

**UNIVERSITÀ DEGLI STUDI DI CAMERINO**

**FACOLTÀ DI SCIENZE E TECNOLOGIE**

*Corso di Laurea in Informatica*

*Dipartimento di Matematica e Informatica*



# **EVOLUZIONE DEI DISPOSITIVI EMBEDDED**

# **XPE**

Tesi di Laurea compilativa  
In Reti di calcolatori

*Laureando*  
**Claudio Di Rosa**

*Relatore*  
**Dott. Fausto Marcantoni**

*Correlatore*  
**Dott. Federico Paoloni**



## Indice

<b>Introduzione</b> .....	<b>1</b>
<b>Capitolo 1 - Windows XP Embedded</b> .....	<b>5</b>
1.1 Panoramica di Windows XP Embedded.....	5
1.1.1 Panoramica.....	5
1.1.2 Scenari di utilizzo .....	6
1.1.3 Funzioni particolari .....	6
1.1.4 Modalità di funzionamento del sistema .....	8
1.1.5 Differenze con Windows XP Professional .....	12
1.2 Storia dei sistemi Embedded di Microsoft.....	13
1.3 Strumenti e requisiti per la costruzione di un sistema Windows XP Embedded.....	23
1.4 Alternative a Windows Embedded .....	27
1.5 Vantaggi e Svantaggi per le aziende .....	29
1.6 A chi è rivolto principalmente il sistema.....	32
1.7 Una soluzione per i moderni NetBook .....	32
1.8 Enhanced Write Filter (EWF) in Windows Embedded.....	33
1.8.1 Panoramica.....	33
1.8.2 Tipi di configurazione .....	34
1.8.3 Configurazione EWF basata su disco.....	35
1.8.4 Configurazione EWF basata su RAM .....	36
1.8.5 EWF Manager Applicazion .....	37
1.8.6 Principali errori.....	39
1.8.7 Suggerimenti .....	39
1.8.8 Considerazioni.....	40
<b>Capitolo 2 – Windows Embedded vs Windows CE Embedded</b> .....	<b>43</b>
2.1 Panoramica Windows CE Embedded .....	43
2.2 Requisiti per l’utilizzo e principali utilizzi .....	47
2.3 I due sistemi a confronto .....	48
<b>Capitolo 3 – Caso particolare di utilizzo</b>	
<b>(IL CHIOSCO INFORMATIVO)</b> .....	<b>55</b>
3.1 Panoramica.....	55

3.2 Una soluzione con PC di HW minimo e SW installato sull'Hard Disk .....	56
3.2.1 Requisiti .....	56
3.2.2 Come realizzare il sistema.....	56
3.2.3 Considerazioni .....	61
3.3 Una Soluzione con PC di HW minimo e SW installato su Server ..	63
3.3.1 Requisiti .....	63
3.3.2 Panoramica sul Preboot Execution Environment .....	64
3.3.3 Panoramica sul Remote Boot Manager e Config. DHCP.....	68
3.3.4 Come realizzare il sistema.....	69
3.3.5 Considerazioni .....	76
<b>Capitolo 4 – Caso particolare di utilizzo</b>	
<b>(IL LETTORE MULTIMEDIALE) .....</b>	<b>79</b>
4.1 Requisiti .....	79
4.2 Panoramica sulle custom shell e sulla Creazione di componenti ...	80
4.3 Realizzazione del sistema.....	86
4.4 Considerazioni .....	93
<b>Capitolo 5 – Conclusioni .....</b>	<b>95</b>
<b>Bibliografia.....</b>	<b>99</b>

## Indice delle figure

FIGURA 1 – SCELTA COMPONENTI IN TARGET DESIGNER.....	5
FIGURA 2 – PROGETTAZIONE COMPONENTI.....	9
FIGURA 3 – PERSONALIZZAZIONE DELLE IMPOSTAZIONI.....	10
FIGURA 4 – VERIFICA DELLE DIPENDENZE.....	11
FIGURA 5 – PROCESSO DI GENERAZIONE DI WINDOWS XP EMBEDDED.....	23
FIGURA 7 – COFIGURAZIONE IMPOSTAZIONI EWF BASATO SU DISCO.....	35
FIGURA 8 – COFIGURAZIONE IMPOSTAZIONI EWF BASATO SU RAM.....	36
FIGURA 9 – SCHERMATA DI UNA INSTALLAZIONE WINDOWS CE.....	43
FIGURA 10 – SCHERMATA DI UNA INSTALLAZIONE WINDOWS MOBILE.....	45
FIGURA 11 – ESECUZIONE PROGRAMMA TAP.EXE.....	56
FIGURA 12– IMPOSTAZIONI PROGETTO WINDOWS XPE.....	57
FIGURA 13– CONFIGURAZIONE TARGET DESIGNER CHIOSCO CON DISCO.....	59
FIGURA 14 – PROCEDURA DI FIRST BOOT AGENT (FBA).....	60
FIGURA 15 – WINDOWS XPE IN ESECUZIONE.....	61
FIGURA 16 – CONFIGURAZIONE EWF IN CHIOSCO CON DISCO.....	63
FIGURA 17 – CONFIGURAZIONE PORTA 60 SU DHCP.....	68
FIGURA 18 – CONFIGURAZIONE COMPONENTE SYSTEM CLONING TOOL.....	69
FIGURA 19 – REBOOT DOPO FBA NELLA CONFIGURAZIONE DELL'IMMAGINE PER CHIOSCO SENZA HARD DISK.....	71
FIGURA 20 – SDI LOADER PER LA CREAZIONE DEL DISCO VIRTUALE.....	72
FIGURA 21 – INIZIALIZZAZIONE DEL DISCO VIRTUALE.....	72
FIGURA 22 – CREAZIONE IMMAGINE RAMDISK.SDI.....	73
FIGURA 23 – REMOTE BOOT MANAGER.....	75
FIGURA 24 – IMMAGINE DEL PROGRAMMA INCTRL5.....	83
FIGURA 25 – IMMAGINE DEL PROGRAMMA REGMON.....	83
FIGURA 26 – FBA RESOURCE CUSTOMIZATION.....	85
FIGURA 27 – PARAMETRI NUOVO COMPONENTE CUSTOM SHELL.....	89
FIGURA 28 – CUSTOM SHELL PER LA VISUALIZZAZIONE DI WINDOWS MEDIA PLAYER.....	92



## Introduzione

Fin dal 2001 Windows Embedded si è presentato come una valida alternativa ai più blasonati sistemi operativi "Full optional", ma orientato ai tecnici, agli amministratori di sistema, a coloro quindi che puntavano soprattutto alla realizzazione di un sistema sicuro, veloce e non eccessivamente costoso, e XPE è stata la risposta a queste esigenze, tra vantaggi e svantaggi questa soluzione si è fatta strada fino ad oggi e con la versione 2009 parecchi passi avanti sono stati fatti.

Spesso negli anni questo sistema ha avuto diversi concorrenti sia interni alla Microsoft come Windows CE o esterni come Linux Embedded, con i quali il sistema si è dovuto progressivamente confrontare.

Nel progetto in analisi tutte questi aspetti sono stati affrontati, facendo una analisi precisa dell'ambiente che andremo pian piano ad analizzare.

Voglio precisare innanzitutto che i progetti sono stati realizzati utilizzando il software Windows Embedded Standard 2009, nonostante spesso le risorse a disposizione sul web sono riferite alla versione 2007. Ho notato comunque che quanto detto per la tale versione è risultato funzionante anche sulla 2009.

Nel primo capitolo, si darà una panoramica del sistema di sviluppo, analizzando i programmi che nei successivi capitoli si andranno ad usare per realizzare il progetto oggetto di studio.

Successivamente, sempre nel primo capitolo si passerà ad analizzare la storia dei sistemi Microsoft e in particolar modo dei sistemi Windows CE e Windows Embedded (XPE) che durante gli anni si sono susseguiti, intralciati e migliorati a vicenda fino a raggiungere l'ottima qualità attuale, a seguire

**Introduzione**

viene fatto un breve confronto tra i due prodotti in modo da vedere i pregi e i difetti di ognuno dei due.

A seguire abbiamo analizzato il principale concorrente che come abbiamo già enunciato è Windows Embedded, e di questi abbiamo messo in evidenza i pregi (l'economicità, la velocità e la bassa richiesta di rilevanti hardware) e dei difetti (pochi componenti a disposizione, costrizione all'uso di determinati hardware supportati, pochi fornitori che producono software libero per tali sistemi, fino ad analizzare quali sono i motivi per cui il sistema XPE potrebbe essere una risposta ai moderni NetBook.

E' stato infine terminato il capitolo primo evidenziando uno dei principali componenti/servizi offerti da Windows XP Embedded ovvero Enhanced Write Filter (EWF) che non è presente nella versione Professional e che è spesso utile in svariati settori e tipi di applicazioni, e che spesso per le sue potenzialità è il principale motivo di utilizzo di un sistema XPE.

Nel secondo capitolo è stato affrontato più approfonditamente il confronto tra Windows CE e XPE, analizzando anche eventuali pregi e difetti dei due sistemi.

Nei successivi due capitoli (3 e 4) vengono analizzati due casi particolari di chiosco, il chiosco informativo e il chiosco multimediale passando per un susseguirsi di passi, che hanno portato alla realizzazione del progetto finale.

Dapprima nel capitolo terzo si è passati dal chiosco con disco e di questo si sono analizzati i pregi e i difetti, passo fondamentale per poter creare un'immagine completa, minimizzando i tempi che nel secondo caso con bootstrap da rete, sono più lunghi, anche se propone altri vantaggi, in questa fase è stato anche analizzato il sistema PXE sia lato server DHCP e lato client.

Infine nell'ultimo capitolo, il quarto è stata data particolare attenzione al sistema utilizzato per il chiosco multimediale,



tramite la realizzazione di una custom shell, altro componente/servizio particolarmente usato nei sistemi Windows Embedded, che permette non solo di avere un sistema customizzato in base all'hardware ma anche un software personalizzato in base all'utente che lo deve utilizzare.



## Capitolo 1 - Windows XP Embedded

### 1.1 – Panoramica di Windows XP Embedded

#### 1.1.1 Panoramica

Windows® XP Embedded è una delle tre versioni a footprint ridotto di Windows, insieme a Windows Embedded CE e a Windows Embedded for Point of Service. Dei sistemi operativi Windows Embedded, soltanto Windows XP Embedded offre la piena funzionalità o almeno il potenziale per la piena funzionalità di un computer client Windows XP Professional, in quanto è basato su Windows XP Professional Service Pack 2 (SP2). Tuttavia Windows XP Embedded è diverso da un normale Windows XP in relazione al livello estremo di modularità. Windows XP Embedded non dispone di quella che si potrebbe chiamare un'immagine di installazione standard.

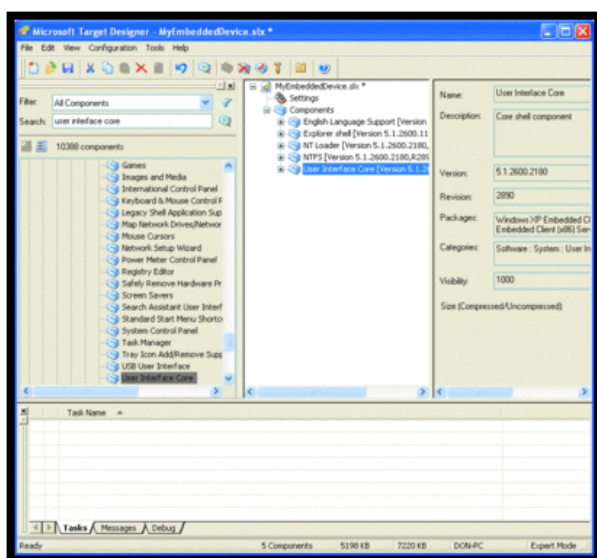


Figura 1 – Scelta componenti in target designer

Viene utilizzato invece un toolkit denominato Windows Embedded Studio per creare un ambiente Windows XP Embedded personalizzato per qualsiasi periferica in fase di installazione e quell'ambiente contiene soltanto le funzionalità di cui la periferica necessita. Ad esempio, se DirectX® o Windows Explorer non sono necessari, semplicemente non

vengono installati. Questa modularità comporta che ogni immagine Windows XP Embedded creata e distribuita avrà meno componenti variabili, per così dire, e questo significa meno lavoro amministrativo e maggiore protezione, come pure la possibilità di essere eseguite su hardware più economici. Alcune delle opzioni disponibili possono essere viste in Target Designer (un componente di Windows Embedded Studio), come illustrato nella Figura 1.

### **1.1.2 Scenari di utilizzo**

Un dispositivo basato su Windows XP Embedded è adatto per un numero crescente di scenari utente. Si considerino, ad esempio, le aziende che gestiscono grandi centri per il servizio clienti. In molti casi, i rappresentanti del servizio clienti necessitano soltanto di una o due applicazioni sviluppate internamente e, forse, di un browser Web. Utilizzano quindi solo una parte delle funzionalità del proprio computer, pertanto Windows XP Embedded consente di ridurre il numero di tali funzionalità al minimo necessario. Minore funzionalità significa meno possibilità di errori o meno possibilità di attacchi alla protezione.

Le periferiche Windows XP Embedded hanno funzionalità limitate, pertanto possono essere eseguite su hardware meno potenti e spesso meno costosi. Numerose aziende producono schede madre che utilizzano i processori compatibili Intel a basso consumo. Queste schede madre sono estremamente piccole, consumano poco e spesso non usano neanche una ventola perché non si surriscaldano. Funzionano piuttosto bene per le periferiche integrate a footprint ridotto, in grado di eseguire soltanto un numero limitato di applicazioni.

### **1.1.3 Funzionalità particolari**

Windows XP Embedded offre alcune opzioni molto interessanti che non sono disponibili nella versione completa di Windows XP. Come ad esempio le funzionalità Enhanced Write Filter

(EFW) e Hibernate Once, Resume Many (HORM) che possono entrambe semplificare le attività di amministrazione di Windows. EWF è un componente facoltativo che è possibile aggiungere alla distribuzione di Windows XP Embedded rendendo di sola lettura un determinato volume su disco. Qualunque attività di scrittura diretta al disco è deviata su un overlay, che può essere posizionato in un altro disco o anche in memoria. Il sistema operativo stesso controlla la zona di overlay EWF e il relativo volume padre come una sola unità, nel senso che li considera come un normale volume del disco. In realtà però il volume padre non viene modificato effettivamente; le modifiche vengono apportate solo all'overlay EWF. In alcuni casi questo può essere un vantaggio. Si supponga di distribuire un'immagine di Windows XP Embedded in una scheda Compact Flash (CF), (le immagini di Windows XP Embedded possono essere anche solo di 5 MB e spesso occupano circa 200 MB, quindi le schede CF hanno tutto lo spazio necessario e anche di più). La scheda CF può essere inserita in un adattatore e collegata ad un connettore IDE della scheda madre, in modo che funga da normale disco rigido. Le schede CF attuali hanno capacità di scrittura limitate (100.000 cicli), pertanto si potrebbe utilizzare EWF affinché il volume CF sia di sola lettura. Si supponga di inserire la sovrapposizione EWF nella RAM: in caso di errori da parte dell'utente, è sufficiente spegnere il computer. La sovrapposizione EWF e tutte le modifiche effettuate al computer vengono perse immediatamente. Al riavvio, si ritorna al punto iniziale memorizzato nella scheda CF. Inoltre, è possibile utilizzare la stessa impostazione per un'immagine su CD o DVD; poiché questo tipo di supporti sono di per sé di sola lettura, EWF cattura tutte le attività di scrittura in un overlay. Anche se l'overlay viene memorizzato in un piccolo disco rigido interno, il volume di avvio originale (su CD o DVD) non viene mai modificato; se è necessario riavviare il

computer nella configurazione originale, è sufficiente eliminare l'overlay EWF.

HORM invece, consente di mettere in stato di ibernazione un computer per poi riprendere l'esecuzione dallo stesso punto diverse volte. Il computer si riattiva quasi immediatamente. È sufficiente configurare il computer ed eseguire tutte le applicazioni necessarie, poi sospenderle. Da quel momento in poi, ogni volta che si accende, il computer diventa operativo in pochi secondi.

Microsoft dispone di un programma Embedded Partner cui partecipano le aziende specializzate nell'uso di Windows XP Embedded. Tuttavia, proprio come molte aziende forniscono ai loro produttori di computer immagini predefinite (che contengono applicazioni aziendali e configurazioni specifiche), è possibile che si desideri creare immagini Windows XP Embedded proprie da condividere con un Embedded Partner che fornisce i sistemi a footprint ridotto e completi. È possibile utilizzare Windows Embedded Studio, che si installa in Windows XP Professional, per creare e gestire le immagini di Windows XP Embedded.

#### **1.1.4 Modalità di funzionamento del sistema**

Per utilizzare il toolkit, occorre dapprima installare la versione completa di Windows XP Professional sull'hardware che si intende utilizzare per il proprio sistema Windows XP Embedded. Ciò consente di adoperare tutte le routine di installazione di Windows XP affinché l'hardware funzioni correttamente e di installare Windows Embedded Studio. In genere, il risultato è l'installazione di un database che viene utilizzato per gestire le immagini incorporate ma è possibile installare lo stesso database altrove, consentendo la creazione di un archivio centralizzato. Successivamente, verrà eseguita un'utilità da riga di comando chiamata Tap.exe, che crea un file speciale utilizzabile da Windows XP Embedded. in

questo caso nel file che viene creato da questa utility vengono rilevate e salvate le periferiche hardware e le impostazioni necessarie per quel particolare sistema.

Nel passaggio successivo, viene eseguito Windows XP Embedded Component Designer e importato il pacchetto di componenti creato dall'utilità Tap.exe. Come illustrato nella Figura 2, Component Designer legge le informazioni dei dispositivi e assembla un componente completo Windows XP

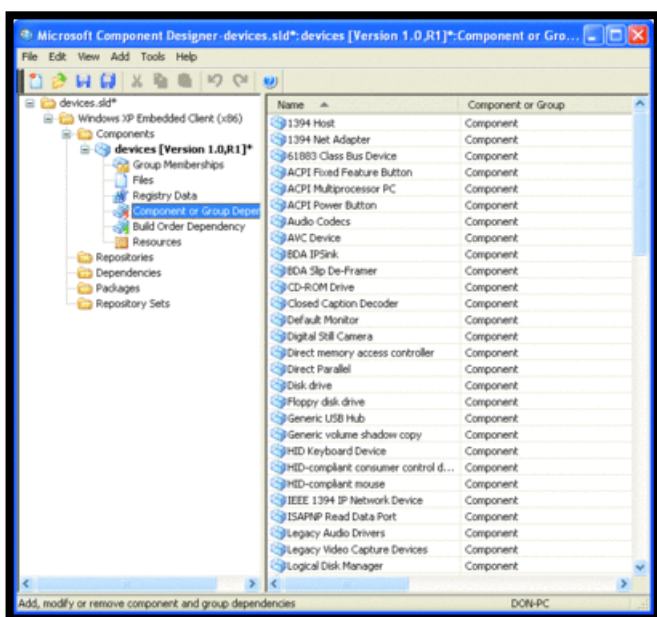


Figura 2 – Progettazione componenti

Embedded che può diventare la base per una nuova immagine Windows XP Embedded. Ciò semplifica enormemente il processo determinando esattamente quali driver di periferica sono necessari per una determinata piattaforma. Dopo aver creato il componente, è necessario utilizzare Component Database Manager per importare il nuovo componente nel database componenti di Windows XP Embedded, un archivio dove vengono memorizzati tutti i componenti noti di Windows XP Embedded. Se il componente è presente nel database è disponibile per l'uso nelle immagini di Windows XP Embedded. Adesso è possibile iniziare a creare una nuova configurazione di destinazione, utilizzando Target Designer di Windows XP Embedded, lo strumento in cui si svolge la maggior parte del

lavoro. Come si è visto nella Figura 1 è possibile aggiungere qualsiasi componente necessario per creare la serie di funzioni richieste dalla periferica. Sarà necessario aggiungere anche alcuni componenti principali di Windows, come Esplora risorse (se la periferica richiede una shell grafica), il file system NTFS (o FAT32), NT Loader, il supporto per una lingua, la parte principale dell'interfaccia utente e così via. Per impostazione predefinita sono disponibili più di diecimila componenti.

È anche possibile pre-configurare tutte le impostazioni del componente. Ad esempio, il componente base dell'interfaccia utente può essere personalizzato per visualizzare elementi specifici nel menu di avvio di Windows, sul Desktop e in altre posizioni. La personalizzazione delle impostazioni di ogni componente assicura che ogni periferica venga pre-configurata per soddisfare le specifiche esigenze (vedere la

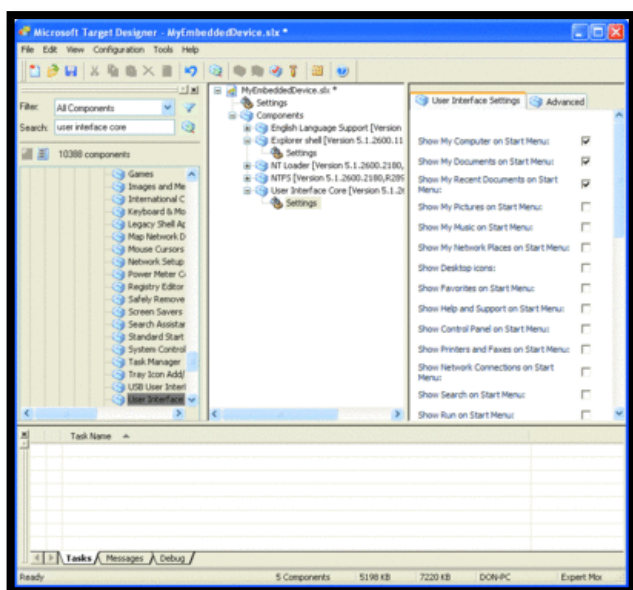


Figura 3).

La parte più complicata dell'immagine di Windows XP Embedded è rappresentata dalle dipendenze. Non è possibile, ad esempio, aggiungere un componente DirectX® all'immagine senza aggiungere diversi componenti che forniscono a Windows le principali funzionalità di gestione



grafici. Tenere traccia di tutte le dipendenze può diventare abbastanza difficile, soprattutto per le immagini più grandi e complesse che possono avere letteralmente migliaia di dipendenze. Questo è il compito del Database componenti. È sufficiente premere F5 in Target Designer e scorrere tutti i componenti inseriti, aggiungendo automaticamente tutte le dipendenze richieste dai componenti elencati. Nella Figura 4 viene illustrata l'analisi in corso; si tratta di un passaggio critico ed è ciò che consente a Windows XP di essere così estremamente modulare. Senza questa intelligenza integrata in Target Designer, sarebbe molto difficile creare correttamente immagini che contengano tutte le dipendenze corrette.

