>,<InitialApplication>,<Reserved>,<Reserved>,<PostInstall>,<Version>,<Obsolete>,<Class>,<InstallAfter>

<State>	R[emoved] | I[nstalled] | C[hanged] | U[nchanged] The following actions will be taken by the Easy Manage Admin Program with the state: *Removed*: Empty the EIF Filename and the DiskSize of that application in the Database. *Installed or Changed*: Update the EIF Filename and DiskSize of that application in the Database. *Unchanged:* Do nothing with the information. The Easy Manage Admin Program will always write the information of the application with State

	UnChanged.
<ApplicationIndex>	Unique Number of the application.
<Common>	Is this a Common Application? 0=No, 1=Yes
<PLATFORM>	This is the binary code number from the database.
<Description>	Description of the application
<EIF File>	Path of the Easy Manage Information File
<DiskUsage>	Disk usage of the application in Mbytes
<InitialApplication>	Is this application used in the initial workspace? 0=No, 1=Yes
<PostInstall>	0 = No Action 1 = Logoff 2 = Restart Windows 3 = Reboot the PC 4 = No Session reboot
<Version>	Version Number
<Obsolete>	Is this application Obsolete? 0=No, 1=Yes
<Class>	Class name of the application

Mac Address mapping file MacInvNr.Dat

The Boot disk will use this file to map a Mac Address to the CI Number. The CI Number will be asked for, If there is no mapping or multiple mappings of the same address to different CI Numbers.
The contents of the file is:
<MacAddress>,<InventoryNumber>
The <MacAddress> consists of a 8 pair hexadecimals separated by a colon ":".

Servers file Servers.INI

This file is used by the Boot disk wizard and will be created by the setup program, when the PXE server and/or the FTP server are installed.

```
[PXEServer]
PXEserver=PLUTO

[FTPServer]
FTPHost=PLUTO
FTPPort=21
FTPUser=installer
FTPPassword=<Encrypted Password>

[HTTPServer]
HTTPHost=Pluto
HTTPPort=8080
```

PC Audit file <CINumber>.ESI

The PC Audit File **<CINumber>.ESI** will be generated by the Easy Manage Audit Agent and will be stored in the LocalDir of the workstation. The Easy Manage Collector Agent moves this file to the SysInfoDir, when a network connection is made to the LoggingDir. The EZMScan program will create the same files and store it directly into the LoggingDir.

The file consists of records, where each record is one line with the following format:

[Files]	Start section
<Entry>	<DiskEntry><DirEntry><FileEntry>[<FileEntry>]...[<Entry>]...
<DiskEntry>	"S"<Drive><SizeLeft> For each partition there will be a DiskEntry followed by the directory entries belonging to this partition.
<Drive>	"A"-"Z"
<SizeLeft>	This is the free space left of the partition in Mbytes.
<DirEntry>	"D"<DirectoryNaam> For each new directory there is a directory entry followed by the file entries belonging to this directory.
<FileEntry>	"F"<FileName>,<Size>,<Date>,<Time>
<Size>	0-99999999 (Bytes)
<Date>	yyyymmdd
<Time>	hhmm

Install Control file UAGENT.ICF

The Easy Manage Admin Program will create an Easy Manage Install Control File **UAGENT.ICF** in the install directory for each Workstation, which will have the following structure:

[<Class>]
<Model>,<AnswerTemplate>,<PLATFORM>,<OSName>,<Label>

<Class>	This value is the Class name of the Application, Device or adapter that was selected in this configuration. This value will be substituted for **{Class}** in the System Template. A Class name must be unique in this file. An empty Class means that this is the COMMON Class, which will describe the System Template. The COMMON Class should be declared as first Class in this file.
<Model>	This value will be substituted for **{Model}** in the System Template for all classes except the COMMON Class. It was declared as Model in the configuration database for each (initial-) application, machine type, adapter or device. For the COMMON Class, this will be the full pathname of the System Template (.**TPL** File).
<AnswerTemplate>	This is the Answer Template File, which will be used as input file to generate a Class Answer File (<Filename>.**ANS**) in the Install directory. The filename of the Answer Template will be used as Filename for the Class Answer file. In the COMMON Class, this field will be the default Answer Template (.INF) for the class with the Operating System Type Name. All answer templates with the extension **INF** will be merged to 1 file (WINSETUP.INF) and will be stored in the Install Directory. If this field is empty, then the Easy Manage Preparation Agent will search for:
<PLATFORM>	This field defines the binary coded Operating System Type and is only valid in the COMMON Class.
<OSName>	This field defines the Operating System Type Name and is only valid in the COMMON Class and will overrule the default Type. This is the value that will be substituted for **{OS}** in the System Template.
<Label>	Unique label name, which will be substituted for **{Label}** in the System Template.

Variables enclosed with a percent "**%**" will be substituted by the DOS environment variable.

Easy Manage Preparator Control file EZDOSPRP.DAT

The Easy Manage Preparation Agent (EZNETPRP.EXE) will use the Easy Manage Preparator Control File **EZDOSPRP.DAT**. It contains all keywords from the .INF Files of Windows, which should have concatenated key values in the WINSETUP.INF (Main) File. It should be located in the same directory as the program.

E.g. If you want to use more then 1 protocol for your network, then the key **Protocols** should have a list of protocols to use. The INF File for the MSTCP protocol has the key **Protocols=MSTCP** and the INF File for the NETBEUI protocol has the key **Protocols=NETBEUI**, then the key **Protocols** in the WINSETUP.INF (Main) File will be **Protocols=MSTCP,NETBEUI.** If the key **Protocols** is defined in the Easy Manage Preparator Control File.

Example of the EZDOSPRP.DAT:

```
; The following keys will be concatenated in the INF File;
; If this appears in an INF file, it will be concatenated in the MAIN INF
; File with a comma "," as separator;
;
; Each key should start on a new line;
; A Key will be compared caseless;
;
AddReg
CopyFiles
;
Protocols
Services
Workstations
RemoveBinding
```

Machine and User Answer file UAGENT.ANS

The Easy Manage Admin Program will create a Machine Answer File **UAGENT.ANS** and a User Answer File **USER.ANS**.

The Machine Answer File will be stored in the InstallDir of the workstation (<InstallDir>\<CI-Number>). The User Answer File will be stored in the NetUserDir of the user (<InstallDir>\USERS\<UserName>). The Easy Manage Agents will use these Answer Files as input.

The contents of these files will be:

ANSWER;
<Key>:= <Value>;
…
END;

<Key>	Key that was defined in the substitution (Classes) table
<Value>	The value that was retrieved from the Key field in the substitution table.

Only those keys will be generated, if that Class in the substitution table was chosen in the configuration and not empty. If the Class is other than COMMON or USER, then <ClassName>_ will be placed before the Key Name E.g. MSTCP_IPAddress$.

The following key types are possible:

- String

If a key ends with a dollar "**$**" then the value is a String type. The value has to be enclosed with a double quote '".

- Boolean

If a key ends with a question mark '**?**' then the value is a Boolean type. The only possible values are TRUE and FALSE or "0" and "1".

- Integer

If a key ends with a percent '**%**' then the value is an Integer type.

Example of a Machine Answer File:

Answer;
Netbeui_ValidateLogon? := "0";
MSTCP_EnableDHCP? := "0";
MSTCP_EnableDNS? := "0";
MSTCP_IPAddress$:= "130.144.1.3";
MSTCP_IPMask$:= "255.255.0.0";
MSTCP_IPHostName$:= "demo4";
MSTCP_EnableWINS$:= "0";
Win98_Key$:= "07296-OEM-0009246-53099";
Ethercrd_Model$:= "3C509";
CompanyName$:= "Easy Manage";
Comments$:= "Anna";
MachineName$:= "Demo4";
UserName$:= "Anna";
WorkGroup$:= "Networks";
FullName$:= "Anna Wolski";
InventoryNumber$:= "Demo4";
UserMailAddress$:= "Anna@Domain";
End;

Substitutions and functions

Easy Manage uses substitutions, when it executes the templates. A substitution is a replacement of the value of a parameter from the Database written in Answer Files or derived from the Local Registry.

