

Image Management with Windows® XP Embedded Image Difference Engine

By Sean D. Liming and John R. Malin
SJJ Embedded Micro Solutions

September, 2008

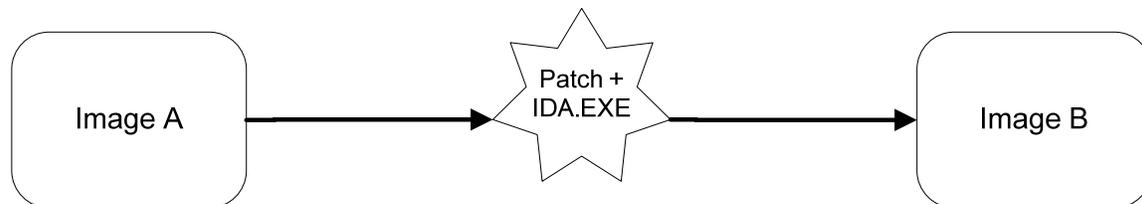
Field upgrades are an import design consideration for any deployed Windows® XP Embedded image. There are two types of upgrades: replace the whole image or create an incremental update patch. How, who, and what upgrades the system are questions that need to be addressed during development, since some update mechanisms, like Device Update Agent (DUA), need to have components built into the image. Replacing the whole OS image can be as simple as replacing the boot media, or as complex as a remote update. Patching can be as simple as updating a single file or registry key, or as complex as updating multiple features or a whole service pack. To address the complex patch, this paper walks through the steps to create a patch using the Windows XP Embedded Image Difference Engine.

In response to customers needing to upgrade systems from Service Pack 1 to Service Pack 2, Microsoft developed the Windows XP Embedded Image Difference Engine (IDIFF). Of course this solution isn't limited to service pack updates. IDIFF can be used for other incremental updates whether it is OS updates, application updates, or both. IDIFF is a solution that manages both image patch creation and application. IDIFF is downloadable from the Embedded Communications Extranet (ECE) website. *Please, see your Microsoft representative or distributor for information about accessing the ECE website.* The utility consists of two tools:

- IDIFF.EXE – Image Difference Engine, creates the patch using two OS images.
- IDA.EXE – Image Difference Applier, applies the patch created by IDIFF to the older of the two images used by IDIFF to create the patch.

1.1 Creating an Image Patch

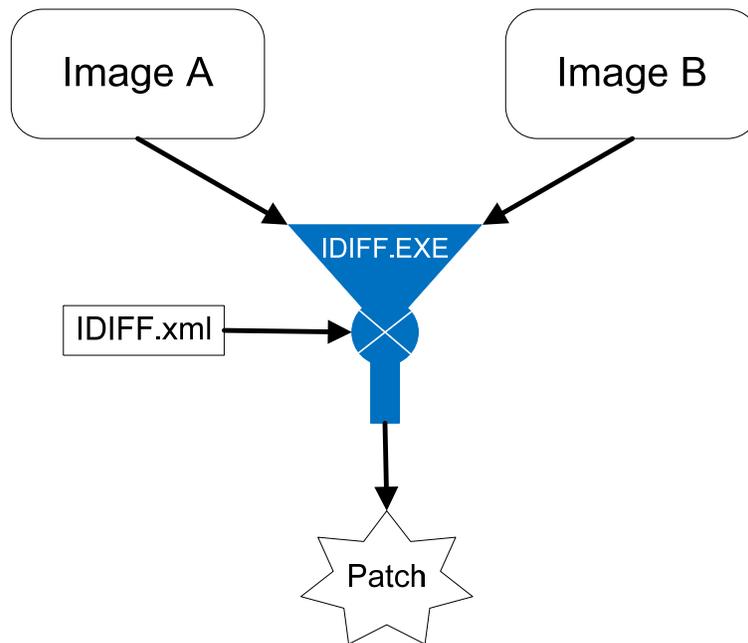
The basic concept is that you have an old image, A, that you want to make look like a new image B, which contains all the updates to the OS and applications.



To create the patch, both images need to be stored in separate folders on the development machine, and both must have completed FBA and cloning.