Una volta aggiunti tutti i componenti e le relative dipendenze, è possibile creare l'immagine di Windows XP Embedded. Le immagini sono semplicemente i file che è necessario copiare nel volume di avvio del dispositivo Windows XP Embedded.

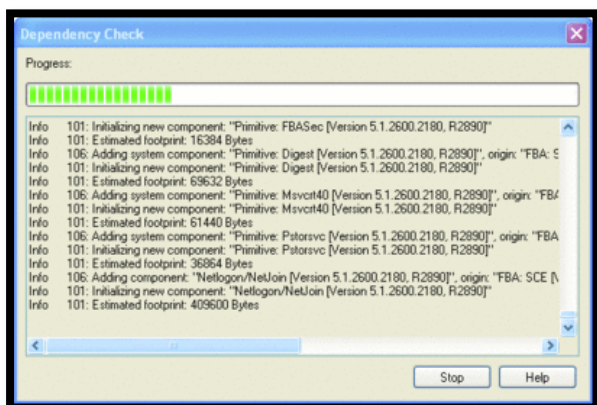


Figura 4 – Verifica delle dipendenze

Non serve alcuno strumento speciale per distribuire le immagini (anche se il toolset Windows XP Embedded contiene un server di avvio remoto simile nella funzionalità ai Servizi di installazione remota; consente la distribuzione delle immagini di Windows XP Embedded sui nuovi dispositivi). Dopo aver completato questi passaggi, è possibile vedere quanto sia grande l'immagine in realtà. Creando un'immagine

semplice con la shell di Esplora risorse e alcune applicazioni, come Blocco note e Paint, si ottiene un'immagine di dimensioni inferiori a 90 megabyte. Certo non è facile trovare molti utenti interessati a una periferica che esegue solo Blocco note e Paint, ma l'esempio illustra quanto piccola può essere un'immagine specifica di un'attività.

### **1.1.5 Differenze con Windows XP Professional**

Windows XP Embedded non è affatto un'opzione ridotta del sistema operativo. Microsoft offre componenti preassemblati che consentono a un computer con Windows XP Embedded di disporre di gran parte delle stesse funzionalità di una versione completa di Windows XP. In questo modo vengono incluse opzioni per installare DirectX® 9.0c, Microsoft® .NET Framework, Windows Media® Player 10 e altro. È disponibile anche molta documentazione per sviluppatori in cui si descrive come creare componenti personalizzati, ad esempio una shell personalizzata per sostituire Esplora risorse, consentendo di creare un ambiente su misura per la specifica periferica Embedded.

Tuttavia, Windows XP Embedded non offre ogni singola funzionalità della versione completa di Windows XP Professional. Ad esempio, Protezione file Windows (WFP) non è presente in Windows XP Embedded e neppure troviamo alcune semplici funzionalità, come Presentazione di Windows XP, le schermate iniziali e le procedure guidate che aiutano un nuovo utente a configurare il proprio computer o Installazione di Windows. Windows XP Embedded non utilizza Attivazione di un prodotto Windows (le immagini di Windows XP Embedded vengono attivate utilizzando un codice "Product Key" di runtime dagli strumenti di Windows Embedded Studio). In particolare, Windows XP Embedded non utilizza affatto Windows Update, sebbene possa utilizzare un server Microsoft Software Update Services nel proprio ambiente. In

Windows XP Embedded Infine, manca anche MSN® Explorer, che è contenuto in Windows XP Professional.

Per il futuro, Windows XP Embedded prevede un ricco percorso di sviluppo sempre più questi miglioramenti continuano ad aggiungere flessibilità e opzioni al sistema operativo Windows XP Embedded, portandolo sempre più a un livello paragonabile con le funzionalità impostate in Windows XP Professional.

I sistemi Embedded possono richiedere maggiore lavoro per la costruzione rispetto a un sistema Windows XP tradizionale, ma offrono vantaggi significativi in determinati scenari. Sebbene ridotti nelle funzionalità, i sistemi Windows XP Embedded possono supportare hardware più piccoli, meno potenti, più facili da gestire e riparare, con costi ridotti. La riduzione del consumo di energia consente di ridurre anche le bollette e l'emissione di calore, favorendo maggiore flessibilità nella progettazione di aree di lavoro compatte. Oltre agli usi previsti di Windows XP Embedded, come nei dispositivi per l'elettronica di consumo, nei dispositivi medicali, nei punti vendita e in altri luoghi dove forse neanche si immagina che ci possa essere Windows XP Embedded.

## **1.2 – Storia dei sistemi Embedded di Microsoft**

Sono ormai passati 10 anni da quando Microsoft ha deciso di entrare nel mondo dei sistemi operativi Embedded e vale la pena tracciare un breve excursus storico di questo periodo.

Per dispositivi Embedded intendiamo un apparato elettronico governato da un computer "dedicato" allo scopo per cui è stato disegnato, senza che l'utente ne percepisce la presenza. Esempi classici sono: il mouse, le tastiere gli hard - disk, le pen-drive, bilance, macchine di misura, strumenti medicali, navigatori satellitari, POS, ecc...

I sistemi Embedded sono realizzati con l'obiettivo di contrapporsi ai "sistemi aperti", tipici dei Personal Computer. Dai primi processori, per un lungo periodo non è emersa l'esigenza di un vero e proprio sistema operativo. Ogni produttore forniva all'utente il proprio ambiente di sviluppo con svariati linguaggi, oltre naturalmente all'Assembler di sistema, per poter sviluppare applicativi e sottosistemi in grado di gestire direttamente il dispositivo.

Per una larga fascia di microcontrollori a 8bit e di CPU a 16bit la situazione non è cambiata di molto, mentre per altri, si è concretizzata l'esigenza di sistemi operativi Embedded con precise esigenze:

- Affidabilità;

Il dispositivo Embedded, dedicato ad un numero finito di applicazioni, ha bisogno di avere un sistema operativo che funzioni sempre, possibilmente con un breve tempo di accensione.

- Performance;

Ai dispositivi Embedded vengono richieste alte prestazioni rispetto alle ridotte risorse di sistema come ad esempio applicazioni con sistemi operativi "hard-real-time", cioè in grado di gestire "eventi" in un tempo determinabile a priori, senza che ci possa essere alcuna perdita di informazioni.

- Economicità;

I prodotti di pubblico consumo che devono avere un prezzo competitivo e l'evoluzione richiesta in molti ambienti Embedded, hanno portato all'uso di ambienti di sviluppo semplici, familiari e performanti, sia per il sistema operativo che per gli applicativi.

Inoltre Non sempre nei dispositivi viene richiesto un sistema operativo, questa esigenza si manifesta quando si presentano una o più di queste necessità:

- Sincronizzazione;

- schedulazione in condizioni di multitasking;
- gestione di un gran numero di dispositivi di I/O;
- gestione dei file, dispositivi di rete, dispositivi video grafici;
- gestione della memoria, della sicurezza, dell'alimentazione.

Dove la presenza di un sistema operativo le rende più facilmente controllabili.

Nella metà degli anni novanta quando i processori Pentium avevano appena pochi anni di vita, Windows 95 era appena nato e la maggior parte dei Personal Computer utilizzava MS-DOS come sistema operativo, non esisteva ancora un'offerta Embedded specifica e lo sviluppo di queste soluzioni veniva effettuato con svariati linguaggi, senza piattaforme generalizzate, e con sistemi operativi quali Ms-Dos, Windows 3.11, Windows NT.

Nel 1996 nasce il primo sistema operativo Embedded Microsoft, Windows CE 1.0, Scritto per processori PowerPC, MIPS e Intel, e utilizzato da pochi sistemi OEM (Original Equipment Manufacture) per realizzare piccoli PC palmari.

Questa versione aveva un ambiente di sviluppo simile a quello standard, che presentava una scelta di chiamate al sistema operativo (API) ricavate da quelle WIN-32, era multithread e multipiattaforma ma non compatibile con il Dos.

Negli anni 1997 / 1998 Vengono rilasciate le versioni 2.0 2.01 2.11 e 2.12, utilizzate da un maggior numero di sistemi OEM. Vennero introdotte le funzionalità per la connessione di rete che primi permisero la nascita dei Thin-client (dispositivi che, sfruttando l'architettura "terminal server", vengono utilizzati come terminali remoti delle applicazioni Windows).

Con questa versione gli sviluppatori potevano acquistare un Kit di personalizzazione del sistema (ETK=Windows CE Embedded Toolkit) infatti il sistema prevedeva soltanto una

piattaforma hardware. Con i "service pack" 2.11 e 2.12 vennero aggiunte:

- la prima interfaccia grafica;
- una revisione della "command-shell" tipo MS-DOS;
- una prima implementazione della sicurezza;
- una versione ridotta di Internet Explorer 4.0,
- l'IrDa.

Successivamente all'esigenza di sviluppare con gli stessi strumenti per i sistemi standard, Microsoft realizzò un ambiente di sviluppo per "costruire" un sistema operativo partendo dai "componenti di base" e aggiungendo altri componenti dipendenti dai primi, così da poter personalizzare il sistema operativo risultante. L'utente di questo ambiente, gestito da un database, può generare facilmente i propri componenti aggiuntivi, determinarne le dipendenze e combinarli per le più svariate esigenze. Nasce Windows NT Embedded.

Nel 2000 nasce Windows CE 3.0, una versione ridisegnata con un'architettura "hard-real-time" dove la piattaforma assume definitivamente una veste grafica, interattiva e comprensiva di tool di sviluppo e di debug.

Il nuovo kernel permetteva di gestire le funzionalità real-time mediante:

- 256 livelli di priorità;
- La programmabilità del "quantum" di ogni thread;
- La gestione degli interrupt nidificati;
- l'object store allargato a 256Mb;
- la dimensione dei singoli file aumentata a 32Mb;
- vennero introdotti e aggiornati i supporti DCOM, PPTP, ICS, RDP 5, ecc ....

Con questo rilascio, si può affermare che Microsoft sia entrata definitivamente nel mondo dei sistemi operativi "Embedded real time".

La release Windows CE .NET 4.xx coincide con la nascita della piattaforma .NET dedicata al mondo "mobile" ed Embedded, ovvero la ".NET Compact Framework", che anche se ancora non la supportava offriva molte delle potenzialità del runtime .NET con una libreria "leggera" per essere inserita anche in sistemi poco performanti.

Questa release si basava sostanzialmente su questi concetti:

- La produttività;  
Puntando su nuovi template e device pre-configurati e con un rapido riciclo tra build, test e modifiche. Inoltre l'ambiente multiplatforma, offriva un cross-compiler interno per le diverse CPU supportate (ARM, MIPS, SH e X86). Vennero anche aggiunte due nuove piattaforme di sviluppo per gli applicativi: Visual Studio .NET e Embedded Visual C++ 4.00.
- L'accesso al codice in forma sorgente  
Tramite due milioni di linee di codice disponibili e con Forum, Faq, chat ed eventi divulgativi.
- Consolidamento strutturale;  
Effettuato tramite il rafforzamento che andava dalla parte "real-time" alle dimensioni del sistema. 350 componenti, una nuova gestione dell'alimentazione, il device per le comunicazioni: (TCP/IP, IPV4, IPV6, NDIS 5.1, Winsock 2.0, ecc) e tanti altri supporti per i più recenti sistemi USB, SMTP, FTP, ecc.
- Soluzioni innovative;  
rivolte ad una mutua connessione dei device, soprattutto per le soluzioni wireless (PAN, LAN, WAN, Bluetooth, 802.11, ecc), ad una ricca piattaforma multimediale (WM 9, DirectX8, I.E. 6.0), e con l'inserimento dei viewer dei documenti fondamentali (Excel, Word, PPT, Image, PDF), oltre al supporto multi-lingue per 12 linguaggi.

La release 5 di Windows CE punta alla ricerca della qualità di ogni componente. nascono così:

- Production-Quality OAL (OEM Adaptation Layer);
- Production-Quality BPS (Board Support Packages)“;
- Production-Quality driver.

Soluzioni orientate alla certificazione dei propri prodotti. Con la stessa versione nascono anch tool di test (CETK = Windows CE Test Kit). Di seguito alcuni aggiornamenti:

- Ambiente di sviluppo per driver e build;  
Possibilità di inserimento di breakpoint e semplificazione della connessione di debug e unificazione dell’interfaccia grafica con la gestione a riga comando, nuove opzioni di compilazione e di link, ecc.
- Elementi di sistema;  
Interrupt di sistema portati da 32 a 64, possibilità di variare la frequenza di schedulazione, Embedded Database (EDB), integrazione di tecnologie grafiche e multimediali (Windows Media, Direct3D Mobile, Direct sound, ecc), ecc.
- Nuovi componenti;  
Internet Explorer 6.0 per Windows CE, una serie di componenti per la sicurezza.

Nella release 6 di Windows CE le due maggiori novità sono il nuovo kernel che ha permesso di superare alcune barriere sul numero dei processi e la disponibilità di memoria per processo e l’integrazione della piattaforma di sviluppo nell’ambiente Visual Studio .NET 2005.

Queste sono state le più importanti novità:

- a livello di sistema;  
2Gb di memoria virtuale per ogni processo per un max. di 32000 processi, spostamento di componenti critici di sistema all’interno del kernel e possibilità di scrivere i driver in “kernel mode” con un incremento delle performance o in “user mode”.



- A livello di tool per lo sviluppo e il test;  
L'integrazione in Visual studio .NET 2005 ha portato immediati benefici: aggiornamento dei compilatori all'ultima release, miglioramento delle performance, aggiunta di editor dedicati, runtime image viewer, ecc.
- Ulteriore disponibilità dei codici sorgente:  
Il codice sorgente presente tramite la licenza shared source arriva al 100% del kernel e del device manager.
- Nuovi componenti per le nuove tecnologie.  
Proiettore wired e wireless per i notebook con Windows Vista, funzionalità legate alla telefonia cellulare (Cellular Network Support), funzionalità multimediali (Network Media Device) e la compatibilità con componenti per il motore DVR in MPEG2, ecc.

La nascita di XPE parte invece dal 2001 e l'interesse di Microsoft per i sistemi operativi Embedded diventa così marcato che passano soltanto 6 mesi tra il rilascio di Windows XP a quello del suo analogo per il mondo Embedded. Il nome dell'intero pacchetto Windows XP Embedded: "Target Designer" (TD) viene preso dal suo elemento più significativo, ma la suite completa è composta da tre ambienti grafici, oltre al TD troviamo il "Component designer" e il "Component Database Manager". Il prodotto è l'evoluzione dell'ambiente NT Embedded. I "componenti" del sistema operativo XP sono stati riordinati in gruppi, ne sono state stabilite le dipendenze e le loro mutue esclusività e da lì è stato ricreato un database ben articolato in modo da permettere all'utente di creare, molto semplicemente, un sistema operativo Windows XP "ritagliato" sulle proprie esigenze. A tutto questo sono stati aggiunti nuovi componenti specifici per l'ambiente Embedded: la personalizzazione grafica, la velocità di accensione, l'aggiornamento, la protezione del sistema, ecc...

Analizzeremo ora quelle che sono le caratteristiche di un sistema Embedded XPE e che sono state presenti fin dalla prima versione.

Il prodotto, che da quel momento è stato aggiornato con tre rilasci sia per mantenere l'allineamento con l'ambiente Windows XP Professional sia per le notevoli migliorie dell'ambiente di lavoro e della granularità dei componenti, è costituito da tre momenti fondamentali:

- la creazione della piattaforma hardware con l'esigenza, in questa fase, di creare una lista di tutti i singoli componenti hardware di cui sarà formato il dispositivo Embedded finale e, per ognuno di questi, di reperirne il driver opportuno. I costruttori di hardware infatti distribuiscono il "pacchetto" dei driver da inserire nel database di lavoro. Per le altre innumerevoli combinazioni di componenti hardware che possono essere scelte per la propria piattaforma, Microsoft ha preparato una serie di tool per aiutare l'utente a creare la propria lista.
- Il secondo passaggio riguarda la scelta del tipo di dispositivo che si vuole produrre. L'ambiente presenta una serie di template organicamente precostituiti, che permettono di partire da una situazione consolidata; l'utente può poi inserire o eliminare, secondo le proprie esigenze, altri componenti specifici. In questa fase, ci sarà eventualmente anche l'integrazione dell'applicativo specifico dell'utente all'interno del sistema (es.: il produttore di un "chiosco informativo" creerà il componente dell'applicativo "chiosco").
- L'ultimo passaggio riguarda l'assemblaggio vero e proprio del sistema:

Una importante prerogativa del sistema è la possibilità autorizzata da Microsoft di sostituire la parte di "presentazione iniziale" del sistema (splash-screen) con un

logo diverso che può essere, ad esempio, quello del produttore del software o, in alcuni casi, quello del cliente finale.

L'ambiente offre una serie di opportunità per gestire al meglio il fatto che spesso i dispositivi Embedded non possiedono tastiera e mouse e, soprattutto l'utilizzatore abituale non può essere il referente delle segnalazioni del sistema: ad esempio su un chiosco informativo è inutile far apparire un pop-up che richiede un intervento sistemistico quando davanti allo schermo c'è il cliente della banca che aveva richiesto un estratto conto.

Nell'ambiente Embedded, le configurazioni degli utenti del sistema, dei privilegi e degli accessi vanno decisi, in gran parte, a livello di produzione del sistema. In genere, l'esigenza richiede due tipi di utenza finale: quella di chi utilizza l'applicativo per cui viene venduto il dispositivo e quella dell'amministratore del sistema o meglio del manutentore. E' possibile organizzare il sistema in modo che l'utente del dispositivo non possa in alcun modo alterare il sistema operativo mentre il manutentore, per definizione, deve essere in grado di farlo.

L'ultimo punto riguarda l'aggiornamento sul campo del dispositivo. Per questo, Microsoft ha previsto sia *soluzioni standard* (aggiornamenti automatici e client di ambienti di distribuzione software tipo SMS), per permettere ai sistemi XP Embedded di coesistere con sistemi XP Professional, sia *soluzioni dedicate*.

Così come dall'ambiente Windows CE sono state create da Microsoft diverse distribuzioni per le specifiche esigenze di mercato (Windows Mobile per PDA, telefonini, navigatori, Windows CE per l'Automotive, ecc ...), così dall'ambiente Windows XP Embedded Microsoft ha preparato un prodotto specifico per i punti di vendita di servizi chiamato Windows Embedded for Point of Services.

Questo prodotto non ha un ambiente di sviluppo come per Windows XP, ma è stato preparato da Microsoft con tutti i componenti specifici dell'ambiente in cui si opererà: chioschi, POS (Point of Sale), ATM ( Automatic Teller Mchine), Pompe di benzina, Self-Checkout, ecc.

L'installazione è simile a quella di Windows XP Professional, ma con i benefici di avere un sistema pensato per l'Embedded con tutti i driver tipici dell'ambiente: PINPAD, Badge reader, stampanti, ecc e con la piattaforma .NET POS per poterli programmare in modo standard.

Negli anni si è susseguito il rilascio di diversi sistemi Embedded come le release Embedded dei "Classici" sistemi Windows XP, 2000, NT, 3.1 e 3.11 e il "buon vecchio" MS-DOS 6.22. Per ognuno di questi sistemi, esiste a listino la versione "for Embedded Systems", che è del tutto identica alla versione corrispondente del prodotto in ambiente desktop, con il vantaggio della reperibilità per quelli più datati.

Sotto al nome di Windows Embedded Server c'è un'intera famiglia di prodotti server. Il concetto è quello dell'Embedded, ovvero singoli server dedicati a servizi specifici: gestione file, telecomunicazioni, protezione, ecc. derivati da Windows Server 2003 R2, con restrizioni esclusivamente sulla licenza e non sulle funzionalità. Il vantaggio è squisitamente economico: i server Embedded costano meno dell'omologo standard edition.

Infine c'è ".NET Micro Framework". Ideato per dispositivi con un basso consumo di energia e CPU a 32, 16 e 8 bit, senza una richiesta specifica di MMU (Memory Management Unit). Venne distribuito da MSN Direct già nel 2004 e attualmente, usato come side-show in alcuni portatili distribuiti con Windows Vista. Windows SideShow è una nuova tecnologia che permette all'utente di visualizzare su un piccolo video sulla parte esterna del portatile delle informazioni anche se il PC è spento o ibernato.

### 1.3 – Strumenti e requisiti per la costruzione di un sistema Windows XP Embedded

I requisiti per l'utilizzo di Windows XP Embedded prevedono che il computer su cui viene installato il sistema di sviluppo di Windows Embedded debba avere installati uno dei seguenti sistemi operativi:

- Windows Server 2003;
- Windows Server 2008;
- Windows Vista;
- Windows XP

Mentre per quanto riguarda il tools di sviluppo i sistemi richiesti sono i seguenti:

Nel caso il sistema installato fosse Windows XP Professional SP2 o Windows Server 2003 Standard la configurazione minima richiesta è la seguente

- CPU, 728 MHz o maggiore
- RAM minima 512 MB anche se è raccomandata 1 GB
- Nel caso il sistema installato fosse Windows Vista or Windows Server 2008 Standard si richiedono:
- CPU, 1 GHz o maggiore
- RAM minima 1 GB

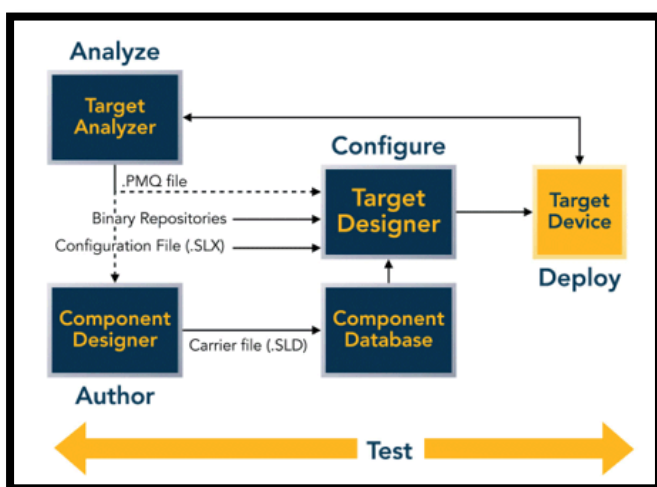


Figura 5 – Processo di generazione di Windows XP Embedded

In ogni caso è richiesta l'installazione di Microsoft Internet Explorer 6 anche se è consigliata una versione successiva del programma.