The substitution of a variable will be done, if the variable is enclosed between a "**{**" and a "**}**". If the substitution fails, because the variable wasn't defined, then the command will not be executed.

The following variables will be substituted in the templates and .UAP files:

Substitution	Derived from	Used in Templates	Remarks
{<Variable>}	UAGENT.ANS, USER.ANS	All	
{Installer}	INSTALL.DAT	All	Decrypted Installer name
{InstallerPW}	INSTALL.DAT	All	Decrypted Installer password
{AdminPW}	INSTALL.DAT	All	Decrypted Local Administrator password
{OS}	UAGENT.ICF	.TPL File	Platform name
{Model}	UAGENT.ICF	.TPL File	Model name
{Class}	UAGENT.ICF	.TPL File	Class Name
{Label}	UAGENT.ICF	.TPL File	Label name
{UserName}	Registry	All	Login UserName
{MachineName}	Registry	All	Machine name
{WorkGroup}	Registry	All	WorkGroup/ Domainname
{1}	Registry	.EIF Files	Windows directory
{2}	Registry	.EIF Files	Windows System directory
{3}	PARAMS.DAT	.EIF Files	LocalDir
{4}	PARAMS.DAT	.EIF Files	LoadDir
{5}	PARAMS.DAT	.EIF Files	AdminDir
{6}	PARAMS.DAT	.EIF Files	MasterDir
{7}		.EIF Files	EZM\<ApplicationIndex>
{8}	Generated	.EIF Files	<LocalDir>\<ApplicationIndex>
{9}	Generated	.EIF Files	EZMAPPS\<ApplicationIndex>
{10}	Registry	.EIF Files	Windows Program Files directory
{11}	PARAMS.DAT	.EIF Files	ProgramDir

The following variables will only be used in the Link Commands section

{a}	Registry	.EIF Files	Desktop Path (Windows 9x)
{b}	Registry	.EIF Files	StartMenu Path (Windows 9x)
{c}	Registry	.EIF Files	StartMenu Programs Path (Windows 9x)
{d}	Registry	.EIF Files	StartMenu Startup Path (Windows 9x)
{e}	Registry	.EIF Files	Desktop Path (Win32)
{f}	Registry	.EIF Files	StartMenu Path (Win32)
{g}	Registry	.EIF Files	StartMenu Programs Path (Win32)
{h}	Registry	.EIF Files	StartMenu Startup Path (Win32)

Functions

In the .TPL, .ROF and .INF files you can use functions, which will return a value that is valuable for the Windows version to install. A function is declared between a tilde **'~'** and will be executed after the substitution is executed. General in all functions: If you use a list, the elements are separated by a comma and it should be declared between a double quote '". A function returns an empty string, when it fails. It is allowed to use another function as parameter of a function. If a row should be deleted, use the key **"{DeleteRow}"** as output parameter of that function.
The following functions could be used:

HexInt(<HexValue>) Returns <Decimal>
Convert from hexadecimal to decimal. This procedure returns the decimal conversion of a hexadecimal number.
Example:
IOAddress = ~HexInt(300)~

This will output :
IOAddress = 768

IntHex(<Decimal>) Returns <HexValue>
Convert from decimal to hexadecimal. This procedure returns the hexadecimal conversion of a decimal number.
Example:
IOAddress = ~IntHex(768)~

This will output:
IOAddress = 300

ParseStr(<List>,<ElementNr>) Returns <Element>
This function returns the nth element from a list of elements given by ElementNr.
Example:
FrameType =~ParseStr("ETHERLINK_2,ETHER_802",1)~

This will output:
FrameType = ETHERLINK_2

GetElmNr(<List>,<Element>) Returns <ElementNr>
This function returns the number of the element in the <List>.
Example:
FrameType =~GetElmNr("ETHERLINK_2,ETHER_802",ETHERLINK_2)~

This will output:
FrameType = 1

BoolToNum(<Boolean>) Returns <Number>
These function returns the number 1 if the Boolean was true, else the number 0. A Boolean is FALSE, when the first character is a "0","N" or "F". A Boolean is TRUE in all other cases.
Example:
UseDHCP=~BoolToNum(Yes)~

This will output:
UseDHCP = 1

YesNo(<Boolean>) Returns <YesNo>
This function returns the string Yes if the Boolean was true, else the string No. A Boolean is FALSE, when the first character is a "0","N" or "F". A Boolean is TRUE in all other cases.
Example:
UseDHCP=~YesNo(Yes)~

This will output:
UseDHCP = Yes

IfTrue(<Value>,<ThenOutput>[<ElseOutput>]) Returns <OutputStr>
This function returns the <**ThenOutput**> if <**Value**> exists and the first character of <**Value**> is not "0","N" or "F".
Example:
Protocols=~**IFTrue**(Yes,ThenOutput,ElseOutput)~

This will output:
Protocols=ThenOutput

IfFalse(<Value>,<ThenOutput>[<ElseOutput>]) Returns <OutputStr>
This function returns the <**ElseOutput>** if <**Value**> not exists or the first character of <**Value**> is a "0","N" or "F".
Example:
Protocols=~**IfFalse**(Yes,ThenOutput,ElseOutput)~

This will output:
Protocols=ElseOutput

EqStr(<Value>,<Condition>,<ThenOutput>) Returns <ThenOutput>
This function returns the <**ThenOutput>** if <**Value**> is equal to the <**Condition**>.
Example:
SelectedKeyboard=~**EqStr**({Keyboard_Language$},"Dutch"," KEYBOARD_00000413")~

This will output:
SelectedKeyboard= KEYBOARD_00000413

NotDef(<Value>,<ThenOutput>,<ElseOutput>) Returns <OutPutStr>
This function returns the value of the <**ThenOutput>** if the variable not is present in the Answer file, otherwise it will return the <**ElseOutput>**.

Universal Agent Profiles

The Universal Profiles are command files to make connections to necessary shares the Agents needs and to execute programs. These are located in the AdminDir and will be copied to the workstation at install time and each time when you modify them.

- The FIRST profile (**FIRST.UAP)** will be used to make an initial connection with a file server and will be called as first action by the Install agent.
- The LAST profile (**LAST.UAP**) will be used to disconnect some connected drives and will be called as last action of the Install agent.
- The USER profile (**USER.UAP**) will be copied (if exists) from the NetUserDir to the workstation and executed after the LAST.UAP.
- The MACHINE profile (**MACHINE.UAP**) will be copied (if exists) from the MachineDir to the workstation and executed after the LAST.UAP.

The following commands could be in a profile:
Only the first character of the command is needed.

- **C**[onnect] Connect a device to a share on a server (Permanently)
- **D**[isconnect] Disconnect a device
- **E**[xecute] Execute a program
- **P**[rofile] Include an other profile
- **M**[ap] Map an UNC path to a device
- All other characters at the beginning of a line will be seen as comment.

General

<Device>	a drive letter or LPT name ended with ':'
<Server>	the name of the server to connect
<Share>	the name of the share on the server to connect. For Novell use the syntax <VolumeName>:<ShareName>.
<Password>	the password of the service to connect, if a password is needed
<Profile>	the full filename of the profile to execute
<Program>	the full filename of a (Windows) program to execute
<NoWait>	N or n: Don't wait until the program is ended.
<Default>	D or d: Default printer.
<Queue>	Printer queue name.
<LogonAs>	Name of the user for the resource.

A command will not be executed, when a substitution variable could not be evaluated.

Connect a device to a share on a server
This command will disconnect the device if already connected and will connect the <Device> to the <Share> defined on the <Server>. If the <Device> is not ended with a semicolon ':' it will be seen as a printer queue name. If a <Queue> is specified, then the <Server> and <Share> will be assigned as UNC to the printer queue on the workstation.
The connection will be permanent during the session, until the command Disconnect is given.

