Easy Manage Reference Guide version 5.4

Restricted Rights Legend

COPYRIGHT

LICENSE

This software is generally on a PER USER base. Please consult the accompanying license agreement for details regarding the licensed use of this product.

YOU MAY NOT USE, COPY, MODIFY, RENT, DISASSEMBLE, REVERSE ENGINEER, CREATE DERIVATIVE WORKS, IN WHOLE OR IN PART, EXCEPT AS EXPRESSLY PROVIDED FOR IN THE LICENSE.

DISCLAIMER

EZManage B.V. makes no representation or warranties of any kind, either expressed or implied for this product, the manual, disks or program, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose or as the quality, utility or performance of the program, all of which are hereby expressly excluded.

Further EZManage B.V. reserves the right to change the software and this publication without any obligation by EZManage B.V. to notify any part of such revisions or changes.

TRADEMARKS

Easy Manage ® is the trademark of EZManage B.V. Other products mentioned in this manual are the trademarks of their respective manufacturers.

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

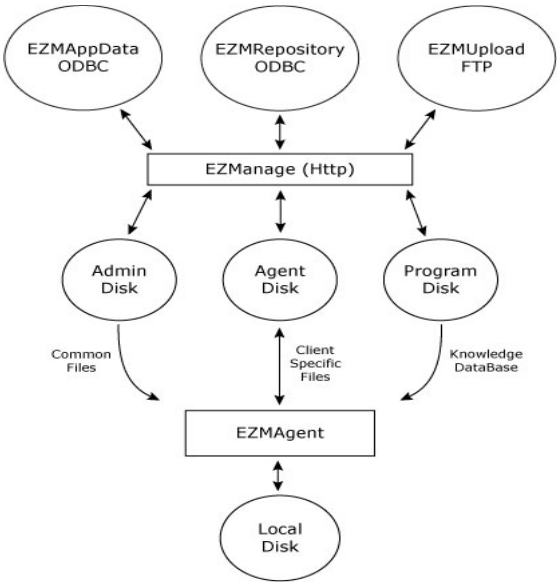
1. Easy Manage Directories

The directories that will be used by Easy Manage are:

Easy Manage Disks	Directory	Name
Web Server		
Apache Tomcat MainDir	\webapps\ezmanage	EZManageDir
ODBC	EZMData	EZMData
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:)	\	AgentDir
StartMenuDisk (S:)	\	RStartmenuDir
FTP Server		
EZMAdmin	\ADMIN	AdminDir
EZMAdmin	\ADMIN\SITES	SitesDir
EZMProgs	\	ProgramDir
EZMProgs	\EZMBASE\EZM	MasterDir
EZMProgs	\EZMAPPS	ApplDir
EZMAgent	\	AgentDir
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:\TEMP	TempDir
Workstation (C:)	C:\Easy Manage	EZMSetupDir

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

Schematic view of Easy Manage:



1.1 EZManageDir

The EZManageDir contains the complete Easy Manage Web Application. The Easy Manage application is configured as “ezmanage” in the root of the Tomcat webapps 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

\docs
This is the documentation directory

admin.htm	Administrator Program Guide
help.htm	Helpdesk Program Guide
conf.htm	Configuration Guide
ref.htm	Reference Guide(This file)

\downloads
This is the directory with files to download. It contains the ezmcopy.exe 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 2003, Windows XP
\images	This is the directory with all the images.
\log	This is the directory, which will be used to create the loggings.
\language	This is the directory, which will be used for language translations.
\repository	This is the directory, which will be used to maintain the repository and the CMDB updates.
\sshd	This is the directory with the Self Service Helpdesk application.
\xls	This is the directory with the templates of the Print function.
\WEB-INF	This is the directory with the compiled java servlets and beans

.

1.2 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 EZMData database. Only needed in the developer mode.

1.3 EZMData

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

1.4 AdminDir

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

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	Updated 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 and WOL commands
Users.UAF	Easy Manage User and Application File
UAGENT.AIF	Easy Manage Application Information File
APPLICATIONS.DAT	Applications file used by the Easy Manage StartMenu program
EZMSITES.DAT	Site Applications and workstations used by the slave synchronizer
ATTENTION.DAT	Operator attention commands on the secondary master
Machines.UAF	Easy Manage Machine and Application File
HDI.Dat	Hardware Device ID's database
HDI.UHD	Hardware Device ID's Update database

The following files could be managed per site and will also be in the Site directory (\Sites\<SiteName>)

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

1.5 LoadDir

This is the directory where the applications are loacted which are started up by the Easy Manage Admin Program from the client view.

DatabaseDir=<AdminDisk>\EXE

EZDRVINS.EXE	Easy Manage Driver Installer Wizard
EZMAccessor.exe	Easy Manage Assessor
EZMImport.exe	Easy Manage Active Directory/Network Inventory Scanner and Import program
PCIDEVS.TXT	PCI devices definition file
PNPID.TXT	PNP devices definition file

1.6 ProgramDir

The ProgramDir or Easy Manage **Knowledge Database**, is a directory tree containing all the operating system distributions, the System scripts and the drivers for adapters and devices for each platform and Application Scripts that have been created by the Easy Manage Application Script 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:

\WIN2003	Windows 2003 Setup Information Files Scripts and drivers.
\WIN2008	Windows 2008 Setup Information Files Scripts and drivers.
\WIN2008x64	Windows 2008 64 bit Setup Information Files Scripts and drivers.
\WINXP	Windows XP Setup Information Files Scripts and drivers.
\WINVISTA	Windows Vista Setup Information Files Scripts and drivers.
\WINVISTAx64	Windows Vista 64 Bit Setup Information Files Scripts and drivers.
\WINDOWS7	Windows 7 Setup Information Files Scripts and drivers.
\WINDOWS7x64	Windows 7 64 Bit Setup Information Files Scripts 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 EZM.INF.

