

ImageUltra Builder
Version 1.1



User Guide

ImageUltra Builder
Version 1.1



User Guide

Note

Before using this information and the product it supports, be sure to read "Notices" on page 83.

First Edition (October 2002)

© Copyright International Business Machines Corporation 2002. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

| | |
|--------------------------------------|----------|
| Preface | v |
| Who should read this guide | vi |
| Additional information | vi |

| | |
|---|----------|
| Chapter 1. ImageUltra Builder capabilities | 1 |
| Understanding images | 1 |
| Definition of a Smart Image | 1 |
| Types of images supported by a Smart Image | 2 |
| The purpose of the service partition | 3 |
| The use and importance of Sysprep. | 3 |
| Which image types are best for you? | 3 |
| Ultra-Portable Image. | 4 |
| Portable-Sysprep Image. | 5 |
| Hardware-Specific Image | 6 |
| Summary of image types | 7 |

| | |
|---|----------|
| Chapter 2. An overview of the image process | 9 |
| Importing pre-existing modules | 10 |
| Building a module | 11 |
| Building a driver module, application module, or add-on operating-system module | 11 |
| Creating an operating system module | 12 |
| Creating or modifying base maps and driver maps | 13 |
| Base maps. | 13 |
| Driver maps | 13 |
| Deploying Smart Images to target computers | 16 |
| Installing images on the target computer | 17 |
| Advanced functions | 17 |
| Filters | 17 |
| Utilities. | 18 |

| | |
|--|-----------|
| Chapter 3. Installing the ImageUltra Builder program. | 19 |
| Minimum requirements | 19 |
| Prerequisites | 19 |
| Considerations | 20 |
| Limitations | 20 |
| Installing the ImageUltra Builder | 21 |
| Getting help after installation | 21 |

| | |
|---|-----------|
| Chapter 4. Working with the ImageUltra Builder interface | 23 |
| Layout of the Main window | 23 |
| Conventions used within the interface | 24 |
| Module repository | 28 |
| The Repository window | 28 |
| Map windows | 29 |
| Module windows | 30 |
| Modifying attributes | 30 |
| Tools | 31 |

| | |
|---|-----------|
| Chapter 5. Map construction overview | 33 |
| Creating a base map tree structure. | 33 |
| Inserting menu items in a base map | 33 |
| Inserting modules in a base map | 35 |
| Using utilities in a base map | 37 |
| Using filters in a base map | 38 |
| Using links in a base map | 40 |
| Defining user information in a base map | 41 |
| Controlling the network-sync function | 42 |
| Controlling the service partition behavior | 42 |
| Simple and complex base maps. | 43 |
| Constructing a device-driver map tree structure | 44 |
| Inserting menu items in a driver map | 44 |
| Inserting modules in a driver map. | 45 |
| Using filters in a driver map | 45 |

| | |
|--|-----------|
| Chapter 6. Creating maps. | 47 |
| Creating and modifying base maps | 47 |
| Creating a new base map. | 47 |
| Adding menu items to a base map | 47 |
| Adding operating-system modules to a base map | 48 |
| Adding application modules to a base map. | 48 |
| Working with base map and menu item attributes | 49 |
| Creating and modifying driver maps | 51 |
| Creating a new driver map | 52 |
| Adding menu items to a driver map | 52 |
| Adding device-driver modules to a driver map | 53 |
| Specifying driver map attributes | 53 |

| | |
|---|-----------|
| Chapter 7. Preparing source files for modules | 55 |
| Creating source operating-system images | 55 |
| A note about Ultra-Portable Images | 55 |
| Required third-party tools | 55 |
| Creating a Portable-Sysprep Image | 57 |
| Creating a Hardware-Specific Image | 61 |
| Preparing source files for application modules and add-on operating-system modules. | 63 |
| Preparing for a silent install | 63 |
| Adding mini-applications. | 64 |
| Making files accessible to the ImageUltra console | 64 |
| After the files are prepared | 64 |
| Setting the attributes for a new application module. | 64 |
| Setting the attributes for a new add-on operating-system module. | 66 |
| Preparing source device-driver files | 68 |
| Adding mini-applications. | 68 |
| Requirements for preparing device-driver files for modules | 68 |
| Setting the attributes for a new device-driver module. | 68 |
| Preparing source utility files. | 70 |
| Preparing source filter files | 71 |

| | | | |
|---|-----------|---|-----------|
| Building a new module | 72 | Device Driver exceptions | 77 |
| Chapter 8. Getting help and support | 73 | Editing Unique Scripts. | 78 |
| Using the documentation and help system | 73 | Including Rapid Restore PC | 79 |
| Using the Web | 73 | Changing the order of module installation | 80 |
| Contacting an ImageUltra Builder technical expert | 74 | Appendix. Notices | 83 |
| Chapter 9. Dealing with exceptions | 75 | Trademarks | 83 |
| Application module exceptions | 75 | Index | 85 |

Preface

The ImageUltra™ Builder package contains this user guide and the ImageUltra Builder CD.

This guide has three purposes:

- To help you understand the concepts behind building, deploying, and installing Smart Images using the ImageUltra Builder program.
- To guide you through the ImageUltra Builder installation process.
- To supplement the ImageUltra Builder help system, which provides detailed step-by-step procedures for the tasks associated with the ImageUltra Builder program.

Before installing or using the ImageUltra Builder program, make sure that you are familiar with the information in Chapter 1, “ImageUltra Builder capabilities” and Chapter 2, “An overview of the image process”.

This guide is organized as follows:

Chapter 1, “ImageUltra Builder capabilities” on page 1 contains an overview of the concepts and functions of the ImageUltra Builder program.

Chapter 2, “An overview of the image process” on page 9 contains an introduction to the image-building process using the ImageUltra Builder program.

Chapter 3, “Installing the ImageUltra Builder program” on page 19 contains installation instructions for the ImageUltra Builder program, as well as prerequisites, limitations, and considerations that you need to understand before you install the program.

Chapter 4, “Working with the ImageUltra Builder interface” on page 23 introduces you to the various windows and conventions used in the ImageUltra Builder interface.

Chapter 5, “Map construction overview” on page 33 focuses on the concepts behind creating a map tree structure and setting some of the map attributes.

Chapter 6, “Creating maps” on page 47 provides step-by-step instructions for creating base maps and driver maps.

Chapter 7, “Preparing source files for modules” on page 55 provides step-by-step instructions for preparing the source files, creating repository entries, setting attributes, and building the module.

Chapter 8, “Getting help and support” on page 73 contains instructions about getting help and support for the ImageUltra Builder program from IBM®.

Chapter 9, “Dealing with exceptions” on page 75 contains helpful information about working with special cases that affect the image building process.

“Notices” on page 83 contains legal notices and trademark information.

Who should read this guide

This guide is intended for information technology professionals who do the following:

- Set up and maintain computer images
- Deploy images by CD or across a network

Additional information

You can obtain additional information regarding the ImageUltra Builder program from the ImageUltra Builder Web site at <http://www.pc.ibm.com/qtechinfo/MIGR-44316.html>.

This User Guide might be updated periodically. The updated versions will be made available at this Web site.

Chapter 1. ImageUltra Builder capabilities

The IBM® ImageUltra™ Builder program provides new methods of creating, deploying, and installing custom images. This chapter provides information to help you understand the concept of building images and the various types of images that the ImageUltra Builder program supports.

Understanding images

An image consists of several components: an operating system, applications, device drivers, and other related files needed to successfully run a computer. In most cases, Information Technology (IT) departments create images on source computers, extract the images and store them in a central location. Then, they deploy the images to multiple computers either across a network or from a set of image CDs. This enables an IT department to develop standardized images and control the versions of software and device drivers used throughout their company.

Developing and deploying images in this way ensures that every computer within a company maintains a high level of image consistency. However, because images created by this method are hardware dependent, a significant amount of time is spent creating, testing, and transferring images from multiple source (donor) computers to accommodate various hardware configurations and variations of supported software. In many cases, in order to accommodate specific software requirements, a base image is deployed and specific application programs are added manually on a system-by-system basis after the image has been installed.

As a result, IT departments often have multiple images to manage. Whenever a change is required to accommodate new hardware, an operating-system update, a device-driver update, an additional language, or a new or changed application program, many images must be rebuilt.

The major challenges most IT departments face are as follows:

- Reducing the amount of time spent creating and testing images.
- Finding a way to build and deploy images that provides increased flexibility to meet specific software requirements that might vary from department to department.
- Finding a way to build and deploy images across a wider range of hardware.
- Reducing or minimizing the total number of images that have to be maintained.

The purpose of the ImageUltra Builder program is to provide IT departments with the tools needed to develop, deploy, install, and maintain a variety of images efficiently. Through new tools and techniques, the ImageUltra Builder program enables you to break down images into reusable subcomponents (called *modules*) and use *maps* to construct Smart Images that can be deployed across a broad range of hardware platforms.

Definition of a Smart Image

The ImageUltra Builder program enables you to store a variety of image-related files as reusable *modules* in a repository. Typically, there is a module for each operating system, application, and device driver. These modules are organized by *maps* to meet the needs of specific users, groups, or an entire enterprise. A single

base map can contain multiple operating systems and applications, as well as variations based on language. A single *driver map* can contain device drivers for a single hardware platform, or through the use of *filters*, device drivers for a variety of hardware platforms. At the beginning of the deployment process, the installer selects which base map and driver map to use. All of the modules defined in the maps are copied to a hidden partition (called the *service partition*) on the target computer hard disk along with a set of installation and recovery tools. The compilation of modules and tools that is contained in the hidden service partition is called a *Smart Image*.

A Smart Image has the following characteristics:

- It is the source from which a specific image is installed on the target computer.
- It establishes recovery capabilities on the target computer.
- A single Smart Image can contain multiple images or components that make up multiple images.
- It can have varying degrees of portability across different hardware platforms. Portability is the degree in which you can use a single image for different types of computers.

The actual installation process can begin immediately after a Smart Image is deployed, or the computer can be prepared to have the installation process performed at a later date. During the installation process, the person performing the installation (IT professional or end user) makes selections from a menu system to define which image to install. The menu system and menu entries are controlled by entries in the *base map*.

Types of images supported by a Smart Image

A Smart Image can contain three different types of images, each achieving different levels of portability:

- **Ultra-Portable Image:** This type of image contains hardware-independent operating-system modules developed by IBM and application modules and device-driver modules that are either developed by IBM or created through the ImageUltra Builder program. These modules enable you to create Smart Images that can be used across the IBM product line of Hardware-Independent-Imaging-Technology-enabled (HIIT-enabled) personal computers. For a listing of IBM HIIT-enabled personal computers, visit the ImageUltra Web site at <http://www.pc.ibm.com/qtechinfo/MIGR-44316.html>.
- **Portable-Sysprep Image:** This type of image is created by a third-party image-cloning tool, such as Symantec Norton Ghost or PowerQuest DeployCenter, but modified by ImageUltra Builder tools to enable the installation of additional modules created and managed by the ImageUltra Builder program. The additional modules can be device-driver modules to enhance portability across various hardware platforms or application modules to provide additional flexibility to meet the needs of individual departments.
- **Hardware-Specific Image:** This type of image is created by a third-party image-cloning tool, such as Symantec Norton Ghost or PowerQuest DeployCenter. It is a self-contained image, which *does not* enable the installation of additional modules. The hardware configuration of the target computer must be the same as the hardware configuration of the source computer used to create the image.

Additional detail about these image types are discussed later in this publication.

The purpose of the service partition

The service partition is a hidden partition on a hard disk that typically contains operating-system modules, application modules, device-driver modules, and a set of recovery tools. It also provides its own operating environment, which has no dependency on the Windows operating system being present on any active partition. Because the service partition is protected from common read and write operations as well as most viruses, it is a practical place to store recovery, backup, and installation files and tools.

The service partition has three major purposes:

- As shipped from the factory on IBM HIIT-enabled computers, the service partition provides a recovery mechanism to restore the contents of the hard disk to its factory-installed state. It also provides a vehicle for distributing IBM-developed modules that the ImageUltra Builder program can import.
- As used by the ImageUltra Builder program, it is a storage area for deployed Smart Images and for the tools required to perform the image-installation process. Smart Images replace the factory-installed image files in the service partition. If a service partition is not on a target computer before a Smart Image is deployed, one is created during the deployment process.
- As used by the IBM Rapid Restore PC program (available separately from IBM), the service partition is a storage area for backup images and the tools required to restore backup images or selected files to the active partitions of the hard disk. The Rapid Restore PC backup images and recovery tools share the service partition with factory-installed images and Smart Images; they do not replace them.

The use and importance of Sysprep

If you intend to use Portable-Sysprep Images or Hardware-Specific Images, you must have a thorough knowledge of the Microsoft Sysprep tool. Unlike Ultra-Portable Images, Portable-Sysprep Images and Hardware-Specific Images are based on images extracted from a source computer that has gone through a Windows setup operation. Therefore, specific user and hardware information must be removed from the source computer using the Sysprep tool before the image can be deployed to other computers.

In contrast, Ultra-Portable Images are created from modules provided by IBM and, optionally, application and device-driver modules developed through the ImageUltra Builder program. If you intend to use Ultra-Portable Images only, the Sysprep tool is not used.

Which image types are best for you?

The types of images you choose to deploy using the ImageUltra Builder program depends on the mix of PCs used by your company, the investment you have in images created by third-party image-cloning software, and the willingness of your IT staff to learn new methods for creating and modifying images.

For example:

- To maintain existing PCs for which you already have images, you might choose to convert existing Hardware-Specific Images into modules that can be stored and deployed using the ImageUltra Builder program.

- If you are planning to install assorted models of IBM PCs, you might consider developing Ultra-Portable Images that can be used across the IBM product line regardless of hardware variations.
- If you plan to install a mix of PCs from various manufacturers, consider using the ImageUltra Builder program in conjunction with third-party image-cloning software to create Portable-SysPrep Images that enable the installation of application or device-driver modules so that the image can be deployed across a wider range of PCs.

Because Smart Images support all three types of images, you can use combinations of images within a single base map.

Each image type has its own characteristics. The following sections provide a brief overview of the function, advantages, and disadvantages of each type of image that can be deployed as part of a Smart Image.

Ultra-Portable Image

The Ultra-Portable Image is the most portable of the three types of supported images. Because an Ultra-Portable Image is deployed in an uninstalled state and uses the full Windows setup for hardware detection during installation, the image is virtually hardware independent. However, the high portability of an Ultra-Portable Image comes with a cost; this type of image takes more than 50 minutes to install. Nevertheless, the life-cycle of an Ultra-Portable Image is typically longer than Portable-Sysprep Images and Hardware-Specific Images because the image can be used across the IBM product line of HIIT-enabled personal computers, regardless of the core chip set and other hardware variations.

Anatomy of an Ultra-Portable Image

An Ultra-Portable Image consists of several modules that are imported from the service partition of a source IBM computer and additional modules that you create using the ImageUltra Builder program. These modules are stored in the ImageUltra Builder repository. One type of module is the *base operating-system module*, which only IBM can provide. The base operating-system module contains all of the files required for a specific operating system. The base module resides in the service partition of the source IBM computer and is part of a *container module* that also holds *add-on operating-system modules* for hot fixes, patches, updates, and service packs. Additional modules, such as *application modules* and *device-driver modules* also are located in the service partition and can be imported into the ImageUltra Builder repository. Additionally, the ImageUltra Builder program enables you to build your own modules and import modules from IBM Recovery CDs. This enables you to use a common base module with various combinations of device-driver and application modules to produce department-unique or user-unique images. Because of the modular construction and reuse of modules, Ultra-Portable Images are much more efficient to store than traditional images.

As you are developing an Ultra-Portable Image, the ImageUltra Builder program also enables you to predefine default user settings or implement a process that prompts the person installing the image to provide all necessary user settings near the beginning of the installation process. The settings are stored and provided to the Windows setup program when needed. This feature minimizes the amount of time that someone needs to be present at a target computer during the installation process.

Portability of an Ultra-Portable Image

The high portability of the Ultra-Portable Image is limited to IBM personal computers only. The same base module and most application modules can be used

across the IBM personal computer product line of HIIT-enabled systems, regardless of hardware variations. For a listing of IBM HIIT-enabled personal computers, visit the ImageUltra Web site at <http://www.pc.ibm.com/qtechinfo/MIGR-44316.html>.

Note: Do not install an Ultra-Portable Image on a non-IBM computer. If you attempt to install an Ultra-Portable Image on a non-IBM computer, the deployment and installation should complete without error; however, you might be violating terms of the Microsoft Windows End User License Agreement and the license agreement for this product. Also, in the case of Windows XP, an Ultra-Portable Image will work for only 30 days after installation on a non-IBM computer. After 30 days, Microsoft will require you to register the operating system and computer to reactivate the image.

An Ultra-Portable Image enables you either to use a driver map to control the level of the device drivers deployed at the target computer or to use the IBM-factory-installed device drivers already contained in the service partition of the target computer. Using the factory-installed device drivers makes an Ultra-Portable Image even more portable and simplifies development because you only have to develop and maintain a base map and its associated modules; you do not have to concern yourself with driver maps and device-driver modules. However, managing your own device drivers enables you to standardize them, thereby reducing diagnostic time if a problem arises.

An added benefit of using your own driver map is that you can use the Image-Ultra Builder program for recovery purposes if a hard disk fails and requires replacement. Because the driver map deploys the device drivers as part of the Smart Image, no preexisting drivers need to be present at the target computer. Therefore, by using a driver map, the Smart Image can be installed on a new hard disk drive.

Portable-Sysprep Image

Using the supported third-party image-cloning tools (Symantec Norton Ghost or PowerQuest DeployCenter) in conjunction with the ImageUltra Builder program, you can create a Portable-Sysprep Image that can be deployed across a wider range of hardware than a traditional image.

Anatomy of a Portable-Sysprep Image

In contrast to an Ultra-Portable Image, a Portable-Sysprep Image is a “snapshot” of a source computer’s hard disk contents. The source computer is set up to match the desired configuration of the target computers. However, because Windows setup has been run on the source computer, specific user and hardware information (such as the user ID, passwords, and network settings) is recorded in the source computer’s registry. The Microsoft Sysprep tool must be run on the source computer to erase this information before third party imaging software is used to extract the image.

By implementing a few minor variations during the development of a traditional image, you can enable the image to use application or device-driver modules to append the image during the deployment and installation process. These variations are discussed in detail in Chapter 7, “Preparing source files for modules” on page 55.

After creating the image using Symantec Norton Ghost or PowerQuest DeployCenter, you use the ImageUltra Builder program to create and build a base operating-system module from the image. The module is automatically stored in the ImageUltra Builder module repository during the module-building process.

After a Portable-Sysprep Image has been deployed to the target computer hard disk, a mini-setup runs during the installation process. Any device drivers that are defined in the driver map are copied to the C drive and made available to the mini-setup. The mini-setup prompts for specific user information and performs a limited hardware query. Although the mini-setup makes allowances for minor variations in the hardware (such as audio, video, and so on), it does not deal effectively with different core chipsets. Because a Portable-Sysprep Image does not have to go through a complete Windows setup, it typically reduces initial Windows setup time to 10 minutes or less. The ImageUltra Builder program enables you to predefine default user settings or implement a process that prompts the person installing the image to provide all necessary user settings near the beginning of the installation process. These settings are stored and provided to the mini-setup program when needed.

Upon the successful completion of the mini-setup, any additional applications defined by the base map are installed. The actual setup time for a complete image installation varies depending on the size and number of application modules defined in your base map.

Portability of a Portable-Sysprep image

Although Portable-Sysprep Images can be used with IBM and non-IBM computers and provide increased portability by enabling additional device drivers and applications to be installed, there are still hardware limitations. As a general rule, a Portable-Sysprep Image is for use on identical or similar computers, and in many cases, the Portable-Sysprep Image is manufacturer unique.

There are various factors in determining what constitutes a similar computer.

- If you are deploying a Portable-Sysprep Image to a target computer that has the identical system board (motherboard) as the source computer, the image will install and work successfully.
- If you are deploying a Portable-Sysprep Image to a target computer that has a different system board, but the same core chip set as the source computer, there is a high probability that the image will install and work successfully.
- If you are deploying a Portable-Sysprep Image to a target computer that has a different system board and a different core chip set from the source computer, there is a low probability that the image will install and work successfully.

Hardware-Specific Image

The least portable image is the Hardware-Specific Image. Hardware-Specific Images can be developed for, deployed to, and installed on IBM and non-IBM computers, but each image must be unique to a specific hardware and software configuration.

Anatomy of a Hardware-Specific Image

A Hardware-Specific Image is similar to a Portable-Sysprep Image, but it does not have the ability to append itself with additional modules. It is essentially a “snapshot” of a source computer hard disk after the Microsoft Sysprep program has been run. When deployed to a target computer, the Hardware-Specific Image performs a limited hardware query, but the hardware it can support is limited to the device drivers that were provided by the source computer. A

Hardware-Specific Image forces its source and target computers to contain the exact same hardware with little or no variations. It also forces any variation in applications to be handled separately from the image-deployment process and image-installation process.

Portability of a Hardware-Specific Image

A Hardware-Specific Image is for use on identical computers. Hardware variations are limited to plug-in devices and peripherals for which device drivers were provided on the source computer. No variation in software content can be handled by the deployment process. If additional device drivers or application programs are required by the target computer, they must be installed manually or through some other software-deployment mechanism, such as the IBM Software Delivery Assistant program.