C[onnect],<Device>|<Queue>,<Server>,<Share>,[<Password>],[<LogonAs>],[<Default>]

Map a device to a share on a server
This command will map the <Share> defined on the <Server> to a drive. This means, that the drive names defined in the PARAMS.DAT file are substituted by \\<Server>\<Share> during executing time.
The connection is temporary during the execution of the agent.

M[ap],<Device>,<Server>,<Share>

Disconnect a device
This command will disconnect the given <Device>.

D[isconnect],<Device>

Execute a program
This command will execute the (Windows) program <Program>. If the NoWait flag not is used, then the next command will be executed without waiting for the called program to be finished.

E[xecute],<Program>,<"Parameter(s)">[,**N**oWait]

Include a profile
This command will read and execute the given <Profile>.

P[rofile],<Profile>

Example
The contents of the **FIRST.UAP** is:

```
! First Universal Agent Profile
!
! This file will be on the workstation
! Map to the Admin Disk
Map,N:,EZMServer,ezmadmin$
! Map to the Programs Disk
Map,P:,EZMServer,ezmprogs$
! Map to the Agents Disk
Map,R:,EZMServer,ezmagent$
! Connect to the Users Home directory
Connect,H:,EZMServer,{UserName}
```

The contents of the **LAST.UAP** is :

```
! Last Universal Agent Profile
!
! Include hook from AdminDir (Example)
! Profile,N:\ADMIN\HOOK.UAP
!
! Include a profile depending on the MachineName
! Profile,N:\ADMIN\{MachineName}.UAP
!
! Disconnect the Agent disk
! Disconnect,R:
! Disconnect the Program disk
! Disconnect,P:
!
! Include a profile depending on the WorkGroup
!Profile,N:\ADMIN\{WorkGroup}\GROUP.UAP
!
! Include a ccMAIL Profile depending from the ccMAIL_PO$ in the UAGENT.ANS
Profile,N:\ADMIN\CCMAIL\{ccMAIL_PO$}.UAP
!
! Include a profile depending on the Username
!Profile,N:\ADMIN\USERS\{UserName}.UAP
!
! Disconnect the Admin disk
! Disconnect,N:
!
! Include a user profile
! Profile,H:\USER.UAP
```

Easy Manage Templates

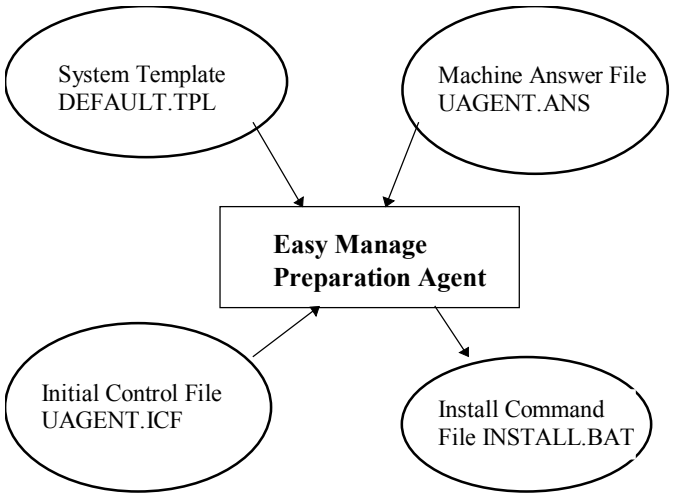
Easy Manage uses templates to generate the Install Procedures.
If the value of a substitution not could be evaluated, then the line will not be copied.

Easy Manage uses the following templates:

- System Template (.TPL)
- Class Answer File (.INF and .ANS)
- System RunOnce template and Class RunOnce templates (.ROF)
- Post install procedure (POSTINST.BAT)
- Distribution procedure (EZMDISTR.DIS)
- Windows Setup Information File and Class Windows Setup Files (.INF)

System Template <Method>.TPL

The System Template will be used to define the sequence of the installation of all-possible Initial Software programs and drivers for an Operating System. For each kind of operating system you will have a separate System Template which is located in the <Platform>\EZM directory. Easy Manage delivers each time a DEFAULT template. If you want to change the System Template, you have to make a copy of the DEFAULT System Template and use your template (See Software -> Operating Systems). This template will be used by the Easy Manage Preparation Agent to create an Install Command File (INSTALL.BAT) for each configuration, which will be created in the InstallDir \<CI-Number> for that Workstation. A System Template will have the extension **.TPL**.



The contents of the System Template is as follows:

- It contains Class and Common Sections to define the commands to execute. A Section starts with a section header and ends when another Section starts or when the end of file is reached. A Section Name is enclosed with a "[" and a "]".
- A Common Section (an empty Section Name) will be copied unconditional to the Install Command File (INSTALL.BAT).
- A Class Section will only be copied to the Installation Command File, if its Section Name is selected as Class in the configuration (It was defined in the Installation Control File).

Easy Manage distributes the following templates:

Platform	**Default.TPL**	**Network.TPL**	**Ghost.TPL**
Win98/ME	Recommended	NOT Possible	Possible
Windows NT	Possible	Recommended	Possible
Windows 2000	Recommended for notebooks	Recommended	Possible
Windows XP	Possible	Possible	Recommended

Post Installation procedure POSTINST.BAT

The Post Installation procedure POSTINST.BAT, is located in the <Platform>\EZM directory. This file will be copied by the System Template to the workstation during the installation and will be used to execute the following tasks, when the BOOT DISK is removed and the system was rebooted the first time:
The Windows Setup procedure will be executed for all platforms. For Windows 9x platforms a Config.SYS and anAutoExec.BAT file will be created after the setup.
For Windows 9x platforms created with Ghost, it will add the Easy Manage Agent in the registry. Easy Manage will maintain and deliver this file for each version. So if you want to change this template, then you have to copy the original and change the DEFAULT System Template.

RunOnce template EZM.ROF

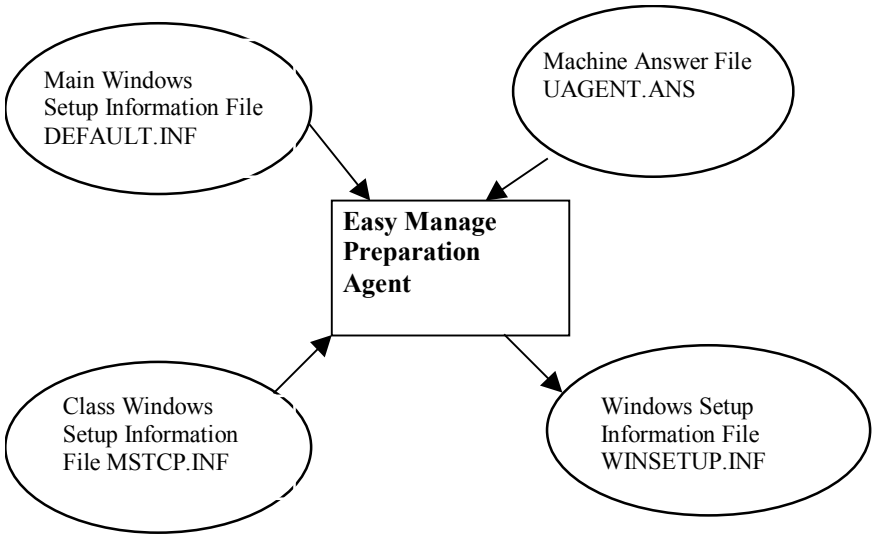
The Main RunOnce template EZM.ROF, is located in the <Platform>\EZM directory. This file will be copied by the System Template to the workstation during the installation as EZM.INF and will be used to execute some registry settings, when the Windows Setup procedure is finished. Easy Manage will maintain and deliver this file for each version. So if you want to make changes to this template, then you have to create a Class RunOnce template .ROF file in the class directory, with the name of the class. E.g. Netware.ROF. The processing will be the same as for the Windows Setup Information Files.

Windows Setup Information Files

The Windows Setup Information Files will be used by the Easy Manage Preparation Agent to generate the answers for a setup of a Windows platform.