An application or adapter software for Windows 7 could look like:

```
PLATFORM =WINDOWS7
CLASS=OS
MODEL=UKPRO
        <ProgramDir>\WINDOWS7\OS\UKPRO

PLATFORM =WINDOWS7
CLASS=MSTCPIP
        <ProgramDir>\WINDOWS7\MSTCP\MSTCP\EZM.INF

PLATFORM =WINDOWS7
CLASS=ETHERCRD
MODEL=3C509
        <ProgramDir>\WINDOWS7\ETHERCRD\3C509\EZM.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.XML/ Server.INF/SERVER.XML	Default Windows setup answer file
DEFAULT.TPL	Default System template (Local install)
NETWORK.TPL	Default System template (Network install)
GHOST.TPL	Default System template (Image 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
VNCViewer.exe	VNC Client viewer to take over the screen of a client

ImageDir

\EZMBASE\IMAGE

This directory contains the Partition Dump/Restore program and the tag files

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

SetupDir

\EZMBASE\Setup
This directory contains the setup program SetupEZM.exe to install the easy manage agent on a workstation that should be managed by Easy Manage and not was deployed by Easy Manage.

MasterDir

\EZMBASE\EZM
This directory contains the Easy Manage Distribution for each workstation and is Operating System independent. All files in this directory will initially be distributed to the client or when the time stamp is different on the client. The Easy Manage agent will always check at startup for updates that has 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).	
\Floppy	Programs that will be copied to the DOS Boot Disk.	
\Bin	Extended programs needed on the DOS Boot Disk.	
\Root	Startup files needed on the root of the DOS Boot Disk.	
\Net	Microsoft MSNET network protocol software for the DOS Boot Disk.	
\Tools	DOS Boot Disk tools.	
\CDRomSup	CDRom drivers for the DOS Boot Disk.	
\Drivers	MSNET NDIS driver Database for the DOS Boot Disk.	
\CD	CD-writer software for the DOS Boot Disk.	
\UNIDRV	Undi Drivers for PXE for the DOS Boot Disk.	
\app	Winbuilder plugins	
\ezm		The Easy Manage Environment plugin script
\EZMPlugin		Example of the Easy Manage plugin
\pebuilder	The PEBuilder Environment from Bart Lagerweij	
\plugins		The plugin directory
	\ezm	The Easy Manage plugin script
\EZMInstaller	The Easy Manage Installer programs used in de PEBuilder and WinPE environment	
\Winpe	The Windows 7 scripts for WinBuilder	
\Windrivers	The drivers for the PEBuilder and WinBuilder environment	
\drivers		The driver directory for the network and harddisk controllers

ApplDir

\EZMAPPS
This directory contains the Application Scripts, generated by the Easy Manage Application Script Wizard and is “Operating System independent”.
Each Application script has its own directory tree identified by its unique Application ID. This complete directory tree is an Application Script. The directory “LocalSite” will not be synchronyzed in a Master Slave configuration. So you can place files like templates of a site in this directory, which you can use in your Easy Manage Information Files.

1.7 AgentDir

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>.
MachineDir=<AgentDir>\Machines\<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=<AgentDir>\Logging

<CI-Number>.LOG	Log File

SysInfoDir

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

<CI-Number>.ESI	Easy Manage System Information File
EZMADScan.ADS	Easy Manage Active Directory scan (Master)
<SiteId>.ADS	Easy Manage Active Directory scan (Per Site)

NetUserDir

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

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

1.8 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

1.9 LocalDir (C:\EZM)

The LocalDir is the root directory on the workstation, where the Easy Manage programs files reside and contains data that is generated by the Easy Manage. This directory will be created at Setup time and will automatically be updated by the Agent.

1.10 StartMenuDir

The StartmenuDir is used by the StartMenu program to manage the desktop of the workstations. A subdirectory will be created for each Group with the link files of the scripts of that group.

\StartMenu	Folder with the common start menu items
\Wallpapers	Folder with all available wallpapers
\GroupId	Folder for the specific group
GroupId\Desktop	Folder with the icons for the desktop of the specific group
GroupId\StartMenu	Folder with the icons for the startmenu of the specific group
GroupId\Wallpaper	Folder with the selected waallpaper of the specific group

2. Easy Manage Main Control Files

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

2.1 Parameter file PARAMS.DAT

The Parameter File contains the parameters for the directories that will be used by al 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>
[Setup]
AgentDir=                      (Default: none)
ProgramDir=                    (Default: none)
AdminDir=                      (Default: none)
RStartMenuDir=                 (Default: none)
```

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

2.2 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.

```
[Databases]
EasyManage=DB_EasyManage

[DB_EasyManage]
Name=EZMData
ODBCName=EZMData
; Type = Access | MySql | SQLServer
Type=Access
; Connection = ODBC | Native
Connection=ODBC

[FTP_Common]
Port=21
HostName=PLUTO.easymanage.local
Directory=ezmprogs
User=installer
Password=ECAD8626466E

[FTP_Attachments]
Port=21
HostName=PLUTO.easymanage.local
Directory=ezmupload
User=installer
Password=ECAD8626466E

[Serverpaths]
AdminDir=D:\EZMDrives\EZMADMIN
ProgramDir=D:\EZMDrives\EZMPROGS
AgentDir=D:\EZMDrives\EZMAGENT
RStartMenuDir=D:\EZMDrives\EZMSTARTMENU
SMTPHost=localhost
POP3Host=
POP3User=
POP3Password=
AttachmentsDir=D:\EZM\EZMDB\upload
DBPath=D:\EZM\EZMDB
; AttachmentsDir=ftp:FTP_Attachments

[Clientpaths]
AdminDir=\\PLUTO\EZMADMIN$
ProgramDir=\\PLUTO\EZMPROGS$
AgentDir=\\PLUTO\EZMAGENT$
```

RStartMenuDir=\\PLUTO\EZMSTARTMENU$

2.3 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.

2.4 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. This file will be maintainded by Easy Manage.

2.5 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.

2.6 Administration files Users.UAF / Machines.UAF

The Administration Files **Users.UAF** and **Machines.UAF** will be generated and stored in the AdminDir by the Easy Manage Admin Program in case one of the keys are changed by a database change action. It describes the registered Easy Manage users with their allowed applications, the configured workstations and the applications to be installed on them. These files will also be copied to the Site directory with the content only valid for that site. If this file is changed, then the Easy Manage Install Agent will copy this file to the LocalDir of the workstation.

Users.UAF

[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 scripts.
	The Installer is allowed to install and use Application scripts.
	The User is only allowed to use Application scripts.

[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.

Machines.UAF

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

<CINumber>,<NonRegUsers>,Reserved,<AskForUpdate>,Reserved,<AutoLogon>,<Secure>,<NrOfCancel>,
<UnblockInput>,<StatusId>,<GroupId>

Where:	
< CINumber>	Unique CI Number of the machine
<NonRegUsers>	Allow Nonregistered Users 0 = No, 1= Yes.

<AskForUpdate>	Show the screen to ask for update 0=No, 1=Yes.
<AutoLogon>	Allow AutoLogon 0=No, 1=Yes.
<Secure>	Secure Workstation 0=No, 1=Yes.
<NrOfCancel>	Number of times that the cancel button could be pressed
<UnblockInput>	Disable the blocking of the input on the workstation 0=No, 1=Yes.
<StatusId>	StatusId of the Configuration 3=Production
<GroupId>	Unique GroupId of the Desktop group

[MachineAppls]
This section describes all Configured Workstations and their applications to install. The application indexes are sorted on the installation sequence. 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

2.7 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. The site specific section will also be created as EZMSites.dat in the Site directory. It will be used as interface for the synchronization process.

<SiteCode>	Code of a defined site
<Count>	Sequence number
<CI-Number>	Unique CI Number
<CI-NetworkName>	Unique CI NetworkName used for Configuration Templates
<ScriptId>	Application Script Id
<TimeStamp>	TimeStamp of the last update

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

The applications, users, workstations and Configuration Templates used by a site are defined within a section like:
WKS<Count>=<CI-Number>,<TimeStamp>
CIT<Count>=<CI-Number>,<TimeStamp>,<NetworkName>
USR<Count>=<UserName>,<TimeStamp>
AWS<Count>=<ScriptId>,<TimeStamp>

2.8 Operator Attention file Attention.dat

The Easy Manage Operator Attention File **Attention.DAT** will be generated and stored in the AdminDir by the Easy Manage Master-Master program in case an application needed Operator attention to load the distribution offline in the knowledge database.

<ApplicationId>	Unique Application Id of the Script correspondig with the ID in the Configuration Management DataBase
<Type>	Type of the Script
<Status>	Status of the command
<RelativePath>	Path relative to the programdir
<Version>	Version of the Operator Attention Command

There is a line for each Operator attention
<ApplicationId>,<Type>,<Status>,<RelativePath>,<Version>

2.9 EZMStartMenu file Applications.DAT

The Easy Manage StartMenu File **Applications.DAT** will be generated and stored in the AdminDir by the Easy Manage Admin Program in case an application script is added to or removed from an application or when an application is added or removed from a desktop group. It describes the scripts per group. It will be used as interface for the Easy manage StartMenu program.

<UserName>	Unique Username of the supervisor
<GroupId>	Unique Id of the Desktop group
<DisableAllUsers>	Do not use the All Users desktop settings No=0, Yes=1
<UseGroupStartMenu>	Use the Group Start Menu instead of the common Start Menu No=0, Yes=1
<UseGPO>	Use the Group Policies of Windows instead of the Easy Manage Desktop Agent to manage the Desktop and the Start Menu No=0, Yes=1
<Directory>	Base Directory of the Desktop group
<GroupName>	Unique group name
<ScriptId>	Application Script Id

The section applications describes the scripts per application.
[Supervisors]
<UserName>[,<GroupId>]...

This section describes the scripts per group
[Groups]
<GroupId>,<site>,<DisableAllUsers>,<UseGroupStartMeny>,<UseGPO>,<Directory>,<GroupName>[,<ScriptId>]...