Per quanto riguarda l'hardware minimale richiesto, comprende:

- connessione di rete per la condivisione del database di componenti
- Drive DVD
- Monitor con risoluzione VGA o maggiore (1024x768 di risoluzione raccomandata)
- 70 MB di disco libero per l'installazione

Il processo di generazione di Windows XP Embedded viene sinteticamente spiegato nella Figura 1, e prevede 4 fasi ossia: Analisi hardware, Creazione componenti, Configurazione versione e Distribuzione immagine.

Il modo migliore di investire il proprio tempo e assicurarsi un processo di generazione affidabile è installare Windows XP Professional (il sistema operativo desktop) nel dispositivo di destinazione x86. L'installazione e l'utilizzo di XP Professional e delle relative applicazioni fornisce un benchmark per valutare l'affidabilità di hardware, driver e applicazioni. Questa pratica di benchmark consente allo sviluppatore di eliminare hardware, file binari del driver e applicazioni che potrebbero causare errori nella versione di Windows XP Embedded.

I due programmi di Target Analyzer, Target Analyzer Pro (TAP.exe) e Target Analyzer (TA.exe), si completano vicendevolmente. TAP.exe è un'applicazione di Microsoft Win32 che richiede Windows 2000 o Windows XP per l'installazione sul dispositivo di destinazione. Se non è possibile utilizzare TAP.exe, è possibile eseguire Target Analyzer (TA.exe) che funziona in ambiente DOS. TA.exe rileva solo la presenza di dispositivi ACPI, USB, 1394, SCSI, PCMCIA e ISA. Inoltre, TA.exe fornisce un suggerimento per

l'Hardware Abstraction Layer (HAL) e non rileva i dispositivi enumerati software. Quando si utilizza TA.exe, potrebbero essere necessarie fonti aggiuntive per la documentazione dell'architettura dei dispositivi di destinazione.

Target Designer consente di sviluppare un'immagine funzionante in tutto e per tutto del sistema operativo a scopo di test o per poterci effettuare eventuali presentazioni senza acquistare la licenza finché il progetto non viene distribuito e quindi finché il cliente non si impegna pagarla. L'immagine infatti resta valida per un periodo compreso tra 90 e 180 giorni dopodiché viene visualizzata semplicemente una "schermata blu".

Per quanto riguarda invece i prezzi delle licenze sono quelli indicati nella tabella delle licenze<sup>[1]</sup>.

Target Designer consente allo sviluppatore di impostare una soglia e scegliere se rendere visibile o meno un componente

**Tabella delle Licenze**

<b>Windows Embedded Standard</b>	<b>Prezzo</b>	<b>Cosa viene incluso?</b>
<b>Evaluation kits</b> <i>(per uso non commerciale)</i>	senza alcun costo	Le immagini di sistema operativo create scadono dopo 120 gg.
<b>Tools di sviluppo</b>	\$995 US*	Questi tools abilitano la creazione di immagini di sistema operative che non scadono.
<b>Licenza di Runtime</b> <i>(costo per unità)</i>	\$90 US*	Licenza Runtime per ciascun sistema operativo che viene installato. Non scade mai.

*\* Tutti I prezzi sopra indicate sono basati su dollari USA e sono stimanti in prezzi al dettaglio, possono però variare da distributore a distributore.*

*\* I prezzi sono basati sull'acquisto di una unità, Sono disponibili secondo Microsoft sconti per quantità, Possono essere applicate inoltre tasse locali, tariffe e altri costi.*

durante la configurazione. L'utilizzo dei livelli di visibilità riduce il numero di componenti con i quali deve lavorare uno sviluppatore. Quando si esegue il debug di una versione e non viene individuato un componente, lo sviluppatore deve verificare l'impostazione di visibilità di tale componente.

Una volta configurata una versione di Windows XP Embedded, Target Designer analizza costantemente le catene di dipendenze lunghe. Ciascun componente ha esigenze e dipendenze specifiche. Target Designer risolve

automaticamente i problemi relativi alle dipendenze tra i componenti se la casella di controllo per la *risoluzione automatica delle dipendenze* è selezionata. Lo svantaggio di questa funzione consiste nel fatto che le dimensioni della versione possono diventare maggiori di quanto desiderato dallo sviluppatore senza fornire la visibilità sui componenti aggiunti per soddisfare le dipendenze. Uno sviluppatore deve individuare delle condizioni di compromesso tra la velocizzazione della versione, l'utilizzo del controllo Automated Dependency e la riduzione delle dimensioni dell'immagine.

Nella preparazione di un'immagine di produzione che verrà duplicata in più dispositivi durante il processo di produzione, uno sviluppatore deve includere un apposito componente che permetta all'immagine che verrà creata di essere successivamente duplicata completamente su più device.

Il componente di clonazione non è richiesto nel caso in cui per ciascun dispositivo venga eseguito un processo autonomo di First Boot Agent (FBA) e quindi quando per ogni device, è presente un'immagine a se stante che viene eseguita. Tuttavia, il processo FBA richiede molto tempo e non è vantaggioso in un ambiente di produzione.

Per il debug di Windows XP Embedded, sono disponibili diverse utilità per i test. InCtrl5<sup>[2]</sup> di PC Magazine<sup>[2]</sup> che consente a uno sviluppatore di acquisire uno snapshot del sistema prima e dopo le modifiche. FileMon<sup>[3]</sup>, strumento gratuito che visualizza l'attività dei file system in tempo reale e segnala i file utilizzati dal programma, compresi le librerie a collegamento dinamico (DLL) e i file di configurazione. RegMon<sup>[3]</sup>, disponibile su Sysinternals<sup>[4]</sup>, che visualizza le applicazioni che accedono ai registri. Microsoft inoltre offre utilità tipo Dependency Walker<sup>[5]</sup>, che consentono allo sviluppatore di assicurarsi che tutte le librerie DLL richieste per un'applicazione o un driver siano incluse.



## 1.4 – Alternative a Windows Embedded

Le principali alternative all'ultimo nato di casa Microsoft sono principalmente due, una all'interno della stessa casa di Redmond ossia Windows CE Embedded, e l'altra è il sistema Linux Embedded<sup>[6]</sup> con il quale in passato XPE si è dovuto confrontare e dal quale si differenzia maggiormente. Il primo è caratterizzato da un approccio diverso oltre che dal differente hardware che è possibile utilizzare nel sistema di destinazione e sul quale, per quanto riguarda XPE, viene posta la maggiore attenzione soprattutto per il fatto che può essere utilizzato Hw x86 con un risparmio a livello economico. Il sistema Linux Embedded<sup>[6]</sup> invece è l'alternativa percorsa soprattutto da coloro che amano i sistemi orientati al software libero o a licenze GNU Public License (GPL), alla quale viene associata, e sulla quale il sistema, rispetto a quello di casa Microsoft, punta per combattere la difficoltà di installazione e gestione da parte dell'utente finale che ne ha frenato lo sviluppo in passato.

<b>Confronto</b>	<b>Windows XPE</b>	<b>Windows CE Emb.</b>	<b>Linux Embedded</b>
Costi Sist. Sviluppo	-	-	+
Costi Immag. Create	-	-	+
Prestazioni	=	=	=
Supporto Sviluppo applicaz.	+	+	-

La prima domanda che un utente si potrebbe fare è perché utilizzare Linux in applicazioni Embedded ? Al giorno d'oggi abbiamo sistemi sempre più potenti e complessi, Ram sempre maggiore, HD sempre più elevati, proliferazione dei protocolli

(USB, Firewire, Bluetooth, TCP/IP, Wap, ...), l'incessante e accelerata obsolescenza dei chipset (6-12 mesi) con il costante ridisegno dei sistemi Hw e Sw, e solo l'hardware ben supportato che fornisce un costante aggiornamento dei driver, sopravvive. Risultato, gli sviluppatori Embedded, contano su sistemi operativi supportati.

Inoltre ci sono diversi svantaggi dei sistemi Embedded tradizionali:

- Codice a volte troppo semplicistico per le funzioni e le risorse necessarie oggi.
- Kernel personalizzato spesso troppo costoso da sviluppare e da mantenere aggiornato.
- Windows che percepito spesso come troppo costoso.
- Sistemi proprietari spesso non standard, con poche norme, e troppo costosi nello sviluppo e nelle licenze.
- UNIX infine non particolarmente "embeddable".

A tal proposito Linux invece ha creato un sistema appunto Linux-Embedded<sup>[6]</sup>, Con Licenza Free, Percepito come robusto e affidabile, con un'eccellente sostegno alla grafica e alle reti, configurabile, scalabile, modulare e personalizzabile, Supportato da migliaglia di sviluppatori e con una crescente scelta di strumenti di sviluppo, da far gola a moltissimi sviluppatori che in alcuni casi lo hanno preferito al rivale della casa di Redmond.

Linux-Embedded<sup>[6]</sup> è disponibile in molte versioni sia commerciali che non; rispettivamente:<sup>[7]</sup>

- (Lineo, MontaVista, LynuxWorks, PalmPalm, Coventive, FSM Labs, TimeSys, REDSonic, ...) per quanto riguarda Linux Embedded/RT commerciale.
- (uClinux, emdebian, RTLinux, RTAI, ELKS, KURT, LEM, LOAF, LRP, LinuxPPC, mLinux, ThinLinux, ...) per quanto riguarda Linux Embedded/RT non commerciale.

Linux pertanto in pochi anni è passato da nuova tecnologia a tecnologia dirompente anche attraverso la creazione del LEC (Linux Embedded Consortium).

Date queste premesse, (Nessun costo per il sistema di gestione, Nessun costo per le immagini distribuite), può considerarsi come uno dei principali concorrenti al sistema embedded di Microsoft, se non fosse che, nonostante il LEC (volto a diminuire il gap), Linux tende ad avere problemi dal punto di vista hardware che non sempre è supportato dalle case produttrici, e che pertanto ne limita l'utilizzo da parte di una grossa fetta di potenziali utenti. Spesso infatti chi si trova a dover scegliere un sistema piuttosto che un altro, preferisce, soprattutto nel campo professionale, la maggiore stabilità e supporto delle aziende produttrici al fatto di non pagare una iniziale "tassa" per l'utilizzo del software. Tale decisione spesso è ripagata dal fatto di non dover fare scelte obbligate perché magari Linux supporta un determinato tipo di hardware, dando quindi la possibilità al progettista del sistema di poter scegliere la soluzione hardware migliore per il progetto sistema Embedded che si va a costruire.

### **1.5 – Vantaggi e svantaggi per le aziende**

I vantaggi che le aziende possono avere dall'utilizzo di questi sistemi Embedded sono diversi:

Windows Embedded assicura tempi rapidi di introduzione sul mercato, grazie a set di funzionalità completi e a strumenti di sviluppo intuitivi e di semplice utilizzo che contribuiscono ad abbattere i tempi di realizzazione dei progetti. La ridotta necessità di sviluppare soluzioni personalizzate contribuisce al completamento dei progetti risparmiando ore di lavoro in termini di sviluppo. Inoltre, le aziende potranno avvalersi delle competenze di ben sei milioni di sviluppatori Visual Studio .NET in tutto il mondo.

Windows Embedded riduce al minimo l'investimento necessario, offrendo la possibilità di valutare e testare il sistema grazie a kit di valutazione gratuiti della durata di 120 giorni. Una volta che l'azienda entra in fase di produzione, gli strumenti di sviluppo di Windows Embedded si rivelano convenienti, perciò gran parte dell'investimento viene effettuato dopo che il dispositivo è disponibile e inizia a generare profitti.

Da una recente ricerca di mercato realizzata dalla società indipendente Venture Development Corp. (VDC)<sup>[8]</sup> è emerso che:

"Il modello di business Microsoft continua a offrire agli OEM una struttura semplificata per la definizione dei prezzi e un modello di consulenza per l'accesso a tutte le risorse della piattaforma necessarie per la progettazione dei dispositivi, tramite licenze runtime a costi unificati per l'utilizzo dei dispositivi."

L'acquisto della licenza viene effettuato al momento dell'effettiva disponibilità del prodotto. Windows Embedded consente di rimandare l'investimento iniziale finché il dispositivo non è disponibile, tramite l'utilizzo di licenze per unità con l'applicazione di sconti in base ai volumi. Questo consente di prevedere la portata dell'investimento e di acquistare solo le licenze effettivamente necessarie.

Shared Source è un'infrastruttura di licensing Microsoft che consente ai membri della comunità di utenti e sviluppatori di accedere al codice sorgente Microsoft nel rispetto della protezione della proprietà intellettuale, essenziale per mantenere la correttezza dell'"ecosistema" software.

Inoltre, gli sviluppatori di tutto il mondo possono creare e rendere disponibili derivati commerciali del sistema operativo senza l'obbligo di notificarlo a Microsoft.

I criteri di licensing di Microsoft Windows Embedded garantiscono la protezione della proprietà intellettuale. Per

alcune piattaforme di sistemi operativi è necessario rendere pubblica la proprietà intellettuale, il cui utilizzo ricade sotto la responsabilità degli utenti.

Microsoft in tal modo e con queste nuove metodologie di approccio al mondo dello sviluppo di nuove tecnologie intende contribuire al successo dei clienti, anche tramite il supporto delle piattaforme Windows Embedded con una roadmap di innovazione a lungo termine e con attività di ricerca e sviluppo all'avanguardia.

Oltre ad un reparto di sviluppo di alto livello, Microsoft mette a disposizione una serie di opzioni e programmi di supporto, per offrire la massima semplicità e scalabilità nell'utilizzo della tecnologia Windows Embedded.

La rete mondiale di "partner leader" del settore costituisce rappresenta un ecosistema diversificato di risorse e servizi efficaci per accelerare i tempi di introduzione dei dispositivi sul mercato.

La rete di partner Windows Embedded include distributori, integratori di sistemi, fornitori indipendenti di hardware e software, ODM, OEM, fornitori di prodotti in silicio, produttori di strumenti e organizzazioni per la formazione professionale. Dalle aree di ricerca e sviluppo fino ad arrivare al supporto, Microsoft è pronta a offrire assistenza per qualsiasi settore.

L'acquisto di interventi di supporto Windows Embedded garantisce al team di sviluppo un'assistenza affidabile e competente, salvaguardando i progetti riguardanti i dispositivi da eventuali ritardi dovuti a problemi tecnici.

Con Windows Embedded, i dispositivi sono supportati dal riconoscimento a livello mondiale del marchio Microsoft Windows. Infatti, questi dispositivi vengono associati dai clienti all'idea di familiarità, facilità di utilizzo e qualità che è stata una delle mosse vincenti di Microsoft.

Windows Embedded mantiene gli utenti costantemente aggiornati sulle tendenze del mercato e l'evoluzione dei

requisiti relativi al software Embedded. Con Windows Embedded, i progetti precedenti possono essere riutilizzati, per consentire il rapido aggiornamento o l'ampliamento delle linee di prodotti tramite l'utilizzo del codice esistente. Questo permette di rilasciare una più ampia varietà di interessanti prodotti Embedded in tempi ridotti rispetto alla concorrenza e di mantenersi sempre all'avanguardia per consolidare la propria posizione di leader nel settore del software Embedded.

### **1.6 – A chi è rivolto principalmente il sistema**

Windows Embedded per la sua particolarità di sistema operativo, configurabile ed adattabile al dispositivo di utilizzo, che è particolarmente adeguato a dispositivi mobili o altamente customizzati e che spesso richiedono un sistema operativo ad hoc con risorse minori rispetto a quelle di un moderno PC a fronte delle quali le funzionalità devono essere paragonabili a PC di moderna generazione. Uno dei principali casi, in cui questo è visibile è nei moderni NetBook oltre a tutti quei dispositivi in cui l'utente deve interagire per modificare impostazioni, o semplicemente per utilizzare il bene in questione, in modo semplice e funzionale.

Esempi concreti, sono I POS, CELLULARI, MACCHINE UTENSILI, ROBOT, LETTORI DVD, VHS, SISTEMI DI CONTROLLO, CHIOSCHI, Ecc...

### **1.7 – Una soluzione per I moderni Netbook**

Come evidenziato dalla rivista Total Linux in un'articolo<sup>[9]</sup>, circa agli inizi del 2008 l'azienda ASUS introduceva nel mercato una nuova gamma di PC, e forse non poteva ancora supporre che quello che stava facendo, ossia introdurre una moda che da lì a qualche mese avrebbe preso piede fino a far pensare che già nel 2010 circa il 40% del mercato mondiale sarà invaso da questa nuova tecnologia di PC, che per la loro

ridotta dimensione sono facilmente trasportabili, e che spesso è rivolta ad un "target" di utenza con non molte pretese (soprattutto a livello hardware) e che spesso utilizza il PC per poche semplici applicazioni che devono funzionare in maniera adeguata e senza "intoppi".

Date queste premesse, possiamo dire Windows XPE potrebbe diventare l'ideale partner di questi sistemi, che pur non avendo un hardware nemmeno paragonabile ai "cugini" Notebook, ne deve allo stesso tempo reggere il confronto in quelle poche semplici operazioni (Navigazione, Utilizzo di Sistemi di Videoscrittura, Wordprocessor, Oltre alla gestione dei principali dispositivi, Webcam, Stampanti, ecc.).

L'utilizzo di sistemi Embedded come Windows XPE potrebbe favorire la velocità del sistema operativo rispetto ai più blasonati Windows XP nelle versioni Home o Professional, soprattutto per il fatto che in questi sistemi come già spiegato, vengono inseriti solo i componenti che effettivamente servono e che pertanto rendono l'immagine da installare molto leggera (in media 200/300 Mb), e che consentono inoltre l'avvio del sistema in pochi secondi (attorno ai 20/25'') che è spesso un'altra delle richieste che l'utente del Netbook tiene in considerazione nella scelta del suo "computerino" portatile.

## **1.8 – Enhanced Write Filter (EWF) in Windows Embedded**

### **1.8.1 Panoramica**

EWF<sup>[10]</sup> provvede alla protezione del disco dalla scrittura. Esso abilita il sistema operativo avviato da sistemi read-only, CD-ROMs, Hard disk protetti in scrittura o flash media. Tutte le scritture che andrebbero fatte su volume EWF protetto vengono re-direzionate in uno spazio riservato (Overlay). Le scritture vengono fatte nella cache resa disponibile come parte del volume. Questo dà l'impressione che il volume sia

scrivibile. La creazione di questo volume "esteso" può essere effettuata sia su Disco che in memoria ad accesso casuale (RAM). Se lo si desidera, i dati memorizzati nel volume di cache possono essere assegnati al volume protetto come possiamo vedere nello schema di Figura 6 che altro non è che una panoramica di EWF<sup>[10]</sup>.

### 1.8.2 Panoramica

Ci sono due tipi di EWF<sup>[10]</sup> supportati da Microsoft® Windows® XP Embedded. Il primo tipo è EWF basato su disco che ridireziona tutte le scritture su una partizione separata dell'hard disk. I dati immagazzinati sulla partizione separata se desiderato possono essere assegnati al volume protetto. Per un singolo volume protetto possono inoltre essere costituiti tipi multipli di dischi in cui riversare i dati in scrittura. Possono esistere per un singolo volume delle caches multiple e questo permette di avere più controlli nel disco, e di effettuare il ripristino del sistema da un determinato punto. Tutto questo viene controllato tramite "EWF Application Manager". EWF<sup>[10]</sup> supporta per un volume fino a 9 overlay.

La seguente tabella sottolinea pros and cons per ognuno dei tipi di overlay:

	Tipo basato su disco	Tipo basata su RAM
<b>Pro</b>	Protezione dei dati del disco da alterazioni o corruzioni.	Protezione dei dati del disco da alterazioni o corruzioni.
	Fornire più istantanee del contenuto del disco	Enable stateless operation.
	Consente la scrittura sul volume protetto delle scritture effettuate	Abilita XPE all'avvio del sistema senza un persistente sistema di archiviazione (Hard Disk).
	Permette il ritorno ad un precedente punto di vista.	
<b>Contro</b>	Richiede una unità di partizionamento per ospitare il sistema.	Richiede una memoria aggiuntiva per memorizzare i dati non salvati in una eventuale partizione
	Addizionali ulteriori richieste possono impattare sulle performance dei devices.	I dati vengono persi quando il sistema viene riavviato.



### 1.8.3 Configurazione EWF basata su disco

I passi successivi mostrano come configurare l'immagine per sostenere un EWF<sup>[10]</sup> overlay basato su disco:

Prima di tutto va aggiunto in Target Designer, il componente "Enhanced Write Filter" all'interno dell'immagine che è stata create (magari tramite il TAP.EXE). Quando si sta proteggendo un volume, si avrà bisogno di includere il componente "EWF NTLDR".

E' necessario configurare il settaggio selezionando DISK in

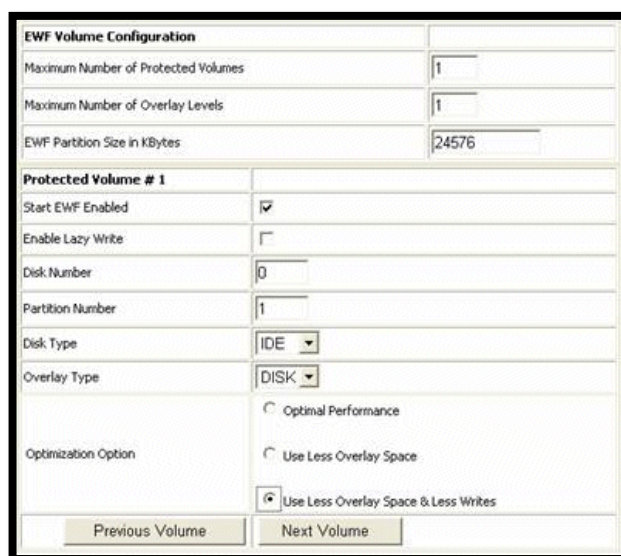


Figura 67 – Cofigurazione impostazioni EWF basato su DISCO

"overlay type" all'interno dell'"EWF Volume Configuration", oltre alla selezione del numero di volumi protetti e del livello di cache "Overlay Level".

La misura della partizione va selezionata secondo la quantità di spazio che si intende dare alla cache. Successivamente ci si deve assicurare di inserire il numero della partizione e il disco, per il volume protetto e si seleziona e attiva il checkbox Start EWF.

A questo punto bisogna configurare, costruire e distribuire l'immagine relativa al sistema da prendere in considerazione, in base al progetto che si andrà a realizzare. Si sarà costretti a partizionare il disco così da avere a disposizione uno spazio libero su una parte estesa del disco. Questa sarà usata da

EFW<sup>[10]</sup> per immagazzinare i dati nel disco "aggiuntivo". Tale spazio deve essere sufficientemente largo per accogliere i dati. Per esempio, se si deve disporre di 100 Mb liberi per il volume protetto, questa partizione dovrà essere almeno di 100 Mb.