Summary of image types

The following table summarizes the characteristics associated with an Ultra-Portable Image:

| Description | Advantages | Disadvantages |
|--|--|--|
| <ul style="list-style-type: none">• Unpacked at the target computer before Windows setup is run• Uses a full Windows unattended setup, which uses plug-n-play hardware detection• User information can be predefined | <ul style="list-style-type: none">• Maximum portability across the IBM product line of HIIT-enabled personal computers• Extremely reliable• Application programs and device drivers are added independent of the base operating-system module. Therefore, a common base operating-system module can be used with various combinations of driver modules and application modules to produce many variations of an image.• Longer life-cycle of the image• No hardware dependencies• Easy to maintain | <ul style="list-style-type: none">• Initial setup at the target computer can take 50 minutes to install the Windows operating system, plus the time it takes to install all applications• For use with IBM computers only |

The following table summarizes the characteristics associated with a Portable-Sysprep Image:

| Description | Advantages | Disadvantages |
|--|---|---|
| <ul style="list-style-type: none"> • Semi-modular (supports the installation of separate application modules and driver modules to append the base image) • Setup is run once on a source computer • Sysprep cleans out user information and some hardware-specific information from the registry • Performs a limited hardware query at the target computer (Windows plug-n-play) • User information can be predefined | <ul style="list-style-type: none"> • Initial setup at the target computer is typically 10 minutes or less for a base image, plus the time it takes to install all applications • Some portability (such as video, audio, network, and so on) • Supports IBM and non-IBM computers • Base operating system module can include some large core applications (such as office suites), which speeds installation time | <ul style="list-style-type: none"> • Image is used only for identical or similar hardware, and in most cases is manufacturer unique. |

The following table summarizes the characteristics associated with a Hardware-Specific Image:

| Description | Advantages | Disadvantages |
|---|--|--|
| <ul style="list-style-type: none"> • Setup is run once at a source computer • Applications and drivers are part of the base image • Sysprep can be run to clean out user information | <ul style="list-style-type: none"> • Initial setup at the target machine is typically 10 minutes or less • Images can be developed for, deployed to, and installed on IBM and non-IBM computers, but each image must be unique to a specific hardware and software configuration | <ul style="list-style-type: none"> • One image per hardware platform • Image is for use on identical computers only • Unique image must be created for each set of applications, or applications must be added manually after installation. • No portability |

Chapter 2. An overview of the image process

This chapter will help you understand the various processes and components associated with the ImageUltra Builder program and how they relate to each other. For step-by-step instructions on how to complete any of the processes described in this chapter, refer to the ImageUltra Builder help system.

The ImageUltra Builder program stores individual image components as modules in the ImageUltra Builder repository. A module is a packaged, compressed, self-contained unit that often can be combined with other modules in the repository to make one or more images. Each application is a module, each device driver is a module, each operating system is a module, and so on.

Note: A module created from a Hardware-Specific Image cannot be combined with any other modules to enhance its function. A module created from a Portable-Sysprep image already contains device drivers and applications, but can be combined with other application modules and device-driver modules to enhance its function.

Modules fall into the following basic categories:

- **Operating-system modules:** There are two types of operating system modules:
 - Base operating-system modules: These modules contain all of the components that are part of the core operating system. Base-operating system modules include modules created from Hardware-Specific Images and Portable-Sysprep Images, as well as base operating-system modules imported from the service partition of a source IBM computer.
 - Add-on operating system modules: These modules include items such as operating-system hot fixes, service packs, patches, and updates.

The deployment and installation of operating-system modules are controlled by *base maps*.

- **Application modules:** Each of these modules contains all of the components associated with a specific application program.
 - If you are building an Ultra-Portable Image or a Portable-Sysprep Image, you can use application modules. The deployment and installation of the applications modules are controlled by *base maps*.
 - If you are building a Hardware-Specific Image, you cannot use application modules in conjunction with that image. All applications must be part of the Hardware-Specific Image.
- **Device-driver modules:** Each of these modules contains all of the components associated with a specific device driver.
 - If you are building an Ultra-Portable Image or a Portable-Sysprep Image, you can use device-driver modules. The deployment and installation of the device-driver modules are controlled by *driver maps*.
 - If you are building a Hardware-Specific Image, you cannot use device-driver modules in conjunction with that image. All device drivers must be part of the Hardware-Specific Image.

During the deployment process, you choose which base map and which driver map to deploy. This enables you to maintain device-driver modules and driver maps independently of the operating-system and application content defined by base maps.

In addition to modules, the ImageUltra Builder program enables you to create *containers* in the repository. A container is a special type of module that enables you to group other modules together under a single identifier. All modules in a container must be of the same type. For example, you might want to group all of the device drivers used for a specific machine type into a device-driver container. Or, you might want to group a Windows 2000 base operating-system module with all of its associated add-on modules into an operating-system container.

The use of containers is optional, but you might find containers helpful when creating maps because you can simply insert the container module into your map instead of inserting each individual module.

There are five basic steps to developing an image:

1. Importing pre-existing modules
2. Building a module
3. Creating and modifying base maps and driver maps
4. Deploying Smart Images to target computers
5. Installing images on the target computer

Importing pre-existing modules

If you want to use IBM-developed modules, you use the ImageUltra Builder import tool to make an *Import Diskette* or *Import CD*, and then use the diskette or CD on the source computer to import modules from the service partition of the IBM source computer over the network to the ImageUltra Builder repository. The import program enables you to view a list of all the operating-system modules, application modules, and device-driver modules that are in the source computer service partition and select only those modules that you want to import. In most cases, operating-system modules are in a container and the entire container must be imported.

The advantages of importing modules, as opposed to building your own, are as follows:

- IBM-developed modules have been tested by IBM and are ready for deployment.
- *Base maps* and *driver maps* are automatically created by the ImageUltra Builder import function based on the modules you select to import. You can use these maps as they are, create new maps based on the maps created by the ImageUltra import program and modify the maps to meet your particular needs, or discard the maps and create your own maps.
- If base maps and driver maps already exist in the service partition of the source computer, you can import those maps.
- IBM-developed modules contain applications and signed device drivers that conform to Microsoft certification requirements, ensuring device-driver compatibility.
- IBM-developed base operating-system modules are the central element of Ultra-Portable Images, which provide increased portability across a broad range of hardware. This type of module must be imported.

You also can use an *Import Diskette* or *Import CD* to import modules and image maps from the service partition of any target computer containing a Smart Image that was deployed by the ImageUltra Builder program.

Building a module

In addition to importing modules, you can create your own modules for operating systems, applications, and device drivers. However, there are a few limitations:

- Application modules and device-driver modules are for use in Ultra-Portable Images and Portable-Sysprep Images only; they cannot be used in conjunction with Hardware-Specific Images. Hardware-Specific Images must have all required applications and device drivers installed on the source computer before the image is created.
- Base operating-system modules can be created from Hardware-Specific Images and Portable-Sysprep Images only; base operating-system modules for Ultra-Portable Images must be imported.

Building a driver module, application module, or add-on operating-system module

Preparing files for a device-driver module is different from preparing files for an application module or an add-on operating-system module:

- **Application files and add-on operating-system modules:** Any application or add-on operating-system component that you intend to use for a module must have a silent-install capability for unattended installation. Before you build a module, prepare the application or add-on operating-system component for unattended installation so that the installation process does not require any user interaction. In most cases, the Microsoft Software Installer (MSI) and InstallShield programs allow for these types of automated installations.

Applications and add-on operating-system components that do not allow for an unattended installation cannot be deployed as modules; they have to be installed after the image has been installed either manually or through some other software distribution method, such as the IBM Software Delivery Assistant program.

- **Device-driver files:** When building a device-driver module, you must use the standard Microsoft Plug-n-Play .INF installation method, which is usually unattended in nature. The device driver itself must be certified by the Microsoft Windows Hardware Qualification Lab (WHQL) and signed. Failure to use signed, WHQL-certified device drivers will result in interruptions during the installation process because of Windows-generated information messages, which require user interaction to bypass.

After the application, add-on operating-system component, or device driver has been prepared, you use the New Module Wizard to create a module entry in the repository and to set the attributes associated with the module. Attributes vary depending on whether the module is a device-driver module, an application module, or an operating-system module. In all cases you must define a name that will identify the module in the repository and the location where the prepared files are currently stored. When you have finished defining the attributes, you can select the entry from the repository and use the build tool to build the module.

Note: The ImageUltra Builder program does *not* automatically scan files for viruses. To help ensure a virus-free environment, use a virus-checking program on the computer used to build the modules.

For step-by-step instructions on preparing source files and creating a module, see Chapter 7, “Preparing source files for modules” on page 55 or refer to the ImageUltra Builder help system.

Creating an operating system module

If you choose to create your own base operating-system modules instead of importing an IBM-developed base operating-system module, you can either build a module from an existing Hardware-Specific Image (an industry-standard image created by Symantec Norton Ghost or PowerQuest DeployCenter) or you can create a Portable-Sysprep Image and build a module from it. When you create a base operating-system module from a Portable-Sysprep Image, the module can use application modules and device-driver modules to enhance its portability across a wider range of hardware.

Creating a base operating-system module from a Hardware-Specific Image

A Hardware-Specific Image is a self-contained entity (operating system, applications, and device drivers) that is prepared using the Microsoft Sysprep tool and created using either Symantec Norton Ghost or PowerQuest DeployCenter. It has limited portability because the source computer and target computer must use identical hardware with very few variations. Many large companies are using Hardware-Specific Images that can be converted to modules and deployed using the ImageUltra Builder program.

Building an operating-system module from an existing Hardware-Specific Image is simple. You use the New Module Wizard to create an operating-system module entry in the repository and to set the attributes associated with the module. When you have finished defining the attributes, you can select the entry from the repository and use the build tool to build the module.

For step-by-step instructions on preparing source files for a Hardware-Specific Image and creating a module, see Chapter 7, “Preparing source files for modules” on page 55 or refer to the ImageUltra Builder help system.

Creating a base operating-system module from a Portable-Sysprep Image

Creating a module from a Portable-Sysprep Image is nearly identical to creating a module from a Hardware-Specific Image. The only difference is in the way you create the Portable-Sysprep Image on the source computer.

By altering how the image is developed, you can enable the image to append itself with other application programs and device drivers during the deployment and installation process. The variations in development include the following:

- Using MS-DOS to format the source computer hard disk as FAT32 and install the DOS system files before installing Windows.
- Running the IBM Customization Program (provided by the ImageUltra Builder program) on the source computer before running the Sysprep program. The IBM Customization Program provides the hooks required for Portable Sysprep Images to make use of applications and device drivers defined in maps.
- Modifying the BOOT.INI file so the computer will initially boot from DOS.

Most other aspects of developing a Portable-Sysprep Image are the same as developing a traditional Hardware-Specific Image including running the Sysprep tool and creating the image files using either Symantec Norton Ghost or PowerQuest DeployCenter. Once you have created the image files, you use the

New Module Wizard to create an operating-system module entry in the repository and to set the attributes associated with the module. When you have finished defining the attributes, you can select the entry from the repository and use the build tool to build the module.

For step-by-step instructions on building a Portable-Sysprep Image, see Chapter 7, “Preparing source files for modules” on page 55.

Creating or modifying base maps and driver maps

To understand the importance and use of base maps and driver maps, you need to have some insight into the deployment and installation processes.

- During the deployment process, the person at the target computer is prompted to select a base map and a driver map from the repository. The modules that are defined in the selected base map and driver map are deployed to the service partition of the target computer.
- During the installation process, the person at the target computer is presented a set of menus. The menu selections define the content of the image that will be installed on the target computer. The menu items and menu hierarchy are defined by menu-item entries in the *base map*. The physical appearance and behavior of the menus are controlled by the attributes assigned to the various menu items.

There are two parts to every map:

- The tree structure: A map tree structure is very similar to the tree structure found in Windows Explorer; however, instead of containing files and folders, a map tree structure contains menu items and modules, and possibly containers.
- Attributes: The root entry of the tree structure and each menu item contained in the tree structure has a set of user-definable attributes associated with it.

Base maps and driver maps have different purposes and use the menu items and tree structure differently. Therefore, an individual overview of each is provided.

Base maps

The tree structure in a base map enables you to define multi-tiered, multi-branch structures containing a series of menu items that control the flow and content of the menus displayed at the target computer during the installation process. Modules are inserted in the map under menu items in strategic locations. As each menu item is selected at the target computer, modules associated with that menu item are collected. When the final menu item is selected from the last menu in the sequence, all of the modules that have been accumulated are installed.

You use the New Map Wizard to create a base-map entry in the repository. Then you populate the tree structure with menu items to establish the flow and content of your menu system. Finally, you insert modules under menu items in strategic locations to meet your particular needs. For detailed information about constructing base maps, see Chapter 5, “Map construction overview” on page 33.

Driver maps

Driver maps are for use with Ultra-Portable Images and Portable-Sysprep Images only; driver maps are *not* used by Hardware-Specific Images. If you select a driver map in conjunction with a base map that contains a Hardware-Specific Image, the driver map is ignored during the installation of the Hardware-Specific Image.

The tree structure in a driver map enables you to define a multi-branch structure containing device-driver modules. The use of menu items is optional. Menu items in a driver map serve two purposes only:

- They enable a developer to annotate or label groups of device-driver modules.
- They enable a developer to assign *filters* to one or more device-driver modules. Using filters is optional, but they might be useful if you create a driver map that contains device-driver modules for multiple computer types. The filters provided with the ImageUltra Builder program enable you to key off of the machine type, machine type and model, hardware platform (desktop or mobile), and whether or not a computer is an IBM computer.

Menu items in driver maps have no effect on the menus displayed at the target computer.

You use the New Map Wizard to create a driver-map entry in the repository. Then you populate the tree structure with menu items, if you choose. Finally, you insert modules in your tree structure and optionally assign filters to menu items. For detailed information about constructing driver maps, see Chapter 5, “Map construction overview” on page 33.

The use of driver maps is optional if the target computer is an IBM HIIT-enabled personal computer. If you do not select a driver map during the deployment of an Ultra-Portable Image or a Portable-Sysprep Image, the installation process will use the device-driver modules that are present in the target computer service partition.

It is important to note that when a driver map is used with an IBM HIIT-enabled personal computer, the driver map does not append the factory-installed device-driver modules; it deletes all factory-installed device-driver modules and replaces them with the device-driver modules defined in the map. Therefore:

- If you use a driver map with an Ultra-Portable Image, the driver map must contain *all* device drivers required for the target computer.
- If you use a driver map with a Portable-Sysprep Image, the driver map must contain all device drivers required to supplement those already contained in the base operating-system module.

Defining service partition behavior

The service partition is a hidden partition on the hard disk that contains a set of recovery tools, operating-system modules, application modules, device-driver modules, and in some cases maps. As shipped from the factory, the modules in the service partition reflect the factory-installed image on the C drive. When deploying an image from the ImageUltra Builder program, the factory-installed operating-system modules and application modules are deleted from the service partition and replaced by the modules defined in your base map. If a driver map is used, all factory-installed device-driver modules are deleted from the service partition and replaced by the modules defined in the driver map. The service partition is resized, if necessary, to accommodate all of the modules defined in the maps. The actual size of the service partition is directly related to the size and number of modules defined in your maps.

As you create your base map, the New Map Wizard enables you to define how the service partition on the target computer is treated after an image has been installed. You have three options:

- Delete none: The service partition size and contents remain the same as when the Smart Image was deployed.

- **Delete some:** All modules not used for this specific computer's image are deleted. The service partition is resized to accommodate only those modules that remain. The C partition is resized to include any free space that was created when the service partition was resized. This option increases space for other data storage and keeps the client-side recovery mechanism intact. This option also extends installation time.
- **Delete all:** All modules, maps, and files in the service partition are removed, the service partition is deleted, and the C partition is resized to include the space previously used by the service partition. Although this option increases space for data storage, the client-side recovery mechanism is lost. This option also extends installation time.

You can change the setting anytime during the map development process. To change the setting, open the Base Map window, select the root entry, select the Options tab, and select the appropriate radio button in the Post-Install Action field.

Note: If you intend to install IBM Rapid Restore PC as an application in your image, be aware that IBM Rapid Restore PC uses the service partition to store backup images. Using either the **Delete all** or **Delete some** setting will prevent the Rapid Restore PC program from operating if the program is installed as part of the normal silent-install installation process. If you need to use either of these settings, install the Rapid Restore PC program after the installation process is complete. (Rapid Restore PC is not part of the ImageUltra Builder program. However, it is provided with many IBM personal computers and is also available as a download from the IBM Web site.)

Enabling the network-sync function

The network-sync function is used to check the repository for updated maps. The check will occur twice if the target computer is connected to the network:

- **At the beginning of deployment:** If a down level map is selected, the network-sync function will locate the latest map and use it in place of the down-level map.
- **At the beginning of installation:** This check is most valuable when the installation process takes place at a later date than the deployment process. In this case, maps that have been updated or replaced since the original deployment process took place are used. In some cases, this check might result in the deployment of a complete new Smart Image.

In order for a network-sync operation to locate a replacement map, two conditions must be met:

- The network-sync function must be enabled in the base map.
- The network-sync settings table (accessible through the ImageUltra Builder Tools menu) must contain an entry defining the deployed map and the replacement map.

As you create your base map, the New Map Wizard enables you to designate whether you want to enable the network-sync function. You can change the network-sync settings anytime during the map development process. To change the settings, open the Base Map window, select the root entry, select the Options tab, and mark or unmark the Use Network Sync checkbox.

Defining user information

The User Information tab in base maps enables you to define default values for user-specific settings, or prompt for user-specific settings near the beginning of the

installation process, or both. You can define the user information settings and prompts anytime during the base-map development. For detailed information about using this feature, see Chapter 5, “Map construction overview” on page 33.

Deploying Smart Images to target computers

Important: Before deploying a Smart Image to a real work environment, test the Smart Image to ensure that it works as expected.

You can deploy Smart Images to target computers using either of two methods:

- Direct deployment from *Distribution CDs*: This method copies all modules defined in the selected base map and driver map to a set of CD-R or CD-RW discs. The first disk of the set is bootable. When started in the target computer, the CD copies the modules from the CDs to the service partition of the target computer and prepares the target computer for the installation process. If a service partition does not exist at the start of the deployment process, one is created.

To create a set of stand-alone *Distribution CDs*, you use the ImageUltra Builder Deploy tool to select the base map, driver map, and location where the CD files are to be stored. The Deploy tool then creates individual folders for each CD in the location specified. Then, you can use your CD-RW software of choice to create the CDs. Because CD 1 must be bootable, make sure you follow the instructions provided in the ImageUltra Builder help system for creating CDs for CD deployment.

- Network deployment: This method uses a bootable, DOS-based diskette to establish a network connection between the target computer and the ImageUltra module repository. Once the connection is established, you select which base map and driver map to use. All of the modules defined in the map are downloaded to the service partition of the target computer, and the target computer is prepared for the installation process. If a service partition does not exist at the start of the deployment process, one is created.

You use the ImageUltra Builder Deploy tool to create a network-deployment diskette. The Deploy tool prompts you for any information required for the target computer to connect to the repository (such as the path to the repository, IP address, user name, and password), and then builds the diskette. If the target computer does not have a diskette drive, you can use your CD-RW software to make a bootable CD using the files on the diskette as source. Because this CD is bootable, make sure you follow the instructions in the ImageUltra Builder help system for creating a Network Deployment CD.

The advantages of CD deployment are speed and the flexibility of not having to be dependent on a network connection. The advantage of network deployment is the assurance that the latest maps and modules will be installed if the network-sync function is enabled.

The division between the deployment and installation is a single boot cycle. In many cases, deployment is done in one location and installation takes place in another location, typically at the end-user’s work area. When deployment is complete, you can shut down the target computer and prepare the computer for delivery to the end-user’s work location.

Note: Some large enterprises that use drive-duplication technology for mass production might want to replicate the drive at the end of the deployment process, prior to installation. In order for this drive-duplication process to work successfully, you must install the IBM Boot Manager on each drive

after the drive has been duplicated. You can get the IBM Boot Manager installation tool (BMGRINST.BAT) from the TOOLS\BMGR folder of your repository. You must run BMGRINST.BAT during a DOS session.

Installing images on the target computer

The first boot cycle following the end of deployment starts the installation process. The following sequence of events takes place during the installation process.

1. An installation menu is displayed on the screen of the target computer. The content and behavior of this menu is controlled by the base map. The person performing the installation makes selections from the menu. The installation program collects modules associated with the selected menu items until all required modules are accumulated.
2. The accumulated operating-system modules and application modules are decompressed and copied to the C partition.
3. The device-driver modules defined in the driver map are decompressed and copied to the appropriate folder in the C partition.
4. Depending on the type of image you are installing, either a complete Windows setup or Sysprep mini-setup operation begins. If you are installing an Ultra-Portable Image, a full Windows setup operation begins. If you are installing a Portable-Sysprep Image or a Hardware-Specific Image, a Sysprep-controlled mini-setup operation begins.

Note: If you add user information to your base map, any prompts that you defined take place near the beginning of setup. If you do not add user information to your base map, prompts for specific user information occur in the normal progression of the setup process. Not taking advantage of the user-information feature might necessitate that an installer be physically present at the target computer longer.

5. Upon the successful completion of the operating-system setup, the setup process begins for each application program that was deployed as a module. The installation order is determined by the *Install Hook* and *Install Slot* selections defined for that specific application.

Note: The *Install Hook* and *Install Slot* are defined in the Options tab of the Application window associated with the application module. For more information about the Install Hook and Install Slot attributes, see the ImageUltra Builder help system.

6. When all applications have been installed, the installation process is complete. If you specified in the base map that the service partition is to be deleted or resized, the operation will take place at this time.

Advanced functions

There are two optional, advanced functions that you can use to enhance the image process: filters and utilities.

Filters

Filters automate a decision during the installation process. You typically use filters to obtain hardware-related information from a target computer to determine if a specific module or set of modules should be installed. The ImageUltra Builder program provides filters that check the machine type, machine type and model number, hardware platform (mobile or desktop), and whether or not a computer is an IBM computer. You can also develop your own filters if needed.

Filters are DOS-based programs that run during the installation process. In most cases filters query the BIOS of the target computer, but they can query other devices that store identification information in a ROM or EEPROM module. For example, you might want to check for the presence of a PCI modem before installing a modem-related application program or you might want to check for a specific machine type to determine which device drivers should be installed.

You can implement filters two different ways:

- Assign a filter to a menu item in either a base map or driver map by using the Filter tab attributes for the menu item.
- Assign a filter directly to an application or device-driver module by using the Filter-tab attributes for that specific module.