Easy Manage uses 2 kinds of Windows Setup Information Files:

- Main Windows Setup Information File Default.INF
- Class Windows Setup Information File <Class>.INF



The Class Windows Setup Information File(s) will be merged with the Main Windows Setup Information File and will be stored in the Windows Setup Information File <InstallDir>\<CI-Number>\WINSETUP.INF. The values that should be substituted will be derived from the Machine Answer file.

The Preparation Agent searches for Class Answer files with the following sequence:
<ProgramDir>\<PLATFORM>\<Class>.ANS,
<AdminDir>\<Class>.INF,
<ProgramDir>\<PLATFORM>\<Class>\<Model>\<Class>.INF,
<ProgramDir>\<PLATFORM>\<Class>\<Model>\<Model>.INF,
<ProgramDir>\<PLATFORM>\<Class>\<Class>.INF.

Main Windows Setup Information File WinSetup.INF

The Main Windows Setup Information File is an answer script with the answers for a minimum Windows setup. Only the necessary answers will be defined in this file and is located in the <Platform>\EZM directory. The Class Windows Setup files will be included in this file. Keys in the same section will be placed in that section. Equal keys in the Class Window setup file overwrite the keys in the Main Windows Setup file unless this key was defined in the **EZDOSPRP.DAT** file.

Example of a Main Windows Setup Information File:

```
; File: Win98.INF
; This is the main Windows 98 setup answer script.
 [Setup]
ProductType=1
Express=1

•••

[System]
DisplChar=8,800,600
Multilanguage=english
 [NameAndOrg]
Display=0
Name="{UserName$}"
Org="{CompanyName$}"
[Printers]
 [Network]
Display=0
```

ComputerName="**{MachineName$}**"
Workgroup="**{WorkGroup$}**"
Description="**{Comments$}**"
Clients=VREDIR
IgnoreDetectedNetCards=0
WorkstationSetup=0
DisplayWorkstationSetup=0
HDBoot=1
RPLSetup=0
Security=share
[VREDIR]
LogonDomain="**{WorkGroup$}**"
ValidatedLogon=0

Class Windows Setup Information file <Class>.INF

The Class Windows Setup Information file is an Additional Information File with the answers for the Windows setup that has to be used for the installed Windows version. These files are located in the Knowledge Database for the specified class and the Windows version. These files will be merged with the Main Windows Setup Information file and copied to the Windows Setup Information File (**WINSETUP.INF**) in the TempDir.
The key of the latest Setup file will replace same keys within a section in sequence. Some keys will be concatenated (depending on the Easy Manage Preparation Data File **EZDOSPRP.DAT**).

Example of a Class Windows Setup Information File: C:\EZMPROGS\WIN98\MSTCP\MSTCP.INF

; File: MSTCP.INF Date: 980608
; This is the answer script for MS/TCP which will be inserted in the
; MAIN file (WIN98INF).
[Network]
; This key will be concatenated in the Main file
Protocols=MSTCP
; This key will be replaced in the Main file
DefaultProtocol=MSTCP
[MSTCP]
; The following keys will be replaced in the MSTCP section of
; the MAIN file
DHCP=**{MSTCP_EnableDhcp?}**
DNS=**{MSTCP_EnableDNS?}**
DNSServers=**{MSTCP_DNSServers$}**
Domain=**{MSTCP_DomainName$}**
DomainOrder=**{MSTCP_DomainOrder$}**
Gateways=**{MSTCP_Gateways$}**
Hostname=**{MSTCP_IPHostName$}**
IPAddress=**{MSTCP_IPAddress$}**
IPMask=**{MSTCP_IPMask$}**
LMHostPath=**{MSTCP_HostPath$}**
PrimaryWINS=**{MSTCP_PrimaryWins$}**
SecondaryWINS=**{MSTCP_SecondaryWins$}**
ScopeID=**{MSTCP_ScopeID$}**
WINS=**{MSTCP_EnableWins$}**

Easy Manage Information File

To control the agents, Easy Manage works with Easy Manage Information Files in each Workspace. The Easy Manage Agents will create the following Information files:

Application Workspace <Application>.EIF
Windows Workspace WINDOWS.EIF
Session Workspace SESSION.EIF
User Workspace USER.EIF

General

- Each row started with a colon ' ;' is comment.
- Fields with comma's and double quotes, should be enclosed with double quotes ".
- If no drive is specified in a source path, then the location of the original EIF File will be used.

History Section

This section describes the history of this Application Workspace.

[History]
All rows of this section will be displayed in the window of the user.

<Information>

Install Section

Install data will be placed in this section

[Install]
User=<UserName>
ApplType=<ApplType>
Installer=<InstallerName>
DiskUsage=<DiskUsage>

<UserName> Active user name.
<ApplType> "SNAPSHOT" | "AUTOIT".
<InstallerName> Name of the installer.
<DiskUsage> Disk space needed to install the application in Mbytes.

File Commands

With the File Commands, you can copy files to the workstation or delete files from the workstation.

The following keys will be used:
<SourcePath> Full path of the directory where to copy from.
<File> Filename to be copied or deleted
<DestPath> Full path of the directory to copy to.
<Path> Full path of the directory to delete.
<Condition> Copy condition
'I' Initial only. The file will be copied only if the destination file does not exist.
'U' Unconditionally. The file will be copied unconditionally.
'C' Conditionally or Empty (Default). The file will be copied only, when the destination file is older then the source file.
'O' Copy Once. Like initial, but the file will never be deleted.

[Copy Files]
This section describes the files that should be copied from the source to the destination directory.
A new directory will be created automatically. The file will be copied according to the condition.
<Reserved>,<SourcePath>,<File>,<DestPath>,[<Condition>],[<ShortFileName>]

[Remove Files]

This section describes the files that should be removed. An empty directory will be removed automatically. A file will only be removed, if no other application needs this file.

<Reserved>,<Reserved>,<File>,<Path>,[<Condition>]

LNK Files Commands

These sections describe all the LNK files from the desktop that should be created or removed.
The following keys will be used:

<Description>	Name of the object.
<FileName>	Full Filename of the object.
<Command>	Command to Execute.
<Arguments>	Arguments to be executed by the command.
<WorkingDir>	Directory to go to before execution.
<IconPath>	Full path of the file with the Icon.
<IconIndex>	Index of the Icon in the Icon file.
<ShortCutKey>	ASCII presentation of the ShortCutKey.
<WindowState>	State of the window 0 = 1 = 2 =
<IdList>	Pif Id list

[Add LnkFiles]
This section describes all the Lnk and Pif files that should be created. A new directory will be created automatically.

<Application>,<Description>,<FileName>,<Command>,<Arguments>,<WorkingDir>,<IconPath>,<IconIndex>,<ShortCutKey>,<WindowState>,<IdList>

[Remove LnkFiles]
This section describes the files that should be removed. An empty directory will be removed automatically. A file will only be removed, if no other application needs this file.

<Reserved>,<Description>,<FileName>[,<Command>,<Arguments>,<WorkingDir>,<IconPath>,<IconIndex>,<ShortCutKey>,<WindowState>]

EIF Files Commands

This section describes all EIF files, which will only be used in the Session Workspace.

The following keys will be used:

<EIF Path>	Full path of the EIF path.
<EIF File>	EIF File name.
<CommonApplication>	"0" No Common Application, "1" Common Application.
<Version>	Version of the application
<Description>	Description of the Application.
<Obsolete>	"0" Not Obsolete, "1" Obsolete.
<ExitCode>	
<SavedState>	

[EIF Files]
<ApplicationId>,<EIF Path>,<EIF File>,<Reserved>,
<CommonApplication>,<Version>,<Description>,<ExitCode>,<SavedState>

Autoexec Commands (WIN9x)

With the Autoexec Commands, you can add commands to the AUTOEXEC.BAT procedure. These commands will NOT be added directly into the AUTOEXEC.BAT, but will be added to the batch file <StartupDir>\<Application>.BAT, which will be called by the AUTOEXEC.BAT.
The <StartupDir> will be retrieved from the DOS Environment EZMStartup and can be overruled by the parameter StartupDir in the **PARAMS.DAT**.
The following keys will be used:

<Command> Command that will be added to the <Application>.BAT file.