2.10 Application Installation file UAGENT.AIF

This 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 updates will be processed by the background program, when the file updaif.tch was detected in the AdminDir.
- The Easy Manage Application Script Wizard also updates the file, when an Application Script 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>,Reserved,<Common>,<Description>,<EIFFile>,<DiskUsage>,<InitialApplication>,
Reserved,Reserved,<PostInstall>,<Version>,<Obsolete>,<Class>,<ScriptType>,<Status>

<State>	C[hanged] | U[nchanged] The following actions will be taken by the Easy Manage Admin Program with the state: *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>	Application is common? 0=No, 1=Yes
<Description>	Description of the application
<EIF File>	Path of the Easy Manage Information File or only .EIF
<DiskUsage>	Disk usage of the application in Mbytes
<InitialApplication>	Application used in the initial workspace? 0=No, 1=Yes
<PostInstall>	0 = No Action 1 = Logoff 2 = Restart Windows 3 = Reboot the PC
<Version>	Version Number
<Obsolete>	Application is Obsolete? 0=No, 1=Yes
<Class>	Class name of the application
<ScriptType>	0 = SnapShot 1 = Silent (Default) 2 = AutoIt
<Status>	Status of the application script (4 = Production)
<SiteDependend>	Is this Sicript site dependend 0=No, 1=Yes
<ProgramName>	Name of the program to monitor

2.11 Mac Address mapping file MacInvNr.Dat

This file will be updated by the Easy Manage Admin program, when a configuration was changed and saved. 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 content of the file is:
<MacAddress>,<CI-Number>,<SiteCode>, <WOLDate>,<WOLTime>,<PXEType>

<MacAddress>	8 pair hexadecimals separated by a colon ":".
<WOLDate>	Wake On Lan date with format yyyymmdd or the days separated with a comma.
<WOLTime>	Wake On Lan time with format hhmmss.
<PXEType>	ID Of the PXE Type to boot with. (1=Dos, 2=WinPE,3=EzmPE)

2.12 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. This file will be synchronized from the slave to the site directory of the master.

```
[Domain]
DomainName=ezmanage.local

[PXEServer]
PXEserver=PLUTO

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

[HTTPServer]
HTTPHost=Pluto
HTTPPort=8080

[Site]
Site=MainSite
MainSite=1
```

2.13 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.

2.14 Checklist file <CINumber>.ECL

The Checklist file <CINumber>.ECL will be generated by the Easy Manage Desktop Agent, when an Application script is installed correctly. It will copy and expand the rows in the section [CheckList] in the Easy Manage Information file (.EIF) of the script.

The file consists of records, where each record is one line with the following format:
<ScriptId>,<Parameter>,<Value>,<HowToSee>

<ScriptId>	Unique Id of the installed script
<Parameter>	The parameter to look for
<Value>	The expanded value
<HowToSee>	Text with the instruction of how to find the parameter manually

2.15 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 name of the System Template (.**TPL** File).
<AnswerTemplate>	This field defines the Answer Template File and is only valid in the COMMON Class. This field will be the default Answer Template (.INF/.XML) 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.
<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.

2.16 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
```

2.17 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;
InventoryNumber$ := "Demo22";
MachineName$ := "Demo22";
Unit$ := "1";
CompanyName$ := "Easy Manage";
Unit$ := "0. ezmanage.local";
Machine_Type$ := "Midi Tower";
Machine_Model$ := "MS-7211";
FullName$ := "Unknown";
UserName$ := "Unknown";
MSTCP_IPMask$ := "0.0.0.0";
MSTCP_EnableDHCP? := "1";
WinXP_ValidateLogon? := "1";
WinXP_LogonDomain$ := "ezmanage";
WinXP_DefaultProtocol$ := "0";
End;
```

2.18 Operator Attention Command file Attention.INI

The operator Attention Command file will be created by the Easy Manage Application Installer Wizard in the directory of the Application Script.

Example of the Operator Attention Command file:

```
[Attention]
Prompt=Place the CD with word
Version=1.0
SourceTag=autorun.inf
DoneTag=autorun.inf
SourceDir=
DestinationDir=
```

3. Substitutions and functions

Easy Manage uses substitutions, when it executes the templates. A substitution is a replacement of the value of a script 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 (Row) 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
{12}	PARAMS.DAT	.EIF Files	EZMStartMenuDir
{13}	PARAMS.DAT	.EIF Files	<AdminDisk>\exe

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

{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)

3.1 Functions

In the .TPL, .ROF, .INF and .XML 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 tildes **'~'** and will be executed after the variables are substituted. 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:

Env(<EnvironmentVariable>) Returns <OutputStr>
Reads the Environment variable.
Example:
AdminDir = ~Env(AdminDir)~

This will output:
AdminDir = n:\admin

Enum(<EnumerateVariable>[,<Start>]) Returns<Number>
Defines or enumerates a defined value. Somtimes you need a number which could be incremented. When this function is used for the first time, it will define the Enumeration Value and increments its start value with 1. The next time that this funtion is used, it will increment the value with 1. The variable start is by default 0.
Example:
Session1 = ~Enum("Session","-1")~
Session2 = ~Enum("Session")~

This will output:
Session1 = 0
Session2 = 1

GetEnum(<EnvironmentVariable>) Returns <Number>
Reads the defined Enumeration variable, created with the function Enum.
Example:
Session1 = ~Enum("Session","-1")~
Session2 = ~GetEnum("Session")~

This will output:
Session1 = 0
Session2 = 0

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**>.

Decrypt(<Value>)Returns<OutputStr>
This function returns the decrypted value of an encrypted value used for passwords.

Enum(<Label>[,<StartValue>])Returns<OutputStr>
This function returns the enumerated value of the label. When this function was called the first time, then the value of the label becomes 1. You can also initialize the label with the integer startvalue. For each next time that the funtion is used the value of label is incremented by 1. This function is only available for XML files.

4. 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 (Temporary)