Using filters enables you to develop one map that works across many computer types. For information about creating your own filters or information about using the filters provided with the ImageUltra Builder program, refer to the ImageUltra Builder help system.

Utilities

Utilities are DOS-base EXE, BAT, or COM files that run during the installation process. For example, an installer might want to run CHKDSK to determine the size and condition of a hard disk before any files get installed. Utilities are copied to the service partition of the target computer during deployment, run during the installation process, but never get copied to the active partition of the target computer.

The ImageUltra Builder program does not provide any utilities. If you want to include utilities, you will have to build your own utility modules.

To use a utility, you assign a utility module to a base-map menu item.

For information about creating your own utilities, see Chapter 7, “Preparing source files for modules” on page 55.

Chapter 3. Installing the ImageUltra Builder program

Before using the information in this chapter, you must have a good understanding of the image-creation process and be familiar with some of the terminology associated with the ImageUltra Builder program. Take the time to read Chapter 1, “ImageUltra Builder capabilities” on page 1 and Chapter 2, “An overview of the image process” on page 9.

Then, review the minimum requirements, prerequisites, considerations, and limitations in this chapter before you install the ImageUltra Builder program.

Minimum requirements

The following information provides the minimum requirements for the ImageUltra Builder console (the computer on which you install the ImageUltra program):

Operating system: Windows XP Professional Edition or Windows 2000

Disk Space: 100 MB of available disk space to install the ImageUltra Builder program. Additional storage is required for the repository, which can be on the ImageUltra Builder console, on a shared disk of another computer, or on a server. The amount of disk space required for the repository depends on the size and number of modules that will be stored in it. Repositories cannot span drives, so make sure you have enough storage for future growth.

Removable-media drives:

- CD or DVD drive for installation
- Diskette drive for the creation of *Network Deployment Diskettes* and *Import Diskettes*
- CD-RW drive for CD deployment

Minimum networking speed: 10 Mbps; Ethernet or token ring

Prerequisites

The following prerequisites must be followed for the ImageUltra Builder to function correctly and for you to create and distribute Smart Images legally:

- Install the ImageUltra Builder interface on a local computer, not a server.
- During the installation of the ImageUltra Builder program, you can establish the repository either locally on the ImageUltra Builder console, on a shared disk of another computer, or on a server. The repository must be accessible to the ImageUltra console, the source computer for import operations, and any target computer that will have a Smart Image deployed over the network.
- Ensure proper licensing:
 - The ImageUltra Builder license agreement applies to the ImageUltra Builder program only. Read the license carefully during the installation process. All other licenses, warranty service, and support for non-IBM products, such as Microsoft Sysprep, MS-DOS, Symantec Norton Ghost, and PowerQuest DeployCenter, must be acquired from the third party, not IBM. IBM makes no representations or warranties regarding non-IBM products. If you need to

review the license after installation, click **Help** in the ImageUltra Builder menu bar, and then click **View the license**.

- IBM requires that you obtain a per-seat license for each computer on which a Smart Image is deployed. For information about ordering licenses, contact your IBM marketing representative.
- You are solely responsible for obtaining any licenses required to distribute operating systems, application programs, and any other licensed software.

Considerations

Consider the following information when using the ImageUltra Builder program:

- When building driver modules, use only certified Microsoft WHQL device drivers. Device drivers must have an .INF extension to install correctly. Building modules with unsigned device drivers might cause Windows to display information messages during installation, thereby preventing an image from installing in an unattended manner.
- Research compatibility between operating systems and versions of Symantec Norton Ghost and PowerQuest DeployCenter before you begin creating Portable-Sysprep Images or Hardware-Specific Images.
- If you intend to use stand-alone *Distribution CDs* for deployment, the ImageUltra Builder program creates folders containing the files required for the CDs that most CD-RW software can use as source to create CDs. However, the first CD of each set is bootable. Ensure that your CD-RW software can create bootable CDs and follow the instructions in the ImageUltra Builder help system for creating CDs for CD deployment.
- Although only one repository is sufficient for most environments, you can create additional repositories using the New Repository Wizard. Before creating multiple repositories, it is a good idea to develop a repository strategy. Consider the following:
 - When you create a new repository, make sure you assign it a unique ID that has not been used for any other repository in your company. If you have duplicate IDs, and import modules from a source computer that has a Smart Image that was deployed from a different repository, there is a chance that modules can get overwritten.
 - All modules defined in a map must be in the same repository as the map.
 - The driver map and base map used for deployment must be in the same repository.
 - You cannot move or copy modules from one repository to another.
- Consider using a separate server to back up your module repository.
- It is possible to create an image that contains modules that collide, or fail to interact correctly. For example, an image that contains two antivirus program modules might not install correctly. As a result, the image might not work. One way to prevent the failure is to read the information provided with devices, device drivers, and applications before beginning to create modules.

Limitations

The following limitations apply to functionality of the ImageUltra Builder program:

- To deploy an Ultra-Portable Image without a driver map, the target computer must be HIIT enabled. For a list of IBM HIIT-enabled computers, visit the ImageUltra Builder Web site at <http://www.pc.ibm.com/qtechinfo/MIGR-44316.html>.

- Ultra-Portable Images and Portable-Sysprep Images can be based on Windows 2000 and Windows XP only. Hardware-Specific Images can be based on any Windows operating system.

Installing the ImageUltra Builder

Use the following procedure to install the ImageUltra Builder program:

1. Insert the ImageUltra Builder CD into the CD drive of the computer that will be used as the ImageUltra Builder console. In most cases the install program will start automatically. If the install program does not start automatically, complete the following procedure:
 - a. Click **Start**.
 - b. Click **Run**.
 - c. Type *d:\SETUP.EXE* (where *d* is the letter of your CD drive).
 - d. Click **OK**.
2. Follow the instructions on the screen.

Getting help after installation

The ImageUltra Builder program has a built-in help system to assist you with most on-screen tasks. You can access the help system using any of the following methods:

- Click the **Help** button provided on the Main window tool bar
- Click **Help** on the Main window menu bar, and then click **View help topics**.
- Press **F1**.

If you need additional assistance, see Chapter 8, “Getting help and support” on page 73.

Chapter 4. Working with the ImageUltra Builder interface

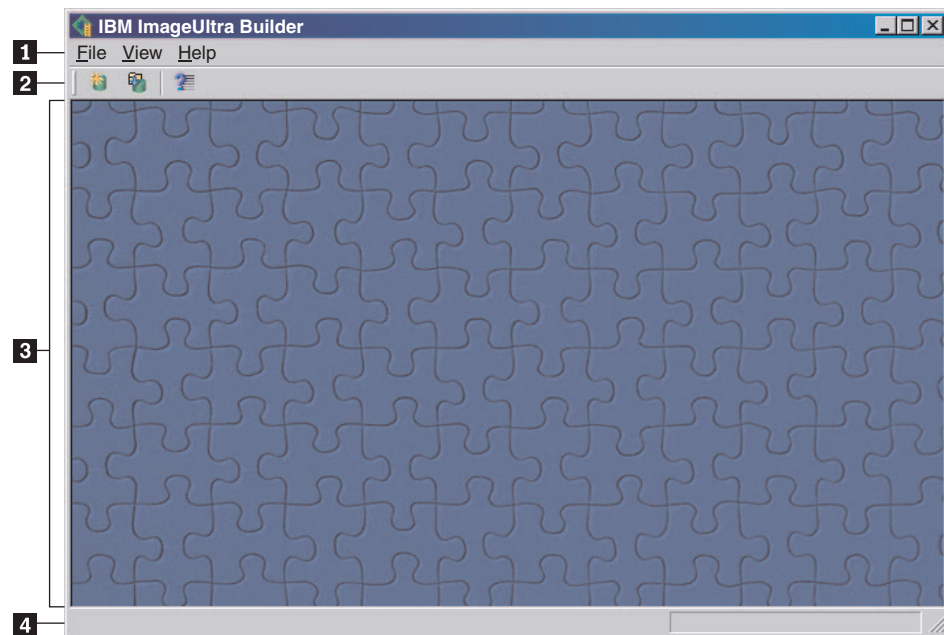
The ImageUltra Builder interface provides several important functions:

- It provides a staging area to prepare and build modules for Smart Images.
- It enables you to manipulate modules within a repository, create additional repositories, and navigate among all defined repositories.
- It provides direct access to tools used to import modules, build modules, promote modules, and deploy Smart Images.
- It enables you to develop base maps and device-driver maps.

This chapter explains the major components of the Image Ultra Builder interface, where the components are located, and how to navigate within the interface.

Layout of the Main window

The Image Ultra Builder interface is made up of a Main window and several child windows. The Main window is divided into the following areas:



1 Menu bar **2** Toolbar **3** Workspace **4** Status bar

- **Menu bar:** The menu bar selections change depending on the type of child window that is currently active in the workspace. You can make menu-bar selections using a mouse or keyboard. For keyboard navigation, the **Alt** key toggles the focus between the menu bar and the active child window. To make menu-bar selections using the keyboard, press **Alt** to shift the focus to the menu bar, and then use your cursor-control keys and Enter key to make selections.
- **Toolbar:** The toolbar contains shortcuts for a number of functions. The contents of the toolbar changes depending on the type of child window that is currently active in the workspace. The state of each shortcut shown in the toolbar changes depending on the item selected within the active child window. When you position the cursor over any of the toolbar icons, a short description of the function is provided. You can choose to hide or display the toolbar through the








menu-bar View menu. You can make toolbar selections using a mouse or keyboard. Any function shown in the toolbar also can be made through menu-bar selections.








- **Workspace:** The workspace displays all child windows generated by the ImageUltra Builder program. The active window controls the contents of the menu bar and toolbar. You can have multiple windows open at the same time. For keyboard navigation Ctrl+F6 toggles among the open windows.
- **Status bar:** The Status bar is located at the bottom of the Master window and provides two major functions:
 - It provides a status of any operation currently being performed by the ImageUltra Builder program, such as a build or import operation. When no operation is in progress, the status bar shows Ready.
 - When the cursor is positioned over a drop-down menu item, the Status bar provides quick help for that item.













You can choose to hide or display the Status bar through the menu-bar View menu.









Conventions used within the interface



The following table displays the various icons used in the ImageUltra Builder program.

| Icon | Name | Description | Keyboard Shortcut |
|---|---------------|---|-------------------|
|  | Save | Saves changes in the active map or module window. | Ctrl+S |
|  | Cut | Cuts the currently selected object. | Ctrl+X |
|  | Copy | Copies the currently selected object to the clipboard. | Ctrl+C |
|  | Paste | Pastes an object from the clipboard. | Ctrl+V |
|  | Build module | Builds the currently selected module. | |
|  | Promote | Promotes the currently selected map or module to the gold state. | |
|  | New OS Module | Launches the New Module Wizard to create new operating system modules. This icon is available in the Repository Window toolbar only when the Operating System category is selected in the Components area of the Repository Window. | |

| Icon | Name | Description | Keyboard Shortcut |
|---|--------------------------|---|-------------------|
|  | New Base Map | Launches the New Map Wizard to create new base maps. This icon is available in the Repository Window toolbar only when the Base Map category is selected in the Components area of the Repository Window. | |
|  | New Driver Map | Launches the New Map Wizard to create new driver maps. This icon is available in the Repository Window toolbar only when the Driver Map category is selected in the Components area of the Repository Window. | |
|  | New App Module | Launches the New Module Wizard to create new application modules. This icon is available in the Repository Window toolbar only when the Application category is selected in the Components area of the Repository Window. | |
|  | New Device Driver Module | Launches the New Module Wizard to create new device driver modules. This icon is available in the Repository Window toolbar only when the Device Driver category is selected in the Components area of the Repository Window. | |
|  | New Filter Module | Launches the New Module Wizard to create new filter modules. This icon is available in the Repository Window toolbar only when the Filter category is selected in the Components area of the Repository Window. | |
|  | New Utility Module | Launches the New Module Wizard to create new utility modules. This icon is available in the Repository Window toolbar only when the Utility category is selected in the Components area of the Repository Window. | |
|  | New Repository | Launches the New repository Wizard to create a new repository. This icon is available in the main window toolbar when the Repository Window is active. | |

| Icon | Name | Description | Keyboard Shortcut |
|---|-----------------------|--|-------------------|
|  | Add Filter | Assigns a filter to a base-map menu item, an add-on operating-system module, an application module, or a device-driver module. This icon appears on the Filter tab associated with a base-map menu item, an add-on operating-system module, an application module or a device-driver module. | |
|  | Import | Launches the Import tool to create import media or to import modules from CD. | |
|  | Deploy | Launches the Deploy tool to create a <i>Network Deployment Diskette</i> , <i>Network Deployment CD</i> , or standalone <i>Distribution CDs</i> . | |
|  | Network Sync Settings | Opens the Network Sync Settings window so you can define replacement maps for maps that have already been deployed. | |
|  | Open Repository | Enables you to open a repository or change to a different repository. | Ctrl+O |
|  | Delete | Deletes the selected map, module, menu item, or link. | Del |
|  | Views | Enables you to change the view of the repository to use large icons, small icons, a list, or details. | |
|  | Move up | Moves a menu item up in base maps and driver maps | Alt+Up Arrow |
|  | Move down | Moves a menu item down in base maps and driver maps | Alt+Down Arrow |
|  | Module category | Used in the left pane of the Repository window to categorize operating-system modules, application modules, device-driver modules, filter modules, and utility modules. | |
|  | Maps category | Used in the left pane of the Repository window to categorize base-maps and driver maps. | |
|  | Base map | Used in the repository Window to identify an object as a base-map. When this icon is gold, the base map is in the gold state. | |

| Icon | Name | Description | Keyboard Shortcut |
|---|-------------------------|--|-------------------|
|  | Driver map | Used in the repository Window to identify an object as a driver map. When this icon is gold, the driver map is in the gold state. | |
|  | Operating-system module | Used in the Repository window, Base Map window, and Driver Map window to identify an object as an operating-system module. When this icon is gold, the operating-system module is in the gold state. | |
|  | Application module | Used in the Repository window, Base Map window, and Driver Map window to identify an object as an application module. When this icon is gold, the application module is in the gold state. | |
|  | Device-driver module | Used in the Repository window and Driver Map window to identify an object as a device-driver module. When this icon is gold, the device-driver module is in the gold state. | |
|  | Filter module | Used in the Repository window, Base Map window, and Driver Map window to identify an object as a filter module. When this icon is gold, the filter module is in the gold state. When a filter is associated with a menu item or other module, a miniature version of this icon overlays the menu-item icon or module icon. | |
|  | Utility module | Used in the Repository window and Base Map window to identify an object as a utility module. When this icon is gold, the utility module is in the gold state. When a utility is associated with a menu item or other module, a miniature version of this icon overlays the menu-item icon or module icon. | |
|  | Help | Opens the help system. | F1 |
|  | Container | Used in the Repository window, Base Map window, and Driver Map window to identify an object as a container. This icon overlays other module icons. For example, if this icon overlays a device-driver module icon, that object is a device-driver container. | |

| Icon | Name | Description | Keyboard Shortcut |
|---|-----------|---|-------------------|
|  | Menu item | Used in Base Map windows and Driver Map windows to identify an object as a menu item. | |
|  | Link | Used in Base Map windows to identify an object as a link. | |

Module repository

During the ImageUltra Builder installation, the New Repository Wizard prompted you to define a location for your repository. Although a single module repository is sufficient for storing all of your modules, you can create additional repositories after the initial installation of the ImageUltra Builder program. When you open the ImageUltra program, the workspace automatically associates itself with the last module repository used and opens the appropriate Repository window. The name of the active repository is shown on the title bar of the Repository window.

To make a new repository, do the following:

1. Click **File**.
2. Click **New Repositories**.
3. Follow the instructions on the screen.

To open a repository or change to a different repository, do the following:

1. Click **File**.
2. Click **Open Repositories**. A list of recently accessed repositories is displayed.
3. Select the desired repository. If the repository you want to use is not listed, use the **Browse** button to navigate to the desired repository.

To change to a module repository that has been recently accessed, do the following:

1. Click **File**.
2. Click **Recent Repositories**.
3. Select the desired repository.

The Repository window

The Repository window is the starting point for every operation in the ImageUltra Builder program. You use the Repository window to view the contents of the repository, add map and module entries to the repository, open and modify maps and modules contained within the repository, build modules, create import media, and create deployment media.

The Repository window is divided into two panes: the left pane is the Components area and the right pane is the details area. The Components area provides a categorical listing of the maps and modules contained in the repository. By clicking on a category in the Components area, you can view all of the components in the repository that are associated with that category type. For example, if you click the Base Map category, all of the base maps in the repository will be displayed in the right pane.

Each module type has an icon associated with it. The module icons serve the following purposes:

- They provide a visual indicator for the type of module (operating-system module, application module, device-driver module, filter module, or utility module).
- They distinguish a container module from a standard module.
- They show whether or not a filter or utility is associated with a module.
- They indicate whether the module is in the test state or gold state. If the icon is gold, the module is in the gold state. When a module is in the gold state, the module is locked and you cannot make changes to it. If the icon is any other color but gold, the module is in the test state. When a module is in the test state, you can make changes to the module attributes as often as is needed.

Map icons serve the following purposes:

- They provide a visual indicator for the type of map (base map or driver map).
- They indicate whether the map is in the test state or gold state. If the icon is gold, the module is in the gold state.

Because creating a module entry in the repository and building the module are two separate operations, the ImageUltra Builder interface provides a means by which you can tell whether or not a module has been built. If the module entry in the repository is shown in grey, the module has not been built. If the module entry is shown in black, the module has been built.

These icon and module entry conventions also apply to all other windows used in the interface.

Map windows

The Components area of the Repository Window provides two map entries: Base Maps and Driver Maps. When you click **Base Maps**, a listing of all base maps that are currently in the selected repository is displayed in the right pane. When you click **Driver Maps**, a listing of all device-driver maps that are currently in the selected repository is displayed in the right pane. When you double-click a map in the right pane, a Map window opens for the selected map.

Map windows are divided into two panes. The left pane contains a tree structure that shows all of the components (menu items and modules) contained within the map. The right pane shows the attributes for each selected component. Within the Map window, you can do the following:

- Delete a menu item from the tree structure
- Insert a menu item in the tree structure
- Insert a module in the tree structure
- Move a menu item up or down in the tree structure
- Use the Filter and Utility tabs to assign filters and utilities to menu items in a base-map tree structure
- Use the Filter tab to assign filters to menu items in a driver-map tree structure
- Use the User Information tab in base maps to define user-specific information or establish prompts for user information
- Use the Submenu tab in base maps to control the visual appearance of the menus to be displayed on a target computer

Module windows

The Components area of the Repository window provides five module category entries: Operating Systems, Applications, Device Drivers, Filters, and Utilities. When you click on any of these categorical entries, a listing of all associated modules in the selected repository is displayed in the right pane. When you double-click a module entry in the right pane, an attribute window opens for the selected module. Within the attribute window, you can view or define certain attributes, such as the language, location of the source files, and associated filters. The attributes that can be viewed or modified vary depending on the category of module selected. The five types of module windows are:

- Operating System window
- Application window
- Device Driver window
- Filter window
- Utility window

The attributes define key information about an associated module that affect content, behavior, and installation. The attributes must be filled in completely before you build a module. If a module is imported from a source computer, all required attributes are already filled in. However, it is a good idea to open the associated attributes window and review the attribute values to ensure that they are suitable for your specific implementation.

When you build a module from scratch, you use the New Module Wizard to create the entry in the repository and fill in the attributes. You must complete the associated attributes before you can build the module.

Modifying attributes

By default, when you import or create a map or module, it is placed in the repository in the test state. As long as a module is in the test state, you can modify the attributes and source files and rebuild the module as many times as is needed. As long as a map is in the test state, you can modify its tree structure and attributes. Once a map or module has been promoted to the gold state, it is locked and you can no longer make changes to it.

It is imperative that you test Smart Images while the maps and modules are in the test state. Once you are comfortable that all of the maps and modules are working correctly, *promote* them to the gold state to ensure that they are not accidentally modified or overwritten in the future.

You can determine whether a map or module is in the test state or gold state by using either of the following methods:

Method 1: Icon identification

To determine whether a module is in the test state or gold state using the icon identification method, do the following:

1. Open the appropriate Repository window.
2. In the Components area, click the appropriate map or module category.
3. In the right pane, locate the desired map or module.
 - If the icon for the module is any color other than gold, the module is in the test state.

- If the icon for the module is gold in color, the module is in the gold state.

Method 2: Properties identification

This method is useful if you have difficulty determining color. To determine whether a module is in the test state or golden state using the properties-identification method, do the following:

1. Open the appropriate Repository window.
2. In the Components area, click the appropriate map or module category.
3. In the right pane, right-click the appropriate map or module and then select Properties. If the Gold checkbox located near the bottom of the tab is marked, the map or module is in the gold state. If the Gold checkbox is not marked, the map or module is in the test state.

Tools

The ImageUltra Builder program makes several tools available from the menu bar to assist with the image building process:

- **Build Module tool**

The Build Module tool enables you to build a module from source files. A module must be highlighted in the Repository window before this tool can be selected.

- **Promote tool**

The Promote tool enables you to promote a map or module from the test state to the gold state. A map or module must be highlighted in the Repository window before this tool can be selected.

- **Import tool**

The Import tool enables you to create an *Import Diskette* or *Import CD*, or to import modules directly from an IBM Recovery CD. The import media prepares the service partition for viewing, provides an interface for selecting modules, and establishes a network connection to the repository. A Repository window must be open before this tool can be selected.

- **Deploy tool**

The Deploy tool enables you to create a *Network Deployment Diskette*, *Network Deployment CD*, or stand-alone *Distribution CDs*. The deployment diskette and CD establish a connection to the repository, provide an interface for selecting the desired maps, and deploy the appropriate maps and modules to the service partition of the target computer. The standalone *Distribution CDs* contain all of the required maps and modules, and deploy them to the target computer without using a network. A Repository window must be open before this tool can be selected.

- **Network-Sync Settings tool**

The Network-Sync Settings tool enables you to define replacement maps for maps that have already been deployed. A Repository window must be open before this tool can be selected.