Se non esiste partizione estesa e si hanno meno di 4 partizioni primarie, si avrà bisogno di lasciare spazio non ripartito nel drive.

Avviando il device, durante il First Boot Agent (FBA), EFW<sup>[10]</sup> si configurerà basandosi sul setting dei registri effettuato in fase di configurazione dell'immagine da distribuire. Quindi creerà e formatterà una Partizione EFW.

#### 1.8.4 Configurazione EFW basata su RAM

I seguenti passi mostrano come configurare l'immagine per supportare un EFW basata su RAM:

Innanzitutto bisognerà aggiungere il componente "Enhanced Write Filter" all'immagine creata tramite il Target Designer.

Andranno configurate le impostazioni per il proprio dispositivo impostando la dimensione della partizione EFW a 0 e selezionando in "Overlay Type" la dicitura RAM. In "EFW Volume Configuration" va selezionato il numero dei volume

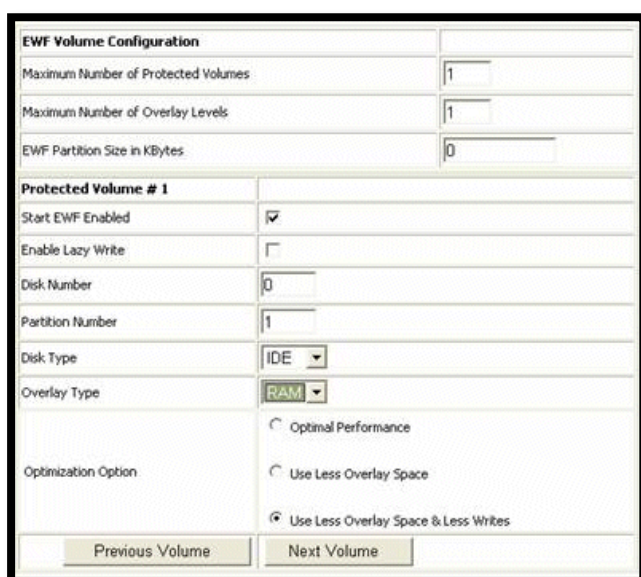


Figura 78 – Configurazione impostazioni EFW basata su RAM

protetti e "overlay levels" va impostato ad 1. Dopo essersi assicurati di aver inserito il numero di disco e di partizione relativi al volume protetto, andrà attivato il check box Start EWF.

Come abbiamo fatto per la configurazione basata su disco, a questo punto si passerà a configurare, costruire e distribuire l'immagine per il dispositivo.

Avremo bisogno a tal proposito di partizionare il disco assicurandosi di avere uno spazio libero disponibile per ogni partizione estesa di almeno 32 Kbyte. Questo spazio è usato da EWF<sup>[10]</sup> per memorizzare i dati di configurazione per la RAM di cache durante l'avvio.

Una volta avviato il dispositivo, durante il FBA (First Boot Agent) EWF<sup>[10]</sup> si configura in base alle impostazioni dei registri precedentemente settate al momento della configurazione dell'immagine del dispositivo, in questa fase il sistema si creerà una partizione EWF minima dove conservare le informazioni di configurazione.

### **1.8.5 EWF Manager Application**

EWF Manager Application è una console di utilità per gestire EWF sul dispositivo. Questo è un component opzionale che va aggiunto alla configurazione e che abilita al controllo delle operazioni EWF. E' possibile verificare lo stato di EWF tramite il seguente comando<sup>[11]</sup>:

#### ***Ewfmgr***

EWF manager visualizzerà un risultato simile al seguente:

```
Overlay Configuration
Volume Size                2048030208
Segments                   8192
Segment Size               249856
Free segments              8192
Max Levels                  3
Max Protected Volumes     1
Protected Volumes         1
Overlay volume percent full 0.00
Protected volumes
Arc Path "\Device\HarddiskVolume1"
```

Si può controllare lo stato del volume di cache EWF e attivare/disattivare EWF, settare i punti di salvataggio e ripristino delle modifiche.

### 1.8.6 Principali Errori

Il sistema permette anche di effettuare il debug dell'installazione, a tal proposito viene creato il file FBALog.txt di cui qui di seguito ne possiamo notare alcune righe di esempio:

```
4.53.43 - [FBASetProgressText] Setting PNP Flag...
4.53.45 - [FBASetProgressText] Initializing...
4.53.46 - [FBASetProgressText] Setting SystemPrefix...
4.53.46 - [FBASetProductKey] Evaluation product key
4.53.46 - [FBASetProgressText] Initializing User Profiles...
4.53.48 - [FBASetProgressText] Installing System Security...
4.54.02 - [FBAInstallSecurity] Successfully set security!
4.54.02 - [FBASetProgressText] Saving Security Settings...
4.54.17 - [FBASetProgressText] Installing Components...
4.54.18 - [FBALaunch] C:\WINDOWS\system32\regsvr32.exe /s
"C:\Program Files\Internet Explorer\ieproxy.dll" (ExitCode: 0x0)
4.54.18 - [FBASetProgressText] Registering Components...
```

Durante il processo di FBA, Quando il processo FBA è terminato si può esaminare questo file per ottenere dettagli sullo stato di EWF oppure per ricontrollare le informazioni ed assicurarsi che magari sia stata creata correttamente la partizione.

Quelli che seguono sono i principali errori che possono verificarsi in caso di una configurazione di Windows XPE basata su EWF.

#### Creazione della partizione EWF fallita

Questo è uno degli errori più comuni, e si verifica quando fallisce la creazione della partizione EWF, in tal caso ci si deve assicurare che ci sia spazio libero in una partizione estesa o spazio su una macchina con meno di 4 partizioni primarie. Questa partizione è usata da EWF<sup>[10]</sup> per memorizzare le informazioni di configurazione. Nel caso di EWF<sup>[10]</sup> basato su disco questa partizione sarà usata per

memorizzare i dati scritti per il volume protetto. Questa partizione la si può visualizzare all'interno della gestione disco nelle utility degli Strumenti di Amministrazione in Pannello di controllo di Windows, anche se in questo caso non ha ancora la lettera assegnata.

#### Riavvio infinito

E' un caso che si può verificare quando si configura FBA EWF in un sistema che contiene già una partizione EWF precedente. Caso tipico di una situazione di sviluppo in cui lo stesso sistema viene utilizzato più volte per lo sviluppo e la sperimentazione. Per risolvere il problema basta rimuovere la partizione utilizzando il seguente comando<sup>[12]</sup>:

#### ***Etprep /delete***

#### Errore durante la scrittura sulla partizione EWF

Di solito si ha questo problema quando si tenta di scrivere in una partizione EWF protetta. In tal caso viene visualizzato il messaggio "Error when writing EWF partition"

#### Delayed write Failed

Indica che Windows non è stato in grado di salvare tutti i dati per il file.

Questo succede quando si scrive fuori dello spazio della partizione EWF. Quando si aumenta lo spazio della partizione EWF (nel sistema basato su disco) o se si installa una memoria addizionale (nel sistema basato su RAM) sono due dei casi che possono causare questo problema.

### **1.8.7 Suggerimenti**

I seguenti argomenti affrontano alcune questioni inerenti il salvataggio e il volume del disco.

#### EWF normalmente deve essere fatto su alcuni storage persistenti

La maggior parte delle configurazioni EWF possono avere alcuni sistemi di memorizzazione assegnati a EWF. Questo è vero anche per il sistema basato su RAM ed è dovuto al fatto

che I driver EWF devono consentire la memorizzazione dei dati tra due boot. Per il sistema basato su disco, questi dati vengono memorizzati nella partizione di cache. Per I sistemi basati su RAM una piccola quantità di spazio (Circa 32 Kb) devono essere resi disponibili.

E' prevista una eccezione per I sistemi basati su RAM con un singolo volume protetto. In questo caso, la partizione EWF<sup>[10]</sup> viene cancellata dopo che è completato l'FBA. Questo permette scenari come EL TORITO BOOT o BOOT da FLASH MEMORY. In questi casi I settaggi vengono inseriti nei registri.

EWF supporterà solamente la quantità di spazio libero segnalata dal volume protetto

In entrambi I sistemi sia quello basato su disco che quello basato su RAM, EWF<sup>[10]</sup> supporterà solamente la scrittura per la quantità di dati trasmessa dal volume protetto. Questo è vero a prescindere dalla quantità di spazio su disco o su RAM disponibile nel sistema. Questo è dovuto al fatto che in EWF si assume che i dati possono essere persistenti alla base del volume.

Avvio del volume EWF basato su disco richiede EWF NTLDR

EWF NTLDR deve essere usato quando si effettua il boot da un volume EWF protetto basato su disco. Il component è usato al posto dello standard NT LOADER (NTLDR). EWF NTLDR non è necessario quando si effettua il boot da El Torito CD con EWF basato su RAM.

### **1.8.8 Considerazioni**

Enhanced Write Filter prevede tools flessibili per lo sviluppatore per proteggere il volume dalla scrittura. Questo abilita il sistema operativo e altri software che richiedono il media in lettura e scrittura di operare su di esso senza modifiche, basandosi quindi sulla configurazione base. Questo permette una flessibilità e viene usato in una varietà di

situazioni compreso il boot da flash media, CD-ROM o da dischi di sola lettura.





## Capitolo 2 – Windows Embedded vs Windows CE Embedded

### 2.1 – Panoramica Windows CE Embedded

La prima domanda che bisogna porsi analizzando Windows CE è che cosa sia questo sistema.

Windows CE è un sistema operativo sviluppato da Microsoft, a partire dal 1996, per dispositivi portatili (PDA, Palmari, Pocket PC), Smartphone e sistemi Embedded.

Come si intuisce dal nome, è un derivato della famiglia di sistemi operativi Windows, ma ha un kernel differente e non è quindi una semplice "riduzione". Le API e l'aspetto grafico sono comunque molto simili.

Il termine "Windows CE" (compact edition)<sup>[13]</sup> è in realtà il nome "tecnico" con il quale viene indicata la piattaforma generale di sviluppo di questo sistema operativo. Essendo "Windows CE" sufficientemente modulare e flessibile, sono state sviluppate delle specifiche versioni per dispositivi differenti (oltre che per processori differenti dal x86, quali MIPS, ARM, Hitachi SuperH, Intel XScale).

Tali specifiche versioni prendono il nome "commerciale" di "MS Handheld 3.0" (e 3.1), "MS Handheld 2000", "Microsoft Pocket PC 2000" (e 2002), "MS Smartphone 2002", "MS

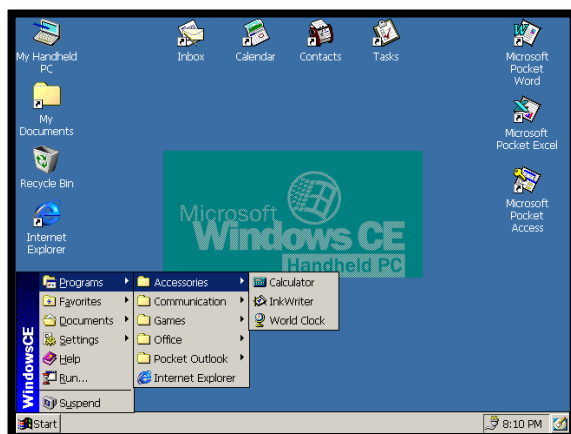


Figura 89 – Schermata di una installazione Windows

Windows Mobile 2003" (e 2003 Second Edition), "MS Windows Mobile 5.0", "Microsoft Auto". Tali varianti fanno tutte riferimento a specifiche evoluzioni della piattaforma di riferimento "Windows CE", passato dalla v1.0 alla v6.0 (del 2007).

Tale moltitudine di nomi non è raro che generi ambiguità, impiegando erroneamente in modo ambivalente i termini "Windows Mobile" o "Windows CE" e simili.

L'ambiguità è accentuata dal fatto che la piattaforma Windows CE è largamente distribuita a tutti i produttori di sistemi Embedded (dai citofoni ai robot industriali), mentre Windows Mobile ha termini di licenza più restrittivi. Purtroppo pur derivando da Windows CE, Windows Mobile differisce notevolmente da esso dal punto di vista della programmazione ed è quindi molto facile che un applicativo scritto per piattaforma mobile non sia eseguibile su CE anche a parità di architettura del processore (Windows Mobile è compilato esclusivamente per ARM). La portabilità del codice può essere garantita scrivendo codice in C# o VB.NET tramite il .Net Compact Framework.

Microsoft ha risolto queste ambiguità con l'ultimo sistema operativo Windows Mobile 6.0, creando apposite release per le varie tipologie di dispositivi. Più precisamente: "Windows Mobile 6.0 Classic" per i palmari privi di modulo telefonico; "Windows Mobile 6.0 Standard" per gli smartphone, ovvero i terminali "data centric" non dotati di touchscreen; infine "Windows Mobile 6.0 Professional", la versione più ricca riservata ai PDA phone, ovvero quei device tascabili caratterizzati dalla presenza del modulo telefonico integrato, spiccatamente "data centric" e contraddistinti principalmente dallo schermo sensibile al tocco.

Occorre precisare che, nonostante la dizione "Windows Mobile 6", lo stesso si basa ancora sulla versione di Windows CE 5.0 e non sull'ultima release denominata Windows CE 6.0.

Windows Mobile è un sistema operativo compatto basato sulle



Figura 910 – Schermata di una installazione Windows Mobile

API Win32 della Microsoft. Fra i dispositivi che ne possono essere equipaggiati ci sono i Pocket PC, gli Smartphone ed i Portable Media Center. La sua interfaccia grafica è pensata in modo da essere il più possibile simile a quella di Windows.

Diverse sono le caratteristiche ad oggi che il sistema ha in comune con il suo "fratello" maggiore Windows:

- La schermata iniziale "**Oggi**" contiene la data corrente, i dati del proprietario, gli appuntamenti più vicini, l'elenco delle e-mail e l'elenco delle cose da fare (la classica "*To Do list*"). Altri elementi sono: la barra con le icone per le notifica dello stato della connessione Bluetooth ed il pulsante "Nuovo". Altri programmi applicativi possono essere installati per aggiungere funzionalità alla schermata "Oggi", come, ad esempio, *Microsoft Money* per Pocket PC. Lo sfondo può essere personalizzato con colori e "temi" diversi.
- Nella taskbar sono visualizzati l'ora, il volume, e lo stato della connessione a Internet. Quando si apre la finestra di un programma o di un messaggio, nello spazio vuoto a destra dell'orologio appare un "OK" oppure un'icona per chiudere la finestra. L'elemento più importante della taskbar è il pulsante "Start", molto

simile a quello del desktop di macchine Windows. Premendo questo pulsante si apre il *Menu Start*, che contiene, in alto, l'elenco dei programmi usati di recente, nove voci personalizzabili, collegamenti ai programmi, impostazioni, ricerca e guida in linea.

- In Windows Mobile sono integrate alcune applicazioni della suite Microsoft Office in versione adattata a Pocket PC, fra cui Pocket Word e Pocket Excel. Nella versione 5.0 di Windows Mobile è inserito anche Pocket PowerPoint. Si tratta di versioni che offrono molte delle funzionalità delle rispettive versioni desktop, ma mancano di altre, come l'inserimento di tabelle e immagini. Con l'utility ActiveSync è possibile convertire i file in formato compatibile con Pocket PC. Anche Outlook è incluso, con le classiche funzioni di agenda, calendario, contatti, posta.
- Anche Windows Media Player è disponibile integrato in Windows Mobile, attualmente nella versione 9, mentre la versione 10 viene pre-installata sugli apparecchi con caratteristiche hardware più recenti. Per i dispositivi esistenti è possibile scaricare da internet e installare la versione 10, che però non è compatibile con i modelli di tutti i costruttori. I formati di file multimediali supportati includono .WMA, .WMV, .MP3, e MPEG, ma non .AVI e WAV, che possono essere riprodotti solo installando un player separato. Alcune versioni di Windows Mobile permettono di riprodurre anche i file audio .M4A.

I vantaggi dell'utilizzo di Window CE sono:

- Connettività (Circa 700 Componenti)  
Utilizzando solo i componenti necessari, è possibile ridurre lo sviluppo globale, l'ottimizzazione del sistema operativo dimensioni, e più bassi costi di hardware.
- La fiducia di programmare con Visual Studio

Windows Embedded CE 6.0 semplifica il processo di sviluppo, permettendo di sfruttare la conoscenza di Visual Studio nativo per creare applicazioni e driver per CE 6.0 rapidamente. Dal codice sorgente condiviso deriva la disponibilità ad offrire una vasta selezione della qualità nella produzione driver, oltre alla disponibilità ulteriori informazioni sulle caratteristiche dei prodotti Windows Embedded CE che danno un vantaggio competitivo.

- Comunità e supporto

Una comunità attiva di sviluppo e una rete di partner leader del settore sono disponibili per aiutare l'utente in ogni passo del ciclo di sviluppo. Lavorare con un computer Windows Embedded Partner presto promuove lo sviluppo e risparmio. Windows Embedded CE è anche economicamente efficace con costi di licenza a runtime determinata per unità spedite.

## 2.2 – Requisiti per l'uso e principali utilizzi

Il PC su cui installare il sistema di sviluppo richiede l'installazione di Microsoft Visual Studio che in base alla versione deve prevedere la versione 2005 o precedenti e di conseguenza i requisiti minimi hardware associati all'installazione di questo sistema di sviluppo.

Windows CE supporta direttamente i vari tipi di periferiche e dispositivi hardware, come tastiere, mouse, touch-panel, porte seriali, Ethernet, modem, dispositivi USB, dispositivi audio, porte parallele, dispositivi di stampa e dispositivi di archiviazione (ATA o flash media).

Allo stesso tempo come Windows CE si estende a nuovi settori e categorie di dispositivi, vi è un enorme potenziale per Embedded sviluppatori di aggiungere facilmente nuovi tipi di dispositivi e periferiche.

Al momento il processore e le architetture supportate comprendono:<sup>[14]</sup>

NEC, Philips, Toshiba MIPS 39xx e 4xxx, Motorola PowerPC 821, 823, 850, 860, Hitachi SH3 e SH4, Intel e Pentium 486 (e compatibili: AMD, Cyrix, SGS Thomson) , ARM, e Intel X-Scale.

Per quanto riguarda invece gli utilizzi che vengono presi in considerazione per i sistemi Windows CE Embedded possiamo delineare alcune principali categorie:

- PC basati su Windows
- PC palmari o H / PC. (H/PC consentono di fornire i milioni di professionisti mobili che utilizzano personal computer basati su Windows un complemento di PC conveniente e semplici da utilizzare, per eseguire le informazioni più importanti quando sono fuori il PC).
- dispositivi di elaborazione mobile piccola del fattore di modulo, ad esempio:
  - "wallet"Computer
  - periferiche di comunicazione senza fili (ad esempio cercapersone)
  - le informazioni digitali e cellulari telefoni intelligenti,
  - console di elementi multimediali e intrattenimento all'avanguardia (inclusi lettori DVD e purpose-built Internet accesso periferiche quali Internet TV, digitale set-top box e Internet"telefoni Web).

### 2.3 – I due sistemi a confronto

Ci sono pregi e difetti in entrambe le tecnologie, pertanto analizzeremo queste tecnologie indicando le motivazioni alla base della scelta di Window CE o Window XPE.

Negli ultimi anni i sistemi operativi sono stati spesso di dimensioni eccessive. Con Windows CE e Windows XP Embedded si è affrontato questo problema trovando una

soluzione adeguata; infatti essendo sistemi basati su componenti, che ne permettono una scelta degli stessi a seconda del dispositivo da sviluppare cosa che ha portato alla diminuzione le dimensioni totali dell'immagine da distribuire. Windows CE include circa 500 componenti, mentre Windows XP Embedded ne ha circa 12.000; di questi, circa 9.000 sono driver di periferica, mentre i componenti rimanenti sono tecnologie del sistema operativo.

La dimensione minima della build di Windows CE è di circa 200 KB. La dimensione dell'immagine di un gateway domestico è di circa 4 MB e include il supporto delle reti LAN o wireless, dell'interfaccia di amministrazione remota (server Web). Un'immagine simile a quella di un PDA che include la shell di Windows, funzionalità di networking, il protocollo RDP, un browser Internet, Windows Media Player, COM/DCOM e così via, richiede circa 18 MB. Ovviamente, tutto questo può essere ottimizzato in base alla specifica applicazione che si sta sviluppando.

La dimensione minima della build di Windows XP Embedded è di circa 5 MB. Si tratta di una build basata esclusivamente sul kernel, difficilmente utilizzabile, anche se è possibile avviarla ed eseguirla. La dimensione di una immagine di Windows XP Embedded si aggira mediamente intorno ai 200 MB o anche oltre, ma comunque molto meno rispetto a un'installazione tipica di Windows XP Professional per sistemi desktop. Durante lo sviluppo dell'immagine del sistema operativo, è possibile selezionare i componenti hardware e software da includere nella piattaforma. Se Windows Media Player, DCOM, RPC o Microsoft Internet Explorer non sono necessari, si eviterà di includerli nell'immagine.

L'architettura del sistema operativo Window CE è completamente diversa da quella dei sistemi operativi desktop Windows. In un'articolo<sup>[15]</sup> di RTC Magazine vengono illustrate le funzionalità real time di Windows CE e nei white paper<sup>[16]</sup>

della società di consulenza Dedicated Systems Experts viene fornita una valutazione di Windows CE 5.0 e, in particolare, di Windows CE 5.0 su piattaforme x86.

Windows XP Embedded non è un sistema operativo real time, ma per renderlo tale esistono numerose estensioni di terze parti<sup>[17]</sup>.

Per quanto riguarda la sicurezza già da Windows CE 5.0 con l'implementazione dell' OEMCertifyModule è stato consentito di bloccare le immagini del sistema operativo in modo da eseguire soltanto il codice desiderato. In Windows CE Inoltre le funzionalità di protezione sono attivate per impostazione predefinita. Per una descrizione di queste funzionalità, si può consultare l'articolo<sup>[18]</sup> di WindowsForDevices. Essendo un sistema operativo basato su componenti, Windows CE permette inoltre di includere nel dispositivo in corso di sviluppo soltanto i componenti e le tecnologie necessari. Un altro aspetto interessante è la differenza a livello di architettura tra Windows CE e i sistemi operativi desktop. Questi ultimi includono tre librerie principali a cui ogni programma deve fare riferimento: Kernel32, GDI32 e User32. Nessuna di queste librerie è invece presente in Windows CE pertanto, anche se il dispositivo viene eseguito con un processore x86, non sarà possibile eseguire neanche l'applicazione desktop più semplice, come ad esempio il Blocco note. Di conseguenza in Windows CE non funzioneranno neanche virus, worm e altri programmi pericolosi per la sicurezza progettati per i sistemi operativi desktop.