- 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
```

5. Easy Manage Templates

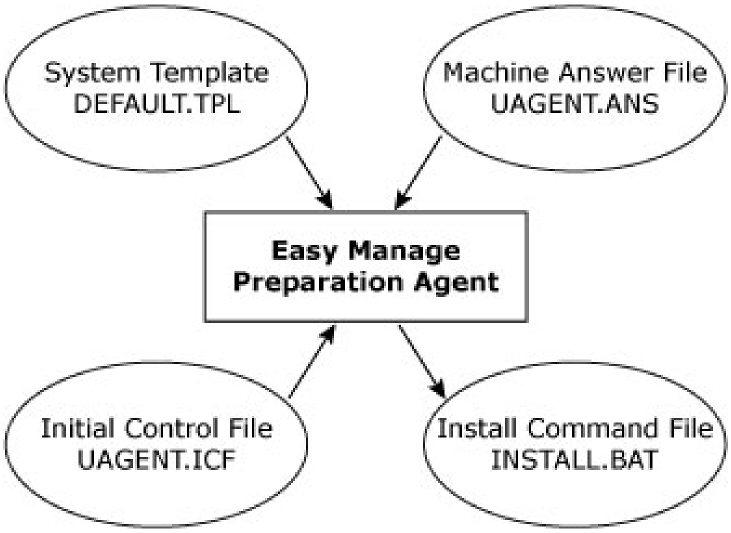
Easy Manage uses templates to generate the Install Procedure.
If the value of a substitution not could be evaluated, then the line will not be copied.

Easy Manage uses the following templates:

- System Template (.TPL)
- Class Answer File (.INF, .XML 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/.XML)

5.1 System Template <Method>.TPL

The System Template will be used to define the sequence of the installation of Operating System, all-possible System Scripts 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 Procedure (INSTALL.BAT) for each configuration, which will be created on the ramdrive of the workstation during installation. 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 Proceduree (INSTALL.BAT).
- A Class section will only be copied to the Install Command File, if its section name is selected as Class in the configuration (It was defined in the Install Procedure).

Easy Manage distributes the following system templates:

Platform	Local (Default.TPL)	Network (Network.TPL)	Image (Ghost.TPL)
All Windows	Recommended for notebooks	Recommended for desktops	Possible

The advantages and disadvantages of the different distribution methods:
Local/Network
Advantages:

- Notebook setup is more smoothly
- Managing updates is easy
- Less network load
- Less synchronisation load

- Recognition of hardware layers is better

Disadvantages:

- Slow

Image

Advantages:

- Fast

Disadvantages:

- Notebook setup is not so smoothly
- Managing updates is difficult (A complete new image has to be build)
- More network load
- More synchronisation load
- Recognition of hardware layers is not always garanteed

5.2 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 Windows setup procedure, when the BOOT DISK is removed and the system was rebooted the first time.

Easy Manage will maintain and deliver this file for each version.

5.3 RunOnce template EZM.ROF

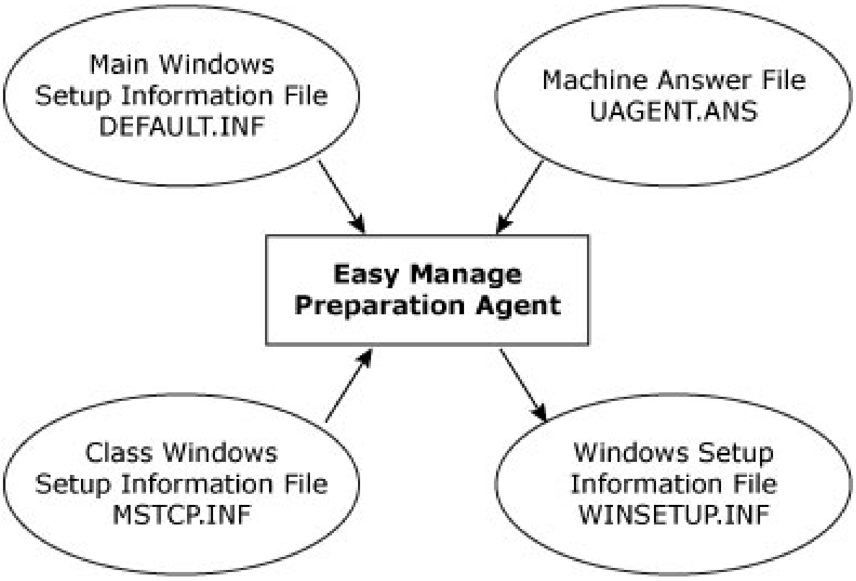
This file will be used to create initial registry settings in the Initial Workspace during setup. The Main RunOnce template EZM.ROF, is located in the <Platform>\EZM directory. This file will be copied by the System Template to the workstation during the installation as EZM.INF and will be used to execute 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.

5.4 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/Default.XML/Server.INF/Server.XML
- Class Windows Setup Information File <Class>.INF/EZM.XML



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 WINSETUP.INF on the ramdisk of the workstation during installation. 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>\EZM.INF,
<ProgramDir>\<PLATFORM>\<Class>\<Model>\<Class>.INF,
<ProgramDir>\<PLATFORM>\<Class>\<Model>\<Model>.INF,
<ProgramDir>\<PLATFORM>\<Class>\<Class>.INF.

5.5 Main Windows Setup Information File WinSetup.INF/Unattended.XML

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 (Default.INF):

```
;
; Microsoft Windows XP Professional
;

[Unattended]
Unattendmode = FullUnattended
OemPreinstall = yes
TargetPath = ~NOTDEF({WinXP_TargetPath$},WINNT,{WinXP_TargetPath$})~
FileSystem = ConvertNTFS
OemSkipEula = yes
OemFilesPath=~NOTDEF({SetupDir$},"C:\I386",{SetupDir$})~\$OEM$

...

[UserData]
```

```
FullName = "~NOTDEF({UserName$},{installer},{UserName$})~"
OrgName = "{CompanyName$}"
ComputerName = "{MachineName$}"
ProductId = "~NOTDEF({WINXP_SerialNumber$},J4YH9-DDCX3-HPYMM-XYHK6-GGHDB,{WINXP_SerialNumber$})~"

...

[Identification]
;CreateComputerAccountInDomain=Yes
DomainAdmin = {installer}
DomainAdminPassword = {InstallerPW}
JoinWorkgroup = "~NOTDEF({WINXP_LogonDomain$},{Workgroup$},{DeleteRow})~"
JoinDomain = "~NOTDEF({WINXP_LogonDomain$},{DeleteRow},{WINXP_LogonDomain$})~"

...
```

5.5 Class Windows Setup Information file <Class>.INF/.XML

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\WINXP\MSTCP\EZM.INF

```
; Microsoft Windows XP Professional Version 4.0
;
[Networking]
; When set to YES, setup will install default networking components. The components to be set are
; TCP/IP, File and Print Sharing, and the Client for Microsoft Networks.
InstallDefaultComponents = NO

[NetProtocols]
MS_TCPIP = TCPIP_Params

[TCPIP_Params]
AdapterSections=params.adapter1
DNSDomain = {MSTCP_DomainName$}
DNSSuffixSearchOrder = {MSTCP_DomainOrder$}

[params.adapter1]
DefaultGateway = {MSTCP_Gateways$}
ScopeID = {MSTCP_ScopeID$}
DHCP = ~YesNo({MSTCP_EnableDHCP?})~
IPAddress = ~IfTrue({MSTCP_EnableDhcp?},{DeleteRow},{MSTCP_IPAddress$})~
SubnetMask = {MSTCP_IPMask$}
Wins = ~YesNo({MSTCP_EnableWins$})~
WINSServerList= {MSTCP_PrimaryWins$},{MSTCP_SecondaryWins$}
DNSServerSearchOrder={MSTCP_DNSServer$}

...
```

6. Easy Manage Information File

To control the agents, Easy Manage works with Easy Manage Information Files in each script. The Easy Manage Agents will create the following Information files:
Application Script <Application>.EIF
Windows Workspace WINDOWS.EIF
Session Workspace SESSION.EIF
User Workspace USER.EIF

6.1 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.

6.2 History Section

This section describes the history of this script. All rows of this section will be displayed in the information window of the user.

[History]
<Information>

6.3 Notify Section

This section describes the Notification messages of this script. When rows are defined in this section, then a window with the defined rows will be displayed always at the top of the desktop during the installation of the script.

[Notify]
<Information>

6.4 Install Section

Install data will be placed in this section

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

<UserName>	Active user name.
<ApplType>	"SNAPSHOT" | "AUTOIT" | "SILENT".
<InstallerName>	Name of the installer.
<DiskUsage>	Disk space needed to install the application in Mbytes.
<Version>	Version of the Application Script. The application script will not be installed, when the version differs from the version field in the AIF File.
<Description>	Description of the application script

6.5 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 (or deleted) 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 or when the condition is "Unconditionally"

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

6.6 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>]

6.7 EIF Files Commands

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

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>	Internal use
<SavedState>	Internal use

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

6.8 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>

6.9 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>

6.10 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]
<Reserved>,<”Command”>,<Directory>]

6.11 Execute commands

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

<Command>	Programname (with parameters) to execute. Use a extra double quote to define a double quote in the parameters.
<NoWait>	Use the key “NoWait” or “Nowait” if the installer shouldn’t wait until the program is finished. (Default Wait)
<Secure>	Run the command in the secure mode. (Obsolete)
<WaitForFile>	Wait until the specified file is present. (Default empty)
<AutoLogin>	Prepare an auto login with the installer account. The autologin will be executed, when the script commands to reboot.
<WaitForWindow>	Window to wait for used for some silent setup programs (InstallShield). The installation will wait until the Window was displayed.

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

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

6.12 Uninstall commands

With the Uninstall commands, you can execute a program when the application is removed. It is recommanded to use this section for each application, because then it is possible to unistall an application unattended.

<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. (Depricated)
<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>]
```

6.13 Verify commands

With the Verify commands, you can verify the installation of a script. You can check if a file or registry key exist after the installation. If not, then the script will be marked as not installed and the next time it will try to install again. You may use more than one command to check.