- **Get IBM Customization Program**

This tool provides instructions on how to locate and copy the IBM Customization Program (IBMCP.EXE) to a diskette or shared drive. The IBM Customization Program is required when building the source for Portable-Sysprep Images. A Repository window must be open before this tool can be selected.

Chapter 5. Map construction overview

Creating a fully functional map involves three fundamental steps:

- Creating a map entry in the module repository
- Defining attributes for a map
- Creating a map tree structure

This chapter focuses on the concepts behind creating a map tree structure and defining certain attributes that affect map behavior. For step-by-step instructions on creating a map entry, defining attributes for a map, or any other process described in this chapter, refer to the ImageUltra Builder help system.

Creating a base map tree structure

The New Map Wizard enables you to create a new base map from scratch or to create a new base map using an existing base map as a template. The New Map Wizard creates the new map entry in the repository, assigns the map a file name, and places the map in the correct location of the repository. To access the New Map Wizard for base maps, click **New** in the Main window menu bar, and then click **Base Map**.

If you create a new map based on an existing map, you simply make the necessary changes to the tree structure and attributes. If you create a new map from scratch, you are responsible for:

- Inserting *all* menu items
- Inserting *all* modules
- Defining *all* menu-item attributes

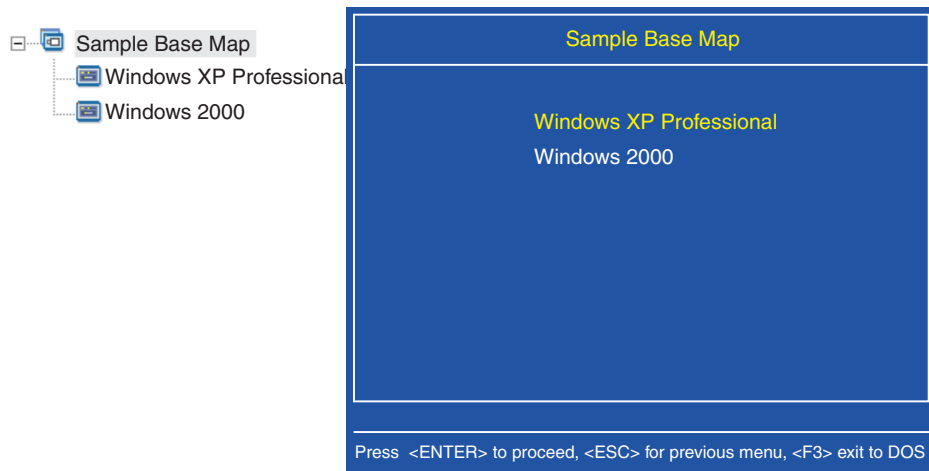
Inserting menu items in a base map

Each base map must contain at least one menu item. Menu items are responsible for the menu choices displayed at the target computer during the installation process. Menu items also determine the hierarchy within the base map, meaning that some menu items are subordinate to other menu items. Each menu item listed in a base map directly corresponds to a choice made at the target computer during the installation process. When a menu item is chosen during the installation process, that action directs the installation program to accumulate modules associated with that menu item or to display a new menu of subordinate menu items. In some cases, both actions occur simultaneously. When the person performing the installation makes the final selection from the final menu, the installation program installs all of the accumulated modules. The accumulation of menu choices during the installation process transforms the Smart Image into a customized image for a specific target computer.

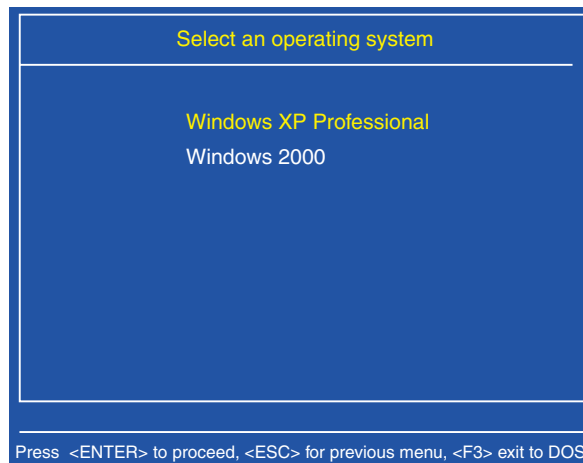
As you develop your tree structure, you might want think of the menu items as an outline for the installation process associated with the entire Smart Image.

Assume that you want to construct a base map that gives the installer a choice of installing Windows XP Professional or Windows 2000. The following illustration shows a base-map tree structure with two menu items (Windows XP Professional

and Windows 2000) before modules or submenu items are added. Also shown is the corresponding default installation screen as it would appear at the target computer.



Notice how the menu items in the base map directly correlate to the menu items on the installation screen at the target computer. Optionally, you can use the Submenu Items tab to override the default title and replace it with an instruction to clarify the action required at the target computer. For example, you might want to replace the default title with “Select an operating system” as shown in the following illustration.

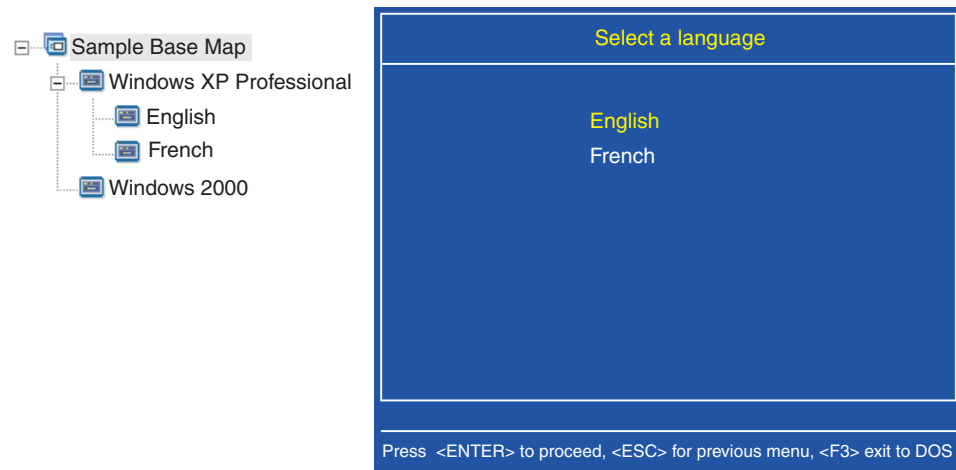


In this example, the Sample Base Map entry was highlighted in the map, the Submenu Items tab was selected, and “Select an operating system” was typed in the Title field. The Submenu Items tab is present only if the selected menu item has subordinate menu items.

Now take the map one step further and give the installer language choices for the Windows XP operating system by adding more subordinate menu items to the base map.

The following illustration shows the tree structure with two subordinate menu items (French and English) under the Windows XP menu item. Also shown is the

corresponding installation screen that the installer will see at the target computer after selecting Windows XP from the first menu.



Notice how the menu items in the base map directly correlate to menu items on the installation screen at the target computer, but the title contains an instruction instead of the default title. In this example, the Windows XP Professional entry was highlighted in the map, the Submenu Items tab was selected, and “Select a language” was typed in the Title field.

You can continue building on this concept and add as many menu items and branches as needed to obtain a menu structure that gives you the desired result.

Inserting modules in a base map

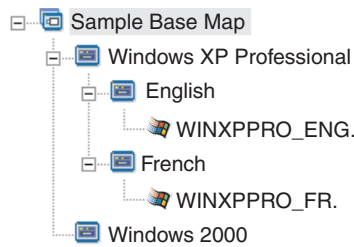
Once you have established the hierarchy of a base map with menu items, you can start populating the map with operating-system modules and application modules.

At least one operating-system module must be associated with each branch of a map tree. Depending on the type of operating-system module associated with a given branch, you might have limitations as to whether you can associate application modules with that branch:

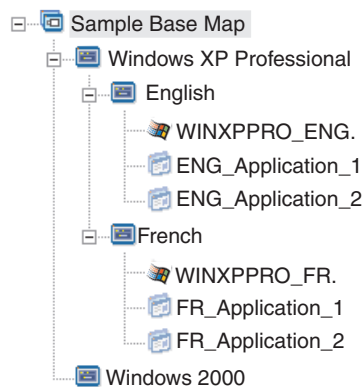
- If the operating-system module is a Hardware-Specific Image, you cannot associate application modules with that branch. If you attempt to associate application modules with that branch, the application modules will be ignored during the installation process. Only those applications contained within the Hardware-Specific Image itself will be installed.
- If the operating system module is an Ultra-Portable Image or Portable-Sysprep Image, you can associate application modules with that branch.

You insert modules in strategic locations of the base map. Module names never appear on the installation screen at the target computer. Typically, modules are associated with specific menu items so the installation of the modules is contingent upon the selection of a given menu item. As the person performing the installation progresses through the menu structure, the modules to be installed accumulate until the final selection is made from the final menu, at which time, all of the accumulated modules are installed.

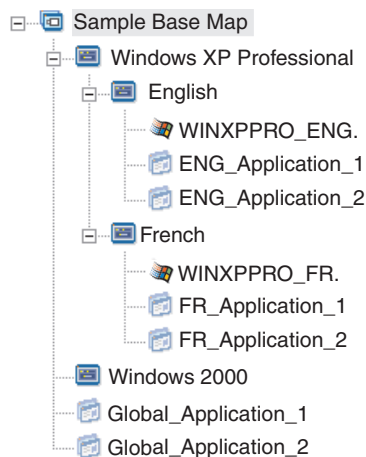
Using the map that was used as an example earlier, the appropriate operating-system modules would be inserted under the French and English menu items as shown in the following illustration.



If the operating-system modules are either Ultra-Portable Images or Portable-Sysprep Images, you can also insert application modules as shown in the following illustration.



There is one important exception to associating modules with menu items. If you want to install one or more modules that are global in nature (that is, modules to be installed on every target computer using this map regardless of what menu items are selected), you can insert those modules at the root level of the map as shown in the following illustration.

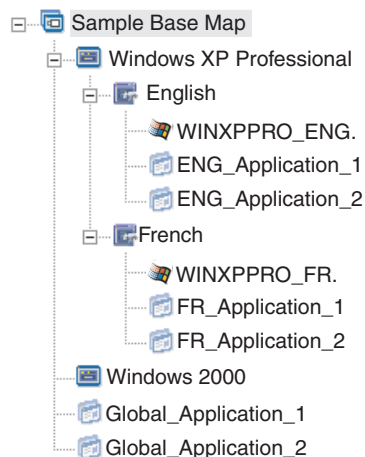


Using utilities in a base map

The use of utilities is optional. Utilities are DOS applications that run before any part of an image is installed on a target computer. For example, you might want to run CHKDSK to determine the current condition of a target computer hard disk or PC-Doctor for DOS to determine the overall condition of a target computer before installing the actual image. One important difference between a utility and an application module is that a utility resides in the service partition only and never gets installed on any other active partition.

Utilities are assigned to menu items in the tree structure of a base map. Each menu item can have a maximum of one utility defined in its attributes. There are two approaches to assigning utilities to menu items:

- **Approach A:** Assign utilities to menu items that already contain modules as submenu items. In this approach, the utility is automatically selected when the installer selects the existing menu item. It is transparent to the installer that a utility is being selected. The following is a sample tree structure with utilities associated with the French and English menu items. If you compare this tree structure to the sample tree structure used earlier, you will notice that the tree structure itself does not change, but the icon associated with the menu item does change to show that a utility is associated with it. Notice the small hammer that is now part of the icons for the English and French menu items.

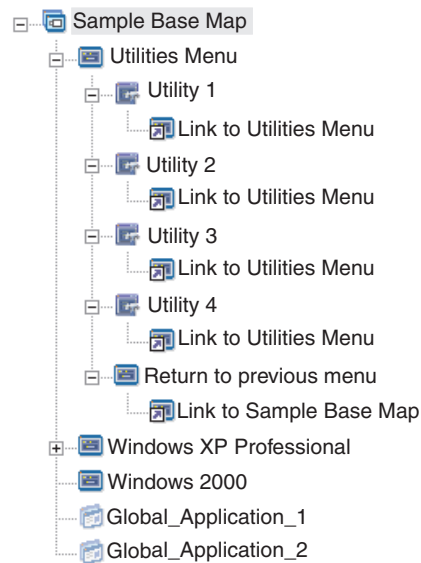


When using this approach, you use the menu-item Utility tab to set the execution state to either of the following settings:

- **Run now:** The utility is run immediately when the menu item is selected at the target computer. This is the default setting.
 - **Queued:** The utility is queued to run after the final selection is made from the final menu in the sequence. All utilities that are queued are run before any files get installed.
- **Approach B:** Create menu items specifically for each utility and assign a utility to each menu item. This approach creates a custom utility menu from which the person performing the installation can manually select the utility needed. The utility must run at the time the menu item is selected; therefore, when you develop your map, you must set the attribute for each utility menu item to the **Run now** setting. If you set the utility-menu-item attributes to the **Queued** setting, an error will occur and the installation process will be terminated. Additionally, a custom utility menu requires a unique structure using links to return to the base menu or to the utility menu after a utility has been run. There are five steps required to develop this unique structure:

1. Create a menu item specifically for the utilities menu.
2. Create subordinate menu items for each utility
3. Associate a utility with each menu item in the utilities menu.
4. Create links under each utility menu item. Each link provides a path after the utility runs that would not otherwise exist.
5. Create a subordinate menu item that enables the person performing the installation to return to the previous menu and create a link under it to the previous menu.

The following illustration shows the structure associated with implementing a custom utility menu.



Using filters in a base map

The use of filters in a base map is optional. The purpose of a filter in a base map is to query the target computer for specific hardware information and then use this information to determine whether a specific menu or menu item should be displayed during the installation process. The hardware information is typically obtained from the computer BIOS. The filters provided with the ImageUltra Builder program enable you to query for the machine type, machine type and model number, platform (desktop or mobile), and whether or not the target computer is an IBM computer. When using filters in a map, filters are assigned to menu items, not to modules. The behavior of the filter is controlled through the Filter tab of the menu-item attributes. Multiple filters can be assigned to a menu item; however, if you assign multiple filters to a menu item, you must define whether only one filter needs to be satisfied or if all filters need to be satisfied for the menu item to be displayed.

The following are a few examples of how you can use a filter in a base map.

Example 1: You have a set of applications that are unique to a mobile computer and you want to use a filter to facilitate the installation of these applications only if the target computer is a mobile computer.

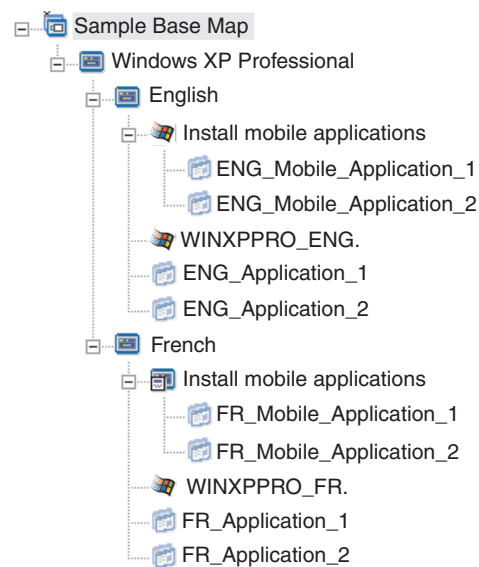
There are several ways to achieve these results based on the current tree structure in your map. For the purpose of this example, you would create a menu with a single menu item at the end of a branch using the following procedure:

1. Create a menu item at the end of one of your branches and name it “Install mobile applications.”
2. Insert all of your mobile-unique applications under the “Install mobile applications” menu item.
3. Double-click the “Install mobile applications” menu item.
4. In the Filter tab, click the **Insert Filter** icon, select the **IBM Hardware Platform Check filter** to determine mobile or desktop, type in the “isMobile” parameter, and then click **OK**.
5. Click the **Save** icon in the Main window toolbar.

When this map is used, the “Install mobile applications” menu item will appear at the target computer only if the target computer is a mobile computer.

Note: Because “Install mobile applications” is the only item in the menu, you can hide the menu and have the mobile-unique applications install automatically by highlighting the menu item one level *above* “Install mobile applications” and checking the **Auto-Select** check box in the Submenu tab. In this example, you would check the Auto-Select checkbox in the Submenu tab for the English and French menu items.

The following shows a base map using this approach.



Example 2: You are creating a map that includes a series of operating-system modules that were built from Hardware-Specific Images, and you want to use a filter to display only the menu items that are appropriate for the target computer.

If you did not use a filter, the person performing the installation would have to make a selection from a menu listing all of the machine types that this map supports. By using a filter, the only choices that are presented at the target computer are choices specific to the target computer.

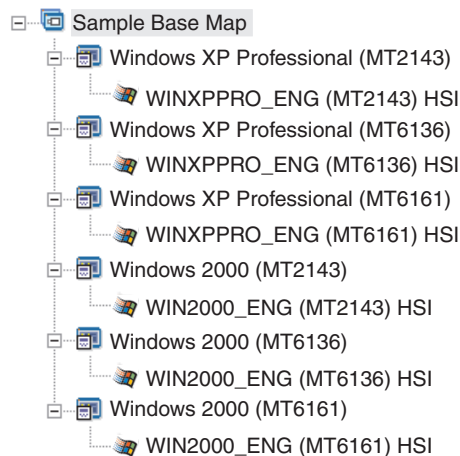
To create a map that includes filters for this example, you would complete the following procedure:

1. Create a menu item below the root entry of the tree for each operating-system module you will include in your map. Give each menu item a name that includes the machine type and operating system.
2. Insert the appropriate operating-system module under each menu item.
3. Do the following for each menu item:
 - a. Double-click the menu item.
 - b. In the Filter tab, click the **Insert Filter** icon, select the Model Check filter, type the appropriate machine type in the parameter field (for example, 2143xxx, where 2143 is the machine type and xxx is a wildcard for all model numbers), and then click **OK**.
4. Click the **Save** icon in the Main window toolbar.

When this map is used on a computer having the machine type of 2143, the only selections that will appear at the target computer will be the menu items for the operating system module developed for that computer. If you had many operating-system modules in your map, but only one Windows XP module and one Windows 2000 module were developed for the target computer, those would be the only two choices shown at the target computer.

Note: If only one operating-system module meets the filter requirements, you can hide the menu and have the correct operating system module install automatically. To enable your map to do this in this example, you would highlight the *root entry* of your map and check the **Auto-Select** check box in the Submenu tab.

The following illustration shows a base map using this approach.



If the target computer is a machine type 2143, the only two menu items that would be displayed at the target computer would be:

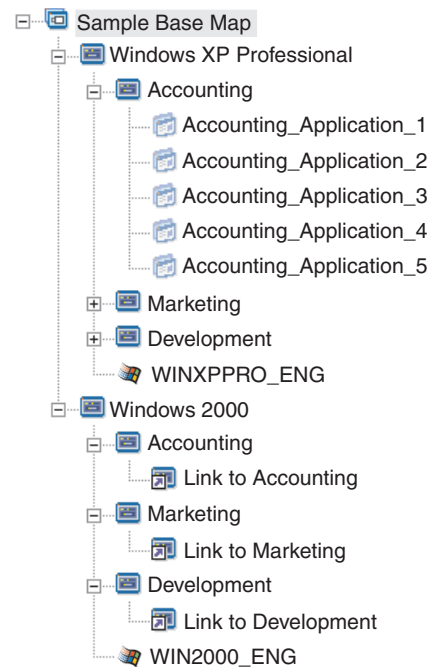
- Windows XP Professional (MT2143)
- Windows 2000 (MT2143)

Using links in a base map

A link is a type of shortcut that can be added to base maps only. Links are useful when you have the same menu structure in multiple places within the same map.

By using links, you can avoid the redundant task of making the same entries, corrections, or changes in multiple places. For example, assume you have a set of five core applications that get installed on most of the computers you maintain. Within your map, you can define those applications under a menu item once and then link to that menu item from elsewhere in the map when those five core applications need to be defined.

The following illustration shows an example of how links can be used in a base map.



You can also use links to create paths for utilities that get selected from a custom utility menu. For details, see “Using utilities in a base map” on page 37.

Defining user information in a base map

During Windows setup or Sysprep mini-setup, you are typically prompted to provide information for a number of user-unique settings. These settings include:

- Administrator password
- Computer name
- DHCP configuration (whether or not an IP address is to be obtained from a DHCP configuration)
- User’s full name
- Gateway
- IP Address
- Workgroup
- Organization name
- Subnet mask

The ImageUltra Builder program enables you to predefine default settings, or prompt the person performing the installation for specific user information near

the beginning of the installation process, or both. Thus, this feature can help minimize the amount of time that an installer needs to be present at the target computer during the installation process.

You can also define your own settings and prompts for other purposes. For example, you can prompt the installer for the office number where the computer is being installed and the telephone number for the user. Then, you can run your own software after installation to collect the office number, telephone number, and user's full name to create or append a company phone book.

User information is implemented in base maps on a map-by-map basis. All predefined user information and installer-provided user information is gathered during the installation process and stored in a file called PERSONAL.INI located in the root folder of the target computer.

To implement user information, do the following:

1. Open the Base map window.
2. Click the root entry of the tree structure.
3. Click the User Information tab.
4. Click the **Add User Information** icon located in the User Information tab, and fill in the appropriate fields. You can use the Information field drop-down menu to work with the typical user settings required by Windows, or type your own setting name in the Information field and define default values and prompts.
5. Click **OK**.
6. Repeat steps 4 and 5 for each setting.
7. Click the **Save** icon located in the Main window toolbar.

Controlling the network-sync function

When you create your base map, the New Map Wizard prompts you to define whether or not to enable the network-sync function. You can change the network-sync setting anytime during the map development process. To change the setting, do the following:

1. Open the Base Map window.
2. Click the root entry of the tree structure.
3. Select the Options tab.
4. Use the **Use Network Sync** check box to define the setting.

For more information about the network-sync setting, see "Enabling the network-sync function" on page 15.