Per quanto riguarda invece Windows XP Embedded la prima cosa da dire è che, essendo basato su componenti, Windows XP Embedded consente di scegliere i componenti da includere nell'immagine del sistema operativo. Se non si necessita di componenti di rete quali MSMQ, DCOM, RPC o altri ancora, semplicemente si eviterà di includerli nell'immagine.



Osservando il comportamento di alcuni virus e worm in un desktop infetto, si nota che la maggior parte di questi sfrutta RPC, DCOM e altre porte aperte nel PC. Windows XP Embedded offre tutti gli aggiornamenti per la protezione presenti nella versione desktop, che includono un firewall (le cui porte sono tutte disattivate per impostazione predefinita, ad eccezione della porta 80) unito al supporto per funzionalità NX (No Execute) e per l'identificazione dei buffer overruns (entrambe sono descritte in un'intervista<sup>[19]</sup> a Joe Morris realizzata dal team di Windows XP Embedded). Sono inoltre disponibili soluzioni di terze parti quali Computer Associates e Trend Micro.

Windows CE Il sistema operativo è basato su componenti e il relativo strumento di sviluppo (Platform Builder) include una procedura guidata che assiste l'utente nella configurazione iniziale della piattaforma. In questa procedura sono disponibili nove configurazioni diverse, tra cui le configurazioni "solo kernel" (200 KB), gateway residenziale, Internet o WebPad, ognuna delle quali può essere considerata il punto di partenza dell'attività di progettazione. È possibile aggiungere o rimuovere caratteristiche dall'area di lavoro della piattaforma, ad esempio aggiungendo il supporto di .NET Compact Framework o rimuovendo il supporto del server Web HTTP. Lo strumento di sviluppo della piattaforma consente inoltre di aggiungere applicazioni e driver personalizzati<sup>[20]</sup>. Windows CE include potenti driver e pacchetti BSP (Board Support Package) che facilitano le operazioni di configurazione, sviluppo, debug e testing. Gli strumenti di sviluppo includono il codice sorgente 2,5 milioni di righe distribuiti in licenza shared source, che può essere utilizzato come supporto per le attività di sviluppo e debug della piattaforma e per comprendere meglio il funzionamento del sistema operativo Windows CE.

Anche Windows XP Embedded, come abbiamo più volte ripetuto, è un sistema basato su componenti. Come Windows CE, Windows XP Embedded offre numerosi modelli "di partenza" per lo sviluppo del sistema operativo, che possono essere applicati a set-top box, terminali Windows, dispositivi Internet e così via. È comunque possibile cominciare da zero e selezionare i componenti hardware e software desiderati. La procedura finalizzata ad eseguire Windows XP Embedded su una nuova piattaforma è molto semplice. L'hardware e l'architettura x86 sottostanti consentono di eseguire uno strumento in grado di analizzare l'hardware e produrre un file di output XML contenente un elenco dei componenti hardware per la reference board. Questa definizione XML permette di visualizzare in modo ordinato la piattaforma hardware per la reference board. A questo punto è sufficiente aggiungere uno dei modelli o singoli componenti software alla definizione dell'hardware e quindi sviluppare il sistema operativo. Tutto con estrema facilità<sup>[21]</sup>.

Per quanto riguarda le differenze e i costi dei due sistemi Windows CE e Windows XP Embedded possono essere scaricati gratuitamente dal sito Web Microsoft<sup>[22]</sup>. Inoltre viene data all'utente la possibilità di valutare il sistema operativo e i relativi strumenti per un periodo di 120 giorni allo scadere dei quali si può acquistare il prodotto completo. La versione di valutazione di Windows CE include gli stessi strumenti per lo sviluppo e i 2,5 milioni di righe di codice sorgente del prodotto completo.

L'acquisto delle licenze per i sistemi operativi Embedded diventa necessario solo quando si inizia a fornire ai clienti il prodotto reale. Fino ad allora, è possibile utilizzare gli strumenti della versione di valutazione (o del prodotto completo) per creare e inviare ai clienti versioni di test dell'immagine del sistema operativo, senza bisogno di alcuna licenza.

Esistono diversi modi per ottenere supporto per Windows CE e Windows XP Embedded, dal quello gratuito ([www.microsoft.com/embedded](http://www.microsoft.com/embedded)) ai corsi di formazione, alla consulenza dei partner al servizio Microsoft Developer Support. Il supporto gratuito include l'accesso a newsgroup monitorati dai partner e team di sviluppo, chat online ed esercitazioni.

In merito al supporto di cui si ha bisogno, soprattutto per quanto riguarda i drivers, Windows CE include almeno un Board Support Package<sup>[23]</sup> per ogni processore supportato (x86, ARM, MIPS, SH4)<sup>[24]</sup>. Windows CE offre a sviluppatori hardware, integratori di sistemi e fornitori di prodotti IT l'interessante capacità di estendere il catalogo di componenti Windows CE tramite l'aggiunta del supporto per reference board, BSP e driver personalizzati.

Windows XP Embedded include lo stesso set di driver della versione desktop di Windows XP Professional, ovvero oltre 9.000 driver disponibili come singoli componenti. Questo elenco può essere facilmente esteso tramite gli strumenti di sviluppo per Windows XP Embedded. Se si dispone di driver di terze parti per Windows 2000 o Windows XP, è possibile includerli nel catalogo di Windows XP Embedded importandone il file INF. Una volta importato il file INF e verificato il componente nel database di componenti, il nuovo driver può essere utilizzato esattamente come qualsiasi altro driver esposto negli strumenti di sviluppo.

Sia in Windows CE che in Windows XP Embedded la shell può avere l'aspetto di un desktop Windows, completo di pulsante Start, icone e così via. In entrambi i casi, però, la shell di Windows può essere interamente rimossa dal sistema operativo, per avere la possibilità di eseguire l'avvio nell'applicazione personalizzata. L'utente del dispositivo non necessita di un'interfaccia Windows, non può accedere ad alcun desktop né al Pannello di controllo (perché non

esistono) e così via. È possibile sviluppare un'applicazione da utilizzare come shell del dispositivo. In Windows XP Embedded sono disponibili numerose shell di esempio, tra cui la shell di Windows Explorer, la shell comandi e la shell di Task Manager.

## Capitolo 3 – Caso particolare di utilizzo (IL CHIOSCO INFORMATIVO)

### 3.1 – Panoramica

Parliamo a questo punto di quello che si è realizzato utilizzando la tecnologia a nostra disposizione.

Un chiosco informativo deve essere realizzato cercando di prevedere tutto ciò che potrà servire al dispositivo già in laboratorio:

- l'aggiornamento guidato. Il prodotto dovrebbe essere certificato di nuovo in laboratorio per qualsiasi modifica o aggiornamento, sia del sistema operativo, che applicativi sviluppati internamente o di quelli di terzi integrati nel prodotto.
- l'affidabilità. Il progetto del prodotto deve curare l'affidabilità in tutte le fasi di lavoro, di aggiornamento e anche di situazioni prevedibili (cadute improvvise di tensioni, aggiornamenti parziali, ecc...), così da mantenere operativo il sistema il più a lungo possibile;
- la reperibilità. Il materiale (sia hardware che software) utilizzato per il prodotto deve essere reperibile per tutta la durata di vita del prodotto. Microsoft garantisce 10 anni di distribuzione per ognuna delle release dei sistemi operativi Windows Embedded.

Un'altra osservazione importante che ci può far capire molte scelte architettoniche è che il destinatario oggetto dei prodotti Embedded non è l'utente finale, ma quasi sempre un "produttore" che, una volta adattato e personalizzato il sistema operativo per le proprie esigenze, crea un "prodotto finale" pronto per il mercato.

Fatte queste premesse si è passati alla realizzazione del prodotto chiosco seguendo delle fasi accurate, che vanno eseguite nella sequenza esatta qui di seguito esposta.

## 3.2 – Una soluzione con PC di HW minimo e SW installato su Hard Disk

### 3.3.1. Requisiti

In questa prima fase non sono richiesti requisiti particolari, poiché il tutto può essere installato su un qualsiasi PC (x86). L'unica considerazione che può essere fatta riguarda l'hard disk in quanto questa soluzione si può percorrere sia utilizzando un hard disk che una pen drive che permette la scrittura di circa un milione di volte. Per ovviare al problema seppur abbastanza fittizio, legato al limite della scrittura della pen drive, bisognerebbe prendere in considerazione quanto detto nel Capitolo dell'EFW e con attenzione alla re-direzionare dell'overlay in RAM nel momento in cui il sistema avesse l'esigenza di scrivere dei dati.

### 3.2.2. Come realizzare il sistema

Si è partiti nell'analisi di quella che potrebbe essere l'implementazione di un chiosco, dal caso più semplice ossia quello di installare all'interno dell'Hard disk del PC tutto il software che potrebbe essere utile per eseguire il chiosco.

Nel nostro caso ci siamo limitati ad installare il sistema in un semplice Hard Disk.

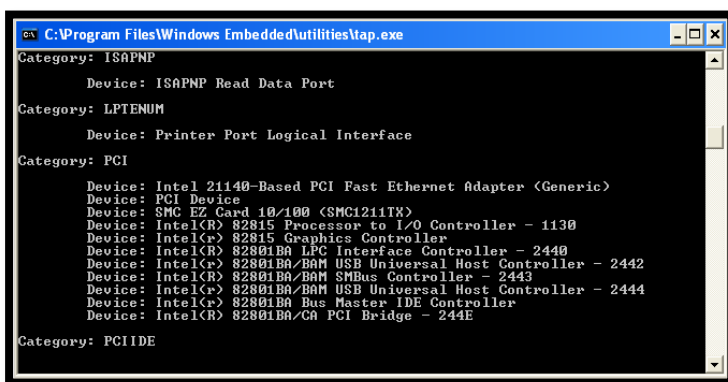


Figura 1011 – Esecuzione programma TAP.EXE

In questa prima fase sono stati eseguiti diversi passi.

Innanzitutto abbiamo copiato il Target Analyzer Probe (TAP.EXE) all'interno del disco che è stato utilizzato per la

**Capitolo 4 – Caso particolare di utilizzo (IL LETTORE MULTIMEDIALE)**

realizzazione del chiosco e montato su questo. Tale programma ci ha permesso reperire tutte le informazioni importanti relative all'architettura hardware del sistema come in figura 13 (Disco, Periferiche, Stampati collegate, Schede di rete, ecc.). Al termine dell'esecuzione del programma infatti è stato generato un file "devices.pmq". A questo punto il file è stato trasferito nel PC di sviluppo in cui era stato precedentemente installato Windows Embedded Studio 2009 e qui è stato importato.

All'inizio è stato creato un nuovo progetto cliccando all'interno dell'applicativo Target Devices in File → New (Per la creazione di un nuovo progetto), digitando il nome del progetto e scegliendo la piattaforma, in questo caso «Windows XP Embedded Client (x86)».

Andando alla schermata di Figura 12 è possibile ultimare la configurazione del progetto indicando quelle che sono le credenziali dell'azienda che realizza il progetto per il cliente.

Name:	Demo Platform		
Version:	1.0	Revision:	1
Owner(s):	Your Name	Author(s):	Your Name
Vendor:	Microsoft Corporation	Copyright:	Copyright (c) 2002 Micr
Created:	5/3/2002 5:57:59 AM	Revised:	5/3/2002 5:57:59 AM
Description:	Configuration of Microsoft(R) Windows XP Embedded Client Platform Definition		
Build type:	Release	Advanced...	

Figura 1142– Impostazioni progetto Windows XPE

Successivamente si è passati alla configurazione dei dati relativi alla partizione che avrebbe ospitato nel disco indicando all'interno del Target devices nella schermata centrale alla voce Setting e utilizzando il tasto destro per visualizzare il contenuto editabile ed è stato settato tutto come nelle impostazioni di seguito indicate:

```
Boot Drive=C:  
Windows Folder=C:\Windows  
Program Files Folder=C:\Program Files  
Document and Setting Folder=C:\Document and Setting  
Boot ARC path=multi(0)disk(0)rdisk(0)partition(1)  
Boot Partition Size (MB)=700
```

Queste impostazioni sono state così configurate in quanto il tutto è stato installato su un Hard disk con una sola partizione e un solo disco.

Nel caso ci fossero state più partizioni o più dischi questa configurazione sarebbe stata leggermente differente, variando magari il Boot Drive (Es. D) e le altre cartelle che non avrebbero puntato a C ma magari a D o variando il numero di partizione, non la prima ma magari la seconda o la terza del disco 0 o del disco 1 configurabile in Boot ARC path, per i dischi IDE utilizzando gli ultimi due parametri (rdisk e partition) mentre se SCSI utilizzando anche gli altri due valori del parametro in questione (multi e disk).

Effettuata questa fase si è passati al caricamento del file "devices.pmq" che era rimasto alla fase di ricerca dell'hardware tramite il TAP. L'importazione dello stesso viene effettuata cliccando su:

*File → Import*

e selezionando il file che precedentemente era stato creato nel Hard disk del device (in questo caso il chiosco).

Dopo una procedura automatica di importazione dei devices analizzati dal TAP, viene prodotta dal sistema di Target designer una schermata all'incirca come quella in figura 13.



## Capitolo 4 – Caso particolare di utilizzo (IL LETTORE MULTIMEDIALE)

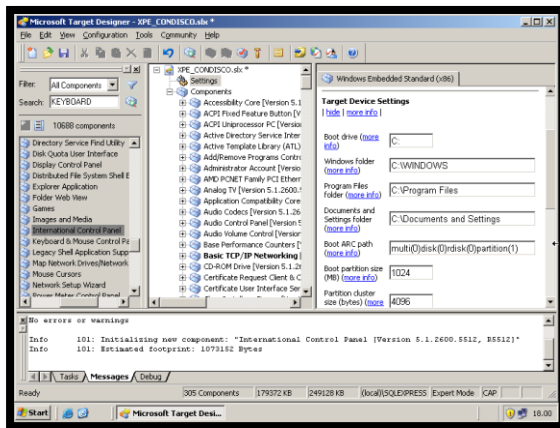


Figura 1243 – Configurazione Target Designer Chiosco con Disco

Nella fase successiva, dopo aver importato il target abbiamo aggiunto quei componenti utili al sistema che la procedura guidata non aveva importato. Questi componenti sono in particolare:

- quello per la gestione dello schermo e delle sue varie risoluzioni (Display Control Panel);
- quello per la gestione della tastiera italiana (Keyboard and mouse Control Panel e Italian Language Support);
- il componente presente tra i template ossia Kiosk/Gaming console che contiene diversi delle impostazioni e dei Componenti utili per un chiosco senza doverli andare a trovare tra la miriade di componenti di XPE;
- I componenti che contengono la dicitura IIS che permettono l'installazione del Web server sulla macchina XPE chiosco (componente importantissimo per la gestione del chiosco informativo);
- il componente Windows Media Player (per quanto riguarda il chiosco multimediale).

A questo punto è stata effettuata la funzione del Target Designer ossia

*Configuration → Check Dependencies*

per controllare se ci fossero problemi di dipendenze tra i vari componenti aggiunti al progetto. Qualora mancasse qualche

dipendenza, o qualche altra fosse errata, il sistema lo avrebbe rilevato fino a darci un progetto privo di errori di dipendenza.

A questo punto il progetto è risultato pronto per la fase di generazione dei file binari che devono essere copiati nel disco affinché il sistema operativo possa essere installato. Questa operazione effettuata tramite la funzione

*Configuration → Build Target Image*

in cui il sistema ci chiede (cliccando il tasto Build in essa presente) se si vuole effettuare il check delle dipendenze qualora non sia stato fatto e successivamente passa alla fase di generazione dell'immagine compressa.

Fatta questa operazione e copiati i file nel disco del chiosco la procedura richiede l'avvio del dispositivo che in questa prima fase visualizzerà delle schermate simile a quelle in figura 15, infatti verrà lanciata la procedura FBA (First Boot Agent) che provvede alla scompattazione dei file binari generati e che come detto in precedenza, si basa su quanto impostato nel Target Designer per configurare il nostro sistema.

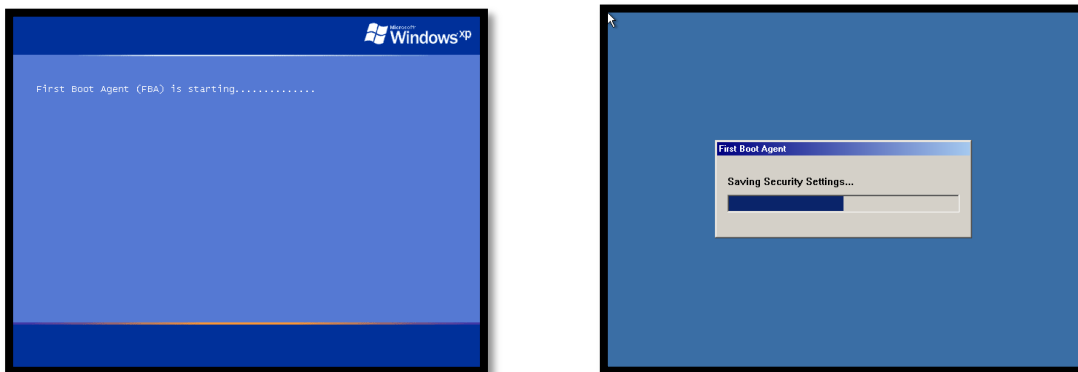


Figura 1314 – Procedura di First Boot Agent (FBA)

Al termine di questa procedura se non si verificano errori, riavviando il sistema si avvierà il nostro sistema Windows XPE Embedded che avrà una schermata simile a quella in figura 15.

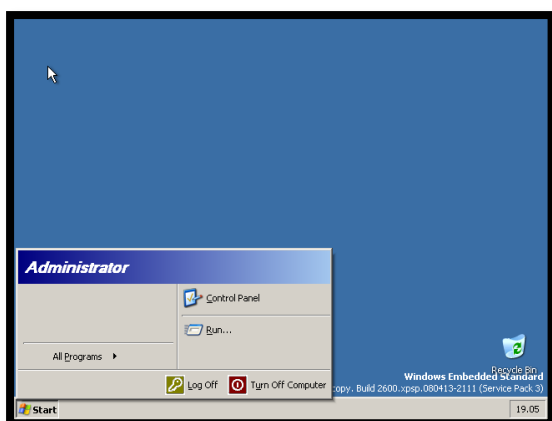


Figura 1415 – Windows XPE in esecuzione

### 3.2.3. Considerazioni

Questa prima parte della realizzazione del sistema (hard disk) andrebbe sempre attuata nella realizzazione di un sistema Embedded, perché spesso la configurazione richiede di effettuare diversi controlli prima di essere realmente funzionante. I motivi possono essere differenti. Ad es. potrebbe mancare un componente o essersi verificato un errore nella fase di configurazione del disco o della partizione. Pertanto in questi casi bisogna ripetere le operazioni più volte, quindi se fatte su hard disk fanno risparmiare un po' di tempo in fase di progettazione.

Questa fase, come preannunciato nel paragrafo precedente, ha il vantaggio di avviare il sistema in un tempo particolarmente veloce (a volte anche in 20/30 secondi), tempo neanche paragonabile a sistemi quali Windows XP Home o Professional. Questa informazione risulta particolarmente utile per pensare di installare XPE su sistemi quali NetBook o PC aziendali che richiedono particolari semplici operazioni e devono essere avviati in poco tempo.

In alcuni casi con l'aiuto dell'opzione EWF si può anche rendere il sistema aziendale più sicuro, impedendo intrusioni esterne nel sistema; infatti le scritture essendo re-direzionate su altre partizioni del disco fanno sì che in caso di attacchi (magari di virus) la parte del sistema operativo non venga toccata.

Uno svantaggio di questo approccio è dato dal fatto che in un sistema come questo basato su disco, le principali attenzioni inerenti il discorso sicurezza vanno rivolte a lui, infatti qualora esso venga danneggiato, il sistema Embedded installato verrà meno, e pertanto andranno ripetute le operazioni, che se il disco non sarà esattamente uguale a quello precedentemente installato si dovrà ripartire dall'esecuzione del TAP.EXE di cui abbiamo già parlato all'inizio della descrizione della procedura.

Per quanto riguarda in nostro progetto, il chiosco, questa soluzione seppur funzionante, non è ideale per il fatto che spesso i chioschi sono più di uno, uguali in ogni pezzo e connessi in rete. A tale scopo si è pensato di sviluppare un'immagine e di metterla su un server da cui ogni client (chiosco) effettua il bootstrap via rete. Analizzeremo nel dettaglio questa opzione nei prossimi paragrafi.

Prima di concludere questa fase del progetto è necessario inoltre parlare di alcuni componenti che risultano utili di cui analizzeremo l'importanza e la configurazione:

- EWF NTLDR
- Enhanced Write Filter

Analizziamo i singoli componenti.

EWF NTLDR come abbiamo già detto in precedenza, serve nel caso in cui, si usa il sistema EWF<sup>[10]</sup>, per la protezione del disco, in questo caso utilizzando la memoria RAM come redirect per la scrittura. In questo caso il componente va semplicemente inserito nel progetto senza particolari configurazioni.

Il secondo componente è invece Enhanced Write Filter, ed è il componente principale di configurazione del sistema EWF<sup>[10]</sup>.

In questa fase il componente è stato utilizzato con la configurazione di figura 16.

## Capitolo 4 – Caso particolare di utilizzo (IL LETTORE MULTIMEDIALE)

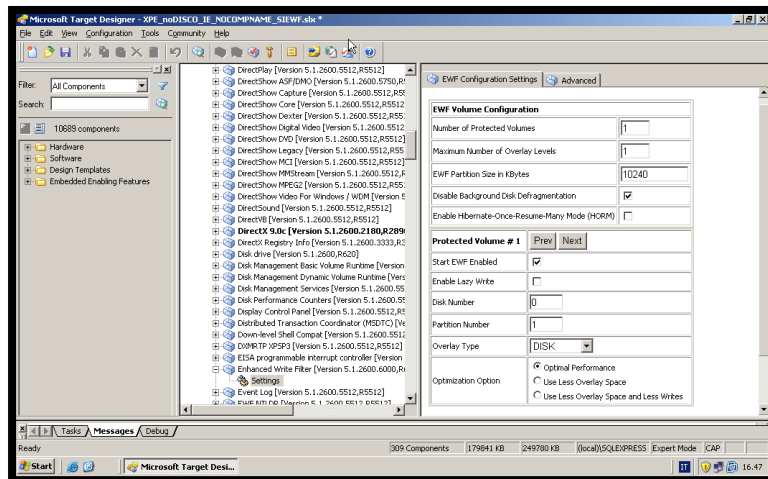


Figura 1546 – Configurazione EWF in chiosco con DISCO

### 3.3 – Una soluzione con PC di HW minimo e SW installato su Server

#### 3.3.1. Requisiti

Per la realizzazione del progetto si devono avere a disposizione diversi strumenti, che permettano di avviare un sistema XPE da un server di rete, Ovvero:

- Un sistema DHCP (Nel nostro caso abbiamo utilizzato Windows 2003 Server Enterprise SP2.
- Un BIOS che preveda l'avvio remoto del sistema operativo ossia PXE (Preboot Execution Environment)<sup>[25]</sup>.
- Un sistema x86 su cui eseguire l'immagine che viene creata.
- Un disco da utilizzare solo nella prima fase di creazione dell'immagine.