[Add Autoexec]
These are the commands to add to the AUTOEXEC.BAT.

<Reserved>,<Command>

Registry Commands

With the registry commands, you can Add, Remove or change the registry entries on a workstation.

The following keys will be used:

<RegistryRoot>	'0' = HKEY_CLASS_ROOT, '1' = HKEY_LOCAL_MACHINE, '2' = HKEY_CURRENT_USER, '3' = HKEY_CURRENT_CONFIG, '4' = HKEY_USERS.
<RegistryType>	'0' = WIN9x, '1' = WIN32.
<RegistryPath>	Path relative to the registry root
<Key>	Key ends with an equal sign '= ' in the INI file.
<FormatField>	String | <IntValue> | <BinValue>
<SearchCondition>	String
<KeyType>	"S' = String 'I' = Integer 'B' = Binary
<Key>	Key ends with an equal sign '= ' in the INI file.
<IntValue>	int:<HexString>, key's will be continued with a '-' sign at the end of the line.

Section Registry Commands
For each Registry Path there will be a separate section. The actions should be defined for each Registry Key in the Path.

[<Reserved>,<RegistryRoot>,0,<RegistryPath>**]**

Add Key
The Key with the FormatField as value will be added to the path.

<KeyType>,**A**,<Key>,<FormatField>

Update Key
The Key with the FormatField as value will be added to the Path.

<KeyType>,**U**,<Key>,<FormatField>,<SearchCondition>

Delete Key
The Key with value will be deleted from the Path, if the SearchCondition meets the value (Caseless).

<KeyType>,**R**,<Key>,< SearchCondition>

INI Key Commands

With The INI Key Commands, you can Add, Remove or change keys in the INI Files.

The following keys will be used:

<IniFilePath>	Path of the INI file. If this field is empty, then the Windows Directory will be used. If the path not directs to the C: drive, the commands will be merged to the file, when the user uses the application for the first time.
<IniFile>	Name of the INI file.
<Section>	Section name in the INI file where the commands in this section has to be executed.
<FormatField>	String. If this field is a tilde '~', then the key has NO equal sign.
<Key>	Key ends with an equal sign '=' in the INI file.
<SearchCondition>	String.

Section INI Key Commands
For each section in a INI File there will be a separate section. The actions should be defined for each key in the section.

A section will be created, if a key should be added in a section and that section does not yet exist.

[<Reserved**>,**<IniFilePath>,<IniFile>,<Section>[,<Initial>]**]**

Add Key
The Key with the FormatField as value will be added to the section, if the Key with its value is unique in the section.

<KeyType>,**A**,<Key>,<FormatField>

Delete Key
The key with value will be deleted from the section, if the SearchCondition meets the value (Caseless).
<KeyType>,**D**,<Key>,<SearchCondition>

Update Key
The FormatField will replace the value, if the Key exists in the section. If the Key was absent in the section, it will be added in the section.

<KeyType>**,U**,<Key>,<FormatField>

Substitute Key
If the Key exists in the section, then the FormatField will substitute the SearchCondition. If the SearchCondition is absent in the Key value, then the FormatField will be appended to value of the Key.

<KeyType>,**S**,<Key>,<SearchCondition>,<FormatField>

Directory commands

With the Directory commands, you can create or delete directories. This action wil be done as last action of an installation.

<Command> “D” | “d” Delete a directory tree
“C” | “c” Create a driectory
<Directory> Directory name to create or to remove.

[Directory]
With this command you can start a program when the application is installed or updated.

<Reserved>,<”Command”>,<Directory>]

Printer Commands

With the Printer Commands, you can Add or Remove a printer.

<ServerName> Name of the server
<PrinterName> Name of the Printer for the user
<PortName>
<DriverName>
<Processor>
<DataType>
<Version>
<Environment>
<DriverPath>
<DataFile>
<ConfigFile>

[Add Printer]
With this command you can add a printer to the workstation.

<Reserved>,<ServerName>,<PrinterName>,<PortName>,<DriverName>,<Processor>,<DataType>,<Version>,<Environment>,<DriverPath>,<DataFile>,<ConfigFile>
[Remove Printer]
With this command you can remove a printer

<Reserved>,<ServerName>,<PrinterName>[,<PortName>,<DriverName>,<Processor>,<DataType>,<Version>,<Environment>,<DriverPath>,<DataFile>,<ConfigFile>]

Execute commands

With the Execute commands, you can execute a program when the application is installed.

<Command> Programname (with parameters) to execute.
<NoWait> Use the key “NoWait” or “Nowait” if the installer shouldn’t wait until the program is finished.
<Secure> Run the command in the secure mode.
<WaitForFile> Wait until the specified file is present.

[Execute]
With this command you can start a program when the application is installed or updated.

<Reserved>,<”Command”>,[<NoWait>],[<Secure>],[<WaitForFile>]

Uninstall commands

With the Uninstall commands, you can execute a program when the application is removed.

<Command> Programname (with parameters) to execute.
<NoWait> Use the key “NoWait” or “nowait” if the installer shouldn’t wait until the program is finished.
<Secure> Run the command in the secure mode.
<WaitForFile> Wait until the specified file is present.

[UnInstall]
With this command you can start a program when the application is removed.

<Reserved>,<”Command”>,[<NoWait>],[<Secure>],[<WaitForFile>]

AutoIt commands

AutoIt is a freeware program and was primarily designed to assist in automatically installing software.
Many software packages such as MS Office, IE4, Mcafee have their own automated installation routines - most applications don't. AutoIt gets around this by using a script file to perform window commands (waiting for, hiding, activating, etc.) and to send keystrokes to specific windows. This allows AutoIt to perform the automatic installation of such software. It can also be used to automate simple tasks.

Most keystrokes can be sent (i.e. not just alphanumerics, but characters like
!"$%^&*()_-+=, using any language keyboard. Also, left and right mouse clicks
can be sent.

For a full explanation of the AutoIt commands see the help file of the distributed AutoIt package.

With the AutoIt commands, you can create a script for the AutoIt program to Setup or Remove an Application, which could not be executed by the Desktop Installer. The script file will be created dynamically and executed by the AutoIt program.

<AutoItCommand> AutoIt command to execute

[AutoIt Setup] **Version 1.8**
[AutoIt2 Setup] **Other Versions**
With this command you can build an AutoIt script which will be executed when the application is installed.

<Reserved>,”<AutoItCommand>”

[AutoIt Remove] **Version 1.8**
[AutoIt2 Remove] **Other Versions**
With this command you can build an AutoIt script which will be executed when the application is removed.

<Reserved>,”<AutoItCommand>”

AutoIT Tips & Tricks

AutoIT is a very powerful tool, which aids in the scripting process for Easy Manage. It fills the gap where Silent Setup just isn't enough or can't be used at all. With a few simple commands you can do so much. Here are some tips & tricks. We assume that you have installed AutoIt version 2 or higher on your system, where you want to script.

Basic commands
In every script there are a few important commands that will be used frequently. These are:

Run	- Runs your program
WinWait	- Waits until a certain window pops up
WinActivate	- Gives a certain window focus
Send	- Sends a keystroke

With these commands, in this particular order, you can script entire applications.

Example AutoIT script:

```
Run,%1%\\AutoIt.exe

WinWait,WinZip Self-Extractor,AutoIt Setup
WinActivate,WinZip Self-Extractor,AutoIt Setup
Send,{ENTER}

WinWait,WinZip Self-Extractor - AutoIt.exe,Please click
WinActivate,WinZip Self-Extractor - AutoIt.exe,Please click
Send,!s
```

How to get accurate window captions, window text etc.
Accuracy is of the most importance thing in scripting. A typing error will halt your script. Therefore it's better to cut and paste the window caption and window text. You can do this with the AutoIT Reveal Mode program in the AutoIT startmenu. Start this program and select your window. AutoIT Reveal Mode program shows all information from that particular window. Just cut and paste what you need.

Losing focus
It sometimes happens that during script execution your main window loses focus due to various circumstances, with the result that your script halts. This can be easily solved. Just add the command **WinActivate** after the **WinWait** command. Before it sends the keystroke it activates the window and gives it focus.