<Action>	The action could be “Skip”, “Abort” or “Logoff”. With skip only this application will be skipped. With Abort or Logoff the whole install session will be aborted. With the logoff action, the session will be logged off and the user has to logon again.
<Operator>	Use the operator “And” or “Or”, if you want to use more then one condition. The default is “And”.
<Type>	“F” or “R” defines if the existence of a file or a registry key should be checked. The default is File.
<FileName>	If the Type is “F”, then this value is the full filename of the file that should be checked for existence.
<RegistryKey>	If the Type is “R”, then this is the registry key in the HKEY_LOCAL_MACHINE that should be checked for existence.
<RegistryValue>	If the Type is “R”, then this is the registry value in the <RegistryKey> that should be checked for existence.

[Verify]
With this command you can verify if the application is installed.

<Reserved>,<Action>,[<Condition>],[<Type>],<FileName>|<RegistryKey>[,<RegistryValue>]

6.14 Checklist commands

With the Checklist commands, you can write a list of parameters to check when the installation was succeeded. The commands will be appended to Checklist file. The Checklist file will be deleted at the start of an initial install of the workstation.

<Parameter>	The parameter to look for
<Value>	The expected value
<HowToSee>	Text with the instruction of how to find the paramater manually

[CheckList]
With this command you can describe the parameters and the values to check and the way how to check.

<Reserved>,<Parameter>,<Value>,<HowToSee>

6.15 Printer commands

With the printer commands, you can add a printer to a configuration. When the script is removed, then the printer will be removed.

<Action>	The action Create or Delete
<PrinterName>	The name of the printer in the format \\server\Printer Name

[Printer]
With this command you can add the printer to the configuration.

<Reserved>,<Command>,<PrinterName>

7. How to create an AutoIt script

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.

Step 1. Create a Master reference PC.

Before you script an application with AutoIt, you have to prepare an environment, which could be recovered from the start situation are a known situation.

- It is recommended to create a virtual PC with VMWare or Microsoft Virtual PC. You can create this PC with Easy Manage or with an OEM Windows distribution.
- Install on the Master reference PC at least the platform, where the application should run.

Install the following applications on this reference PC:

- The latest version of AutoIt. You can install it from the Easy Manage distribution CD or download it from http://www.autoitscript.com/autoit3/downloads.shtml.
  Create a shortcut on the desktop to start **AutoIt** easily. (C:\Program Files\AutoIT3\AutoIt3.exe).
  The program AutoIt will be used to make an exe file (compile) of the script.
  Create a shortcut on the desktop to start **WindowInfo** easily. (C:\Program Files\AutoIT3\Au3Info.exe). The program will be used to display the names of the controls.
- The latest version of SciTE. You can install it from the Easy Manage distribution CD or download it from http://www.autoitscript.com/autoit3/downloads.shtml
  Create a shortcut on the desktop to start **SciteWriter** easily. (C:\Program Files\AutoIT3\SciTE\ScriptWriter\AU3Record.exe). The program will be used to start the application to script and to record all actions in a script file.
  Create a shortcut on the desktop to start **SciteEditor** easily. (C:\Program Files\AutoIT3\SciTE\SciTE.exe). The program will be used to edit the AutoIT script on a convenient way.
- Create a shortcut on the desktop to start **Wordpad** easily. (C:\Program Files\Windows NT\Accessories\wordpad.exe). The program will be used to save all actions, so you will have a storyboard to script the application offline.
- Shutdown the Virtual PC and make a snapshot of this Master reference PC. (Call it Clean PC)

Step 2. Create the Application script

Each time when you want to script an application from scratch follow the following steps:

- Create an Application Script with Easy Manage Admin Program. Software – Application Scripts – New.
  Tab General. Fill the mandatory fields
  Tab Script – Script Type = AutoIt
  Tab Platforms. Add the supported platforms. Add at least the platform of the Master reference PC.
- Load the application distribution in the Knowledge database with Easy Manage to create a template and to load the complete distribution in the knowledge database. Software – Application Scripts – Load
  The Application Wizard will start.
  Select - Load Silent Setup Script – Next
  Block ”Install” – Command : Browse to the executable on the distribution to start the setup of the application and click Next.
  Block ”Source and Destinations” – Select Copy the distribution to the Knowledge Database. Enter the desired parameters and click the button Next. **Remember the Identification of the Application script**.

Step 3. Script the application

- Create a virgin environment from the Master reference PC (Clean PC)
  Copy the master to you work environment or start with the snapshot you just have created in **Step 1**. (So you can revert to the beginning situation)
- Prepare the scripting on the reference PC
  Use Wordpad to save a screen dump of each step, where the active window of the application and the active control by **WindowInfo** is displayed.
  a. Start Wordpad
  b. Start **WindowInfo**
- Start the setup of the application
  Start **SciteWriter**
  Record Mouse Off
  Record Windows text On
  Run Application (Browse on the ProgramDisk\ezmapps\Identification\setupprogram), which will start the setup of the application.
- Click the button **Click to record**
- Record the steps
  a. When the application is asking for an action, move with the mouse over the control that you want to touch. Create a print screen and save this in Wordpad (Ctrl Print and then Ctrl V in Wordpad). Be sure that the control and the WindowInfo both are shown.
  b. Give the Answer with the keyboard (Use the HotKeys of the Setup program)
  c. and register the given Answer in Wordpad.

d. Repeat step a until step c the application is finished.

- Stop Recording
  a. Click the button Stop Record of the **SciteWriter**
  b.Save the file as ProgramDisk\ezmapps\Identification\EZMInstall.au3
  c. Close the **SciteWriter**
  d. Save the Wordpad file as ProgramDisk\ezmapps\Identification\EZMInstall.doc
- Tune the EZMInstall.au3 file
  Update the au3 file with the functions of the Ezmlib common library

Step 4. Test the script

Step 5. Compile the script

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.

Step 6. Edit the Easy Manage Information file

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.

AutoIT Tips & Trics

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
```

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!

8. 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).

9. Adding a driver in the Knowledge Database by hand

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

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 hardware type or system script 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.

RunOnce.TXT

This file is needed 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.

EZM.INF

A EZM.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 XP platform.

```
; EZM.INF
; RT8139 (A/B/C/810x/813x/C+) Fast Ethernet Adapter
; Model:10EC8139
[Unattended]
OemPnPDriversPath="Drivers\ethercrd"
[PnpDrivers]
C:\Drivers\ethercrd=ethercrd
[Netadapters]
Adapter1=Params.Adapter1
[Params.Adapter1]
INFID=*
```

<Model>.ROF

A <Model>.ROF File will be used to merge Registry 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
[DefaultInstall]
AddReg=Add.RunOnce.Test

[Add.RunOnce.Test]
HKLM,Software\Microsoft\Windows\CurrentVersion\RunOnce\Setup,"Test -Setup.exe",,"~NOTDEF({SetupDir$},"C:\I386",
{SetupDir$})~\Test\Setup.exe -s"
```

10. Using an image and sysprep with Easy Manage

Easy Manage uses Microsoft imagex in the WinPE Boot disk environment to save and restore an image in the rescue partition and to save and restore an image in the Knowledge database after the sysprep procedure. Microsoft imagex can not be used in the DOS Boot disk environment because imagex is a 32 bit application. You can use Ghost or a simular 16 bit program in the DOS Boot disk environment.

Boot disk Environment	**Windows 2003/XP**	**Windows Vista / Windows 2008 / Windows 7**
DOS (only for systems with < 512 MB)	Ghost or an other 16 bit image program	Not possible
EZMPE (recommanded for systems with >= 512 Mb)	Imagex and Ghost32 for compatibillity reasons	Imagex

10.1 Sysprep

The Microsoft sysprep method has to be used if you want to distribute an image with Easy Manage or reseal a PC for reselling. Sysprep will remove all machine dependent entries and remove the license information. Easy Manage distributes the sysprep environment for Windows 2003 and XP. The sysprep environment is already available in the Windows Vista distribution.

10.2 Prepare the image environment (once)

To setup the Ghost environment, copy the Ghost.exe for the DOS Boot environment to the <ProgramDir>\EZMBase\Image directory. If you want to used previously created images with Ghost and you want to use it in the WinPE bootdisk environment, then you have also to copy the Ghost32.exe, Ghostcdr.dll.