Se si è in possesso di tutti i requisiti per l'installazione del chiosco e dopo essersi assicurati che l'immagine con disco sia priva di errori e quindi utilizzabile, si passa alla fase che verrà descritta nei prossimi capitoli. Prima bisogna che vengano dedicati alcuni istanti a capire come va configurato il sistema per l'utilizzo, e che cosa sia il PXE (utile per l'avvio remoto).

### 3.3.2. Panoramica sul PXE (Preboot Execution Environment)

Preboot Execution Environment (PXE)<sup>[25]</sup>, ovvero ambiente di esecuzione pre-boot, è un metodo per eseguire il boot di un computer utilizzando una connessione di rete ethernet ed il supporto di un server, senza bisogno di una unità di memoria di massa.

PXE<sup>[25]</sup> fu introdotto come parte del Framework Wired for Management di Intel, ed è descritto nella specifica (v2.1) pubblicata da Intel e Systemsoft il 20 settembre 1999.

Questa voce descrive la modalità di funzionamento prevista nella specifica ma anche alcune varianti comunemente implementate.

PXE<sup>[25]</sup> utilizza molteplici protocolli di rete:

- IP
- UDP
- DHCP
- TFTP

Inoltre utilizza concetti come GUID/UUID e Universal Network Device Interface ed estende il firmware del client PXE (il computer che viene avviato via PXE con un ulteriori Application programming interface).

Il boot via PXE viene utilizzato generalmente nei seguenti scenari:

- installazione via rete di un sistema operativo. Il sistema operativo viene installato in modo tradizionale, ovvero sulla memoria di massa. Questa modalità permette di installare un sistema operativo senza dover acquistare o masterizzare un cd-rom, e funziona anche in assenza delle periferiche spesso usate a questo scopo (lettore cd-rom, memorie USB, possibilmente anche monitor e tastiera)

**Capitolo 4 – Caso particolare di utilizzo (IL LETTORE MULTIMEDIALE)**

- possibilità di caricare su un calcolatore un sistema operativo diverso da quello normalmente usato, ad esempio per riparare una installazione mal funzionante o per utilizzarlo per funzioni particolari.
- network computing, ovvero utilizzo primario di un sistema operativo caricato dalla rete.
- Ed ora anche per la gestione del dispositivo chiosco.

PXE può essere considerato uno dei modi per caricare il sistema operativo su un calcolatore.

Quando è presente la funzionalità PXE, il bios presenta all'utente la possibilità di scegliere, tra i possibili dispositivi, quello da cui caricare il sistema operativo, il boot da rete.

Se viene selezionata questa opzione, il firmware PXE utilizza DHCP per configurare un indirizzo IP sulla scheda di rete ed individuare un server ("boot server") da cui scaricare via TFTP un file, immagazzinandolo nella memoria RAM. Questo file viene poi eseguito, avviando così un sistema operativo.

Nella modalità di funzionamento prevista dallo standard, il server DHCP non fornisce direttamente il boot server, ma agisce da "PXE redirection service", ovvero redirige il client ai server PXE disponibili. I server PXE sono server DHCP modificati, che utilizzano generalmente la porta UDP 4011, e forniscono al client l'indirizzo IP del boot server ed il nome del network bootstrap program (NBP) da scaricare.

Nella pratica, normalmente il firmware PXE è in grado di ricevere direttamente dal server DHCP i parametri necessari per identificare il boot server e il file da scaricare, ed è quindi possibile configurare un normale server DHCP per fornire i parametri necessari al boot via PXE.

PXE è stato progettato per essere utilizzabile su molte architetture. La specifica v2.1 definisce identificatori di architettura per 6 tipi di sistemi, compresi IA64 Itanium architecture e Alpha. Tuttavia, la specifica copre competentemente solo Intel 32 Bit (IA-32). Intel ha incluso PXE

**Capitolo 4 – Caso particolare di utilizzo (IL LETTORE MULTIMEDIALE)**

nella Extensible Firmware Interface per IA-64, creando uno standard de facto con la sua implementazione.

Il client PXE si identifica come tale verso il server DHCP, inserendo nel pacchetto DHCPDISCOVER, il parametro Class Id con un valore della forma

*"PXEClient:Arch:xxxxx:UNDI:yyyzzz"*

Se è disponibile a ottenere i parametri di boot direttamente dal server DHCP, richiede i seguenti parametri:

Tag	Name	Length	Meaning	Reference
---	----	-----	-----	-----
66	Server-Name	N	TFTP Server Name	[RFC2132]
67	Bootfile-Name	N	Boot File Name	[RFC2132]

Il server DHCP decide se inviare i parametri necessari al boot da rete, e quali parametri inviare, sulla base dei dati presenti nella richiesta DHCP. Ad esempio, può essere necessario fornire a calcolatori differenti diversi boot file.

In particolare il formato dell'opzione Class Id è costruito per codificare in modo standard i dati necessari ad identificare un client.

Il protocollo PXE<sup>[25]</sup> permette inoltre al server DHCP di inviare al client gli elementi per offrire all'utente un menù con diverse opzioni di boot. Questo menù viene implementato dal firmware PXE, e permette di proseguire nel processo di boot con diversi server o file a seconda delle scelte dell'utente.

Dopo aver identificato il boot server da usare ed il nome del file da scaricare, il firmware PXE scaricherà questo file via TFTP, memorizzandolo nella propria RAM, ne verificherà la correttezza e lo eseguirà. A questo punto il compito del PXE in senso stretto è finito, ed è il NBP a prendere il controllo del calcolatore.

Il protocollo PXE<sup>[25]</sup> prevede la possibilità di far scaricare il NBP in modalità multicast, in modo da ridurre il carico sulla



rete e sul server nel caso molti client tentino di avviarsi contemporaneamente.

In pratica spesso l'NBP è in effetti un boot loader, ovvero un programma che a sua volta scarica via TFTP il kernel del sistema operativo e possibilmente un "initial ramdisk", ovvero l'immagine di un filesystem che viene copiata in memoria RAM (detta comunemente initrd).

Per quanto riguarda la sicurezza PXE eredita tutte le debolezze di DHCP, ma ne amplifica la portata, in quanto al protocollo viene demandata non solo l'assegnazione di un indirizzo IP, ma anche la selezione di un sistema operativo da eseguire, che avrà il controllo completo dell'hardware.

Se non è garantita la sicurezza fisica del computer, e quindi è possibile che estranei lo colleghino a reti diverse da quella prevista, lasciare abilitata l'opzione di boot via PXE è equivalente a permettere l'avvio da dispositivi rimovibili: avendo accesso alla console e la possibilità di riavviare il computer un estraneo potrà avviare un sistema operativo di propria scelta, con il quale tra l'altro avrà libero accesso a tutti i dati memorizzati sulla memoria di massa del computer.

Inoltre, visto che è possibile che su una rete venga installato un server DHCP malevolo, tale server può anche offrire ad eventuali client PXE un sistema operativo malevolo. Per questo motivo, su un computer che dovrebbe eseguire un sistema operativo dalla propria memoria di massa, l'opzione di boot via PXE dovrebbe essere disabilitata.

Un client PXE dovrebbe assicurarsi che l'immagine che avvia venga da una sorgente di fiducia. Il protocollo prevede funzionalità in questo senso.

Se il NBP contiene dati sensibili, come ad esempio password o chiavi crittografiche per accedere ad altri servizi, è possibile che un estraneo collegato abusivamente alla rete ne ottenga

una copia, pertanto bisogna assicurarsi che tale file venga inviato solo a client di fiducia.

D'altro canto se ben configurato e ben protetto riduce i costi di sicurezza per i vari componenti della rete (nel nostro caso di chioschi), in quanto prevede che per avere sicuri tutti i dispositivi (chioschi) della rete, basta che si focalizzi l'attenzione solo sul PC con DHCP Server.

### 3.3.3. Panoramica su Remote Boot Manager e Configurazione DHCP

Prima di addentrarci nella spiegazione di quello che è il client TFTP del sistema, ovvero Remote Boot Manager, passiamo ad analizzare come va configurato il server DHCP del nostro sistema.

Nel DHCP (nel caso oggetto del progetto Windows 2003 Server Enterprise) vanno inserite tramite la shell di comandi le seguenti impostazioni:<sup>[26]</sup>

```
c:|>netsh
netsh>dhcp
netsh dhcp>server ||server_name
netsh dhcp server>add optiondef 60 ClassID String 0
netsh dhcp server>set optionvalue 60 STRING PXEClient
```

Con queste impostazioni, il sistema abilita la porta 60 alla gestione del client PXE dando successivamente la possibilità al Remote Boot Manager di poter far effettuare il download dell'immagine creata e questo lo si può notare nell'immagine 16.

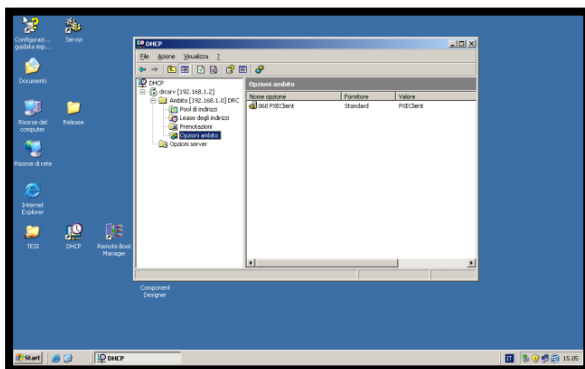


Figura 1647 – Configurazione porta 60 su DHCP

Remote Boot Manager, appartiene al gruppo di programmi utili per Microsoft Windows Embedded Studio e permette la configurazione di un'immagine e dei client ai quali tale immagine è destinata. Questo non è l'unico software disponibile per l'operazione; altri con le medesime funzionalità sono sviluppati da diverse case di software. La scelta è caduta su questo software in quanto è inglobato nel Windows Embedded Studio, riducendo eventuali errori e tempi dovuti a far collimare software di differenti software house.

Ora che però sono stati analizzati quelli che sono gli elementi essenziali di tale configurazione passeremo ad analizzare il processo completo con cui può essere effettuato tale sistema, tralasciando le specifiche di realizzazione del processo di cui abbiamo parlato nel caso precedente, ma indicheremo solo quelle che sono le differenze tra i due processi di costruzione dell'immagine.

### 3.3.4. Come realizzare il sistema

In questa fase verranno tralasciate quelle informazioni che o sono state già date nei paragrafi o capitoli precedenti e sarà rivolta l'attenzione alle fasi che caratterizzano questo caso particolare, menzionando e spiegando magari quei componenti che sono utili per la realizzazione del progetto.

Effettuata la fase di configurazione del DHCP si può passare alla creazione dell'immagine che passa attraverso le seguenti fasi:

Il componente principale attraverso il quale passa la

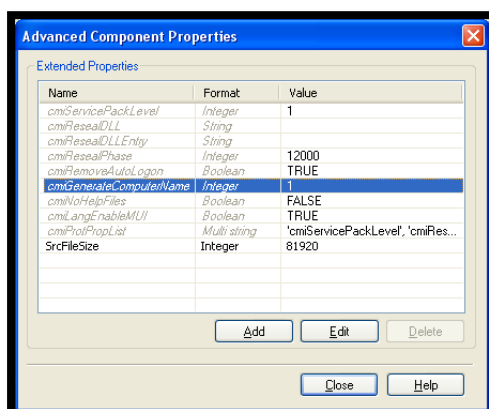


Figura 1718 – Configurazione componente System Cloning Tool

possibilità di far scaricare l'immagine ai client è il:

*System Cloning Tool*

Che permette l'assegnazione ad ogni computer di un codice SID (combinazione alfanumerica) durante l'operazione di FBA inserito al posto del nome del PC client. (ovviamente ogni computer su cui viene clonata l'immagine creata deve avere le stesse caratteristiche hardware del PC su cui è stata creata l'immagine). Tale componente va aggiunto all'immagine creata nel precedente capitolo 3.2. Per abilitare la funzione di rinominazione, il sistema prevede la disabilitazione di un parametro di tale componente, ossia del:

*cmiGenerateComputerName*

impostando tale parametro a 0 anziché a 1, come visualizzato nella figura 18. Questa opzione va configurata affinché la stessa immagine creata possa essere impostata per più chioschi con hardware identico, senza dover ripetere le operazioni effettuate fino a qui per ogni client.

Oltre a questo, un altro utile componente per poter effettuare la giusta configurazione per il progetto, è il:

*Windows Ram Disk Driver*

che abilita il boot remoto in RAM sul client.

All'interno del Target Designer, il Windows Ram Disk Driver, si può trovare nei componenti in

*Hardware → Other*

mentre il System Cloning Tool invece lo si può cercare in:

*Software → System → System services → Base*

Aggiunto il componente, va effettuato il check delle dipendenze e il salvataggio dell'immagine che deve essere copiata sul disco che, ovviamente nel nostro caso, va messo nella macchina chiosco per il FBA e per la conseguente decompressione dell'immagine.

Durante il First Boot Agent ad un certo punto grazie al componente aggiunto nel primo avvio del sistema, ovvero la

procedura Fbreseal, nell'ultimo step dell'FBA verrà visualizzato il messaggio in figura 19.

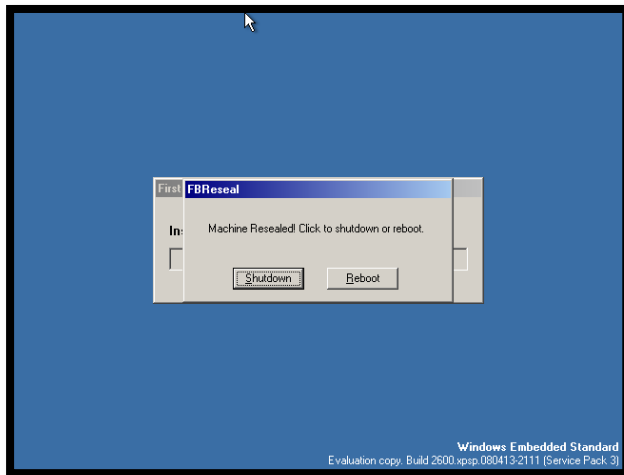


Figura 1849 – Reboot dopo FBA nella configurazione dell'immagine per chiosco senza hard disk

A questo punto riavviando il sistema, si può configurare l'immagine creata in modo che parta con una risoluzione piuttosto che un'altra o magari che avvii all'inizio un determinato programma, o ancora, che permetta la configurazione di IIS, come nel nostro caso, cosicché quando il sistema verrà avviato nel client, il web server sarà già pronto e funzionante e quindi l'utente non dovrà provvedere ad effettuare tali operazioni.

La fase successiva consiste nella creazione di un'immagine (che consiste in un file) che una volta presente nel server, permetterà l'avvio dei client (chioschi). La sua configurazione consiste nel riposizionare il disco in precedenza montato sul chiosco, nel PC di sviluppo (impostandolo come secondario). Questa operazione permetterà la copia dei file (non più compresso – dopo l'operazione di FBA) presenti all'interno del disco aggiunto, in un drive virtuale di cui ora descriveremo la creazione.

La generazione del disco virtuale viene effettuata tramite il programma SDI Loader, che come altri programmi è presente in Windows Embedded Studio 2009.

**Capitolo 4 – Caso particolare di utilizzo (IL LETTORE MULTIMEDIALE)**

Come si può vedere in figura 20, tramite l'opzione Add Disk si può creare un nuovo disco o aprirne uno creato in precedenza, magari per modificarlo.

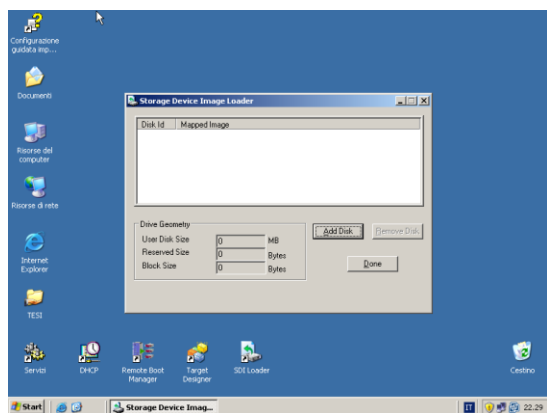


Figura 1920 – SDI Loader per la creazione del disco virtuale

Quando si genera un nuovo disco, innanzitutto bisogna specificarne la dimensione. A tal fine risulta utile vedere qual è l'attuale grandezza dell'immagine scompattata e presente sul disco aggiunto come secondario al PC di sviluppo. E' consigliabile poi incrementare tale valore per evitare problemi di spazio.

Il disco virtuale non è ancora visibile all'interno del sistema operativo. Per renderlo tale bisogna assegnargli la partizione e formattarla tramite:

*Pannello di controllo → Strumenti di amministrazione → Gestione computer → Gestione disco*

come visualizzato in figura 21.

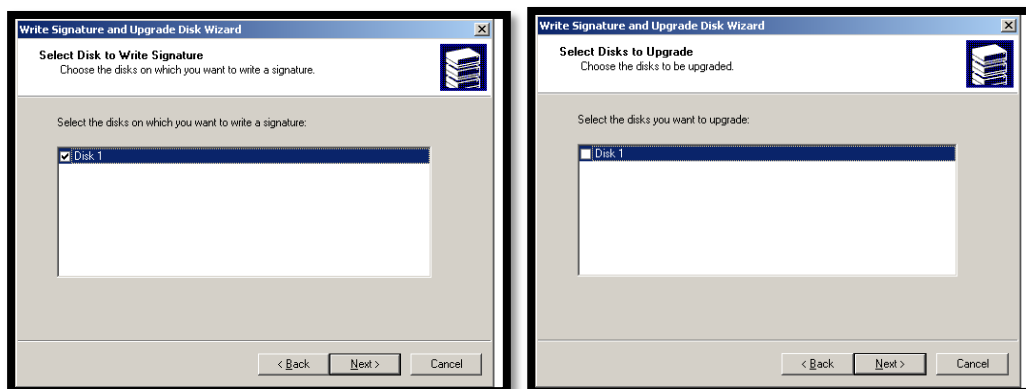


Figura 2021 – Inizializzazione del disco virtuale

**Capitolo 4 – Caso particolare di utilizzo (IL LETTORE MULTIMEDIALE)**

Una volta inizializzato il disco, bisogna formattarlo tramite la selezione e il Click con il tasto destro del mouse e con la scelta delle rispettive opzioni:

*Nuova Partizione*

*Partizione primaria*

Altri parametri da impostare sono:

- *Dimensione della partizione*
- *Tipo file sistem: in questo caso NTFS*
- *Lettera con cui identificare l'unità (utile in futuro per la creazione dell'immagine).*

Fatto questo, tutti i file presenti nel disco di appoggio, impostato come secondario nel PC di sviluppo, vengono copiati all'interno del drive virtuale appena creato. Al termine della copia dei file sarà necessario andare nella shell dei comandi e digitare le seguenti righe:

```
cd "C:\Programmi\Windows Embedded Utilities"
```

```
cscript SDIMgr.wsf /new C:\ramdisk.sdi
```

```
cscript SDIMgr.wsf c:\ramdisk.sdi /readpart:(Lettera part.):
```

Le due righe in che iniziano per cscript, servono rispettivamente a creare un file di nome "ramdisk.sdi" e a copiare il contenuto del disco virtuale identificato da "Lettera part.". Al termine dell'operazione il file pronto per l'uso sarà presente su C:\.

Durante questa fase è importante (onde evitare errori nella creazione dell'immagine) che tutte le finestre presenti, ad eccezione di quella della shell dei comandi (figura 22), siano chiuse.

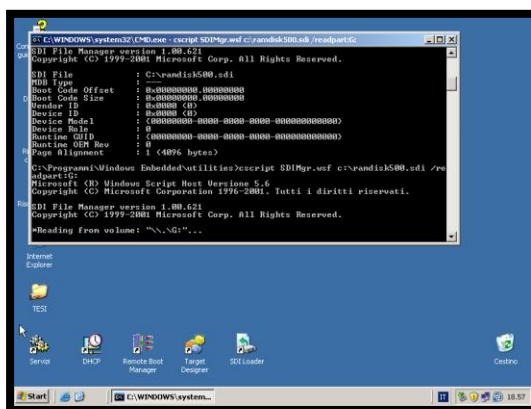


Figura 2122 – Creazione immagine ramdisk.sdi

Al termine di questa fase è necessario copiare il file C:\ramdis.sdi da questa posizione in:

*C:\Programmi\Windows Embedded\Remote Boot Manager\Download*  
è infatti in questa cartella che il Remote Boot Manager vuole che siano posizionati i files immagine che il programma deve gestire.

L'ultima fase nell'escursus delle operazioni da effettuare per la configurazione del chiosco è quella della configurazione del Remote Boot Manager che illustrerò qui di seguito.

Innanzitutto è possibile trovare tale software tra gli altri strumenti ovvero in:

*Start → Windows Embedded studio → Remote Boot Manager*  
che possiamo notare nella figura 23.

E' necessario d'apprima procurarsi il MAC address composto da 12 caratteri alfanumerici, relativo alla scheda di rete del PC remoto su cui deve essere installato il chiosco e solo successivamente, aggiungere una nuova riga sul software in questione.

La nuova riga dovrà contenere i valori:

- MAC address
- Boot Server → che indica il numero IP del server su cui è posizionato il PXE per lo scaricamento del file. Nel caso in cui (come il nostro) PXE e Remote Boot Manager risiedono sullo stesso server questo valore può essere impostato anche a 0.0.0.0.
- Boot program → Questa opzione prevede diversi valori che qui indicheremo nel dettaglio:
  - o Startrom.com → Abilita il boot remoto, e attende che l'utente abbia premuto F12 per l'avvio di tale servizio sul pc client.
  - o Startrom.n12 → Abilita il boot remoto, ma non attende che venga premuto nessun tasto per



**Capitolo 4 – Caso particolare di utilizzo (IL LETTORE MULTIMEDIALE)**

inizializzare il servizio, questo è quello che è stato utilizzato nel progetto.