Controlling the service partition behavior

When you create your base map, the New Map Wizard prompts you to define how the service partition on the target computer will be treated after the image is installed. You have three options: Delete none, Delete all, and Delete some. (See "Defining service partition behavior" on page 14 for more information about these settings.)

You can change the setting anytime during the map development process. To change the setting, do the following:

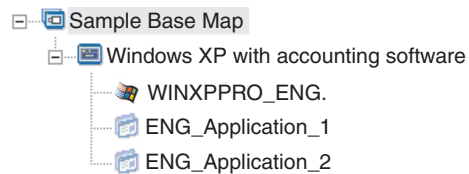
1. Open the Base Map window.
2. Select the Options tab.

3. Use the Post-Install Action field drop-down menu to change the setting.

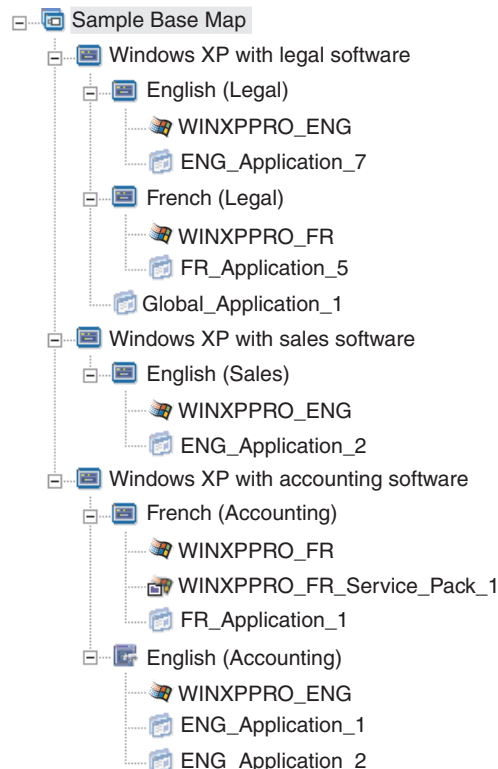
Simple and complex base maps

When you develop your base maps, you can develop simple maps that contain a small number of modules, have limited choices, and pertain to one group; or, you can develop complex maps that contain many modules, a larger number of choices, and apply to many groups.

A simple map is constructed to meet the needs of one specific group of users, who have similar requirements. For example, you can construct a simple map designed specifically for an accounting department that consists of one main menu item called “Windows XP with accounting software” and all of the required operating system and application modules inserted under it. The following illustration shows the tree structure associated with a simple map constructed solely for an accounting department.



A complex map is constructed to meet the needs of several groups of users, each of which have unique requirements. For example, you can construct a complex map that contains menu items for languages, operating systems, and department names. The following illustration shows the tree structure associated with a complex map constructed for a several departments in several languages.



Constructing a device-driver map tree structure

The New Map Wizard enables you to create a new driver map from scratch or create a new driver map based on an existing driver map. The New Map Wizard handles creating the new driver map entry in the repository, assigning the map a file name, and placing the map in the correct location of the repository.

There are some fundamental differences between base and driver maps. Unlike base-map tree structures, the only components that are added to driver-map tree structures are menu items and device-driver modules. Menu items in driver maps have no effect on the menus displayed at the target computer. The only purposes for menu items in a driver map are:

- To provide the developer with a mechanism for annotation
- To assign filters to one or more device-driver modules

It is not always necessary to include a driver map as part of a deployable Smart Image. If you deploy an Ultra-Portable Image or a Portable-Sysprep Image to an IBM HIIT-enabled target computer, you have the choice of:

- Using a driver map and managing *all* of the device-driver modules required for that target computer
- Not using a driver map and relying on the IBM-provided device-driver modules in the service partition of the target computer

If you deploy a Hardware-Specific Image to a target computer, any driver map deployed as part of that Smart Image will be ignored during the installation of the Hardware-Specific Image. The target computer requires all device drivers be included in the Hardware-Specific Image.

If you decide to use driver maps, there are two approaches to constructing them:

- **Create simple driver maps that support a single computer machine type.**

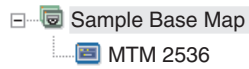
This approach requires creating and managing multiple maps, each customized with device drivers suited for a particular target computer. If you create a simple driver map, you are responsible for determining all of the device drivers that are required for that computer and inserting all of the appropriate device-driver modules into the tree structure.
- **Create a complex driver map that supports many computer machine types.**

This approach uses one menu item for each machine type that your organization supports and then assigns a filter to each menu item to determine which set of device drivers are to be used during the installation process. This approach requires the use of filters. If you create a complex driver map based on an existing map, you can simply make the changes to the tree structure required to support the additional machine type and define the filter for the new or changed menu item. If you create a new complex driver map from scratch, you are responsible for:

 - Inserting *all* menu items
 - Inserting *all* device-driver modules
 - Defining filter attributes for *all* menu items

Inserting menu items in a driver map

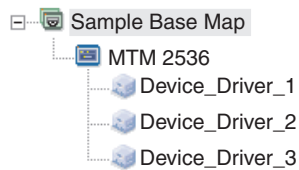
The use of menu items is only required when you plan to use filters. Unlike base maps, you cannot nest menu items. All menu items in a driver map are inserted at the root level. The following illustration shows a driver map tree structure with one menu item listed at the root level.



Inserting modules in a driver map

If you are not using menu items in your driver map, device-driver modules are inserted at the root level. If you are using menu items, device-driver modules are nested under the appropriate menu items. A filter impacts only those device-driver modules that are nested under the menu item to which the filter is assigned.

Device-driver modules are the only type of modules that can be added to a driver map. Typically, one or more device-driver modules are inserted under each menu item listed in the driver map. The following illustration shows a simple driver map tree structure with one menu item and a series of associated device-driver modules nested below it.



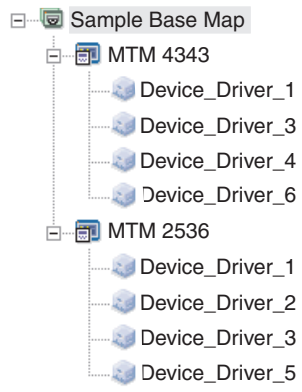
Using filters in a driver map

The purpose of a filter in a driver map is to automatically determine which set of drivers to make available to the Windows setup program (or mini-setup program in the case of a Portable-Sysprep Image) for a specific target computer. The filters provided with the ImageUltra Builder program enable you to query for the machine type, machine type and model number, platform (desktop or mobile), and whether or not the target computer is an IBM computer. To reduce the number of menu items and filters in a map, you can provide multiple drivers and let the Windows setup program (or mini-setup program) sort them out based on plug-n-play hardware detection.

You can also create your own filters if you need to query for some other hardware condition. Filters are DOS programs that run during the installation process. In most cases, filters query the computer BIOS for information, but filters can query any installed hardware that stores identification information in a ROM or EEPROM module.

When using filters in a driver map, filters are assigned to menu items, not to modules. The behavior of the filter is controlled through the Filter tab of the menu-item attributes. If you assign multiple filters to a menu item, you define whether only one filter needs to be satisfied or if all filters need to be satisfied to use the module.

When a filter is assigned to a menu item, a small filter icon is displayed in the lower-left corner of the menu-item icon. The following figure shows the tree structure of a device-driver map that has a filter assigned to each menu item.



Chapter 6. Creating maps

This chapter provides basic information about creating base maps and driver maps as well as how to define the attributes associated with each. For additional information, refer to the ImageUltra Builder help system.

Creating and modifying base maps

Base maps control which operating systems, applications, and utilities are deployed as part of a Smart Image. The tree structure of a base map varies depending on the type of image or images contained in the map and the needs of your organization. You must create a map module before you can construct the map tree structure. Detailed information about constructing the map tree structure, see Chapter 5, “Map construction overview” on page 33.

Creating a new base map

To create a new base map from scratch, complete the following procedure:

1. Open the repository where you want the new base map to reside.
2. In the Components area of the Repository window, click **Base Maps**.
3. From the Main window menu bar, click **New** and then click **Base Map....** The New Map Wizard opens.
4. Click **Next**.
5. Verify that the **Base Map** radio button is marked, and then click **Next**.
6. In the Description field, type the name you want to use to identify your new base map. Be as specific as possible. This name will be used to identify your map in the repository.
7. Click **Next**.
8. If you want the target computer to do a net-sync operation (check for updated maps before installation), click the **Yes** radio button; if not, click the **No** radio button.
9. Click **Next**.
10. Do one of the following:
 - If after installation you want to keep the service partition at the target computer exactly the way the Smart Image is deployed, click the **Delete none** radio button.
 - If after installation you want to resize the service partition at the target computer and keep only those files that were used on the target computer during installation, click the **Delete unused** radio button.
 - If after installation you want to eliminate the service partition at the target computer, click the **Delete all** radio button.
11. Click **Next**.
12. Click **Next** again.
13. Click **Finish**. The New Map Wizard closes and the Base Map window opens.

Adding menu items to a base map

To add menu items to a base map, complete the following procedure:

1. Open the repository where the map resides.

2. In the Components area of the Repository window, click **Base Maps**. A list of base maps is displayed.
3. Double-click the appropriate base map. The Base Map window opens.
4. Do one of the following:
 - If the map tree already has menu items in it, expand the map tree and click the entry one level above the location where you want to add the new menu item.
 - If the map tree does not have any menu items in it, click the root entry.
5. In the Main window menu bar, click **Insert** and then click **Menu Item**. A “New Item” box is added to the tree.
6. In the “New Item” box, type the text for the new menu item and then press Enter.
7. Repeat steps 4 through 6 until all menu items are added.
8. To move a menu item up or down, click on the menu item; then, click the Move Item Up or Move Item Down icon located in the Main window toolbar.
9. When you are finished, click the **Save** icon in the Main window toolbar.

Note: To delete modules or menu items within a map, select the module or menu item and click the **Delete** icon located in the Main window toolbar.

Adding operating-system modules to a base map

To add an operating-system module to a base map, complete the following procedure:

1. Open the repository where the map resides.
2. In the Components area of the Repository window, click **Base Maps**. A list of base maps is displayed.
3. Double-click the appropriate base map. The Base Map window opens.
4. Expand the map tree.
5. Click the menu item above where you want to add the operating-system module.
6. In the Main window menu bar, click **Insert** and then click **Operating System...**
7. In the Add Operating Systems window, click the module you want to add. If you want to add more than one module at a time, use the Shift key or Ctrl key in conjunction with the mouse to highlight multiple modules.
8. Click **OK**.
9. Repeat steps 5 through 8 for each operating-system module you want to add.
10. When you are finished, click the **Save** icon in the Main Window toolbar.

Adding application modules to a base map

To add an application module to a base map, complete the following procedure:

1. Open the repository where the map resides.
2. In the Components area of the Repository window, click **Base Maps**. A list of base maps is displayed.
3. Double-click the appropriate base map. The Base Map window opens.
4. Expand the map tree.
5. Click the menu item above where you want to add the application module.
6. In the Main window menu bar, click **Insert** and then click **Application...**

7. In the Add Applications window, click the module you want to add. If you want to add more than one module at a time, use the Shift key or Ctrl key in conjunction with the mouse to highlight multiple modules.
8. Click **OK**.
9. Repeat steps 5 through 8 for each application module you want to add.
10. When you are finished, click the **Save** icon in the Main Window toolbar.

Working with base map and menu item attributes

During the installation process on a target computer, the person performing the installation typically makes selections from a set of menus to specify the image that is to be installed. The appearance of these menus and the individual menu items are controlled through the base map attributes. Detailed information about working with the base map attributes can be found in Chapter 5, “Map construction overview” on page 33.

The following procedures describe how to specify new attributes or modify existing attributes. To use these procedures, you must have already created your base map and set up your map tree structure.

Specifying base map attributes

Base map attributes are attributes associated with the root entry of the tree structure and control the following:

- The title and physical appearance of the first menu that the installer sees at the target computer during the installation process
- The post-install behavior of the service partition at the target computer
- Default values and prompts for user information.

To define the base-map attributes, complete the following procedure:

1. Open the repository where the map resides.
2. In the Components area of the Repository window, click **Base Maps**. A list of base maps is displayed in the right pane.
3. Double-click the appropriate base map. The Base Map window opens with the root entry highlighted.
4. In the Submenu items tab, do the following:
 - a. In the Title field, type a title or instruction for the first menu. For example, if your first menu contains a list of operating systems, you might want the title to be “Select an operating system.”
 - b. In the Spacing field, set the number of blank lines that you want to use to separate menu items.
 - c. Mark the Auto-Select check box if you want the installation program to automatically make the selection if only a single menu item is listed. A single-menu-item condition will exist if only one submenu item is listed under the selected menu item in the tree structure, or if you assigned a filter to multiple submenu items and only one submenu item satisfies the condition of the filter.
 - d. Mark the Multi-Select check box if you want to enable the installer to choose more than one item from the menu. For example, you might want to give the installer the ability to choose one or more items from a list of application programs.
5. If you want to predefine user information with default values or prompt the installer for user information near the beginning of the installation process, do the following in the User Information tab:

- a. Click the **User Information** icon located in the User Information tab.
 - b. In the Information field, use the drop-down menu to select the type of setting you want to predefine or for which you want to prompt.
 - c. If you want to predefine a value for this setting, type the information in the Default field.
 - d. If you want to prompt the installer for a value for this setting, click the Display the following prompt check box, so a checkmark is visible; then, in the space provided, type the prompt you want the installer to see.
 - e. Click **OK**.
 - f. Repeat these steps for each type of setting.
6. In the Options tab, do the following:
 - a. If you want the target computer to perform a network sync operation (check for updated maps before installation), click the **Use Network Sync** check box so a checkmark is visible.
 - b. In the Post Install Action field, do one of the following:
 - If after installation you want to keep the service partition at the target computer exactly the way the Smart Image is deployed, click the **Delete none** radio button.
 - If after installation you want to resize the service partition at the target computer and keep only those files that were used on the target computer during installation, click the **Delete unused** radio button.
 - If after installation you want to eliminate the service partition at the target computer, click the **Delete all** radio button.
 7. When you are done modifying the attributes, click the **Save** icon in the Main window toolbar.

Specifying base-map menu-item attributes

Menu item attributes in a base map control the following:

- The title and physical appearance of the menus that follow the first menu that the installer sees at the target computer during the installation process.
- Whether or not filters are used to dynamically build the menu
- Whether or not utilities are to be run when a menu item is selected.

To define attributes for menu items listed in the base map, complete the following procedure:

1. Open the repository where the map resides.
2. In the Components area of the Repository window, click **Base Maps**. A list of base maps is displayed in the right pane.
3. Double-click the appropriate base map. The Base Map window opens.
4. Expand the map tree and click on the appropriate menu item. Several tabs are displayed in the right pane.
5. Click the **Utility** tab if you want to associate a utility with the menu item, then do the following:
 - a. In the Utility name field, use the drop-down menu to select a utility from the module repository.
 - b. In the Parameters field, type any parameters required to run the utility.
 - c. In the Execution State area, do one of the following:
 - Click the **All filters must be satisfied** radio button if you have multiple filters associated with the menu item and you want the menu item to be displayed only if the conditions set by all filters are met.

- Click the **One or more filters must be satisfied** radio button if you want the menu item to be displayed if the conditions are met for at least one filter.
6. Click the **Filter** tab if you want to associate a filter with the menu item, then do the following:
 - a. To add a filter to the menu item, click the **Add Filter** icon. An Add Filter window is displayed.
 - b. Use the drop-down menu to select a filter from the module repository.
 - c. Add parameters for the filter. For information about the parameters used with the filters provided as part of the ImageUltra Builder program, refer to the ImageUltra Builder help system.
 - d. In the Multiple Filters field, do one of the following:
 - Click the **All filters must be satisfied** radio button if you have multiple filters associated with the menu item and you want the menu item displayed only if the conditions set by all filters are met.
 - Click the **One or more filters must be satisfied** radio button if you want the menu item displayed if the conditions set by at least one filter is met.
 7. Click the **Submenu items** tab and do the following:

Note: The Submenu items tab is present only if submenu items have been inserted under the selected menu item.

 - a. In the Title field, type a title or instruction for the menu controlled by this menu item. For example, if your menu contains a list of languages, you might want the title to be "Select a language."
 - b. In the Spacing field, set the number of blank lines that you want to use to separate menu items.
 - c. Mark the **Auto-Select** check box if you want the installation program to automatically make the selection when only a single menu item is listed. A single-menu-item condition will exist if only one submenu item is listed under the selected menu item in the tree structure, or if you assigned a filter to multiple submenu items and only one submenu item satisfies the conditions defined by the filter.
 - d. Mark the **Multi-Select** check box if you want to enable the installer to choose more than one item from the menu. For example, you might want to give the installer the ability to choose one or more items from a list of application programs.
 8. Click the Save icon in the Main window toolbar.

Creating and modifying driver maps

Driver maps control which device drivers and hardware-specific applications are deployed as part of a Smart Image. Driver maps can be simple (specific to one machine type) or complex (designed for many machine types), depending on the needs of your organization.

There are three approaches to creating a driver map:

- If you import device-driver modules from a source computer, the ImageUltra Builder program automatically creates a simple driver map for you based on the modules you import. You can use this device-driver map as is, modify the device-driver map to suit your needs, or make a copy of the map and modify the copy.
- You can use the New Map Wizard to create a driver map from scratch.

- You can use the New Map Wizard to create a driver map based on an existing driver map, and then make whatever modifications are needed.

You can also replace an existing map with a modified copy by updating the Network Sync Table.

Detailed information about constructing the map contents is in Chapter 5, “Map construction overview” on page 33. The information in this section describes the methods used to create and modify maps.

Creating a new driver map

Use the following procedure to create a new driver map from scratch or to create a new driver map based on an existing driver map:

1. Open the repository where you want the new driver map to reside.
2. In the Components area of the Repository window, click **Driver Maps**.
3. From the Main Window menu bar, click **New** and then click **Device Driver...** The New Map Wizard opens.
4. Click **Next**.
5. Verify that the **Driver map** radio button is selected, and then click **Next**.
6. In the Description field, type the name you want to use to identify your map. Be as specific as possible. This is the name that will be used to identify your map in the repository.
7. Click **Next**.
8. If you want to enable the network-sync function, click the **Yes** radio button; if not, click the **No** radio button.
9. Click **Next**.
10. Do one of the following:
 - If you would like to base your new map on an existing map, click the map that you want to use as the source, and then click **Next**.
 - If you do not want to base your new map on an existing map, just click **Next** without selecting a map.
11. Click **Finish**.
12. The New Map Wizard closes and the Driver Map window opens.
 - For information about working with the tree structure, see Chapter 5, “Map construction overview” on page 33.
 - For step-by-step instructions on inserting a menu item, see “Adding menu items to a driver map”.
 - For step-by-step instructions on inserting device driver module, see “Adding device-driver modules to a driver map” on page 53.
 - For step-by-step instructions on modifying the attributes in a driver map, see “Specifying driver map attributes” on page 53.

Adding menu items to a driver map

Menu items in a driver map provide a mechanism for annotation and the means by which filters are assigned to individual device-driver modules or groups of device-driver modules. Unlike base maps, menu items in a driver map have no effect on the menu items shown at the target computer.

To add a menu item to a device-driver map, complete the following procedure:

1. Open the repository where the map resides.

2. In the Components area of the Repository window, click **Driver Maps**. A list of driver maps is displayed.
3. Double-click the appropriate driver map. The Driver Map window opens.
4. Click on the root entry of the tree structure.
5. In the Main window menu bar, click **Insert** and then click **Menu Item**. A “New Item” box is added to the tree.
6. In the “New Item” box, type the text for new menu item and then press Enter.
7. Repeat step 4 through step 6 until all menu items are added.
8. To move a menu item up or down, click the menu item; then, click on the **Move Item Up** or **Move Item Down** icon in the Main window toolbar.
9. When you are finished, click the **Save** icon in the Main Window.

Note: To delete modules or menu items within a map, select the module or menu item and click the **Delete** icon located in the Main window toolbar.

Adding device-driver modules to a driver map

1. Open the repository where the map resides.
2. In the Components area of the repository window, click **Driver Maps**. A list of driver maps is displayed.
3. Double-click the appropriate driver map. The Driver Map window opens.
4. Expand the map tree.
5. Highlight either the root entry or the menu-item above where you want to add the device-driver module.
6. In the Main window menu bar, click **Insert** and then click **Device Driver...**
7. In the Add Device Drivers window, click the module you want to add. If you want to add more than one module at a time, use the Shift key or Ctrl key in conjunction with the mouse to highlight multiple modules.
8. Click **OK**.
9. Repeat step 5 through step 8 for each device-driver module you want to add.
10. When you are finished, click the **Save** icon in the Main window toolbar.

Specifying driver map attributes

You can specify attributes for the root entry and you can specify attributes for each menu item. The only attribute you can set for the root entry is whether or not you want to enable the network-sync function. The only attribute you can set for a menu item is associating a filter with the menu item. To define these attributes, complete the following procedure:

1. Open the repository where the map resides.
2. In the component area of the Repository window, click **Driver Maps**. A list of device-driver maps is displayed in the right pane.
3. Double-click the appropriate device-driver map. The Driver Map window opens.
4. To set the root entry attribute, do the following:
 - a. Click on the root entry.
 - b. In the General tab, mark the **Use Network Sync** checkbox if you want to enable the network-sync function for that map.
5. To set a menu-item attribute, do the following:
 - a. Click on the menu item.

- b. Click on the Filter tab.
 - c. To add a filter, click the Insert Filter icon located on the Filter tab. The Add Filter window opens.
 - d. Use the drop-down menu to select a filter from the repository.
 - e. In the Parameters field, type the parameters required for the filter to provide the desired result. For information about the parameters associated with the filters provided with the ImageUltra Builder program, refer to the ImageUltra Builder help system.
 - f. Click **OK**.
 - g. If you assign multiple filters to a menu item, click either the **All filters must be satisfied** radio button or the **One or more filters must be satisfied** radio button.
6. When you are finished making the changes, click the **Save** icon in the Main window toolbar.

Chapter 7. Preparing source files for modules

This chapter provides information about preparing source files and creating modules from source files.