Step 1. Create the class **"Image Script "** (if not yet exists).
The class defines the directory in the knowledge database and the tag in the system template.
With the Easy Manage Admin Program (Setup ->Classes -> New)

- Enter class **Image Script** as name.
- Select **System Script** as main class type.

Step 2. Create the System Script "**Sysprep**" (if not yet exists).
This application removes the funtion to join a domain, disables the "convert to NFS" (DOS Boot disk) and disables the "automatic register" for Windows XP/Vista/Windows 7. It will copy the files needed for sysprep to the workstation.
With the Easy Manage Admin Program (Software -> System Script-> New).

- Enter **Sysprep** as description
- Select **Initial Application** as class
- Tab Script, Enter **Sysprep** as Model. Easy Manage distributes the scripts in the **Sysprep** Model directory.
- Tab Platforms, select the platforms, where to use this application

Setup the distribution method for each specific platform

Step 3 and **Step 4** have to be executed for each specific platform (Windows 2003/XP/Vista), which has to be distributed.

Step 3. Create a Platform Distribution **"<Platform> Image "** with image as ditribution method.
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 e.g. **Windows XP Image** as Description
- Select **Image** as System Template

Step 4. Create a Platform Distribution **"<Platform> Network"** with network as distribution method.
This Operating system distribution type will be used to setup the reference PC from the network.
With the Easy Manage Admin Program you have to create an Operating System (Software -> Operating Systems -New).

- Enter e.g. **Windows XP Network** as description
- Select **NETWORK** as System Template

Setup for each distribution

Step 5. Create an Operating System **"<Platform> <xx>"**.
This Operating System is needed to intstall the OEM distribution in the Knowledge Database and to transfer the License Key and the Domainname to the workstation.
With the Easy Manage Admin Program you have to create an Operating System (Software -> Operating System -> New).

- Enter E.g. **Windows XP Prof UK** as description
- Select the operating system as class
- Tab Script; enter e.g. **PROFUK** as the Model of the distribution.
- Tab Network; enter the Domainname if your workstation should added to a domain.
- Tab License; enter the license key in case you have a volume license.
- Tab Drivers; Load the distribution into the Knowledge Database.

10.3 Make an image for each specific distribution

Execute **step 6** until **step 10** to create an image for each specific distribution.
Step 6. Create a System Script e.g. "**Image <Platform> <xx>"**, where <Platform> <xx> is the platform to distribute**.**
This application is needed to contain the Image file to distribute.
With the Easy Manage Admin Program (Software -> System Script -New).

- Enter E.g. **Image Windows XP Prof UK** as description
- Select **Image** as class
- Tab Script; Enter the directory name e.g. **PROFUK** as Model.
- Tab Platforms, select the platform

Step 7. Create a Reference PC for a specific distribution
With the Easy Manage Admin Program (Configurations -> Workstations-> New).

- Enter E.g. **REFPC** as CI Number and Machine Name.
- Select **"<Platform> Network"** as platform, created in **step 4.**
- Tab System Scripts -> Add the Applications:
  "**Sysprep**", created in **step 2,**
  "<Platform> <xx>", created in **step 5,**
  "**Image <Platform> <xx>"**, created in **step 6.**
- Add **ALL** (SCSI/SATA) Adapters, which are using a txtsetup.oem file to this reference PC.

Step 8. Install the reference PC and tune the PC to your own wishes.
You can use the sysprep method for PC's installed with Easy Manage or with the OEM procedure.
If you want to make an image of an OEM distribution, then execute **step 8a** until **step 8c**, else execute **step 8d**.
Step 8a. Install the PC with the OEM procedure
Step 8b. Install Easy Manage with the SetupEZM procedure explained in Configuration Guide
Step 8c. Copy the Sysprep directory to the C:\sysprep directory of the reference PC.
Step 8d. Install the PC with the Easy Manage Boot disk and install applications or change global settings on this reference PC

Step 9 Use the SYSPREP method
The Sysprep procedure will remove all machine dependent entries on the machine.
If you are using a CD Rom or floppy to boot with, then you have to place the boot disk in it, because the Reference PC will shutdown and power off (if supported).

- Run the command C:\SYSPREP\TOIMAGE.BAT with the Start -> Run menu.

Step 10. Make an image of the Reference PC.

- Start the Reference PC with the Easy Manage Boot disk
- Select the **Maintain this workstation** menu option in the startup procedure to make an image.
- Use the **Model** name as defined in **step 6**. This will (re) create the image file GHOST.IMG in the model direcory in the knowledge database (<ProgramDir>\<PLATFORM>\Ghost\<**Model**>).

10.4 Update the reference PC (optional)

Step 11. Prepare the reference PC for updating
Reset the platform of the reference PC if you want to continue with the state where you have left it.
With the Easy Manage Admin Program (Configurations -> Workstations- Select **REFPC** -> Open).

- Select **"<Platform> Image "** as platform, created in **step 5.**
  The next time when you install this workstation, the image will be used as base.

Step 12. Install the Reference PC for updating

- Restore the image with Bootd disk procedure or Install the reference PC again
- Update the reference PC to your wishes and follow **step 9** and **step 10** when you are finished.

10.5 Distribute the image

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

- Select **"<Platform> Image "** as platform, created in **step 3**
- Tab System Scripts -> Add the Applications:
  "<Platform> <xx>", created in **step 5**
  "**Image <Platform>"**, created in **step 6**
  Add the other Applications and hardware.

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

Note:
If you have decided in **step 10** not to convert to NTFS, then you can do this by hand. This step is only needed for the DOS Boot disk environment, because WinPE works with NTFS. 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

11. Seal a PC for reselling

If you sale PC's with Windows XP/Vista/Windows 7 preinstalled, then you have to seal the PC before you deliver it to your customer. When the customer starts the PC for the first time, it will ask for the license key and that license will be activated.

You can use all Easy Manage distribution methods to install a PC and then reseal it for shipping. Execute always Step 1, then decide which distribution method you want to use.

11.1 Prepare the Reseal (Once)

Step 1. Create the System Script "**Reseal (XP and Vista Only)**".
This application will copy the needed files to the workstation to start the reseal procedure.
With the Easy Manage Admin Program (Software -> System Scripts -> New).

- Enter **Reseal (XP and Vista Only)** as Name.
- Select **Initial Application** as class.
- Tab Script; Enter **Reseal** as Model. Easy Manage distributes the scripts in the **Reseal** Model directory.
- Tab Platforms; select the platforms, where to use this application.

11.2 Reseal a default or network distribution

If you want to reseal a PC, which you completely created, with Easy Manage, then execute **step 2** and **step 5.**

Step 2. Configure the PC
With the Easy Manage Admin Program (Configurations -> Workstations -> Open or New)

- Select a Default or Network platform
- Tab System Scripts -> Add the Applications:
  "<Platform> <xx>",
  "Reseal".
  Add the other Initial Applications, devices and adapters.
- Install the PC with the Easy Manage Boot Disk

11.3 Reseal with an image

If you want to distribute an image for reselling, then you have to execute **step 3** until **step 5**.

11.4 Make a Sysprep image

Step 3. Create an image as explained in chapter **Using an image and sysprep with Easy Manage**.

11.5 Distribute a Sysprep image

Step 4. With the Easy Manage Admin Program (Configurations -> Workstations -> Select Workstation -> Open)

- Select **"<Platform> Image "** as platform.
- Tab System Scripts -> Add the Applications:
  "<Platform> <xx>",
  "**Image <Platform> <xx>"**,
  "Reseal".
  Add the other Initial Applications, devices and adapters.
- Install the PC with the Easy Manage Boot Disk

11.6 Reseal the PC

Step 5. Run the command C:\RESEAL\RESEAL.BAT from the Start -> Run menu**.** This procedure will remove all Easy Manage agents, prepare the system for reseal and shutdown after it is finished.