- Reboot.com → E' usato per scopi di collaudo, e causa il riavvio del pc client.
- Abortpxe.com → Annulla il boot del computer client. Il sistema procede con il boot del prossimo device.
- Boot Image → contiene il nome dell'immagine copiata nella cartella download in precedenza
- Il resto dei valori vanno configurati come in figura, Action con boot, Boot Parameters vuoto e Description con una descrizione magari del client ricevente il file di bootstrpat (come possiamo notare in figura 23).

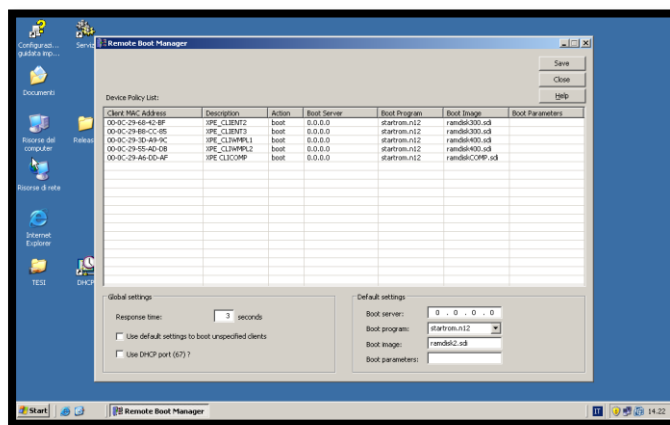


Figura 2223 – Remote Boot Manager

Per quanto riguarda il resto dei valori, vanno tutti deselezionati come in figura, infatti qualora avessimo utilizzato la porta 67 nel DHCP server, allora quel flag sarebbe stato spuntato, ma nella premessa a questo paragrafo, abbiamo detto che era necessario configurare il tutto con delle righe da shell di comando e che in questo caso hanno abilitato la porta 60 alla gestione del PXE, mentre il valore "Use Default setting to boot unspecified client" anch'esso va utilizzato ma quando si usa anche in questo caso la porta 67.

Terminata questa fase per rendere effettive le modifiche all'interno del Remote Boot Manager, si procederà cliccando sul tasto Save e per sicurezza si riavvierà il servizio.

L'ultima fase per rendere effettivo il tutto prevede il settaggio del client che andrà effettuato nel seguente modo:

Come abbiamo già detto innanzitutto bisogna:

- settare da BIOS la funzionalità di avvio tramite PXE;
- riavviare il chiosco e effettuare un test;
- digitare magari F12 per poter far avviare il chiosco da PXE,

Nel giro di qualche secondo il sistema avvia una schermata in cui viene visualizzato Loading Ramdisk Image.

In questa fase va inoltre utilizzato il componente EWF NTLDR di cui abbiamo già parlato, per abilitare la scrittura in RAM dove viene caricato il sistema operativo XPE.

### **3.3.5. Considerazioni**

Analizzando questo caso, ciò che va messo in evidenza è che può essere usato laddove si richieda una certa sicurezza, e per questo è l'ideale nei casi come questo del chiosco, in cui chiunque, magari può a livello software effettuare, i più svariati attacchi al client, ma a nulla possono contro il fatto che il sistema parte da Remoto, è utile anche nei casi dove l'utente finale deve usare delle applicazioni che vanno sempre lanciate e che magari sono ripetitive. Potrebbe essere il caso di uffici pubblici, dove spesso sono utili poche semplici applicazioni, da utilizzare e senza che possano essere modificate. In entrambi i casi infatti, riavviando il sistema, riparte sempre allo stesso modo caricando la stessa immagine, rendendo vane eventuali modifiche allo stesso o attacchi esterni.

Il lato negativo di questo sistema sta nel fatto che l'avvio e la copia del sistema operativo in RAM, richiede tempo sconsigliato pertanto il sistema a quegli uffici in cui il

bootstrap deve essere assolutamente veloce. Il tempo per l'esecuzione del sistema è infatti di qualche minuto (4/5 circa, anche se dipende molto dal tipo di PC utilizzato come client e dalle prestazioni della rete).

Risulta invece una buona scelta per un sistema chiosco come quello analizzato, che viene avviato solo nei casi in cui si deve fare manutenzione e così viene limitato l'aspetto negativo e sfruttata al massimo la velocità di lavorare in RAM. Questo sistema inoltre risulta particolarmente utile nei casi in cui l'amministratore di un sistema deve ripristinarne un altro danneggiato. In questa situazione si può avviare il sistema in RAM (che non è attaccabile, ad esempio da virus), magari caricando all'interno opportuni software ad hoc per il ripristino. Si può così utilizzare il sistema attivo per riconfigurare quello danneggiato.



## Capitolo 4 – Caso particolare di utilizzo (IL LETTORE MULTIMEDIALE)

### 4.1. – Requisiti

Per analizzare questo caso particolare del lettore multimediale, si è presa in considerazione la funzionalità di Windows XP Embedded che potrebbe anche essere associata anche agli altri casi già analizzati in precedenza, ovvero quella di poter eseguire al posto della shell un software particolare. Nel nostro caso abbiamo ritenuto utile utilizzare Windows Media Player, in quanto già inglobato in XPE.

All'avvio del sistema operativo, pertanto, verrà eseguito il programma utilizzato senza che l'utente possa in alcun modo interagire con il sistema, se non per utilizzare il software in questione.

Per la realizzazione di questo caso particolare, non sono richiesti particolari hardware, se non quanto stabilito per i casi precedenti a cui esso può essere ricondotto. Questo caso infatti può essere analizzato sia con la presenza di un disco sul dispositivo (potrebbe essere anche un chiosco) che tramite PXE con l'immagine da scaricare da un server DHCP. (per questo rimandiamo la spiegazione a quanto analizzato nei capitoli precedenti). Qui ci limiteremo a spiegare le differenze che intercorrono con i casi precedenti soprattutto per il lancio di questa shell customizzata.

A livello software in questo caso, viene analizzata quindi la creazione di un nuovo componente custom shell e pertanto, unico requisito per la realizzazione sarà la presenza dei software necessari esistenti in Windows XP Embedded Studio, ossia, Component Designer (per la creazione di componenti personalizzati per Windows Embedded) e Component Database Manager (Per l'importazione di un componente personalizzato già creato in precedenza all'interno della lista di componenti

in Microsoft SQL Server (Server di database che viene utilizzato da XPE per gestire il sistema).

Passiamo ora ad analizzare la parte principale di questo “caso particolare” ovvero come realizzare un nuovo componente, che nel nostro caso è una shell personalizzata (custom shell).

#### **4.2. – Panoramica sulle Custom Shell<sup>[27]</sup> e sulla creazione di componenti<sup>[28]</sup>**

In alcuni casi può essere utile non solo customizzare il proprio sistema operativo, collegandolo in modo stretto al suo hardware, come fanno i sistemi Embedded come XPE, ma può risultare utile anche creare un'interfaccia customizzata, come nel caso di applicazioni RPOS in cui l'utente ha a che fare con una shell ad hoc per il sistema con cui interagisce.

Solitamente un file eseguibile per la creazione di una custom shell (un programma eseguibile) fa riferimento ad una serie di risorse, DLL (Dinamik Link Libraries), Registry Entries, o altro ancora, che potrebbero produrre delle dipendenze addizionali in Windows XPE.

Nella maggior parte dei casi le custom shell sono dei programmi personalizzati che spesso vengono forniti di un relativo programma di installazione, pertanto bisogna innanzitutto definire quelle che sono le risorse che esse utilizzano affinché funzionino correttamente. A tale scopo possono essere utilizzati software per limitare errori di dipendenza, come ad esempio Regmon<sup>[3]</sup> e Filemon<sup>[4]</sup>.

Prima di tutto ora passeremo ad analizzare come si crea un nuovo componente XPE.

Ci sono diversi modi per creare un componente, ad esempio importando i files .INF all'interno del component designer oppure creato dal file Devices.pmq creato con TAP.exe o TA.exe.

Sfortunatamente non ci sono applicazioni che possono automatizzare la costruzione di un componente Windows

Embedded, pertanto bisogna rimboccarsi le maniche ed iniziare.

La creazione di componenti stata ideata per dare maggiore flessibilità al sistema, soprattutto laddove un fornitore debba fornire le specifiche per il proprio dispositivo hardware.

Innanzitutto esistono due seri di tools, per costruire e per testare il componente personalizzato Windows Embedded. La maggior parte dei software è presente nel PC di sviluppo mentre alcuni sono presenti nel "target devices" per poter permettere il debug.

Il comportamento delle installazioni non cambia, ma il sistema operativo può monitorare il processo di installazione. Dal momento in cui le installazioni richiedono il riavvio del sistema, strumenti ideali di analisi sono diventati quelli che analizzano lo stato dei registri dopo tale riavvio e tali strumenti possono essere tipo InCtrl5<sup>[2]</sup> (ZDNet) oppure SysDiff (Microsoft). Questi files analizzano i file presenti prima e dopo l'installazione e ne permettono la stampa, così da sapere quali files devono essere inclusi nel componente Embedded. Questa operazione richiede comunque diverso tempo perché con queste applicazioni, noi notiamo solo le modifiche ma potrebbero esserci servizi o applicazioni già attivi nel sistema che potrebbero influire nella determinazione di questa lista, mettendo chi la analizza nella condizione di discernere ciò che ha a che fare con l'installazione da ciò che ne è estraneo.

Una lista completa dei tools è la seguente:

InCtrl5<sup>[2]</sup> → Usato per avere una istantanea del file system e dei registri, provvede a fare un elenco delle differenze (ZDNet).

Dumpbin<sup>[29]</sup> → Per analizzare le DLL usate in un'applicazione o in un'altra DLL (tools a linea di comando di Microsoft Visual Studio® .NET).

Depends<sup>[5]</sup> → Utilizzato in alternativa per determinare DLL o API usate in un'applicazione.

Regmon<sup>[3]</sup> → Per analizzare le modifiche del registro (NT Systems Internals).

Filemon<sup>[4]</sup> → Per analizzare le modifiche del file system (NT Systems Internals).

Questi programmi che abbiamo analizzato risultano utili per analizzare le dipendenze su un PC di sviluppo per la creazione del componente personalizzato associato ad un programma preesistente o magari da noi creato per un determinato scopo.

Di tutti i programmi a disposizione alcuni non risultano utili, in fase di debug, infatti può capitare che manchi una DLL o un oggetto COM non sia registrato. Per controllare questi problemi sono sufficienti Dumpbin<sup>[29]</sup> o Depends<sup>[5]</sup>, pertanto prima di creare l'immagine definitiva del sistema, possiamo provvedere a rimuovere programmi come Regmon<sup>[3]</sup> o Filemon<sup>[4]</sup> che non risultano più utili in questa fase.

Il primo step per la creazione di un componente è reperire le informazioni del programma di installazione.

Utilizzando il programma InCtrl5<sup>[2]</sup>, viene catturata un'immagine del PC su cui viene eseguito. Questa immagine è comprensiva sia dei files movimentati che delle chiavi di registro modificate in luogo dell'esecuzione di un determinato software. Il programma InCtrl5<sup>[2]</sup> analizza il sistema, prima del lancio del software collegato e ricontraola successivamente le stesse informazioni, permettendo così di mettere in evidenza ciò che viene modificato. Tutto questo avviene successivamente alla pressione del tasto GO della figura 24.



## Capitolo 4 – Caso particolare di utilizzo (IL LETTORE MULTIMEDIALE)

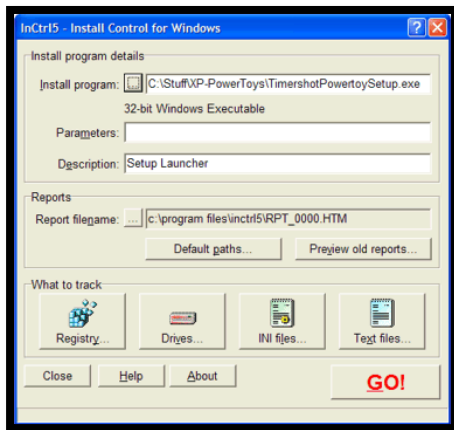


Figura 2324 – Immagine del programma InCtrl5

Possiamo così analizzare chiavi di registro, file di testo e non, file di inizializzazione, e molto altro; ovviamente quanto rilevato comprende non solo il nostro file di analisi ma anche eventuali altre applicazioni attive al momento in cui viene lanciato InCtrl5<sup>[2]</sup>. E' necessario tener conto di questo in fase di realizzazione del componente. Ogni programma inoltre può avere delle opzioni configurabili che possono leggere valori in chiavi di registro o in file .INI di configurazione.

Con Regmon<sup>[3]</sup> invece si analizzano le richieste di lettura e scrittura nel registro da parte del programma in analisi, come in figura 25.

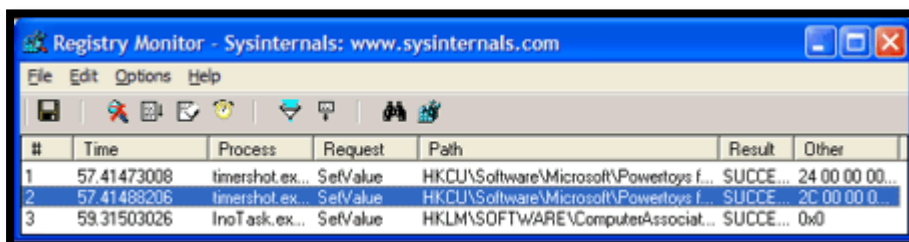


Figura 2425 – Immagine del programma Regmon

Ovviamente tutta questa parte va eseguita nel PC di sviluppo in cui magari si è creato il programma da aggiungere come componente nuovo.

L'ultima fase prima della creazione del componente è l'analisi delle dipendenze che avviene tramite Depends<sup>[5]</sup>, il quale permette di evidenziare non solo il programma che stiamo analizzando ma anche tutti i programmi o DLL ad esso

collegati. Questi file dovranno essere presenti in una cartella dell'immagine di windows Embedded (quindi in essa ci andranno copiati).

Se un componente non è presente o è già presente non lo si deve includere; questo viene determinato tramite la selezione del campo:

*Strumenti → Gestione Filtri*

Con cui digitando il nome del componente (Es. Ole32.dll) il programma lo aggiunge nell'immagine se il componente viene trovato, mentre nel caso contrario è consigliabile Controllare in:

*Strumenti → Opzioni → Advanced-si*

per vedere se è impostata l'opzione predefinita per i componenti ovvero

*visibilità > 1000*

che se impostata con il valore indicato o superiore, ci accorgeremo infatti che tale Ole32.dll non viene visualizzato perché ha una visibilità bassa, applicando pertanto una visibilità minore (Es. 100) noteremo che il filtro questa volta ci trova dei componenti.

Analizzati tutti i dati si può passare alla realizzazione del componente tramite il Component Designer (figura 26).

In questa fase ci possono essere oggetti COM o controlli Microsoft ActiveX® che richiedono prima dell'utilizzo la registrazione, così tramite Dumpbin/Export, analizzando l'esportazione effettuata, se in questa troviamo voci come:

- DllRegisterServer,
- DllUnregisterServer
- DllCanUnloadNow

si avrà sicuramente a che fare con un oggetto COM/ActiveX®. Per ovviare a questo problema durante l'operazione First Boot Agent, come si può vedere nella schermata (figura 26) ci sono diverse impostazioni che si può scegliere di abilitare/disabilitare e far eseguire durante tale fase. Una di

**Capitolo 4 – Caso particolare di utilizzo (IL LETTORE MULTIMEDIALE)**

queste è per l'appunto FBA DLL/COM Registration utile nel caso evidenziato dei controlli COM/ActiveX®.

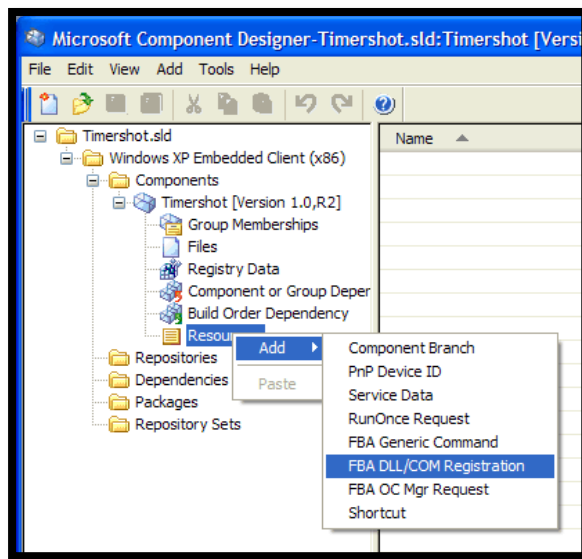


Figura 2526 – FBA Resource customization

La creazione di un componente a questo punto è simile alla creazione di un'immagine, in cui vanno inseriti file, dipendenze, ecc..., in base alle informazioni che abbiamo ottenuto (vedi figura 26).

Il sistema XPE ha scelto di far creare un componente all'utente per far sì che alla una volta ultimata la sua costruzione il processo di realizzazione dell'immagine XPE sia sempre lo stesso.

La fase di creazione di un nuovo componente risulta utile per capire come costruire una custom shell utile nella realizzazione del progetto chiosco o in questo caso del lettore multimediale, in quanto non si discosta molto dalla creazione di un componente. Qui di seguito andremo ad analizzare le operazioni che sono state effettuate per la realizzazione della custom shell per il lettore multimediale, tralasciando o sorvolando sulle fasi qui spiegate della realizzazione di un componente.

### 4.3. – Realizzazione del sistema

Per la realizzazione del sistema innanzitutto si è partiti dalle precedenti immagini create per la gestione del chiosco con bootstrap remoto, infatti in questa immagine erano già presenti tutti i componenti e le configurazioni per tale gestione. A questo abbiamo aggiunto le funzionalità proprie di questo caso particolare del gestore multimediale, e in particolar modo tutto quello che concerne la custom shell creata per Windows Media Player.

Nella realizzazione del sistema tralascieremo le fasi di configurazione dell'immagine, già ampiamente spiegate nei capitoli precedenti, e ci soffermeremo sulla parte principale che differenzia questa fase ossia la realizzazione di una shell ad hoc per il lettore multimediale, dove l'utente interagisce con il sistema, senza poter fare altro che quello che gli si permette di fare in base alle configurazioni, evitando operazioni dannose o che utenti "smanettoni" rendano instabile il sistema.

Andiamo ora ad analizzare questa fase.

Windows XP Embedded fornisce una serie di "Shell" standard come Command Shell, Explorer Shell, Task Manager Shell. Ciascuno di questi serbatoi è disponibile come componente nell'ambito di Target Designer, e può essere trovato sotto:

*Software → Sistema → Interfaccia utente → Shell.*

Ci sono passi coinvolti nella creazione di un componente della shell personalizzata ma in qualche modo possiamo derivare questa operazione da quella della creazione di un componente personalizzato.

Generalmente, il componente che conterrà l'eseguibile e le eventuali risorse, librerie di collegamento dinamico (DLL), voci di registro, e le risorse potrà anche avere dipendenze con componenti aggiuntivi di Windows XP Embedded. Nella maggior parte dei casi, una custom shell sarà un programma personalizzato con il proprio programma di installazione.

A questo punto, come evidenziato in precedenza, bisognerà determinare quali file e le voci del Registro di sistema sono necessarie per far funzionare correttamente l'applicazione.

Questo processo va effettuato prima di creare il componente della shell, usando Explorer Shell e la creazione di uno standalone di componenti per la vostra applicazione. È quindi possibile utilizzare strumenti come Regmon<sup>[3]</sup> e Filemon<sup>[4]</sup> per aiutarsi nel debug l'applicazione componente.

Dopo aver determinato il file di registro e componenti necessari per l'uso della shell, il processo di creazione della shell componente è abbastanza semplice.

Costruire una Custom Shell non differisce molto dalla costruzione di qualsiasi altro componente. Dopo la fase di reperimento dell'elenco di file richiesti, delle voci di registro e delle dipendenze, informazioni conosciute o per il fatto che si è analizzato il programma da eseguire come shell (nel nostro caso Windows Media Player) o per il fatto di essere noi i programmatori del software utilizzato come Custom Shell.

Nel caso del programma Windows Media Player utilizzato per la realizzazione del box multimediale, prevede i seguenti file associati:

- *ADVAPI32.DLL;*
- *KERNEL32.DLL;*
- *USER32.DLL;*
- *OLE32.DLL;*
- *SHELL32.DLL;*

Tali files devono assolutamente essere presenti nel progetto di installazione del box multimediale altrimenti questo non funzionerà.

In primo luogo, dobbiamo lanciare il Component Designer. Dal menu File, fare clic su Nuovo. Questo crea un nuovo componente di lavoro. Il componente di default sarà chiamato "Nuovo Documento 1.sld". Durante il processo di

creazione di componenti, è necessario aggiungere un repository e un componente al file SLD.

E' necessario espandere il componente in modo da poter vedere il nuovo documento 1.sld, Windows XP Embedded Client (x86) e le sottocartelle: Components, Repository, Dependencies, Packages, e Repository Sets. Il componente SLD è vuoto, perciò abbiamo bisogno di creare un nuovo componente selezionando la cartella Componenti e cliccando con il tasto destro del mouse Aggiungi componente, oppure utilizzando la voce di menù:

*Aggiungi → Componente*

Un nuovo componente è stato aggiunto al lavoro, chiamato "New Component 1", al quale saranno associati un certo numero di SubItems, come ad esempio:

- *Memberships Group,*
- *File,*
- *Registry Data,*
- *Component or Group dependency,*
- *Build order dependency,*
- *Resources.*

Selezionando la voce "New Component 1" e utilizzando la voce del Componente Proprietà (a destra), è possibile modificare il nome del componente in modo che corrisponda al nome del file eseguibile, o alla funzione che l'applicazione effettuerà. WMPL è quello da noi utilizzato per il progetto.

Ci sono un proprietà che possono essere impostate per il componente, che comprendono la posizione della guida del componente, i repository utilizzati, e il prototipo del componente. Windows XP Embedded possiede tra gli altri i seguenti tipi di prototipi:

- *Shell Prototype Component,*
- *Keyboard layout prototype,*
- *MUI prototype,*
- *Font prototype*
- *Selector Prototype.*

## Capitolo 4 – Caso particolare di utilizzo (IL LETTORE MULTIMEDIALE)

Il componente potrebbe essere già stato creato basandosi su una esportazione di Target Analyzer Probe (TAP), o Target Analyzer (TA), che in effetti, sono macro componenti che reperiscono le informazioni di un dispositivo in questo caso ci si limiterà a aggiungere o rimuovere alcune delle funzionalità dalla lista già presente.