Creating source operating-system images

The method you use to create the source for a base operating-system module depends on the type of image you intend to deploy to the target computer. The methods for creating the source for a Portable-Sysprep Image and Hardware-Specific Image differ.

A note about Ultra-Portable Images

The ImageUltra Builder program does not provide the tools necessary to create a base operating-system module for an Ultra-Portable Image. You must *import* an IBM-developed operating-system module from either a HIIT-enabled IBM personal computer or from a set of IBM Recovery CDs. For more information about importing modules, see “Importing pre-existing modules” on page 10 or consult the ImageUltra Builder help system for step-by-step instructions.

Required third-party tools

If you intend to create either Portable-Sysprep Images or Hardware-Specific Images, you will need to use the Windows Sysprep tool and do the following:

- Understand the requirements for running the Sysprep tool.
- Familiarize yourself with the several languages that the Sysprep tool supports.
- Understand which versions of the Sysprep tool are compatible with the operating systems you will use for your images.
- Familiarize yourself with at least one of the supported third-party image-cloning tools: Symantec Norton Ghost or PowerQuest DeployCenter.

For more information about using the Sysprep tool, go to the Microsoft Windows Web site at <http://www.microsoft.com/ms.htm> and search on the term Sysprep.

For more information about Symantec Norton Ghost, go to the Symantec Web site at <http://www.symantec.com/> search on the term Ghost.

For more information about PowerQuest DeployCenter, go to the PowerQuest Web site at <http://www.powerquest.com/> and search on the term DeployCenter.

Limitations of Symantec Norton Ghost

When Ghost is used in conjunction with the ImageUltra Builder program, certain limitations must be considered.

Symantec Norton Ghost can clone partitions and drives. Because the ImageUltra Builder program uses a service partition to store the Smart Image, Symantec Norton Ghost might encounter problems when attempting to set the size for multiple partitions or when installing a cloned drive. When defining partition sizes as a percentage, Ghost always uses the full hard-disk capacity and does not take the presence of a service partition into account. As a result, Ghost might delete the service partition on the target computer. Defining absolute sizes can be equally problematic because drive sizes vary on target computers.

When using a cloned drive, two problems can occur:

- If a service partition exists on a source computer, Ghost includes this service partition as part of the image and will overwrite the service partition on the target computer. This results in a mismatch between the contents on the active partition of the target computer and the service partition of the target computer.
- If a service partition does not exist on the source computer, Ghost will assume that it can use the entire hard disk for the image, and will delete the service partition on the target computer.

Because it is so difficult to manage multiple partitions and cloned drives created by Ghost when a service partition is present, IBM supports the following implementations only:

- Restoring single-partition images produced by Ghost
- Restoring a single-partition, cloned-drive image as a single partition, not as a cloned drive.

The Silent Install command in the Source tab of the Operating System window must be GHOST.BAT.

Inside GHOST.BAT, you must use the following statements:

```
Gdisk.exe 1 /cre /pri /sure  
ghost.exe -clone,mode=pload,src=image.gho:1,dst=1:1 -auto -sure -quiet -fx -batch
```

Where *image.gho* is the name of your image.

Limitations of PowerQuest DeployCenter

The ImageUltra Builder program uses a service partition at the end of the hard disk to store the Smart Images. The service partition is hidden and must be protected during the installation process. PowerQuest DeployCenter has a scripting function that allows for the protection of partitions.

Before any files are installed, the ImageUltra Builder program deletes all partitions from the target computer hard disk except for the service partition. Therefore, when creating scripts, you do not have to include any commands to delete partitions.

The following is a sample PowerQuest DeployCenter script that can be used with the ImageUltra Builder program to protect the service partition and restore two partitions:

```
SELECT DRIVE 1  
PROTECT PARTITION LAST  
SELECT FREESPACE FIRST  
SELECT IMAGE 1  
RESIZE IMAGE PROPORTIONAL  
SELECT IMAGE 2  
RESIZE IMAGE 2048  
RESTORE  
SELECT PARTITION 1  
SET ACTIVE  
SELECT PARTITION 2  
UNHIDE
```


In the sample script, two partitions are established: partition 1 uses all available free space and partition 2 uses 2 GB. By default, when PowerQuest DeployCenter sets partition 1 as active, it hides any remaining primary partitions. Therefore, the script must “unhide” any primary partitions that were hidden when partition 1 was set as active. The sample script “unhides” partition 2. If a third primary partition were being used, you would add the following lines to the end of the script:

```
SELECT DRIVE 1
SELECT PARTITION 3
UNHIDE
```

When building a module for a PowerQuest image, use a batch file and use the batch file name as the Silent Install command in the Source tab of the Operating System window. For example, use PQIMAGE.BAT as the Silent Install command.

Inside the PQIMAGE.BAT file, you must use the following line:

```
pqdi.exe /cmd=restore.scp /I24 /NRB /img=image.pqi
```

where *restore.scp* is the script shown earlier and *image.pqi* is the name of the image that you want to restore. The /NRB parameter instructs the PQDI.EXE program not to reboot the computer so the ImageUltra Builder installation program can regain control after the PowerQuest image has been installed.

Before you build the module, you must put the image files, the script file, and the batch file in the same folder.

For additional information about developing PowerQuest DeployCenter scripts, refer to your PowerQuest DeployCenter documentation.

Creating a Portable-Sysprep Image

There are four main steps to creating a base operating-system module based on a Portable-Sysprep Image:

1. Making the IBM Customization Program accessible to the source computer.
2. Creating the Portable-Sysprep Image on a source computer and using Symantec Norton Ghost or PowerQuest DeployCenter to create the image files.
3. Setting the module attributes.
4. Building the module from the image files.

Making the IBM Customization Program accessible

Before you can create a Portable-Sysprep Image, you must first make the IBM Customization Program (IBMCP.EXE) accessible to the source computer. The ImageUltra Builder program enables you to copy the IBM Customization Program on a diskette, shared drive, or any other media to which the ImageUltra console has direct-write access.

1. Use Windows Explorer or My Computer to open the IBMCP folder in your repository.

Note: If you cannot remember the path to your repository, in the ImageUltra Builder Main window, click **File** and then click **Open repository...** The repository paths will be listed. You can use any repository.

2. Copy the IBMCP.EXE program to a diskette or to a shared drive that the source computer can access during the development of the image. If you use a diskette, label the diskette "ImageUltra IBM Customization Program."
3. Do one of the following:
 - Provide an *ImageUltra IBM Customization Program* diskette to any personnel responsible for creating Portable-Sysprep Images.
 - Provide the path to the IBMCP.EXE program to any personnel responsible for creating Portable-Sysprep Images.

Creating and preparing the image files

Before you begin, make sure you put the IBM Customization Program either on portable media that is compatible with the source computer or a shared drive that the source computer is able to access after the operating system and other programs have been installed. See "Making the IBM Customization Program accessible" on page 57 for more information.

To create and prepare Portable-Sysprep Image source files, do the following:

1. On a source computer, use MS-DOS to format the hard disk as follows:

Note: You must use an MS-DOS diskette for this step. Do not use Windows or PC-DOS to partition or format the hard disk. You can create an MS-DOS diskette from Windows 98, but be sure to also copy the FORMAT.COM and FDISK.EXE programs on the diskette.

- a. Insert the MS-DOS diskette into the source computer diskette drive and restart the computer.
 - b. Run FDISK.EXE and create a FAT32 primary partition. In most cases you will want to use the entire hard disk except for the service partition.
 - c. Leave the MS-DOS diskette in the drive and restart the computer.
 - d. Format the primary partition using the `FORMAT C: /S`
2. Install the Windows operating system (Windows XP or Windows 2000) that you want on your target computers. Do not allow the Windows installation program to reformat the hard disk.
 3. Access and run the IBM Customization Program (IBMCP.EXE). This program might be on portable media or a shared drive. See "Making the IBM Customization Program accessible" on page 57 for details.
 4. Obtain the Microsoft Sysprep program designed for your operating system:
 - The Windows 2000 version of the Sysprep program is available from the Microsoft Web site at <http://www.microsoft.com/windows2000/downloads/tools/sysprep/license.asp>. You will need the following files: SYSPREP.EXE and SETUPCL.EXE.
 - The Windows XP version of the Sysprep program is on the Windows XP CD in the \SUPPORT\TOOLS\DEPLOY.CAB file. You will need the following files: SYSPREP.EXE, SETUPCL.EXE, and FACTORY.EXE.
 5. Copy the Sysprep files to the source computer as follows:
 - Windows 2000: Copy the SYSPREP.EXE and SETUPCL.EXE files into the C:\SYSPREP folder.
 - Windows XP: Copy the SYSPREP.EXE, SETUPCL.EXE, and FACTORY.EXE files into the C:\SYSPREP folder.
 6. If you are creating a Windows 2000 image, obtain the SHUTDOWN.EXE file from one of the following sources and copy it into the C:\IBMWORK folder:

- Windows NT 4.0 Resource Kit Support Tools at <http://www.microsoft.com/ntserver/nts/downloads/recommended/ntkit/default.asp>.
 - Windows 2000 Resource Kit at <http://www.microsoft.com/windows2000/techinfo/reskit/default.asp>.
7. The IBM Customization Program creates an empty TAG file for Windows 2000 (C:\IBMWORK\WIN2000) and Windows XP Professional (C:\IBMWORK\XPPRO). If you are making a Windows XP Home image, rename C:\IBMWORK\XPPRO to C:\IBMWORK\XPSHOME.
 8. Install any application programs that will be common across all of your target computers. If you have any applications that you cannot make into a module, now is the time to install them.

Note: This step is optional. You can include common applications as part of your operating-system module or you can build separate application modules and control them through your base map.

9. Make any other modifications to the image that you feel are necessary.
10. Use the following procedure to modify the BOOT.INI file so by default, the source computer will boot to DOS instead of Windows:
 - a. From the Windows desktop, right-click **My Computer** and then click **Properties**.
 - b. Click the **Advanced** tab.
 - c. Click the **Startup and Recovery** button.
 - d. In the Default Operating System field, use the drop-down menu to select either **"DOS"** or **"Microsoft Windows"**, whichever is present.

Note: Do not select "Microsoft Windows 2000 Professional"/fastdetect, "Microsoft Windows XP Professional"/fastdetect, or Microsoft Windows XP Home"/fastdetect.

- e. Click **OK**.
11. Make any required changes to the C:\IBMWORK\SYSPREP.INF file for your image. This file is used to re-enter the audit-boot mode. You might change a password, set PID, and so on.

Note: You might have to update PID information. In Windows XP, the easiest place to modify the PID information is in C:\IBMWORK\PROKEYP.FM.

12. Make any required changes to customize Sysprep in the C:\SYSPREP\SYSPREP.INF file.
13. Run SYSPREP.EXE as follows:
 - Windows 2000: Run C:\SYSPREP\SYSPREP -quiet
 - Windows XP: Run C:\SYSPREP\SYSPREP.EXE -factory -quiet
14. **Important:** Before you continue to the next step, read "Limitations of Symantec Norton Ghost" on page 55 or "Limitations of PowerQuest DeployCenter" on page 56 to learn about special preparation procedures associated with your image-cloning tool.
15. Use either Symantec Norton Ghost or PowerQuest DeployCenter software to create the image files. If you intend to deploy the image using CDs, make sure you use the tools provided with Ghost and DeployCenter to segment the image into files that can fit on CDs.

Note: Set the maximum file size for CDs at 600MB. If you are using Symantec Norton Ghost version 7.5, make sure you use the -cms parameter when creating the image.

16. Store the image file in its own folder. In the case of CD deployment, where you had to segment the image into CD-sized files, store all of the files in a single folder. Make sure the folder is in a location that is accessible to the ImageUltra Builder console.
17. Store the Image-cloning tool you used to create the image (Symantec Norton Ghost or PowerQuest DeployCenter) in the same folder as the image file.
18. Store any batch files or script files you may have created in the same folder as the image file.
19. Continue with "Setting the attributes for the new base operating-system module".

Setting the attributes for the new base operating-system module

To set the attributes for the new module, do the following:

1. Open the repository where you want the new base operating-system module to reside.
2. In the Components area of the Repository window, click **Operating Systems**.
3. From the Main Window menu bar, click **New** and then click **Operating System....** The New Module Wizard opens.
4. Click **Next**.
5. Verify that the **Operating System** radio button is selected, and then click **Next**.
6. In the Description field, type a name that you want to use to identify this module. Be as specific as possible. This name will be the name that appears in the repository.

Note: It is a good idea to include the image type as part of the name, such as (PSI) for Portable-Sysprep Image. This will help you when you create your maps.

7. Click the **No** radio button to indicate that this module is not a container.
8. Click **Next**.
9. Do one of the following:
 - If you want to base the attributes for this module on a pre-existing operating-system module select the appropriate module and then click **Next**.
 - If you do not want to base the attributes for this module on another module, just click **Next** without selecting a module.
10. Click the **Base Operating System** radio button, then use the drop-down menu to select **Portable-Sysprep Image**.
11. Click **Next**.
12. Click **Finish**. The New Module Wizard closes and the Operating System window for the new module opens.
13. In the General tab, type the full version of the operating system in the Version field.
14. In the OS/Languages tab, do the following:
 - a. In the left pane, put a check mark next to each language for which the module is intended to be used. In most cases, base operating-system modules are language specific and only one language is typically selected. However, you can select multiple languages, if needed.

- b. In the right pane, put a check mark next to the operating system for which this module applies.
15. In the Source tab, do the following:
 - a. In the Source Directory field, type the complete path to the folder that contains the image files.
 - b. In the Silent Install Command area, type the name of the batch file you developed to control the installation of your image in the Command field. (See “Limitations of Symantec Norton Ghost” on page 55 and “Limitations of PowerQuest DeployCenter” on page 56 for information about creating this batch file.)
 16. Click the **Save** icon in the Main window tool bar.
 17. Continue with “Building a new module” on page 72.

Creating a Hardware-Specific Image

There are three main steps to creating a base operating-system module based on a Hardware-Specific Image:

1. Creating the Hardware-Specific Image on a source computer and using Symantec Norton Ghost or PowerQuest DeployCenter to create the image files.
2. Setting the module attributes.
3. Building the module from the image files.

Creating and preparing the image files

To create and prepare Hardware-Specific Image source files, do the following:

1. On a source computer, install the Windows operating system that you want on your target computers. You can use any file system that you want.
2. Install any application programs that will be common across all of your target computers.
3. Install the Microsoft Sysprep program designed for your operating system in the C:\SYSPREP folder. Do not run SYSPREP.EXE at this time.

Note: The Windows XP version of the Sysprep program is available from the Microsoft Web site at <http://www.microsoft.com/windows2000/download/tools/sysprep/license.asp>. The Windows XP version of the Sysprep program is on the Windows XP CD in the \SUPPORT\TOOLS\DEPLOY.CAB file.

4. Make any required changes to the C:\SYSPREP\SYSPREP.INF file and then run SYSPREP.EXE.
5. In most cases the Sysprep program shuts down the computer when it has finished running. If the Sysprep program does not shut down the source computer, shut down the source computer manually.

Important: Before you continue to the next step, read “Limitations of Symantec Norton Ghost” on page 55 or “Limitations of PowerQuest DeployCenter” on page 56 to learn about special preparation procedures associated with your image-cloning tool.

6. Use either Symantec Norton Ghost or PowerQuest DeployCenter software to create the image. If you intend to deploy the image using CDs, make sure you use the tools provided with your image-cloning tool to segment the image into files that can fit on CDs.

Note: Set the maximum file size for CDs at 600MB. If you are using Symantec Norton Ghost version 7.5, make sure you use the -cms parameter when creating the image.

7. Store the image in its own folder. In the case of CD deployment, where you had to segment the image into CD-sized files, store all of the files in a single folder. Make sure the folder is in a location that is accessible by the ImageUltra Builder console.
8. Store the image-cloning tool that you used to create the image (Symantec Norton Ghost or PowerQuest DeployCenter) in the same folder as the image file.
9. Store any batch files or script files that you may have created in the same folder as the image file.
10. Continue with "Setting the attributes for the new base operating-system module".

Setting the attributes for the new base operating-system module

To set the attributes for the new module, do the following:

1. Open the repository where you want the new base operating-system module to reside.
2. In the Components area of the Repository window, click **Operating Systems**.
3. From the Main Window menu bar, click **Insert** and then click **Operating System**. The New Module Wizard opens.
4. Click **Next**.
5. Verify that the **Operating System** radio button is selected, and then click **Next**.
6. In the Description field, type a name that you want to use to identify this module. Be as specific as possible. This name will be the name that appears in the repository.

Note: It is a good idea to include the image type as part of the name, such as (HSI) for Hardware-Specific Image. This will help you when you create your maps.

7. Click the **No** radio button to indicate that this module is not a container.
8. Click **Next**.
9. Do one of the following:
 - If you want to base the attributes for this module on a pre-existing operating-system module, select the appropriate module and then click **Next**.
 - If you do not want to base the attributes for this module on another module, just click **Next** without selecting a module.
10. Click the **Base Operating System** radio button, then use the drop-down menu to select **Hardware-Specific Image**.
11. Click **Next**.
12. Click **Finish**. The Operating System window for the new module opens.
13. In the General tab, type the full version of the operating system in the Version field.
14. In the OS/Languages tab, do the following:
 - a. In the left pane, put a check mark next to each language the module will use. In most cases, base operating-system modules are language specific and only one language is selected.
 - b. In the right pane, put a check mark next to the operating system that applies to this module.

15. In the Source tab, do the following:
 - a. In the source directory field, type the complete path to the folder that contains the image files.
 - b. In the Silent Install Command area, type the name of the batch file you developed to control the installation of your image in the Command field. (See “Limitations of Symantec Norton Ghost” on page 55 and “Limitations of PowerQuest DeployCenter” on page 56 for information about creating this batch file.)
16. Click the **Save** icon in the Main window tool bar.
17. Continue with “Building a new module” on page 72.

Preparing source files for application modules and add-on operating-system modules

Any application or add-on operating-system component that you intend to deploy as a module must be prepared to perform a silent installation. In addition, you can develop and include mini-applications that run immediately before or after an application or add-on operating-system component installs. Because the method of preparing applications and add-on operating-system components are nearly identical, the term application will be used throughout this section to describe both. The terms *add-on operating-system component* and *add-on operating-system module* will be used when these items need to be distinguished from an *application* or *application module*.

Preparing for a silent install

The method you use to prepare an application for silent installation varies depending on the installation mechanism being used. For example, if the application uses InstallShield as the installation mechanism, the application might already be enabled for silent installation if a SETUP.ISS file exists in the same folder as the SETUP.EXE file. If a SETUP.ISS file does not exist, or if the provided SETUP.ISS file does not produce the desired result, InstallShield enables you to create a new SETUP.ISS file by doing the following:

Note: Perform the following steps on a test computer.

1. Copy all source files for the application into an empty folder. Make sure you maintain the folder structure that was provided on the software manufacture distribution media. In the case of a .ZIP file, make sure you maintain the folder structure when you unzip the file.
2. Open a command prompt window.
3. Change to the folder that contains the application SETUP.EXE file.
4. At the command prompt, type **SETUP -R** and then press Enter.
5. Follow the steps to install the program. As you install the program, the keystrokes and mouse clicks are recorded in the new SETUP.ISS file. Because the values you type will be used on all target computers, you should use company-related information in place of any personal information for which you might be prompted.
6. When the installation is complete, the new SETUP.ISS file will be in the Windows folder. Copy the new SETUP.ISS file from the Windows folder into the folder that contains the application SETUP.EXE file.

Most commercially-available installation products, such as Microsoft Installer, also enable you to do a silent install, but the steps are somewhat different from the

InstallShield steps provided here. New versions of InstallShield might also have different steps. See the documentation associated with the specific installation product if you encounter any problems.

Adding mini-applications

The use of mini-applications is optional. Mini-applications are useful for creating special folders, cleaning up registry entries, or performing any other service related to the application being installed. The content and format of a mini-application is entirely up to you. You are limited to running one mini-application before the application is installed and one mini-application after the application is installed.

The only requirement related to the preparation of mini-applications is that you must copy all of the mini-application files into the same folder as the application SETUP.EXE file or into one of its subfolders.

Making files accessible to the ImageUltra console

The last step of preparation is making the files accessible to the ImageUltra console. Simply copy the folder and subfolders containing the application source files to any drive that can be accessed by the ImageUltra Builder console. You must keep the folder structure intact.

After the files are prepared

After the files are prepared, you can create a module entry in the repository if one does not exist, and then build the module. When you create the module entry, you are given the opportunity to define a number of attributes associated with the module. The attributes that are directly related to the preparation steps described in this section are on the Source tab. Those attributes include:

- The source directory of the application
- The silent-install command and parameters
- The preinstall command and parameters (for a mini-application)
- The postinstall command and parameters (for a mini-application)

For additional information about setting the attributes, see one of the following:

- “Setting the attributes for a new application module”
- “Setting the attributes for a new add-on operating-system module” on page 66

Setting the attributes for a new application module

To set the attributes for the new module, do the following:

1. Open the repository where you want the new application module to reside.
2. In the Components area of the Repository window, click **Applications**.
3. From the Main Window menu bar, click **New** and then click **Application...** The New Module Wizard opens.
4. Click **Next**.
5. Verify that the **Applications** radio button is selected, and then click **Next**.
6. In the Description field, type a name that you want to use to identify this module. Be as specific as possible. This name will be the name that appears in the repository.
7. Click the **No** radio button to indicate that this module is not a container.
8. Click **Next**.
9. Do one of the following:

- If you want to base the attributes for this module on a pre-existing application module, select the appropriate module and then click **Next**.
 - If you do not want to base the attributes for this module on another module, just click **Next** without selecting a module.
10. Click **Finish**. The New Module Wizard closes and the Application window for the new module opens.
 11. In the General tab, type the full version of the application in the Version field.
 12. In the OS/Languages tab, do the following:
 - a. In the left pane, put a check mark next to each language for which the application will be used. For example, you might choose to use an English application module with a different language operating-system module.
 - b. In the right pane, put a check mark next to the operating system for which this application module is intended to be used. If the application is not specific to any Windows operating system, you can check all of the boxes. If the application is specific to an operating system, check only the operating system that applies.
 13. In the Source tab, do the following:
 - a. In the Source Directory field, type the complete path to the folder that contains the application files.
 - b. In the Silent Install Command area, type the command that is to be used to install the application in the Command field and any associated parameters in the Parameters field. For example, your command might be `SETUP.EXE` and your parameters might be `/S>`.