Your script would look like this:

```
WinWait,Window Caption,WindowText
WinActivate,Window Caption,WindowText
Send,!n
```

Compile your scripts
It can be handy to compile your scripts to EXE. The main advantage is that you can combine different scripting techniques in one EIF-file. For example when you script Acrobat Reader you have a part which is Windows Installer (silent setup) and an AutoIT script which accepts the Licence Agreement for your users. The

Compiling can be done in two ways: right-click your script and select Compile from the menu or select Convert Script to EXE from the AutoIT startmenu. In the latter stage you can even customize your icon.

Adding parameters
In many cases you want to use the Easy Manage substitutions in your scripts. You can do this by adding them in the EIF-file on the commandline of your compiled AutoIT script. In the AutoIT script you can use these parameters with the %1%, %2%, %3% etc. parameter. %1% is the first commandline parameter, %2% the second and so on.

Your EIF-file would look like this:

```
[Execute]
,{11}\{9}\radeon.exe {11}\{9},wait,Secure,
```

Your AutoIT script would look like this:

run,%1%\\Setup.exe

When you use parameters for paths, don't worry about the double backslashes you normally need. AutoIT will take care of that. Those are only needed inside the scripts.

Beware when you use Easy Manage substutions as parameters when they have spaces in them, for example CompanyName$. When your companyname is Easy Manage B.V. it is treated as 3 different parameters.

Trapping occasional windows
Sometimes you get different windows during script execution, than you had during testing. This is mostly due to other installed software. But you can trap this. First pinpoint the location in your script where it halts. Then add the following code here:

WinWait,Window Caption,WindowText,**30**
IfWinExist,WinActivate,Window Caption,WindowText
IfWinExist,Send,!n

The script waits, in this case, 30 seconds for the occasional window to popup. If it exists the WinActive and Send will be processed, otherwise it proceeds with the next line.

Browse the AutoIT help file
The most important tip. Just browse the help file. It has an extensive list of commands you can use and absolutely more than you would think. Just read it!

Test the Preparation

Easy Manage supplies a very useful Tool to test the preparation of a configuration. You can start the tool and view the files with the file menu commands of the Easy Manage Admin Program when the (Add/Modify) Workstations function is selected. This Tool creates the configuration INSTALL.BAT, the .ANS and .INF Files in a very quick way without actually installing the configuration. Normally you would test a configuration by inserting the BOOT DISK in a machine and follow the standard procedures, but it's much faster to make a Test Preparation.

The Easy Manage Preparation Agent will generate the files in the directory C:\TEMP\EZMINST. These files are identical to the files created by a normal installation.

The following files will be created:

1) The file INSTALL.BAT is the merge file from the System Template (*.TPL) defined in the AdminDir put now with the parameters of the UAGENT.ICF file.
2) The file WINSETUP.INF and/or *.ANS, depending on the operating system, for combining answer files, original from the AdminDir, with the answers from the UAGENT.ANS into a file that is used within the windows setup.
3) The file EZM.ROF, this is the common Easy Manage Run Once File.
4) The file <LoggingDir>\<CI-Number>.log with all actions, warnings and errors.

You can check these files before installing a configuration. This might be handy when configuring a new type of one of the components. Of course it is necessary to have some basic technical knowledge to interpret the generated files. For inspecting the files you can use any kind of Text Editor (i.e. Notepad).

Adding a driver in the Knowledge Database by hand

Easy Manage supplies a Driver Installer Wizard to add drivers and initial software in the knowledge database (See Easy Manage Configuration Guide), but it is possible to place the drivers by hand.

Boot Disk:
The Boot Disk drivers will be handled different then other platform drivers.
The drivers and its heuristics needed for the Network Operating System, will be placed in the Boot Disk database <ProgramDisk>\Bootdisk\Drivers \DEVICES.INI and the driver program will be placed in the <ProgramDisk>\Bootdisk\Drivers directory.

The Boot Disk Database DEVICES.INI has the folowing format:
<Enum> An enumerated field started with the number 0.
<Model> modelname of the network card adapter.
<VendorId> PCI vendor id of the network card.
<DeviceId> PCI device id of the network card.
<Description> description of the network card.
<Driver> driver program of the card.
<Name> binding name in the PROTOCOL.INI file.
<DriverName> drivername in the PROTOCOL.INI file.

The detection program to detect a PCI network card will use the section “Detect”. Each PCI network card that could be decteted has a row in this section.
[Detect]
DEVICE<Enum>=<Model>,<VendorId>,<DeviceId>

The “NETMENU” section will be used to select the network card in the Boot Disk Creation Wizard program.
[NETMENU]
ITEM<Enum>=<Description>,<Model>

There will be a section for each network card with the card specific heuristics.
[<Model>]
Driver=<Driver>
Name=<Name>
DriverName=<DriverName>

Other Platforms:
The Driver Installer Wizard uses Distribution templates for each platform. This Distribution Template (**EZMDISTR.DIS**) is located in the <Platform>\EZM directory and will be copied as EZMDISTR.BAT file to the Model directory of the adapter, device, machine or Initial Software directory.

The following files could be created in the model directory too:
EZMClass.TXT
This file has 1 line in it with the new sub directory name to the local distribution point. This file is only needed, if the distribution point should be different then the CLASS.

EZMInfs.TXT
This file is only needed for Windows 9x/ME systems, to add the OEM Inf Files to the CUSTOM.INF file. Each line in this file contains the name of the OEM Setup file.

RunOnce.TXT
This file is only needed for Windows NT/2000 and XP systems, to add commands to run as runonce during Windows setup. Each line has the following contents:
“{SetupDir$}\<Model>\<RunOnce Program with parameters>” and will be added to the CMDLINES.TXT file.

<Class>.INF
A <Class>.INF File will be used to merge in the Main Windows Setup Answer file. The content of the file depends on the platform and driver specifications.

The following example is an Ethercrd.INF file for a Windows 9x platform.
; Ethercrd.inf
; 3Com 3C920 Integrated Fast Ethernet Controller (3C905C-TX Compatible)
; Model:3C920

```
[Network]
IgnoreDetectedNetCards=1
ValidateNetCardResources=0
Netcards="PCI\VEN_10b7&DEV_9200"
[PCI\VEN_10b7&DEV_9200]

[Install]
AddReg=DisablePNP

[DisablePNP]
HKLM,\System\CurrentControlSet\Services\Class\Net\0000,"DisableWarning",,"1"
```

The following is an example of an ETHERCRD.INF file of a Windows NT platform.

```
; Ethercrd.INF
; 3Com EtherLink PCI NIC
; Model:3C920
[Network]
InstallAdapters=AdapterSection
[AdapterSection]
3C90X= AdapterParams,~NOTDEF({SetupDir$},"C:\I386",{SetupDir$})~\$OEM$\Ethercrd
[AdapterParams]
```

<Model>.INF

A <Model>.INF File will be used by the Windows 9x/ME platforms to merge RunOnce commands in the Main Windows Setup Answer file. The content of the file could have the following content:

E.g. This is an example of

```
; CP200L.INF File
[Install]
AddReg=Add.RunOnce.CP200L

[Add.RunOnce.CP200L]
HKLM,Software\Microsoft\Windows\CurrentVersion\RunOnce\Setup,"CP200L-
Setup.exe",,"~NOTDEF({SetupDir$},"C:\WIN95",{SetupDir$})~\CP200L\Setup.exe -s"
HKLM,Software\Microsoft\Windows\CurrentVersion\RunOnce\Setup,"CP200L-
dcom95.exe",,"~NOTDEF({SetupDir$},"C:\WIN95",{SetupDir$})~\CP200L\dcom95.exe -r:n -q:a"
```

Import/Export Drivers and Applications

The drivers and applications in the Configuration Management Database and the Knowledge database could be exported from one Easy Manage environment and imported into another Easy Manage Environment.