Nel caso del chiosco multimediale, si desidera basare la componente relativa su:

*Shell Prototype Component.*

Pertanto è stata selezionata la voce:

*Prototype*

e cliccando su Browse si è passati a selezionare:

*Software → Systems → User Interface → Shell → Windows Shell*

Per poi arrivare sulla voce indicata (figura 27).

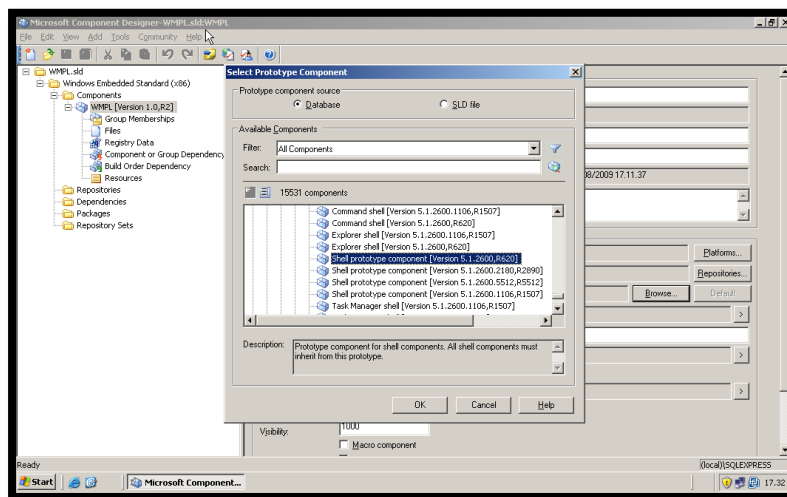


Figura 2627 – Parametri nuovo componente Custom Shell

Successivamente alla scelta è risultato necessario impostare una proprietà estesa per questa componente. Facendo clic sul pulsante Avanzate sottostante le proprietà. Verrà così visualizzata una finestra di dialogo "Advanced Properties". Qui si potranno aggiungere estensioni alle proprietà. Come nel caso del chiosco in cui abbiamo aggiunto la proprietà `cmiShellPath`.

Questa finestra di solito è utilizzata per impostare parametri opzionali per le informazioni di configurazione di un

componente, o per attivare una funzione condizionale. Ad esempio, nel caso di sviluppo di un componente di rete può essere utilizzata una proprietà estesa per memorizzare il Transmission Control Protocol / Internet Protocol (TCP / IP). La proprietà estesa è dotata dei seguenti valori:

- *nome,*
- *formato di dati*
- *valore dei dati.*

Il tipo di dati è determinato dal formato.

La finestra di dialogo "Extension Properties" contiene tre campi: Nome, Formato e valore. Formato voce può essere impostato con diversi tipi a scelta tra String, Int, Raw Binary. Nel caso del box multimediale abbiamo utilizzato i seguenti valori:

Formato: Strings.

Nome: cmiShellPath

Campo: "%16422%\Windows Media Player\wmplayer.exe"

Campo infatti deve puntare alla posizione del file eseguibile della shell customizzata, che nel nostro caso è alla posizione di Windows Media Player.

Windows Media Player è già un componente di XPE e pertanto è stata utilizzata la cartella "C:\Program file\..." ovvero la posizione dove viene installato il player dopo l'operazione di FBA. La posizione C:\Program files è indicata dalla variabile d'ambiente %16422%.

Ci sono diverse variabili d'ambiente nel sistema XPE, come %24% che si risolve in "C:\\" oppure %11% che si risolve in "C:\Windows\System32".

I valori %#%, sono elencati nel Component Designer, all'interno della voce Files nelle cartelle del componente che si sta creando dove, caricato un file, alla voce proprietà si potrà visualizzare la lista. Lo stesso elenco completo è disponibile anche sul sito microsoft<sup>[30]</sup>.



Ci sono due passi da seguire per la configurazione finale del componente custom shell. Per prima cosa bisogna inserire nella shell il gruppo di appartenenza e questo si ottiene selezionando la voce "Group Membership", cliccando sul tasto aggiungi, oppure selezionando il menù Aggiungi e la voce Group Membership, successivamente navigando tra le varie voci fino a raggiungere:

*Dipendences → Shell.*

La seconda e ultima cosa da fare è quella di aggiungere la categoria a cui sarà associato il componente e questo lo si fa cliccando sempre sul "Group Membership" ma questa volta cliccando su:

*Categories → Software → Systems → User Interface → Shells.*

Per rendere completa la creazione del componente, sarà in alcuni casi necessario aggiungere file al componente e creare e selezionare le opportune Repository.

L'operazione di creazione del componente personalizzato per la shell è stata completata. Al fine di utilizzare il componente, è necessario importare il componente di database utilizzando Component Database Manager, semplicemente cliccando sul tasto Import una volta aperto il programma.

La fase finale per la realizzazione del chiosco multimediale, si effettua lanciando il Target Designer e dopo aver rimosso il componente Explorer Shell dalla configurazione già creata seguendo i passi descritti nei precedenti capitoli, deve essere importato il componente appena creato selezionandolo dalla lista delle shell, ossia da:

*Software → Systems → User Interface → Shell*

e come viene fatto per qualsiasi altro componente, infine effettuare il check delle Dipendenze e l'esportazione dei file compressi per poter installare effettivamente il sistema, nell'operazione di FBA. In alcuni casi può essere utile prima di inserire il nuovo componente cancellare il componente

**Capitolo 4 – Caso particolare di utilizzo (IL LETTORE MULTIMEDIALE)**

Windows Logon, perché qualora non presente, viene inserito in automatico se impostata la relativa opzione all'interno delle impostazioni per il controllo delle dipendenze in:

*Tools → Options*

oppure viene richiesto quando si effettua il Dependency Check successivamente alla visualizzazione la dipendenza mancante. Cliccando in basso sopra alla descrizione della dipendenza mancante, si potrà selezionare la shell a noi utile, che in questo caso è WMPL (così abbiamo chiamato il componente). Un'immagine di come si presenta Windows Embedded con la custom shell per visualizzare Windows Media Player 11 la possiamo vedere nella figura 28.



**Figura 2728** – Custom Shell per la visualizzazione di Windows Media Player

C'è inoltre anche la possibilità di non visualizzare la schermata iniziale di Windows. Impostando all'interno di Settings alla destra della lista di componenti nel progetto che è stato creato la voce:

*"Do not display GUI boot screens"*

Tale opzione provvederà ad inserire nel Boot.ini il parametro /NOGUIBOOT il quale permetterà l'avvio del sistema operativo senza questa schermata che può risultare utile per minimizzare i tempi di avvio del sistema.

La custom shell è stata usata per far lavorare l'utente solo con il software a sua disposizione, in questo caso Windows Media Player.

#### 4.4. – Considerazioni

Le uniche considerazioni che possiamo fare in questa fase, sono relative all'utilizzazione delle Windows custom shell.

Come si è già detto esse risultano utili per far fare all'utente solo quello che si decide fargli fare, magari realizzando un Software ad hoc per quella postazione di lavoro.

Questo permette di migliorare così:

- il sistema di sicurezza, (meno software su un PC lo rende meno suscettibile ad attacchi dall'esterno, o da virus);
- i tempi di avvio del sistema (se non viene avviato da remoto, ci sono meno programmi attivi nel sistema e quindi essendo più leggero il sistema è più veloce a partire).

In base a queste considerazioni, non risulta consigliabile l'utilizzo delle custom shell se si devono eseguire diversi programmi sul PC in questione o se non si sa a priori quello che dovrà essere installato su quel PC. Spesso realizzare un software ad hoc non è utile in termini di tempo rispetto al vantaggio di non far eseguire all'utente qualche programma. Potrebbe essere consigliabile in quest'ultimo caso, realizzare un'immagine di Windows Embedded con qualche componente in meno.

Si è dimostrata molto utile invece nei casi analizzati dei chioschi (Informativo e Multimediale). in entrambi i casi infatti a priori, si sa quale sarà l'unico programma utilizzato:<sup>[31]</sup>

- Internet Explorer nel primo caso (con l'opzione "-k «sito»" per visualizzare Internet Explorer a tutto schermo ovvero in modalità kiosk)
- Windows Media Player nel secondo caso.

e inoltre essendo chioschi rivolti ad un largo pubblico, non va tralasciato il fatto di limitare al minimo le intrusioni di utenti "smanettoni".

## Capitolo 5 – Conclusioni

L'evoluzione di Windows XP Embedded, avvenuta dal 2000 ad oggi, ha prodotto un sistema funzionale, versatile e assai potente per la realizzazione di immagini, atto alla creazione di un sistema operativo personalizzato in base all'hardware di un determinato device. Lo sviluppo dei dispositivi Embedded e dei software che negli anni si sono susseguiti hanno fatto sì che il sistema Windows XP Embedded si adeguasse di conseguenza, fornendo agli utenti sempre nuovi software per meglio rispondere alle mutate esigenze che di anno in anno, modificano lo scenario informatico.

Nella realizzazione di componenti ad esempio fondamentale è stato l'ausilio di software come Dependency Walker (Depends)<sup>[5]</sup> o InCtrl5<sup>[2]</sup>, senza i quali oggi, avendo a che fare sempre più con programmi di installazione o con software che utilizzano in maniera massiccia Api o oggetti COM o ActiveX®, il lavoro sarebbe molto più lungo.

Per quanto riguarda la realizzazione di componenti, molto importante è stato l'inserimento nelle ultime versioni della possibilità di attivare oggetti COM o ActiveX®, in quanto la loro assenza limitava molto la generazione di nuovi componenti. La presenza di tali oggetti costituisce un punto di forza importantissimo del sistema XPE perché permette a quei fornitori che ancora non hanno inserito i loro driver tra i 12000 componenti del sistema, di poterlo fare ugualmente con tale sistema, evitando così che l'utente che deve realizzare un sistema Embedded si muova verso altri fornitori magari più costosi (che si possono permettere di fornire a Microsoft i loro driver).

Il lavoro effettuato nell'analisi di XPE ha portato a fare delle riflessioni. Innanzitutto su quello che è lo stato dei lavori rispetto alle esigenze del mercato. Se oggi il mercato cerca di offrire un Computer a basso costo, veloce a partire, con poche e semplici applicazioni, allora la risposta può essere un sistema come quello XPE ?

A seguito dell'analisi effettuata, la risposta può essere sicuramente affermativa.

Quali vantaggi offre questo sistema ?

**Bibliografia**

- Configurazione di un PC in base al proprio hardware, quindi tanti componenti presenti e superflui sui più blasonati sistemi come XP Pro, non vengono caricati, pertanto viene realizzato un sistema con un kernel minimo e quindi molto veloce.
- Kernel minimo produce un sistema che al minimo cambiamento hardware debba essere aggiornato per la sopravvenuta esigenza, quindi può risultare utile laddove pochi sono i cambiamenti e dove è richiesta velocità, soprattutto all'avvio, come avviene nel caso analizzato in cui abbiamo installato un chiosco in un PC con su un disco (Caso base - di partenza) in cui l'avvio è stato valutato attorno ai 20/30) secondi (Vedi Tabella delle prestazioni).

D'altro canto XPE non può ridursi a essere utilizzato solo in sistemi come NetBook, per i quali non è stato progettato, anche se come abbiamo visto può sicuramente portargli dei vantaggi in termini di prestazioni, ma può anche risultare utile in molti settori e aziende laddove magari non è richiesta velocità di avvio del sistema, come nei casi considerati di chiosco informativo e multimediale, ma che prevedono una certa sicurezza del sistema, magari perché utilizzato da utenti "smanettoni" che possono pregiudicarne il funzionamento. In questa situazione è utile il componente EWF, un componente che non era stato mai preso in considerazione prima di XPE e che è eccezionale, soprattutto laddove si vuole salvaguardare la parte di un sistema informatico più facilmente corruttibile ed usurabile nel tempo, ovvero l'hard disk.

L'utilizzo di EWF non elimina i problemi di sicurezza ma li centralizza all'interno dell'azienda, nei server aziendali dove vengono scaricate le immagini come nel nostro caso.

Il chiosco ha al suo interno tutti componenti hardware, "senza memoria", quindi, chiunque voglia attaccarlo può farlo ma al suo riavvio, questo ripartirà con la sua immagine "vergine". Ogni tentativo di attacco di virus o di malintenzionati pertanto avrebbe vita breve, poiché il sistema non scrive nulla su disco (che non c'è) ma in RAM, dove viene re-direzionato da EWF.

Nei moderni PC inoltre è importantissimo nell'utilizzo di queste tecnologie, e soprattutto lo è stato nel caso in studio, il sistema PXE (Preboot Execution Environment) analizzato per effettuare il bootstrap da

remoto, su di un server con su caricato un DHCP e un TFTP (nel nostro caso Remote Boot Manager), spesso utilizzato nei sistemi per ripristinare un sistema danneggiato ma che come visto, può risultare di particolare importanza come nel caso del chiosco, dove la presenza di un hard disk non è in questo caso richiesta.

Dallo studio del sistema è risultato un punto importante, soprattutto a livello aziendale, per quei sistemisti che possono considerare XPE un sistema per abbattere i costi aziendali, infatti questo non richiede hardware cospicui, in quanto è il sistema che si adatta all'hardware e non il contrario, ed è per questo che, come abbiamo detto, si adatta anche ai NetBook che per essere economici devono avere al loro interno hardware non proprio di "ultimo grido".

E' sicuramente evidente in XPE la versatilità delle custom shell, che pur non essendo facilmente configurabili, soprattutto se non siamo stati noi a costruire il software che va installato nel sistema Embedded e che quindi prevede collegamenti con altro software, ma che permettono di avere a disposizione un sistema ad hoc per l'utente, migliorando le condizioni di assistenza (l'utente non ha tutto Windows davanti ma solo la shell customizzata) o di interfacciamento utente/macchina (l'utente può fare solo quello che gli si vuol far fare). Anche questo si può valutare in termini economici. Un utente che deve imparare solo determinate cose sarà produttivo prima di uno che ha a disposizione tutto Windows, e magari di questo deve utilizzare solo dei programmi, come anche l'azienda che fa assistenza a un sistema con una custom shell spesso farà prima a spiegare all'utente cosa deve fare.

Analizzando le prestazioni e la consistenza delle immagini che vengono create in un sistema Embedded notiamo che (come si può vedere dalla tabella delle prestazioni) i sistemi Embedded sono di molto superiori. Anche se va a scapito delle funzionalità e dei programmi a disposizione dei sistemi XPE.

Nei casi analizzati si è considerato un sistema abbastanza completo, seppur minimale.

Tipo	Spazio Compresso	Spazio Non Compresso	Tempo Avvio	Tempo installa.
XPE Con Disco	240 Mb	379 Mb	25 secondi	4 minuti
XPE Con Disco no Virtual Machine	240 Mb	480 Mb	18 secondi	4 minuti
XPE senza Disco Net Boot + Internet Explorer con sito interno	240 Mb	480 Mb	5 minuti	5 minuti + 2 riavvio e configurazione
XPE senza Disco Net Boot + Win. Media Player	240 Mb	370 Mb	4 minuti	5 minuti + 2 riavvio e configurazione

Tabella delle prestazioni

Come si può notare, lo spazio occupato ci permette di restare, anche nei casi peggiori, al di sotto dei 512 Mb per ogni macchina client su cui viene caricato il sistema, magari con bootstrap da rete; pertanto questo aumenta il vantaggio di costo rispetto all'utilizzo di altri sistemi.

Una ulteriore considerazione che scaturisce dall'analisi di questa tabella, sta nel fatto, che seppur diverso tempo viene impiegato nella realizzazione di un'immagine ad hoc in base alle esigenze, l'utilizzo di un sistema XPE permette di avere dei vantaggi in termini di tempi di re-installazione qualora un sistema, debba essere ripristinato (caso di un sistema con disco integrato). Infatti come si può notare, i tempi sono di 5 max. 10 minuti nel peggiore dei casi analizzati. Questo fa sì che si possa ripristinare il sistema in tempi relativamente brevi. Questo nel caso di applicazioni particolarmente critiche e importanti è un fattore da non sottovalutare.



## Bibliografia

[9] Fabio Valenza, (Aprile 2009), "NetBook e sistemi operative: quale via percorrere?", *Total Linux*, **12**:20-24.

[8] Venture Development Corporation, "The Embedded Software Strategic", *Market Intelligence Program*, luglio 2004

### Siti WWW

[1] Microsoft Staff, How to buy Windows Embedded Standard, <http://msdn.microsoft.com/en-us/windowseembedded/standard/dd940301.aspx>

[2] Neil J. Rubenking, "InCtrl5 - Stay in control",  
<http://www.pcmag.com/article2/0,2817,25198,00.asp>

[3] Mark Russinovic e Brice Cogswell, "Regmon for Windows v7.04",  
<http://technet.microsoft.com/en-us/sysinternals/bb896652.aspx>, (Pubblicato: 1/11/2006)

[4] Mark Russinovic e Brice Cogswell, "Filemon for Windows v7.04",  
<http://technet.microsoft.com/en-us/sysinternals/bb896642.aspx>, (Pubblicato: 1/11/2006)

[5] Dependency walker Staff, "Dependency Walker 2.2", "<http://www.dependencywalker.com>, (ultima revisione: aprile 2006)

[6] Linux for Devices Staff, "Linux for devices - Hosted by Hostway", [www.linuxdevices.com](http://www.linuxdevices.com)

[7] Wikipedia, "Distribuzioni per il mercato Embedded",  
[http://it.wikipedia.org/wiki/Embedded\\_Linux](http://it.wikipedia.org/wiki/Embedded_Linux), (ultima revisione: 27/06/2009)

[10] Stephen Berard, "Using the Enhanced Write Filter (EWF) in Windows XP Embedded",  
[http://msdn.microsoft.com/en-us/library/ms838511\(WinEmbedded.5\).aspx](http://msdn.microsoft.com/en-us/library/ms838511(WinEmbedded.5).aspx), (ultima revisione: 01/2003)

[11] Microsoft staff, "EWF Manager Commands", [http://msdn.microsoft.com/en-us/library/ms940853\(WinEmbedded.5\).aspx](http://msdn.microsoft.com/en-us/library/ms940853(WinEmbedded.5).aspx), (ultima revisione: 27/03/2009)

[12] Microsoft staff, "ETPrep Commands", [http://msdn.microsoft.com/en-us/library/bb521523\(WinEmbedded.51\).aspx](http://msdn.microsoft.com/en-us/library/bb521523(WinEmbedded.51).aspx), (ultima revisione: 27/03/2009)

[13] Wikipedia, "Windows CE", [http://it.wikipedia.org/wiki/Windows\\_CE](http://it.wikipedia.org/wiki/Windows_CE), (ultima revisione: 21/08/2009)

[14] Emac inc. staff, "Windows CE",  
[http://www.emacinc.com/operating\\_systems/windows\\_ce.htm](http://www.emacinc.com/operating_systems/windows_ce.htm)

**Bibliografia**

[15] Ronald Dicklin, "Windows CE .NET Combines Real-Time Control, HMI and Networking | RTC Magazine", <http://www.rtcmagazine.com/articles/view/100190>, (ultima revisione: luglio 2003).

[16] Doug, "Reports evaluate Windows CE 5.0 performance and capabilities", <http://www.windowsfordevices.com/c/a/News/Reports-evaluate-Windows-CE-50-performance-and-capabilities/>, (ultimo accesso: 03/11/2004)

[17] Staff, "Adding real time to Windows Embedded", <http://www.windowsfordevices.com/c/a/Windows-For-Devices-Articles/Adding-Realtime-to-Windows-Embedded/>, (ultima revisione: 26/02/2004)

[18] Staff, "Security features in Windows CE", <http://www.windowsfordevices.com/c/a/Windows-For-Devices-Articles/Security-features-in-Windows-CE/>, (ultima revisione: 06/10/2003)

[19] Joe Morris, "Developer security features in XP embedded demoed", <http://channel9.msdn.com/posts/TheChannel9Team/Joe-Morris-Developer-security-features-in-XP-embedded-demoed/>, (ultimo accesso: settembre 2009)

[20] Mike Hall – Steve Maillet, "Extending the Windows CE .NET Platform Wizard" <http://msdn.microsoft.com/en-us/library/aa459160.aspx>, (ultimo accesso: giugno 2009)

[21] Msdn, "Windows Embedded Standard Videos", <http://msdn.microsoft.com/en-us/windowseembedded/standard/dd742236.aspx>

[22] Staff Microsoft, "Windows Embedded Download Center", <http://www.microsoft.com/windowseembedded/en-us/downloads/default.mspx>

[23] Msdn, "Windows Embedded CE Tools", <http://msdn.microsoft.com/en-us/windowseembedded/ce/dd402108.aspx>

[24] Msdn, "Supported processors", <http://msdn.microsoft.com/en-us/windowseembedded/ce/aa714536.aspx>

[25] Wikipedia, "Preboot Execution Environment", [http://it.wikipedia.org/wiki/Preboot\\_Execution\\_Environment](http://it.wikipedia.org/wiki/Preboot_Execution_Environment), (ultima revisione: 9/3/2009)

[26] Jskfan, "script the NETSH command", [http://www.experts-exchange.com/Networking/Misc/Q\\_24348980.html](http://www.experts-exchange.com/Networking/Misc/Q_24348980.html)

[26] Acronis Staff, "Running Acronis PXE Server On The Machine With DHCP Server", <http://kb.acronis.com/content/2292>, (ultimo accesso: maggio 2009)

[27] Mike Hall, "Creating a Windows XP Embedded Shell", [http://msdn.microsoft.com/en-us/library/ms838335\(WinEmbedded.5\).aspx](http://msdn.microsoft.com/en-us/library/ms838335(WinEmbedded.5).aspx), (ultima revisione: Agosto 2002)

[28] Mike Hall, "Creating Custom component for Windows XP Embedded", [http://msdn.microsoft.com/en-us/library/ms838332.aspx#customxpecomp\\_topic1](http://msdn.microsoft.com/en-us/library/ms838332.aspx#customxpecomp_topic1), (ultimo accesso: maggio 2009)

[29] Microsoft Staff, "Descrizione dell'utilità Dumpbin", <http://support.microsoft.com/default.aspx?scid=kb;en-us;154780>, (ultima revisione: 1/9/2005)

[30] Msdn, "Predefined System Environment Variables", [http://msdn.microsoft.com/en-us/library/ms933062\(WinEmbedded.5,printer\).aspx](http://msdn.microsoft.com/en-us/library/ms933062(WinEmbedded.5,printer).aspx), (ultima revisione: 11/09/2006)

[31] Microsoft Staff, "How to use Kiosk Mode in Microsoft Internet Explorer", <http://support.microsoft.com/default.aspx?scid=kb;en-us;154780>, (ultimo accesso: aprile 2009)