Note: The following steps apply to the use of mini-applications.

Mini-applications are useful for creating special folders, cleaning up registry entries, or performing any other service related to the installation of the application. For more information about the use of mini-applications, see “Preparing source files for application modules and add-on operating-system modules” on page 63.

- c. If you have a mini-application contained in this module that needs to run immediately *before* the actual application is installed, use the Preinstall Command area to type the command and parameters associated with starting the mini-application.
 - d. If you have a mini-application contained in this module that needs to run immediately *after* the actual application is installed, use the Postinstall Command area to type the command and parameters associated with starting the mini-application.
14. If you want to assign a filter to the application module, do the following in the Filter tab:
 - a. Click the **Insert Filter** icon. The Add Filter window opens.
 - b. In the Name field, use the drop-down menu to select the filter you want to use.
 - c. In the Parameters field, type any parameters required to get the desired result. For information about the parameters associated with the filters provided with the ImageUltra Builder program, see the ImageUltra Builder help system.
 - d. Click **OK**.
 - e. If you assign multiple filters to a module, click either the **All filters must be satisfied** or **One or more filters must be satisfied** radio button to define how the filters are to be used.

15. In the Options tab field, do the following:
 - a. The Path to copy installable files field defines whether the application files should be copied to the C partition in an installable format.
 - If you type a path in this field and you provide a Silent Install Command in the Source tab, the files will be copied to the C: partition in addition to the application being installed.
 - If you type a path in this field and you do not provide a Silent Install Command in the Source tab, the files will be copied to the C: partition, but the application will not be installed automatically.

Type a path only if you want the installable files to be copied to the C: partition of the target computer.
 - b. All other fields on the Options tab are used to handle special cases. For information about these fields, refer to the ImageUltra help system.
16. Click the **Save** icon in the Main window tool bar.
17. Continue with “Building a new module” on page 72.

Setting the attributes for a new add-on operating-system module

To set the attributes for a new add-on operating-system module, do the following:

1. Open the repository where you want the new add-on operating-system module to reside.
2. In the Components area of the Repository window, click **Operating Systems**.
3. From the Main Window menu bar, click **New** and then click **Operating Systems....** The New Module Wizard opens.
4. Click **Next**.
5. Verify that the **Operating System** radio button is selected, and then click **Next**.
6. In the Description field, type a name that you want to use to identify this module. Be as specific as possible. This name will be the name that appears in the repository.
7. Click the **No** radio button to indicate that this module is not a container.
8. Click **Next**.
9. Do one of the following:
 - If you want to base the attributes for this module on a pre-existing add-on operating-system module, select the appropriate module and then click **Next**.
 - If you do not want to base the attributes for this module on another module, click **Next** without selecting a module.
10. Click the **Add-on operating system module** radio button.
11. Click **Next**.
12. Click **Finish**. The New Module Wizard closes and the Operating System window for the new module opens.
13. In the General tab, type the full version of the add-on component in the Version field.
14. In the OS/Languages tab, do the following:
 - a. In the left pane, put a check mark next to each language for which this module is intended to be used. For example, you might choose to use an English add-on operating system component with multiple languages, or you might want to limit its use to English only.

- b. In the right pane, put a check mark next to the operating system for which this module applies. For example, if you are creating a module for the latest version of Windows Media Player, that module might apply to more than one operating system. If you are creating a module for a service pack, that module would apply to one operating system only.
15. In the Source tab, do the following:
- a. In the source directory field, type the complete path to the folder that contains the image files.
 - b. In the Silent Install Command area, type the command that is to be used to install the add-on operating system component in the Command field and any associated parameters in the Parameters field. For example, your command might be `SETUP.EXE` and your parameters might be `/S>`.

Note: The following steps apply to the use of mini-applications.

Mini-applications are useful for creating special folders, cleaning up registry entries, or performing any other service related to the installation of the application. For more information about the use of mini-applications, see “Preparing source files for application modules and add-on operating-system modules” on page 63.

- c. If you have a mini-application contained in this module that needs to run immediately *before* the add-on operating-system component is installed, use the Preinstall Command area to type the command and parameters associated with starting that mini-application.
 - d. If you have a mini-application contained in this module that needs to run immediately *after* the add-on operating-system component is installed, use the Postinstall Command area to type the command and parameters associated with starting that mini-application.
16. If you want to assign a filter to the module, do the following in the filters tab:
- a. Click the **Insert filter** icon located in the Filter tab. The Add Filter window opens.
 - b. In the name field, use the drop-down menu to select the filter you want to use.
 - c. In the Parameters field, type any parameters required to get the desired result. (For information about the parameters associated with the filters provided as part of the ImageUltra Builder program, refer to the ImageUltra Builder help system.)
 - d. Click **OK**.
 - e. If you assign multiple filters to the module, click either the **All filters must be satisfied** or the **One or more filters must be satisfied** radio buttons to define how the filters are to be used.
17. In the Options tab, do the following:
- a. The Path to copy installable files field defines whether or not you want to copy the add-on operating-system component to target computer in an installable format.
 - If you type a path in this field and you provide a Silent Install Command in the Source tab, the files will be copied to the C: partition in addition to the application being installed.
 - If you type a path in this field and you do not provide a Silent Install Command in the Source tab, the files will be copied to the C: partition, but the application will not be installed automatically.
- Type a path only if you want the installable files to be copied to the C: partition of the target computer.

- b. The rest of the fields in the Options tab are for handling special cases or debugging problems. Do not make changes to these fields at this time. See Chapter 9, “Dealing with exceptions” on page 75 for information about the use of these fields.
18. Click the **Save** icon in the Main window tool bar.
 19. Continue with “Building a new module” on page 72.

Preparing source device-driver files

Device drivers must use the standard Microsoft Plug-n-Play .INF installation method, which is usually unattended in nature. The driver itself must be certified by the Microsoft Windows Hardware Qualification Lab (WHQL) and signed. Failure to use signed, WHQL-certified device drivers will result in interruptions during the installation process by Windows-generated information messages, which require user interaction to bypass.

Adding mini-applications

The use of mini-applications is optional. Mini-applications are useful for creating special folders, cleaning up registry entries, or performing any other service related to the device-driver being installed. The content and format of a mini-application is entirely up to you. You are limited to running one mini-application before the device driver is installed and one mini-application after the device driver is installed.

The only requirement related to the preparation of mini-applications is that you must copy all of the mini-application files into root folder of your device-driver source files.

Requirements for preparing device-driver files for modules

The following are the only requirements for preparing the device-driver source files to be made into a module:

- The device driver must be in a ready-to-install format. It cannot require an unpacking process.
- The device-driver files must be placed in a dedicated folder on a drive that is accessible by the ImageUltra Builder console.
- The device driver folder structure must be maintained.

When the source files meet these requirements, you can set the attributes for the new device-driver module.

Setting the attributes for a new device-driver module

To set the attributes for a new device-driver module, do the following:

1. Open the repository where you want the new driver module to reside.
2. In the Components area of the Repository window, click **Device Drivers**.
3. From the Main Window menu bar, click **New** and then click **Device Driver...** The New Module Wizard opens.
4. Click **Next**.
5. Verify that the **Device Driver** radio button is selected, and then click **Next**.
6. In the Description field, type a name that you want to use to identify this module. Be as specific as possible. This name will be the name that appears in the repository.

7. Click the **No** radio button to indicate that this module is not a container.
8. Click **Next**.
9. Do one of the following:
 - If you want to base the attributes for this module on a pre-existing driver module, select the appropriate module and then click **Next**.
 - If you do not want to base the attributes for this module on another module, just click **Next** without selecting a module.
10. Click **Finish**. The New Module Wizard closes and the Device Driver window for the new module opens.
11. In the General tab, do the following:
 - a. In the Version field, type the full version of the device driver.
 - b. In the Family field, use the drop-down menu to select the type of device for which this device driver is designed.
12. In the OS/Languages tab, do the following:
 - a. In the left pane, put a check mark next to each language for which this device driver will be used.
 - b. In the right pane, put a check mark next to the operating system with which the device-driver module will be used.
13. In the Source tab, do the following:
 - a. In the Source Directory field, type the complete path to the folder that contains the source device-driver files.
 - b. In the Silent Install Command area, leave the Command and Parameters fields blank.

Note: The following steps apply to the use of mini-applications.

Mini-applications are useful for creating special folders, cleaning up registry entries, or performing any other service related to the installation of the device driver. For more information about the use of mini-applications, see “Adding mini-applications” on page 68.

- c. If you have a mini-application contained in this module that needs to run immediately *before* the device driver is installed, use the Preinstall Command area to type the command and parameters associated with starting that mini-application.
- d. If you have a mini-application contained in this module that needs to run immediately *after* the device driver is installed, use the Postinstall Command area to type the command and parameters associated with starting that mini-application.
14. If you want to assign a filter to the driver module, do the following:
 - a. Click the **Insert Filter** icon. The Add Filter window opens.
 - b. In the Name field, use the drop-down menu to select the filter you want to use.
 - c. In the Parameters field, type any parameters required to get the desired result. For information about the parameters associated with the filters provided with the ImageUltra Builder program, see the ImageUltra Builder help system.
 - d. Click **OK**.
 - e. If you assign multiple filters to a module, click either the **All filters must be satisfied** or **One or more filters must be satisfied** radio button to define how the filters are to be used.
15. In the Options tab field, do the following:

- a. The “Path to copy files on the preload” field defines whether or not the device-driver installable files should be copied to the C: partition.
 - If you type a path in this field and check the **INF Installable** checkbox on this tab, the installable files will be copied to the C: partition and the device drivers will be made available to the Windows setup or mini-setup program.
 - If you type a path in this field and do not check the **INF Installable** checkbox on this tab, the installable files will be copied to the C: partition, but the device driver will not be made available to the Windows setup or mini-setup program.

Type a path only if you want the installable files to be copied to the C: partition of the target computer.

- b. In the For INF Install Only area, check the **INF Installable** check box. If the location of the .INF file is not in the source directory you defined on the Source tab, use the “Relative location of the INF files in the source” field to type the relative path to the subdirectory that contains the .INF file. For example, if the source directory is X:\SOURCE_FILES\DRIVER_1, but your .INF file is in X:\SOURCE_FILES\DRIVER_1\WINXP, the relative path is \WINXP.
 - c. All other fields on the Options tab are used to handle special cases. For information about special fields, refer to Chapter 9, “Dealing with exceptions” on page 75.
16. Click the **Save** icon in the Main window tool bar.
 17. To build a driver module, see “Building a new module” on page 72.

Preparing source utility files

The use of utilities is optional. Utilities are DOS-based EXE, BAT, or COM programs that run during the installation process. For example, an installer might want to run CHKDSK to determine the condition or size of a hard disk before any files get installed.

Utilities reside in the service partition and do not get installed on an active partition. Little preparation is required for the source files. The following are the only requirements:

- The utility must be in a ready-to-run format. It cannot require an installation process.
- The utility must be placed in a dedicated folder on a drive that is accessible by the ImageUltra Builder console.
- The utility folder structure must be maintained.

When the source files meet these requirements, you can set the attributes for the new utility as follows:

1. Open the repository where you want the new utility module to reside.
2. In the Components area of the Repository window, click **Utilities**.
3. From the Main Window menu bar, click **New** and then click **Utility...** The New Module Wizard opens.
4. Click **Next**.
5. Verify that the **Utility** radio button is selected, and then click **Next**.
6. In the Description field, type a name that you want to use to identify this module. Be as specific as possible. This name will be the name that appears in the repository.

7. Click **Next**.
8. Do one of the following:
 - If you want to base the attributes for this module on a pre-existing utility module, select the appropriate module and then click **Next**.
 - If you do not want to base the attributes for this module on another module, just click **Next** without selecting a module.
9. Click **Finish**. The New Module Wizard closes and the Utility window for the new module opens.
10. In the General tab, type the full version of the utility.
11. In the Languages tab, put a check mark next to each language for which the utility is intended to be used.
12. In the Source tab, do the following:
 - a. In the Source Directory field, type the complete path to the folder that contains the source utility files.
 - b. In the Silent Install Command area, type the command that is to be used to start the utility in the Command field and any associated parameters in the Parameters field. For example, your command might be `CHKDSK.EXE` and your parameters might be `/F>`.
13. Click the **Save** icon in the Main window tool bar.
14. To build a utility module, see “Building a new module” on page 72.

Preparing source filter files

You typically use filters to obtain hardware-related information from a target computer to determine if a specific module or set of modules should be installed. The ImageUltra Builder program provides filters that check the machine type, machine type and model, computer hardware platform (desktop or mobile), and whether or not the target computer is an IBM computer. You can also develop your own filters if needed.

Filters are DOS-based EXE, BAT, or COM programs that run during the installation process. In most cases, filters query the target computer BIOS for information, but filters can query any installed hardware that stores identification information in a ROM or EEPROM module. For example, you might want to check for the presence of a PCI modem before installing a modem-related application.

The filter must return a “1” for a true condition and a “0” for a false condition. A true condition results in the installation of the modules associated with that filter.

If you decide to create your own filters, you must prepare the filter files as follows:

- The filter must be able to be run from a DOS command prompt using a single command and, optionally, a set of parameters.
- The filter must be in a ready-to-run format. It cannot require an installation or unpacking process.
- The filter must be placed in a dedicated folder on a drive that is accessible by the ImageUltra Builder console.

When the source files meet these requirements, you can set the attributes for the new filter module as follows:

1. Open the repository where you want the new filter module to reside.
2. In the Components area of the Repository window, click **Filters**.

3. From the Main Window menu bar, click **New** and then click **Filter...** The New Module Wizard opens.
4. Click **Next**.
5. Verify that the **Filter** radio button is selected, and then click **Next**.
6. In the Description field, type a name that you want to use to identify this module. Be as specific as possible. This name will be the name that appears in the repository.
7. Click **Next**.
8. Do one of the following:
 - If you want to base the attributes for this module on a pre-existing filter module, select the appropriate module and then click **Next**.
 - If you do not want to base the attributes for this module on another module, just click **Next** without selecting a module.
9. Click **Finish**. The New Module Wizard closes and the Filter window for the new module opens.
10. In the General tab, type the full version of the filter in the Version field.
11. In the Languages tab, put a check mark next to each language for which the filter module is intended to be used.
12. In the Source tab, do the following:
 - a. In the Source Directory field, type the complete path to the folder that contains the source filter files.
 - b. In the Silent Install Command area, type the command that is to be used to start the filter in the Command field and any associated parameters in the Parameters field. For example, your command might be `FILTERX.EXE` and your parameters might be `/S`.
13. Click the **Save** icon in the Main window tool bar.
14. To build a filter module, see “Building a new module”.

Building a new module

Before you can build a module, your source files must be prepared and accessible to the ImageUltra Builder console, and the attributes for the new module must be filled in. When these conditions are met, you can use the following procedure to build your new module:

1. Open the repository where the module entry resides.
2. In the Components area, click the applicable module category (**Operating Systems, Applications, Device Drivers, Filters, or Utilities**).
3. In the right pane, click the applicable module entry.
4. In the Main window menu bar, click **Tools**, and then click **Build Module...**
5. Follow the instructions on the screen.

Chapter 8. Getting help and support

The ImageUltra Builder program is supported by IBM. If you have a problem with the ImageUltra Builder or have questions about a specific feature, a variety of sources are available to help you including documentation, an integrated help system, Web pages, and Web-based technical support.

Support for third-party tools that might be used in conjunction with the ImageUltra Builder (such as Symantec Norton Ghost, PowerQuest DeployCenter, or the Microsoft Sysprep program) is provided by the manufacturers of those tools, not IBM.

Refer to your IBM ImageUltra Builder license for a full explanation of IBM support, terms and duration. To access the license, do the following:

1. Start the ImageUltra Builder program.
2. From the master window, click **Help**, and then click **View the license**.

Be sure to retain your proof of purchase.

Note: ImageUltra Builder is not a warranted product. IBM will provide assistance to resolve problems and answer questions about the ImageUltra Builder, but IBM is not obligated to update this version of the code, to enhance its functionality, to resolve problems, or to make future versions of this product available at no charge. Any decision to make updates available or to charge fees for updates is solely at the discretion of IBM.

Using the documentation and help system

Many problems can be solved without contacting IBM for assistance. If you experience a problem or have a question about the operation or functionality of the ImageUltra Builder, begin with the online help system or this publication, the *ImageUltra Builder User Guide*.

For example, if you need assistance with conceptual information, use the *ImageUltra Builder User Guide*. If you need help with a specific task or procedure, use the help system. To access the help system, either press **F1** or click **Help** in the Main window menu bar, and then click **View help topics**. The help system will open to a topic related to the active window in the ImageUltra Builder workspace. You can select other topics from the contents, or use the search or index functions to find a particular word or phrase.

Using the Web

The IBM ImageUltra Web site provides the latest technical information, any downloadable updates that IBM chooses to distribute, and a listing of HIIT-enabled IBM personal computers. To visit the ImageUltra Web site, go to <http://www.pc.ibm.com/qtechinfo/MIGR-44316.html>.

Contacting an ImageUltra Builder technical expert

Technical assistance for the ImageUltra Builder is available through the IBM ImageUltra Web site. You can contact an ImageUltra technical expert through the e-mail service provided on the IBM ImageUltra Web site. To access the Web site, go to <http://www.pc.ibm.com/qtechinfo/MIGR-44316.html>.

Chapter 9. Dealing with exceptions

The ImageUltra Builder program is designed to be extremely flexible. This design encourages exploration as you and your team become comfortable building images and modules. As your image-building process evolves, there might be times when you want to accomplish tasks not covered in the previous chapters or in the ImageUltra Builder help system. The purpose of this chapter is to provide you with solutions to special exceptions that you might encounter as you develop your image-building process.

The information in this chapter provides an overview on how to deal with exceptions. For detailed step-by-step instructions about working with map or module attributes, refer to the ImageUltra help system.

Application module exceptions

There are several ways that application modules are created to store important information that is not directly related to application source files. The following conditions explain some of the unique exceptions pertaining to applications.

Condition: Adding non-installable application files (or files of any kind) to Ultra-Portable and Portable-Sysprep Images.

Solution: The ImageUltra Builder program enables you to build an application module that simply copies files to a specific location on the target computer hard disk. This method is optional and cannot be used for Hardware-Specific Images.

The process for creating this type of application module is similar to creating a standard application module. However, there are slight differences when you specify the attributes. After you have your source files prepared, use the New Module Wizard to create a new application module as you would a standard application module, but allow for the following exceptions when you fill in the attributes:

- In the **Source tab**, fill in the **Source Directory** field with the complete path to the folder containing the non-installable files for the module you are creating
- In the **Source tab**, do not enter any commands for the **Silent Install Command** field, the **Preinstall Command** field, or the **Postinstall Command** field.
- In the **Options tab**, type the full path to where the files are to be copied in the **Path to copy installable files** field.

After specifying attributes for the application module, click the **Save** icon located in the Main window toolbar. Then, build the module.

Condition: Creating batch files to install applications for Ultra-Portable and Portable-Sysprep Images.

Solution: Certain applications require significant logic in order to install correctly. The ImageUltra Builder program enables you to install these types of applications using a batch file. This method is optional and cannot be used for Hardware-Specific Images.

The process for creating an application module that uses a batch file for installation is similar to creating a standard application module. However, there are slight differences when you specify the attributes. After you have your source files prepared, use the New Module Wizard to create a new application module as you would a standard application module, but allow for the following exceptions when you fill in the attributes:

When creating an application module that contains a batch file, create the module as if it were a standard application module, taking into consideration the following exceptions:

- In the **Source tab**, complete the **Source Directory** field with the entire path to the folder containing the batch file and related application files for the module you are creating
- In the **Source tab**, complete the **Silent Install Command** field with the name of the batch file
- In the **Options tab**, type the full path to where the files are to be copied in the **Path to copy installable files** field.

Note: Using a batch file to install an application requires that the application files and the batch file are the only files that exist in the directory. You can have as many batch files as needed, but only one batch file will be executed from the silent install command.

After specifying attributes for the application module, click the **Save** icon located in the Main window toolbar. Then, build the module

Condition: Providing hardware detection for applications that do not have it.

Solution: You might have an application that is required for the proper operation of a specific hardware device, but does not have hardware detection capabilities. For example, many CD-RW applications do not have hardware detection. If you already have a separate program to detect hardware, or if you have the resources to develop one, the ImageUltra Builder program enables you create an application module that includes a program that does hardware detection and then installs the application, if it is needed.

The process for creating this type of application module is similar to creating a standard application module. However, there are slight differences when you specify the attributes. After you have your source files prepared, use the New Module Wizard to create a new application module as you would a standard application module, but allow for the following exceptions when you fill in the attributes:

- In the **Source tab**, complete the **Source Directory** field with the entire path to the folder containing the hardware detection program and related application files for the module you are creating.
- In the **Source tab**, complete the **Silent Install Command** field with the name of your hardware detection program that first searches for the hardware and then installs the application if the hardware is present. Do not enter commands for the **Preinstall Command** field, and the **Postinstall Command** field.
- In the **Options tab**, type the full path to where the files are to be copied in the **Path to copy installable files** field.

After specifying attributes for the application module, click the **Save** icon located in the Main window toolbar. Then, build the module.

Device Driver exceptions

You can manipulate device-driver modules for various types of device drivers *and* applications. The following conditions explain some of the exceptions pertaining to device drivers.

Condition: Pointing to an .INF file that exists in a source file subdirectory.

Solution: Typically, when building a device-driver module, the .INF file is located in the root directory with other source files used for that device driver. In some cases, a device driver might contain an .INF file in a subdirectory. The ImageUltra Builder program allows for .INF files in subdirectories.