The following requirements are needed for Export/Import:

1. An FTP server with the virtual directory "**ezmprogs**". The virtual directory ezmprogs should be mapped to the same directory as the share ezmprogs$. The virtual directory should have the **full** user rights.
2. The directive FTPEzmProgs=ezmprogs should be defined in the EasyManage.cfg file.

   The setup of Easy Manage configures the Ceasar FT Server according the requirements.

3. The user is authorized to use the export and/or import function.
   Only Assignees where the Import and/or Export function are enabled could use the function.

The following objects could be exported/imported:
Initial Software, Application Workspaces, Machine Types, Adapter and Devices.

Export
Select the specific object to export and click the export button. Now a popup screen will ask you for the directory to store the contents. The Database object and the structure in the Knowledge Database will be exported to the given destination directory on a recursive way. It is possible to export more objects to the same export directory.

Import
Go to the list screen of the object and click the import button. Now a popup screen will ask you for the object.xml file to import. Search the object.xml file in the directory where the export did store the object.

Using Ghost with Easy Manage

Easy Manage can work with other distribution systems. You can use the image file creator program Ghost, if you have a lot of workstations with the same hardware configuration.

N.B. Cloning a Ghost workstation will only be supported for FAT16 or FAT32 drives at this moment.

Setup the Ghost environment (once)

To setup the Ghost environment, follow **step 1** until **step 4**.

Step 1. Install the Ghost program.
a. Copy the Ghost.exe program to the <ProgramDir>\EZMBase\Image directory.
b. Copy the tag files **Save.bat** and **Restore.bat** from the <ProgramDir>\EZMBase\Ghost to the <ProgramDir>\EZMBase\Image directory.
c. Tune these files for other versions of Ghost or even for another image file creator program.

Step 2. Create the class **"Ghost"** (if not yet exists).
With the Easy Manage Admin Program (Setup ->Classes -> New)

- Enter class **Ghost** as name
- Select **Initial Application** as class type.

Step 3. Create the Initial Application "**Reseal (XPOnly)**". (Only for Windows XP)
You can skip this step, if you don't have to reseal the PC for reselling.
This application will copy the needed files to the workstation to start the reseal proceure.
With the Easy Manage Admin Program (Software -> Initial Software -> New).

- Enter **Reseal (XP Only)** as Description.
- Select **Initial Application** as class
- Tab Script, Enter **Reseal** as Model
  Easy Manage has already placed the **InitAppl.INF** and the **Distrib.bat** file in the **Reseal** Model directory.
- Tab Platforms, select the platforms, where to use this application

Step 4. Create the Initial Application "**Sysprep**".
This application disables the "convert to NFS" and disables the "automatic register" for Windows XP. It will copy the files needed for sysprep to the workstation.
With the Easy Manage Admin Program (Software -> Initial Software-> New).

- Enter **Sysprep** as description
- Select **Initial Application** as class
- Tab Script, Enter **Sysprep** as Model
  Easy Manage has already placed the **InitAppl.INF** and the **Distrib.bat** file in the **Sysprep** Model directory.
- Tab Platforms, select the platforms, where to use this application

Setup for each specific platform

For each specific platform, which has to be distributed with the Ghost program, **step 5** until **step 12** have to be carried out.

Step 5. Create an Operating System **"<Platform> Ghost"** with GHOST.TPL as System Template, where <Platform> is the platform to distribute.
This Operating system distribution type will be used to setup each PC every time from the Image for a specific platform.
With the Easy Manage Admin Program you have to create an Operating System (Software -> Operating Systems -> New).

- Enter **<Platform> Ghost** as Description
- Select **GHOST.TPL** as System Template

Step 6. Create an Operating System **"<Platform> Network"** with Network.TPL as System Template, where <Platform> is the platform to distribute.
This Operating system distribution type will be used to setup the reference PC once from the network.
With the Easy Manage Admin Program you have to create an Operating System (Software -> Operating Systems -New).

- Enter **<Platform> Network>** as description
- Select **NETWORK.TPL** as System Template

Step 7. Create an Initial Software Application **"<Windows xx>"**.
This Initial Software Application is needed to transfer the License Key and the Domainname to the workstation.

With the Easy Manage Admin Program you have to create an Initial Software Application with class Operating System (Software -> Initial Software -> New).

- Enter **<Windows xx>"** as description
- Select the operating system as class
- Tab Network, enter the Domainname if your workstation should added to a domain.

Make an image for each specific distribution

Follow **step 8** until **step 11** to create an image for each specific distribution.
Step 8. Create an Initial Software Application e.g. "**<Platform> Ghost Image"**, where <Platform> is the platform to distribute**.**
This application is needed to contain the Ghost Image file to distribute.
With the Easy Manage Admin Program (Software -> Initial Software -New).

- Enter **<Platform> Ghost Image** as description
- Select **Ghost** as class
- Enter the model directory name as **Model**.
- Tab Platforms, select the platform

Step 9. Create a Reference PC for a specific distribution
Create the workstation with the OEM setup and don't change the filesystem to NTFS or setup the Workstation with the Easy Manage Admin Program.
With the Easy Manage Admin Program (Configurations -> Workstations-> New).

- Enter **REFPC** as CI Number and Machine Name
- Select **"<Platform> Network"** as platform, created in **step 6**
- Tab Initial Software -> Add the Applications:
  "**Sysprep**", created in **step 4**
  "<Windows xx>", created in **step 7** and remove the Domain parameter **(Tab Network)**
  "**<Platform> Ghost Image"**, created in **step 8**
- Place the Boot Disk in the floppy or CD-Rom drive, reboot and install the reference PC

Step 10. Tune the Reference PC to your own wishes and reset the platform.
The next time when you install this workstation, it will be installed in the state where you have left it.
With the Easy Manage Admin Program (Configurations -> Workstations- Select **REFPC** -> Open).

- Select **"<Platform> Ghost "** as platform, created in **step 5**

Step 11. Use the SYSPREP method (only for Windows NT, Windows 2000 and Windows XP)

- Place the Boot Disk in the floppy or CD-Rom drive. The Reference PC will shutdown and power off (if supported)
- Run the procedure from the Start -> Run menu depending on the actions after imaging the PC:
  A. Prepare the PC for network distribution
  Run the command C:\SYSPREP**TOIMAGE.BAT**

  B. Prepare the PC for reselling (only for Windows XP)
  Run the command C:\SYSPREP**2FACTORY.BAT**

Step 12. Make an image of the Reference PC.
Start the Reference PC with the BOOT DISK and use the **Maintain this workstation** menu option in the startup procedure to make a ghost image.
Use the Model name as defined in **step 8**.

This will (re)create the image file GHOST.IMG in the model direcory in the knowledge database (<ProgramDir>\<PLATFORM>\Ghost\<Model>).

Update the reference PC

- Place the Boot Disk in the floppy or CD-Rom drive, boot and install the reference PC
- Update the reference PC to your wishes and follow **step 11** and **step 12** when you are finished.

Reseal or distribute the Ghost image

Step 13. Configure the Ghost configuration.
With the Easy Manage Admin Program (Configurations -> Workstations -> Select Workstation -> Open)

- Select **"<Platform> Ghost"** as platform, created in **step 5**
- Tab Initial Software -> Add the Applications:
  "<Windows xx>", created in **step 7**

"**<Platform> Ghost Image"**, created in **step 8**
"Reseal", if this workstation should be sealed for reselling. (Only for Windows XP)

- Add the other Initial Applications, devices and adapters.

Step 14. Copy the **Ghost** workstation for batch distribution.
With the Easy Manage Admin Program (Configurations -> Workstations -> Copy).