12. 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 -> System Scripts -> 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
```

13. Change the Easy Manage Logo

If you are not satisfied with the Easy Manage Logo (Octopus), you can change the logo for the agent as well for the Admin Program and the Self Service Helpdesk. It is also possible to change the splash pages and menu's in the PXE environments.

Follow the following steps to change the logo for the Agent:

1. Create your own logo EZMLogo.bmp with a width of 192 and a height of 334 pixels and a resolution of 96 x 96 dots/inch.
2. Place your EZMLogo.bpm in the folder <ProgramDisk>\EZMBase\EZM
   The Agent will copy the EZMLogo.bmp file to the client and displays 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 MyImage.default.png with a width of 240 and a height of 417 pixels and a resolution of 72 x 72 dots/inch.
2. Place your MyImage.default.png in the folder <Tomcat>\webapps\ezmanage\image.
   Your logo will be displayed the next time that you start the Admin Program or Self Service Helpdesk.

Follow the following steps to change the splashpage in the EZMPE environment:

1. Create your own splash page Setup.bmp with a width of 800 and a height of 600 pixels and a resolution of 94 x 94 dots/inch.
2. Place your Setup.bmp in the folder <ProgramDisk>\bootdisk\winpe\32bit\windows\system32. Rebuild your EZMPE boot disk and your splash page will be displayed the next time that you start the boot disk.

Follow the following steps to change the splashpage of the Easy Manage PXE boot menu:

1. Create your own splash page mysplash.jpg with a width of 640 and a height of 480 pixels.
2. Place your mysplash.jpg in the folder <PXEServerpath>\tftproot. Your splash page will be displayed the next time that you start PXE boot.

Follow the following steps to change the menu of the Easy Manage PXE boot menu:

1. Copy the file <PXEServerpath>\tftproot\pxelinux.cfg\ezmdefault to <PXEServerpath>\tftproot\pxelinux.cfg\mydefault
2. Change the file mydefault with any text editor according the syntax described in<PXEServerpath>\tftproot\PXEHelpMenu.txt and save it.
3. Copy the file <PXEServerpath>\tftproot\pxelinux.cfg\mydefault to <PXEServerpath>\tftproot\pxelinux.cfg\default. Your menu will be displayed the next time that you start PXE boot.

14. Enable the Trace option

Easy Manage has a trace option for the support group. Sometimes there are strange effects, which could not be explained on a normal way, but has to be reported to the support group. When a trace is enabled, then a trace file will be created and could be sent to the support group.

There are two ways to enable this trace option:

1. Temporary.
   Only one program does not fucntion and needs further investigation. The program could be started from a MSDOS command prompt.
   A. Open a MSDOS Command box
   B. Change the directory to the C:\EZM directory
   C. Create the environment variable EZTRACE with the trace level (0 means NO trace, 9 means highest trace level) E.g. SET EZTRACE=9
   D. Start the program. The program will now create the file EZManage.trc in the C:\EZM directory.
   If you want to change the name of the file before you start the program, then you can set the environment variable EZTRACE_FILE with the full path of the trace file. E.g. set EZTRACE_FILE=C:\EZM\MYTRACE.trc

2. Permanent.
   Some programs are started immediately after a reboot and could not be started in a MSDOS box. You can create a Trace.ini file in the same directory of the program(s) to command the program(s) to create the trace. The contents of that file should be like:

   ```
   [Debug]
   eztrace=9
   eztrace_file=c:\ezm\ezmanage.trc
   ```

   Be Aware, if you place this file in the <ProgramDir>\EZMBase\EZM directory, then each Client PC will create a trace file.
   Disabling the permanent trace could be done by deleting the trace.ini file or change the value of eztrace into 0 or empty.

15. How to automate the security updates of Microsoft?

With Easy Manage you can manage the security updates of Microsoft on a very simple way. The concept is the following:

A. Centralize the downloads from Microsoft in the Easy Manage Knowledge Database.
Be aware! The updates from Microsoft are language and platform dependent.

1. Create the folder EZMWINUPD in the platform and distribution Model. This folder will become an extention of the operating system, which was used to install the PC.
   E.g. <ProgramDisk>\WINXP\OS\UKSP3\EZMWINUPD.
2. Use a reference PC, which was created with that distribution, to retrieve the updates for the specified platform with the specified language.
3. Start the Windows Standard update program. http://update.microsoft.com. The Genuine Advantage Validation program should always be executed on this reference PC.
4. Select the "Custom option" to see which updates are needed.
5. Select "Review other updates".
6. Start the Microsoft catalog update program to download the updates. http://catalog.update.microsoft.com.
7. Search for the KBxxxxxxx found in Step 4 and add them to the basket.
8. Download the updates to a temporary folder.
9. Move from the temporary folder and/or rename the files to the correct EZMWINUPD folder.

B. Each time the Agent is running on the client (workstation or server), it looks in the central knowledge database (EZMWINUPD folder) to see if there are updates that have to be done. It will also look in the Registry if the Update already was executed. If a new program is detected in the knowledge database and it was not yet installed, then it will be copied to the client. The update program will be executed in silent mode (with the default parameters).

The path to the central knowledge database will be retrieved from the <AgentDisk>\Machines\<CINumber>\UAGENT.ICF file and will be placed in the C:\EZM\EZMAgent.INI file.
The updates are copied on the client in the folder C:\EZM\EZMWINUPD

C. There is one file (<ProgramDisk;gt;\ezmbase\ezm\winupdate.txt) which have to be managed to control the parameters for the update program. This file will be copied to the client, when it was changed. The syntax of this file is: <Begin string of the file that needs other parameters>, <parameters>[,once]

Example of the WINUPDATE.TXT