The process for creating a device-driver module that points to an INF file subdirectory is similar to creating a standard device-driver module. However, there is one slight difference when you specify attributes. After you have your source files prepared, use the New Module Wizard to create a new driver module as you would a standard device-driver module, but allow for the following exception when you fill in the attributes:

- In the **Options tab**, complete the **Relative location of INF files in the source** field with the relative path to the subdirectory that contains the INF file. For example, if the source file is X:\SOURCE_FILES\DRIVER_1, but your .INF file is in X:\SOURCE_FILES\DRIVER_1\WINXP, the relative path is \WINXP.

After specifying attributes for the device-driver module, click the **Save** icon located in the Main window toolbar. Then, build the module.

Condition: Preparing a device-driver module that installs using an .INF file, but also has an application executable packaged with it.

Solution: Some applications are hardware dependent. These applications are sometimes designed to contain *both* device-driver *and* application files. For example, you might have source files that contain a wireless device driver intended for a wireless network card and a configuration utility intended to work in conjunction with the network card.

In this case, if you make an *INF-only device-driver module* from the application source files, the application might fail to install correctly. Conversely, if you make an *application module* from the application source files, the device driver might fail to install correctly.

Although there are several ways to address this exception, the easiest and most successful solution is to make a device-driver module that is *both* an INF installable *and* an executable. This method is optional and cannot be used for Hardware-Specific Images.

The process for creating this type of device-driver module is similar to creating a standard device-driver module. However, there are slight differences when you specify the attributes. After you have your source files prepared, use the New Module Wizard to create a new device-driver module as you would a standard device-driver module, but allow for the following exceptions when you fill in the attributes:

- In the **Options tab**, check the **INF installable** check box.
- In the **Source tab**, complete the **Silent Install Command** field with the command used to install the application.

After specifying attributes for the device-driver module, click the **Save** icon located in the Main window toolbar. Then, build the module. After building the module, insert the module into the appropriate driver map. When the module is installed with the image, the INF files are copied first into the Windows driver directory, and then the silent install command is executed during the Windows audit boot.

Note: Because you are putting a device-driver module with an executable in a driver map, there are some special considerations.

If the device is a non-Peripheral Component Interconnect (non-PCI) device, such as a USB or PCMCIA, the following guidelines apply:

- The INF files are put in the Windows driver directory.
- When the non-PCI device is used for the first time, Windows plug-n-play hardware detection occurs, assuming the module was built correctly.
- The SETUP.EXE file does not get executed unless the manufacturer designed the .INF file to execute the program when Windows plug-n-play installs the device. If the manufacturer did not design the .INF file to execute the program when Windows plug-n-play installs, then it is the responsibility of either the person who performs the installation process, or the end user, to install the application the accompanies the device they are using.

Building a module using this method enables you to use the same device-driver module in a base map that was used in a driver map, whether or not the non-PCI device is present. The contents of the .INF file determines whether or not the associated application gets installed automatically.

If the device is a PCI device, then the following guidelines apply:

- The .INF files are put in the Windows driver directory if you checked the **INF installable** checkbox in the **Options tab** when you specified attributes for that module and if the device is detected on the target computer.
- The PCI device executes the silent install command, if you entered a **Silent install** command on the **Source tab** when you specified attributes for that module and if the device is detected on the target computer.

If the device is not detected on the target computer, then the .INF files *are not* put in the Windows driver directory and the PCI device *does not* execute the silent install command.

Editing Unique Scripts

The ImageUltra Builder program enables you to specify whether or not you want to view, and edit, build scripts whenever you build a module. In most cases, the program generates the build scripts by processing information that is contained in the module attribute tabs. Editing scripts is optional, and is typically done to troubleshoot problems or specify installation options that are not provided by the ImageUltra Builder program.

Important: Base operating-system modules for Portable-Sysprep Images and Hardware-Specific Images do not generate a build script when you build a module. All other types of modules do generate a build script during the build process.

Condition: Editing Unique Scripts

Solution: If you choose to edit build scripts, some of the file extensions might be familiar while others might not. The following is a list of unique files that contain build scripts and a brief description of each:

- PL and BAT files: These files are run during DOS mode (immediately before the Windows installation begins, even before the audit boot). Editing these scripts requires using DOS commands only.
- CMD files: These files are executed during the Windows audit boot. Editing these scripts requires using Windows commands only. CMD files apply to Windows XP installations only, and only if Windows XP is checked in the OS/Language tab for that particular module.
- FM and FM2 files: Generally, these files are used by the Fmodify.exe program. This program is used to automatically modify the contents of other files. In most cases, an FM2 file functions similarly in Windows 2000 as a CMD file functions in Windows XP.

Notes:

1. If you are not familiar with the Fmodify.exe program, it is available in the tools directory of the module repository and is made when the repository is created. To get more information regarding editing FM file scripts, go to the tools directory and enter the following DOS command: `fmodify /?`
2. If you rebuild a module after it is edited, then you have to edit the scripts again the next time you build the module. It is a good idea to track your edits carefully and record them in a separate file for safekeeping.

Including Rapid Restore PC

When you create a base map, the ImageUltra Builder program enables you to choose how you want the target computer service partition to behave during the installation process. If you plan on including the Rapid Restore PC program in your Smart Image, there are some “best practices” you might want to follow to help ensure the integrity of your service partition and possibly cut down on installation time.

Condition: Including Rapid Restore PC in a Smart Image.

Solution: When using the New Map Wizard to create a map, you are given three choices that define the behavior of the target computer’s service partition during the installation process: **Delete none**, **Delete unused**, and **Delete all**.

Because the Rapid Restore PC program reserves space in the service partition, choosing any action that affects the size of the service partition can have an adverse affect on the Rapid Restore PC program, either in performance or function. If you plan to include the Rapid Restore PC program as an application module in your Smart Image, it is a good idea *not* to install the Rapid Restore PC program as part of the installation process, but instead, copy the Rapid Restore PC installable files to the target computer hard disk during installation and provide an icon on the Windows desktop so it can be installed after the normal installation process is complete. The ImageUltra Builder provides accommodations for this type of installation. For the following reasons, this is a good practice to observe regardless of which service-partition behavior action is defined in your base map:

- In most cases, you want the first Rapid Restore PC backup operation to reflect the state of the hard disk after the complete image has been installed.

- By installing the Rapid Restore PC program after the image has been installed, it does not make any difference which service-partition behavior you choose. The Rapid Restore PC installation program will either resize an existing service partition or create a service partition of the correct size if a service-partition does not already exist.

This practice reduces installation errors, installation time, and provides a true backup image.

To make Rapid Restore PC an installable application from an icon on the desktop, create the application module as if it were a standard application module, but do the following:

- In the **Options tab**, use the drop-down menu in the **Install hook** field and select “Install when customer chooses via desktop shortcut.”

Changing the order of module installation

The ImageUltra Builder program enables you to change the order in which modules are installed. Changing the order of module installation is optional, but can be useful if you have modules that are dependent on one another, or if you have modules that collide during the installation process.

Condition: Resolving installation problems by changing the order of module installation.

Solution: The installation order for modules is automatically controlled by the ImageUltra Builder program. However, you can override the default installation order for add-on operating-system modules, application modules, and some device-driver modules by using the **Install hook** and **Install slot for audit boot** fields located in the **Options tab**. Typically, it is not necessary to change the install slot or install hook unless you know one of the following to be true:

- One or more modules have dependencies on other modules to be installed first
- You have modules that are colliding
- The image is not installing correctly

By default, all application modules and all add-on operating-system modules are installed during the audit boot. The audit boot is the first boot cycle after Windows setup is complete and occurs before the customer first boot. In some special cases, you can configure a device-driver module to install during the audit boot if it has components that have a silent-install command.

The ImageUltra Builder program automatically assigns install slots to control the order of installation during the audit boot. The install-slot number defined in the **Install slot for audit boot** field determines when a module is installed in relation to other modules that are installed during the audit boot. Install-slot numbers range from 1 to 9. The higher the install-slot number, the later the module will be installed during the audit boot. Install slots 1 through 5 are most commonly assigned by the ImageUltra Builder program. So, if you need to ensure that a module is installed late in the audit boot, you should assign an install slot higher than 5. If you have a dependency on one module being installed before installing another module, assign a lower install-slot number to the module that needs to be installed first; then, assign a higher install-slot number to the module that needs to be installed later. In order for the install slot number to be recognized, the **Install hook** field must be set to **Install during audit boot**.

If necessary, you can use the **Install hook** field to change when a specific module will be installed in the overall installation process. The following choices are available from the Install hook drop-down menu:

- **Install during audit boot (default):** This selection enables the module you are creating to be installed during the first boot cycle *after* Windows setup is complete.
- **Install during customer first boot:** This selection enables the module you are creating to be installed after the audit boot, at the time you enter personal identification information.
- **Install when customer chooses via desktop shortcut:** This selection enables users to install the module anytime after the image is installed by clicking a shortcut icon on the desktop.

If you choose any option other than **Install during audit boot**, the install slot number defined in the **Install slot for audit boot** field will be ignored.

The process for creating a module with a unique install slot is similar to creating a standard module. However, there are slight differences when you specify the attributes. After you have your source files prepared, use the New Module Wizard to create a new module as you would a standard module, but allow for the following exceptions when you fill in the attributes:

- In the **Options tab**, make sure the **Install hook** field is set to **Install during audit boot**.
- In the **Options tab**, use the drop-down menu provided in the **Install slot for audit boot** field to select the desired install slot.

After specifying attributes for the module, click the **Save** icon located in the Main window toolbar. Then, build the module.

Tip: If you are installing an Ultra-Portable Image or Portable-Sysprep Image that contains a Windows XP operating system, a log is generated that identifies the modules that were installed and when the modules were installed. This log can be helpful in determining if your image has modules that are dependent on one another, or if modules are colliding. The *Modules.log* file is located in the C:\IBMTOOLS directory. This function is not available on target computers that contain a Windows 2000 operating system.

Appendix. Notices

IBM may not offer the products, services, or features discussed in this document in all countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.*

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM
ImageUltra
Rapid Restore
ThinkPad

Microsoft, Windows, and Windows NT are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Index

A

- access the license 73
- add-on operating-system modules 4, 9
- adding
 - application modules to a base map 48
 - device-driver modules to a driver map 53
 - menu items to a base map 47
 - menu items to a driver map 52
 - non-installable application modules to Ultra-Portable and Portable-Sysprep Images 75
 - operating-system modules to a base map 48
- additional device drivers 7
- advanced functions
 - filters 17
- advantages of importing modules 10
- append the image during the deployment and installation process 5
- application 9
 - modules 9
- application module exceptions 75
- application modules, location of 4
- application programs 7
- application window 30
- assigning filters 14
- attribute, setting 30
- attributes, definition of 30
- attributes, map 13

B

- backing up repository 20
- base map
 - assigning filters 14
 - capabilities 1
 - creating 33
 - inserting menu items 33
 - inserting modules 35
 - menu items, base map purpose 33
 - New Map Wizard 13, 14
 - simple and complex 43
 - steps 33
 - tree structure 33
- base map links
 - definition 40
 - purpose 40
- base maps 13, 29
- base module
 - contents 4
 - location 4
- base operating-system module 9
 - definition 4
 - storing 4
- building a module 11
 - New Module Wizard 11
 - preparing files 11

- building a module (*continued*)
 - add-on operating-system modules 11
 - application files 11
 - device-driver files 11
 - virus checking 11
- building a new module 72

C

- categories, module 9
- change the network-sync setting 42
- CMD files 79
- combining modules 9
- common base module 4
- compatibility between operating systems 20
- complex base map 43
- constructing a device-driver map tree structure 44
- constructing maps 33
- contacting an ImageUltra technical expert 74
- containers 10
- controlling the network-sync function 42
- controlling the service partition behavior 42
- conventions used within the interface 24
- create and build a base operating-system module 5
- creating
 - a new base map 47
 - a new driver map 52
 - a new repository, considerations 20
 - base and driver maps 13
 - base map tree structure 33
 - base operating-system module from a Portable-Sysprep Image 12
 - base operating-system module from Hardware-Specific Image 12
 - batch files to install applications 75
 - complex driver maps 44
 - Hardware-Specific Image 61
 - maps 47
 - Portable-Sysprep Image 57
 - simple driver maps 44
 - source operating-system images
 - limitations of PowerQuest DeployCenter 55
- creating and modifying driver maps 51
- creating or modifying base maps and driver maps 13
- creating source operating-system images
 - limitations of PowerQuest DeployCenter 56
 - limitations of Symantec Norton Ghost 55
 - required third-party tools 55
 - Ultra-Portable Images 55

D

- defining service partition behavior 14
- defining user information in a base map 41
- definition, containers 10
- definition, module 9
 - operating system, base 9
- deploying images
 - directly from CD 16
 - network 16
- deploying images, traditional method 1
- deploying Smart Images to target computers 16
- developing an image
 - building a module 11
 - creating or modifying maps maps 13
 - deploying Smart Images 16
 - importing pre-existing modules 10
 - installing images 17
- device drivers, IBM-factory-installed 5
- device-driver 9
 - exceptions 77
 - maps 29
 - modules 9
 - window 30
- device-driver map
 - constructing 44
 - create 44
 - difference from base map 44
 - inserting filters 44
 - inserting menu items 44
 - inserting modules 45
 - modify 44
 - new 44
 - using filters 45
- device-driver modules, location of 4
- direct deployment from CD 16
- distributing IBM-developed modules 3
- documentation 73
- driver map
 - benefit of creating 5
 - capabilities 2
- driver maps 13

E

- editing unique scripts 78
- enabling the network-sync function 15
- establishing a repository 19
- exceptions 75

F

- factory-installed image, replacing 3
- filter
 - creating 38
 - examples of 38
 - purpose of 38
- filter window 30
- filters 17

- FM and FM2 files 79
- Fmodify.exe program 79
- formatting source computer as FAT32 12

G

- getting help and support 73

H

- hardware limitations of
 - Portable-Sysprep Image 6
- Hardware-Specific Image 12, 35
 - anatomy of 6
 - creating and preparing the image files 61
 - definition of 6, 7
 - installing additional drivers and applications 7
 - limitations 6
 - portability of 7
 - setting the attributes for the new base operating-system module 62
- Hardware-Specific Images 3
- help system 21
- HIIT-enabled IBM personal computers, listing of 73
- HIIT-enabled systems 5
- hot fixes, patches, updates, and service packs 4

I

IBM

- Customization Program 57
- Customization Program, running 12
- factory-installed device drivers 5
- HIIT-enabled personal computers 2
- Image Builder program, purpose of 1
- Software Delivery Assistant program 7
- support, terms and duration 73

IBM HIIT-enabled target computer 44

- icons 24
- identifying icons 24
- image
 - definition 1
 - understanding 1
- image process, overview 9
- image types
 - choosing 3
 - dependencies 3
 - Hardware-Specific Images 3
 - Portable-SysPrep Images 4
- images
 - advantages 7
 - characteristics 7
 - disadvantages 7
 - extracted 3
- ImageUltra Builder import tool 10
- ImageUltra Builder interface
 - contents 23
 - functions 23
 - layout 23
 - working with 23

- ImageUltra Builder New Map Wizard 33
- ImageUltra Web site 74
- import tools
 - Network Import Diskette 10
- importing modules, advantages 10
- importing pre-existing modules 10
- including Rapid Restore PC 79
- inserting menu items in a base map 33
- inserting modules in a base map 35
- inserting modules in a driver map 45
- Install hook 80
- Install slot for audit boot 80
- installation process 2
- installation process menus 13
- installing an Ultra-Portable Image on a non-IBM computer 5
- installing assorted models of IBM PCs 4
- installing images 17
- Installing the ImageUltra Builder
 - getting help after installation 21
- Installing the ImageUltra Builder program 19
 - considerations 20
 - limitations 20
 - limitations Windows 2000 and Windows XP 21
 - minimum requirements 19
 - disk space 19
 - minimum networking speed 19
 - operating system 19
 - removable-media drives 19
 - prerequisites 19
- integrated help system 73
- IT departments, challenges 1

L

- licensing 19
- links
 - definition of 40
 - example of 40
 - purpose of 40

M

- main menu
 - menu bar 23
 - status bar 24
 - tool bar 23
 - workspace 24
- main window
 - layout 23
- make RRPC an installable application
 - located on the target computer's desktop 80
- map
 - assigning filters to base 14
 - attributes 13
 - base 13
 - driver 13
 - tree structure 13
- map construction 33
- map windows
 - attributes 29
 - components area 29

- map windows (*continued*)
 - contents 29
 - using 29
- map, installation menus 13
- menu items, using MTMs 44
- menus 13
- Microsoft Plug-n-Play 68
- Microsoft Sysprep 19
- Microsoft Sysprep tool 3, 5
- Microsoft WHQL device drivers 20
- mini-setup 6
- minimizing installation time 4
- modifying attributes 30
 - determining status 30
 - icon identification 30
 - promoting, maps 30
 - promoting, modules 30
 - properties identification 31
- modifying base and driver maps 13
- modifying base maps 47
- module categories 9
- module repository
 - changing 28
 - defining 28
 - opening 28
- module, creating an operating system 12
- module, definition 9
- modules, application 9
- modules, combining 9
- modules, device-driver 9
- modules, importing 10
- modules, operating system base
 - add-on 9
- MS-DOS 19

N

- network deployment 16
- Network Import CD 10
- Network Import Diskette 10
- network-sync function, enabling 15
- network-sync setting, change 42
- new base map 33
- New Map Wizard 13, 14, 44
 - accessing 33
 - functions 33
- New Module Wizard 13
- New Repository Wizard, considerations 20
- non-PCI device 78

O

- object attribute windows
 - components area 30
 - object entries 30
- operating system window 30
- operating system, add-on 9
- operating-system modules 9
- overview
 - base maps 13
 - driver maps 13
- overview of the image process 9
- overQuest DeployCenter 19

P

- PCMCIA 78
- PL and BAT files 79
- pointing to an INF file that exists in a source file subdirectory 77
- portability, definition of 2
- Portable-Sysprep image
 - portability 6
- Portable-Sysprep Image 5, 35, 44
 - anatomy of 5
 - creating and preparing the image files 58
 - definition 5
 - deployed 6
 - initial setup time 6
 - making the IBM Customization Program accessible 57
 - setting attributes for the new base operating-system module 60
- Portable-SysPrep Image 12
- Portable-Sysprep Images 3
- PowerQuest DeployCenter 2, 57
- PowerQuest DeployCenter, considerations 20
- PowerQuest DeployCenter, for Portable-Sysprep Images 5
- predefine default settings 41
- predefine default user settings 4, 6
- preparing
 - adding mini-applications 64
 - after files are prepared 64
 - for a silent install 63
 - making files accessible to the ImageUltra console 64
 - setting attributes for a new add-on operating-system module 66
 - setting attributes for a new application module 64
 - source device-driver files 68
 - source files for application modules and add-on operating-system modules 63
 - source filter files 71
- preparing a device driver module that works as an INF installable, and as an application executable 77
- preparing source files 55
- providing hardware detection for applications 76

R

- Rapid Restore PC program 3
- Rapid Restore PC, including 79
- reducing diagnostic time 5
- repository window 28
 - building the module 29
 - components area 28
 - contents of 28
 - creating an object module entry 29
 - icon, purpose 29
 - purpose of 28
- repository, backing up 20
- running the IBM Customization Program 12

S

- service partition
 - creating 3
 - definition of 3
 - purpose of 3
 - recovery mechanism 3
- service partition, change behavior 42
- service partition, defining 14
- setting attributes, object attribute window 30
- similar computer
 - deploying Portable-Sysprep Image to 6
 - factors 6
- simple and complex base maps 43
- simple base map 43
- Smart Image 33, 44
 - advantages 4
 - characteristics 2
 - definition 1
 - disadvantages 4
 - function 4
 - Hardware-Specific Image, definition of 2
 - Portable-Sysprep Image, definition of 2
 - supported images 2
 - types 2
- Ultra-Portable Image, definition of 2
- Smart Images, deploying 16
- specifying
 - base map attributes 49
 - base-map menu-item attributes 50
 - driver map attributes 53
- stand-alone CDs, considerations 20
- steps to creating a base operating-system module 57
- steps to develop an image
 - building a module 11
 - creating or modifying maps 13
 - deploying Smart Images 16
 - importing pre-existing modules 10
 - installing images 17
- storage area for deployed Smart Images 3
- summary of image types 7
- support for third-party tools 73
- Symantec Norton Ghost 2, 19, 57
- Symantec Norton Ghost, for Portable-Sysprep Images 5
- Sysprep
 - importance of 3
 - using 3
- SysPrep mini-setup 41

T

- technical assistance 74
- test Smart Images 30
- test state 30
- third-party image-cloning tool 2
- tools
 - build tool 31
 - get IBM customization program 31
 - import tool 31
 - promote tool 31

tools (*continued*)

- update network sync settings tool 31
- tree structure, map 13

U

- Ultra-Portable Image 35, 44
 - anatomy of 4
 - characteristics 4
 - definition 4
 - hardware detection 4
 - hardware independent 4
 - install time 4
 - life-cycle 4
 - modules 4
 - portability of 4
 - using with driver maps 5
 - Windows setup 4
- unique user settings 41
- USB 78
- user and hardware information 5
- user and hardware information, removing 3
- user information
 - implementation 42
 - located 42
- user settings 4
- using
 - a driver module in a base map 77
 - an application module in a driver map 77
 - Microsoft Sysprep tool 3
 - Sysprep 3
 - using filters in a base map 38
 - using filters in a driver map 45
 - using links in a base map 40
 - using the documentation and help system 73
 - using the Web 73
 - using utilities in a base map 37
- utilities 17, 18
 - assigning to base maps 37
 - assigning to menu items 37
 - setting execution of 37
- utility window 30

V

- virus checking 11

W

- web-based technical support 73
- WHQL-certified device drivers 88
- Windows 41
- working with base map and menu item attributes 49



Part Number: 31P9058

Printed in U.S.A.

(1P) P/N: 31P9058