If you have decided in **step 12** not to convert to NTFS, then you can do this by hand. You have to create the file Ghost.INF in the <ProgramDir>\<PLATFORM>\Ghost\<**Model**> directory with the following lines:

```
[OEMRunOnce]
"Convert FAT32 to NTFS","NTFS.Standard", APP
[NTFS.Standard]
InstallType = Standard
SetupFile   = "cmd"
SourcePath  = "%%Windir%%\System32"
CmdLine     = "/c convert c: /FS:NTFS /X< ~NOTDef({EZMDir$},C:\EZM,{EZMDir$})~\EZMNTFS"
 [GuiRunOnce]
"cmd /c convert c: /FS:NTFS /X< ~NOTDef({EZMDir$},C:\EZM,{EZMDir$})~\EZMNTFS"
```

You have also to create the file Yes.txt in the <ProgramDir>\<PLATFORM>\Ghost\<**Model**> directory with 2 lines. The first line should contain the confirmation character of your operating system. The second line should be empty.
E.g. for an English version:

```
Y
```

Using Microsoft's Sysprep for Windows

What is Sysprep

Sysprep is the preinstallation tool from Microsoft, which is used by computer manufacturers to preinstall large numbers of pc's. It consists of a base image, which has all the drivers and software needed to install a range of nearly identical pc. After the image is written to the new machine, a mini-setup will install all the drivers and software according to the required specifications. It is basically a faster method then the normal OEM installation when used on nearly identical pc.

This method can be used in Easy Manage, although it is not the preferred method.
Easy Manage supports this method for Windows NT (Simulates Sysprep), Windows 2000 and Windows XP.

Advantages

- The user environment can be easily tuned by hand. What You See Is What You Get.

Disadvantages

- Because of the diversity of mass storage controllers used today, it is necessary to do a complete re-enumeration of the plug-and-play devices. This makes it not much faster then the preferred OEM installation method.
- Fixed IP settings is not possible, only DHCP.
- Only PnP devices and adapters will be supported.
- It takes a lot of time in making the Sysprep image and in testing it, before it can be rolled out.
- Driver installations, which are using a silent setup to install, have to be installed before the image is created.

Prequisites

- Sysprep, which is located in the <ProgramDisk>\<PLATFORM>\SYSPREP directory.
- Norton Ghost Non Personal Edition (not supplied).

How to make a Sysprep image

Follow **step 1** until **step 12** defined in the **Using Ghost with Easy Manage** chapter and don't forget to execute **step 11.**

How to distribute a Sysprep image

Follow **step 13** defined in the **Using Ghost with Easy Manage** chapter and don't forget to add the initial Software **"Reseal"**, if you want to seal the PC for reselling.

How to seal the PC for reselling

Install the PC with Easy Manage and after succesfull test, run the command C:\RESEAL**RESEAL.BAT** from the Start -> Run menu**.**

This procedure will prepare the system for reseal and shutdown after it is finished.

Easy Manage and Novell Netware

Easy Manage could be used in a Novell Netware environment. Novell Netware Client32 should be installed on each workstation and if you want to use roaming profiles, you have to install ZENWorks for Desktops 3.2 on your Novell server.

Novell Netware and a Windows environment for the Easy Manage File Server

- Create a user Installer in the Novell Netware and Windows environment with the same password.
- Create the user Admin in the Windows environment with the same password as on the Novell server.
- Create each user, which should have Installer rights, in the Windows environment.

Configure the Dynamical User environment if you want to use local profiles and Windows security.
Create a Dynamic Local User (DLU) package with ConsoleOne.

- Start the ConsoleOne and Select the Contents in the Tree where you want to create a DLU Package
- Select the Object "Policy Package" and click OK
- Select the Policy Package "User Package" with Policy "Dynamic Local User" and click Next

- Create a package for admins and for users.
- Add the Admin as member in the Admin package.
- Configure the DLU for the admin package with the properties that the members will become Administrators

Do **NOT** add the Installer into a DLU package.

Follow the following steps to use Novell Netware Client32:
Step 1. Place the Novell Netware Client32 in the Easy Manage Knowledge database.
Download the Client32 software and unextract it.

- For Windows 9x based platforms, copy the extracted directory **ibm_enu** to the <ProgramDir>\<PLATFORM>\Netware**Client32\ibm_enu** directory.
- For Windows NT based platforms, copy the extracted **I386** to the <ProgramDir>\<PLATFORM>\Netware**Client32\I386** directory.

Step 2. Create the Class **Netware** (if not yet exists).
With the Easy Manage Admin Program add the class **Netware** (Setup ->Classes -> New)

- Enter **Netware** as Name
- Select **IPX** as classtype.

Step 3. Create the initial application"**Netware Client32**".
With the Easy Manage Admin Program add the initial Software "Netware Client32" (Software -> Initial Application -> New)

- Enter **Netware Client32** as description
- Select class **Netware**
- Tab Script, Enter **CLIENT32** as Model
- Add the platforms for the platforms to support.
- Pack the distribution with the Easy Manage Driver Loader Wizard. (Tab Drivers -> Select the Platform -> Load)
- Select **Pack distribution** and click Next.

Step 4. Tune your Novell Netware Client32.
Start the NCIMAN.EXE program in the <ProgramDir>\<PLATFORM>\Netware\Client32\ibm_enu\Admin or <ProgramDir>\<PLATFORM>\Netware\Client32\I386\Admin directory.
Open the file in the <ProgramDir>\<PLATFORM>\Netware\Client32\Netware.ans and tune your changes. Use the variables {Netware_Tree$} and {Netware_Context$}. If you want to use IP variables, use the variables of the MSTCP Initial Software. E.g. {MSTCP_IPAddress$}.

Step 5. Assign the Initial application to the workstation
With the Easy Manage Admin Program assign the initial Software "**Netware Client32**" to a workstation
If you don't want to use a Microsoft Domain environment, be sure you have entered an empty domain in the assigned platform Initial Application.

Use your own agent

If you want to install another agent (which will start the EZMAGENT.EXE), you have to create an Initial Application by following the next steps:

Step 1. Copy your own agent to the MasterDir <ProgramDir>\EZMBASE\EZM

Step 2. Create an Initial Application
With the Easy Manage Admin program (Software -> Initial Software -> New)

- Enter **Own Agent** as Name
- Select **Initial Application** as class
- Tab Scripts, enter **OwnAgent** as script

Step 3. Create the Model Directory for each platform
<ProgramDir>\<Platform>\InitAppl\OwnAgent

Step 4. Create the batchfile EZMDISTR.BAT in this Model and copy the Next text in this file and change the line with the OWNAGENT.EXE with the name of your own agent.

```
----------Copy From here
@ECHO OFF
Rem File:EZMDistr.BAT  24-08-2004
Rem Usage: ezmdistr.bat SourcePath
Rem
Rem This procedure will insert an Own Agent in the registry
Rem
rem Pre  : %ProgramDir%  Program Directory
Rem       %EZMDIr%       Directory where the Easy Manage Client distribution is stored
Rem
Set Agent=OWNAGENT.EXE
%Ramd%\Tools\EZMCopy %1\%Agent% %EZMDIR%
Rem Install the Agent in the Registry
echo."Easy Manage"="%EZMDIR%\%AGENT%">C:\TEMP\ezm.txt
Echo.REGEDIT4>%EZMDir%\ezm.reg
Echo.>>%EZMDir%\ezm.reg
Echo.[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run]>>%EZMDir%\ezm.reg
%ProgramDir%\TOOLS\RSubStr C:\TEMP\ezm.txt %EZMDir%\ezm.reg \=\\
----------Until Here
```

Change the Easy Manage Logo

If you are not satisfied with the Easy Manage Logo (Man), you can change the logo for the agent as well for the Admin Program and the Self Service Helpdesk.

Follow the following steps to change the logo for the Agent:
1. Create your own logo EZMLogo.bmp with a width of 187 and a height of 191 pixels and a resolution of 81 x 81 dots/inch.
2. Place your EZMLogo.bpm in the directory <ProgramDisk>\EZMBase\EZM
The Agent will copy the EZMLogo.bmp file to the client and display it, the next time when it starts.

Follow the following steps to change the logo for the Admin Program and the Self Service Helpdesk:
1. Create your own logo EZMLogo.jpg with a width of 367 and a height of 219 pixels and a resolution of 96 x 96 dots/inch.
2. Place your EZMLogo.jpg in the directory <Tomcat>\webapps\ezmanage\images.
Your logo will be displayed the next time that you start the Admin Program or Self Service Helpdesk.