```
; WinUpdate.txt Date: 19-11-2008
; This file will be used if you want to use other parameters for the Microsoft Windows Update program
; then the default.
; The default for .exe (Win200x/WinXP)files is: /quiet /passive /norestart
; The default for .msu (Windows Vista) files is: /quiet /norestart
; The Syntax is the beginning string of the FileName,Parameters[,once]
; Use the Once parameter if the update not is registered in the registry, so it will only executed once
; when the file not yet exists in the local WindowsUpdate directory
; E.g.
; KBD00,/quiet /passive /promptrestart
```

16. How to setup ThinStation

With Easy Manage you can deploy PC's as thin station. This means that you can use PC's with little resources (Memory/CPU or Harddisk) as dedicated workplace. ThinStation is an open source project on sourceforge see http://www.thinstation.org. In the distribution of Easy Manage is ThinStation as platform defined in the Knowledge Database. Some usefull examples of applications are already defined E.g. Remote Desktop (\initappl\rdesktop) and Internet Box (\initappl\internet) and of course you can define your own. You have to setup the ThinStation environment before you can use it. The linux kernel (vmlinuz and initrd) have to be downloaded in the ThinStation\OS\<model> folder in the knowledge database.

Follow the follwing steps to setup once

1. Create the platform **ThinStation** (if not yet exists)
   Software -> Platforms -> New
   Enter as name : **ThinStation**
   Select ThinStation as class
2. Create The Operating System ThinStation 2.2.2d (if not yet exists)
   Software -> Operating Systems -> New
   Enter as Name : **Thinstation 2.2.2d**
   Select ThinStation as platform
   Click the Tab Script and enter TS2d as Model. You have to create the folder TS2d in <ProgramDir>\ThinStation\OS manually.
3. Create the distribution ThinStation Network (if not yet extsts)
   Software -> Distributions -> New
   Enter as name : **ThinStation Network**
   Select the Operating System created in step 2 as Operating System
   Select Network as distribution method
4. Build the ThinStation kernel

Building the ThinStation kernel
The quick way is:
1. Create your linux kernel on the TS-O-Matic site see http://tsom.gelix.org/
2. After the build (takes about 15 minutes) go to the tab PXE and download the kernel files (initrd and vmlinuz).
3. Place the kernel files in the model folder \ThinStation\OS\TS2d

The slow way, but you can more customize the kernel, is:
1. Deploy an Ubuntu workstation
2. Download the http://sourceforge.net/projects/thinstation/files/Thinstation-<version>.tar.gz file
3. Place the file in your home directry and unpack the file with the command: tar xvf Thinstation-<version>.tar.gz
4. Go to the unpacked folder with the command: cd Thinstation-<version>
5. Change the file build.conf according your needs
6. Start the build with the command: ./build
7. Go to the folder boot-images/pxe and transfer the initrd and vmlinux files with FTP (in binary mode) to your /ezmprogs/thinstation/os/ts2d folder.

17. Easy Manage Internal Mail Interface

With Easy Manage you can create calls and manage assignments in the helpdesk via email. Easy Manage will check a POP3 mailbox each minute and processes the messages. The email interface (EZMIMI) is an extra option (**you need a license for it**) within Easy manage.

There are 2 types of email formats:

1. **Unformatted**
   - The email **from address** will be used as search key in the email field of the user. The found user which will be saved as Requested By and Request For in the call. If the user not was found, then the message will be lost and a reply to the sender will be sent.
   - The Body of the message will be saved as Problem Details in the call.
   - Attached files will be added to the call.
   - The Subject of the Message will be saved as Problem Summary in the call. If the call number was placed between "[" and "]" in the Subject of the message and the call does not have the status "Closed", then the message will be added as a journal to the progress of the call, else a new call will be created.
2. **Formatted**
   - A formatted message uses tags to define the fields and values in a call.
   - A message is of the type formatted, if it has the tag **#Formatted#** started at the first character of the subject.
   - The body text consists of tag rows and value rows. The tags are case less and could be defined in any order.
   - A tag will be recognized when it starts on the first character of a tag row in the body text of a message. It defines the field in the call and indicates that the next row is a value row.
   - The value of a field starts on the first row after the tag row.
   - Only the last non empty value row will be saved in single fields.
   - The following tags could be used:

Tagname	**Single Field**	**Meaning**
#Summary#	Yes	Problem summary (Mandatory)
#Details#	No	Problem details
#Status#	Yes	Number 1, 2 or 3 1 = Open 2 = Solved 3 = Closed
#Service#	Yes	The name of the defined service
#Category#	Yes	The name of the defined category. The service should also be filled, because the program checks if the category was linked to the service.
#Urgency#	Yes	Number 1, 2, 3 or 4 4 = High 3 = Medium 2 = Low 1 = Very low
#Impact#	Yes	Number 1, 2 or 3 1 = Small 2 = Medium 3 = Big
#Configuration#	Yes	Existing Configuration CINumber
#ComponentType#	Yes	Number 1, 2 or 3 1 = Application 2 = Application Script 3 = System Script
#Component#	Yes	The name of the existing component
#SolutionSummary#	Yes	Solution summary
#SolutionDetails#	No	Solution Details
#UserName#	Yes	The unique username as RequestBy and RequestFor. If the user not could be found, then the message will be lost.
#CallType#	Yes	Number 1, 2, 4 or 5 1 = Incident 2 = Change 4 = Problem 5 = Service Request (Default) 6 = Project
#ExternalReference#	Yes	CallReference of the third party. For EZM – EZM communication the format will be IMI:CallReference.AssignmentId.

#eMailRecipients#	Yes	eMail address(es) that will be handle as RequestFor separated by semicolons.
#Journal#	No	A journal record will be created in the progress of the call.
#PropertyAnswer1# #PropertyAnswer2# #PropertyAnswer3#	Yes	3 Extra fields in the properties tab of the call.
#END#	No	All rows will be flushed until the next tag

Use Cases:

1. An End user wants to raise a call to the Easy Manage Helpdesk via email.
   Pre-conditions
   - An IMI POP3 mailbox was created (E.g. imihelp@mycompany.com)
   - The password of the mailbox user is made persistant
   - Easy Manage is configured with the setup procedure to listen to the POP3 mailbox.
   - The user is defined in Easy Manage with a unique email address.
   - The user can send the message to the IMI POP3 mailbox with subject and body text.

   Process
   Easy Manage receives the message and creates a call with the subject as Problem Summary, the body as Problem Detail and the attached files.
   The user will receive a notification, when the notification event was created and the field (User – Tab Easy Manage) No Notification is disabled.
   When [CallId] is found in the subject and the field and the call is not closed, then the reply of the user will be saved as journal (Progress) in the call (Place [{CallId}] in the subject of the Notification Template).

2. The Easy Manage Helpdesk (EZM site) wants to use another Easy Manage Helpdesk (Third Party site) as Third Party. The EZM Site will use an assignment to create a call in the Easy Manage Helpdesk of the Third Party site. When the call is closed at the Third Party site, then the assignment will be finished at the EZM site.

   Pre-conditions
   At the EZM Site
   - An IMI POP3 mailbox was created (E.g. imihelp@mycompany.com)
   - The password of the mailbox user is made persistant.
   - Easy Manage is configured with the setup procedure to listen to the IMI POP3 mailbox.
   - An assignee EZM-ThirdParty1 was created.
   - An expertise group (Third Party) was created
   - The assignee is a member of the expertise group (Third Party)
   - The email address of the assignee is the IMI POP3 Address of the Third Party site.
   - The field (User – Tab Easy Manage) Formatted Notification is enabled for the assignee.
   - The field (User – Tab Easy Manage) No Notification is disabled for the assignee.
   - An Assignment Notification Event has to be created with the values in the field From "<none>" and the field To is "Assigned".
   - An Assignment Notification Event has to be created with the values in the field From "Waiting" and the field To "Assigned".

   At the Third Party site
   - An IMI POP3 mailbox was created (E.g. imihelp@thirdparty.com)
   - Easy Manage is configured with the setup procedure to listen to the IMI POP3 mailbox.
   - The user EZM-ThirdParty1 was created (The name of the user should be exactly the same as the name of the assignee of the supported site).
   - The email address of the user is the IMI POP3 address of the EZM site.
   - The field (User – Tab Easy Manage) Formatted Notification is enabled for the user.
   - The field (User – Tab Easy Manage) No Notification is disabled for the user.

   Process
   Easy Manage at the EZM site sends a formatted assignment notification to the third part site, when an assignment is assigned to EZM-ThirdParty1.
   The tag #ExternalReference# will be filled with "IMI:CallId.AssignmentId".
   The tag #Status# will be filled with "1" (Which means Open).
   The tag #UserName# will be filled with "EZM-ThirdParty1".
   The tag #Summary# will be filled with the summary of the assignment.
   The tag #Details# will be filled with the details of the assignment.

   Easy Manage at the Third Party site reads the formatted notification and creates a call with the given External Reference.

   Easy Manage at the Third Party site sends a Formatted Notification, that the call is registered to the User "EZM-ThirdParty1" of the EZM site.
   The tag #ExternalReference# will be filled with the External Reference.
   The tag #Status# will be filled with "1" (Which means Open).
   The tag #UserName# will be filled with "EZM-ThirdParty1".

   Easy Manage at the EZM site reads the formatted notification and changes the status of the assignment to "Accepted".

   When the call is solved or closed at the Third Party site, then a formatted notification will be send to the User "EZM-ThirdParty1" of the EZM site.
   The tag #ExternalReference# will be filled with the External Reference.
   The tag #Status# will be filled with "2" (Which means Solved) or with "3" (Which means Closed).
   The tag #UserName# will be filled with "EZM-ThirdParty1".

Easy Manage at the EZM site reads the formatted notification and changes the status of the assignment to “Finished”.

3. The Easy Manage Self Service Helpdesk portal is not suitable for your company. You can create your own portal and use the Internal Mail Interface to register calls.

   Pre-conditions
   - An IMI POP3 mailbox was created (E.g. imihelp@mycompany.com)
   - The password of the mailbox user is made persistant.
   - Easy Manage is configured with the setup procedure to listen to the POP3 mailbox.
   - Your portal can sent emails to the POP3 mailbox.

   Process
   The user fills the form of your portal with the needed fields. You can define which fields should be filled or predefined.
   The portal formats a mail and sends the email to POP3 mailbox.