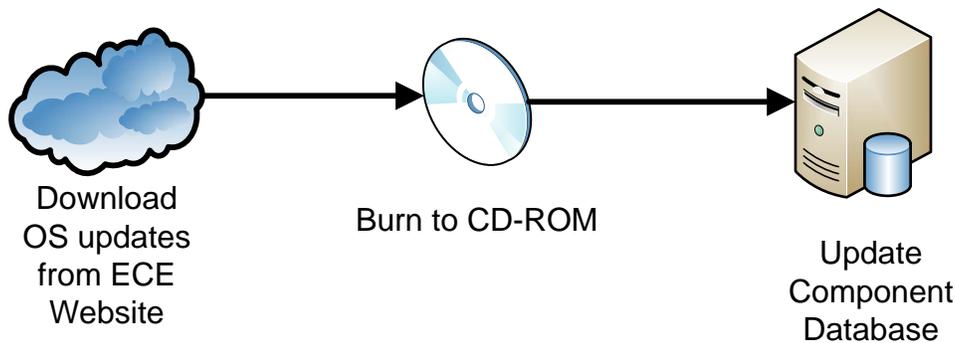
- When IDIFF runs, it compares the two images and generates a patch in another folder, the patch folder.
- When IDA compares the two images, if a file, folder, or registry key exists in both images but is newer in Image B, then it will replace the file, folder, or registry key in Image A. If there is a file, folder, or registry key in Image A that is not in Image B, then the file, folder, or registry key will be deleted. If a file, folder, or registry key exists in Image B that does not exist in Image A, then the file, folder, or registry key will be added as part of the patch to Image A.

If you need to preserve any data, file, folder, or registry key, from the old image A, then you need to modify the idiff.xml file and add the file, folder, or registry key to an exclusion list. The exclusion list tells IDIFF / IDA not to apply changes to the files, folders, or registry keys it finds in the list.



1.2 Applying Updates to the Database and Creating the Update Image

The first step is to create a new image that contains all the OS and/or custom application/driver updates. OS updates are downloaded from the ECE website in the form of .ISO or .IMG files that must be burned to CD-ROM. The OS updates are for updating the Component Database.



Once the component database has been updated, the next step is to build a new image with the updated components and test the new image on a target device. The image must NOT be resealed for creating the patch. Once the image has passed all the testing, then the next step is to create the patch.

1.3 Creating a Patch

You must have a copy of the original image that has shipped with the unit. Like the updated image, the original image cannot be resealed. It is a good practice to always save copies of images that have been shipped. Here are the steps to create a patch with IDIFF:

1. Copy the Image Difference Engine tools to the development machine. For this example, we will add them to the folder c:\IDIFF.
2. Create two folders under c:\IDIFF – c:\IDIFF\old_image and c:\IDIFF\new_image. You can name the folders anything you like.
3. Create another folder under c:\IDIFF\ create a folder called: IDA. Under c:\IDIFF\IDA create a folder called os_patch_<date>, where date is the date of the patch. (*Pick a date convention that is compatible with characters allowed for folder names.*)
4. Copy the old image to the c:\IDIFF\old_image folder
5. Copy the new updated image from the target to the c:\IDIFF\new_image folder.
6. Modify the sample IDIFF.XML file and add any files, folders, or registry keys to the exclusion / keep lists that you do not want modified. The instructions that come with IDIFF explain the different sections in the XML file.
7. Open a CMD window and run IDIFF.EXE to create the patch as follows (*all one line*):

```
IDIFF /Source:c:\IDIFF\old_image /Dest:c:\IDIFF\new_image  
/Output:c:\IDIFF\IDA\os_patch_<date> /BinaryCompare
```

The patch is generated to the c:\IDIFF\IDA\os_patch_<date> folder. An IDIFFIOUT.XML is generated as part of the patch. This XML file provides the directives for IDA.EXE. Everything in the c:\IDIFF\IDA folder can be copied to the system to be updated or to a server for downloading to the targets to be updated.

1.4 Applying the Patch

The patch can be applied to an system that is booted and running. You can also apply the patch when booted into another Windows OS such as WinPE or an embedded image that is booted from USB flash disk. You will need a mechanism to download the patch and enough space to store the patch. Here are some example steps to apply the patch:

1. Since the patch was built to the c:\IDIFF\IDA\os_patch_<date> directory, everything that is need to apply the patch is in the c:\IDIFF\IDA directory. For this example, copy the IDA folder and all its contents to a target running the old image. Copy the folder to the root of the old image – c:\IDA
2. Run the IDA.EXE to apply the patch:

```
Ida c:\ida\os_patch_<date>\idiffout.xml /dest:c:
```

3. Once the patch has been applied, reboot the system. The system should reboot with all the updates applied.

Testing the patch first is important. You may have to iterate applying the patch to locate all the information for the IDIFF.XML. Make sure that you clean up the os_patch_<date> directory, by deleting all of its contents, before rerunning IDIFF.

1.5 Other Notes

The IDA utility is very small in size so it is possible to build IDA into the image with very little impact to the final image size. Combined with DUA, you have a small patch management solution in the image. If EWF and FBWF are in the image, these filters would have to be disabled before applying the update.