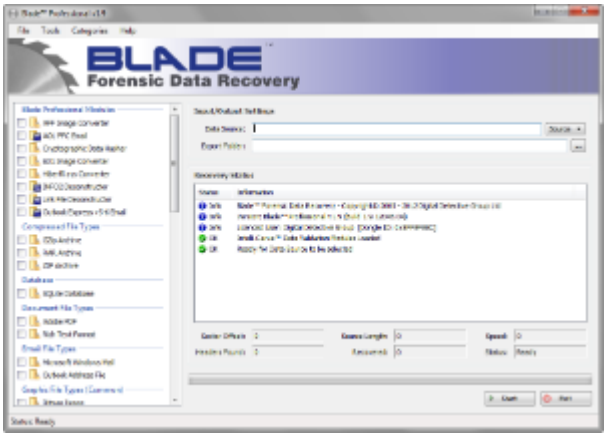
Blade v1.9



Introduction to the New Features of Blade v1.9

This release of Blade (Change Log v1.9) brings a number of fixes and some great new features. This is the first release of Blade to have evaluation capabilities which allow the user to test and evaluate our software for 30 days. When Blade is installed on a workstation for the first time (and a valid USB dongle licence is not inserted) the software will function in evaluation mode.

We have made some changes to the standard data recovery profiles, which provide additional capability through new configuration parameters for recovering data. This allows for more accurate recovery of data in certain scenarios, which are highlighted below. We have also added an option for setting a codepage, which enhances our multi-language support; this means that these strings can now be converted into a readable form using the same code page that was used by the source system when the data was originally saved to disk.

For this release, we have added two new Professional Recovery Modules:

- AFF® to dd image converter
- Hiberfil.sys converter

We have also modified some of the standard recovery profiles to make them more accurate, as well as adding new recovery profiles such as:

- SQLite database recovery

We have also been working on the data recovery engines to make them more efficient and much faster than before. The searching speed has been significantly increased. For a full list of all the changes in this release, please see: Change Log v1.9.

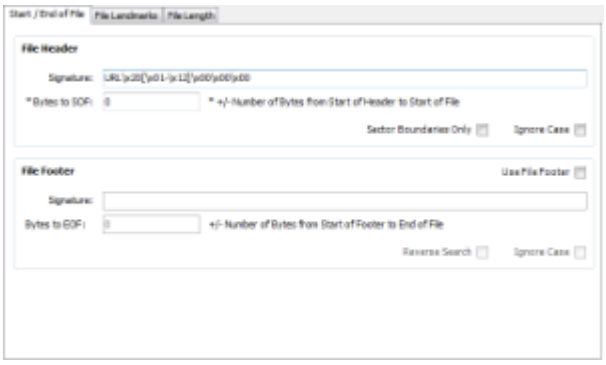
Evaluation Version

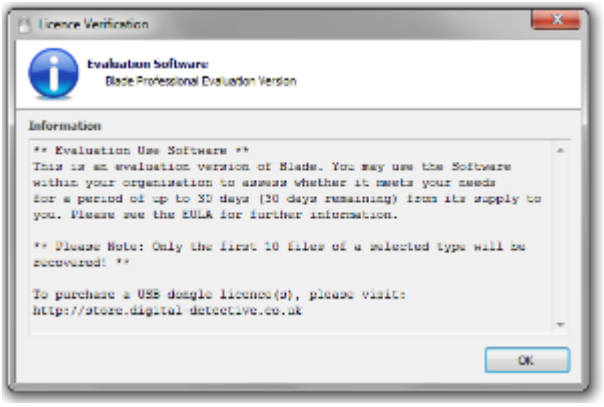
In this release of Blade, we have added the ability to work in demo or evaluation mode.

For a period of 30 days from installation, Blade will function with many of the features of the Professional version (with some limitations). During this time, Blade will be able to process any of the supported data sources, but will only recover the first 10 of any identified file type.

For a full list of the limitations, please see: Evaluation Version Limitations.

Recovery Profile Update

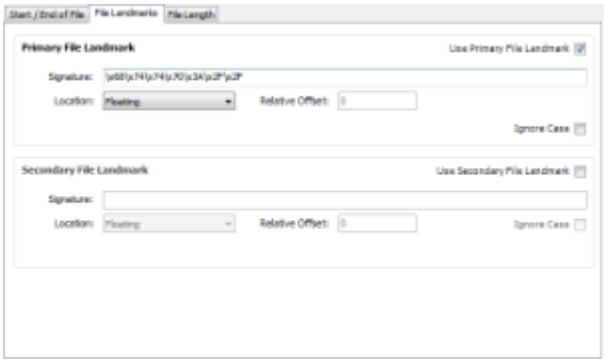




The recovery profile database has been updated and improved and now includes new fields to assist with accurate data recovery. The layout has also been changed to a tabbed format to make it easier to update and review on screen.

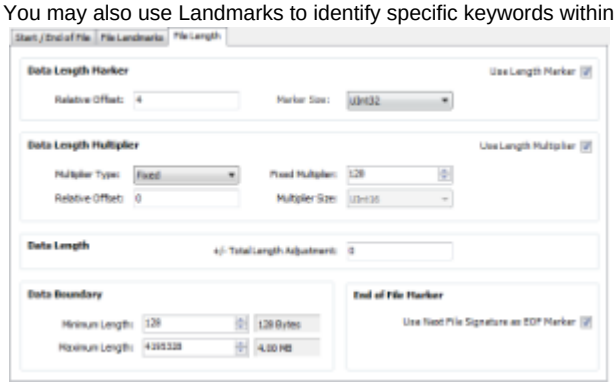
The first tab relates to the file (or record) header and footer. Having both parameters in the same location assists with creating recovery profiles quickly.

> **Note**
> It is worth remembering that specifying a footer may result in a limitation of accurate files recovered. For example, GIF images have a specific footer; however, as the footer is so small, there is a chance that byte sequence could naturally appear within the file and therefore truncate the output.



The second tab relates to the data landmarks within the file (or record) and their locations within the data. Data landmarks are specific bytes (or patterns of bytes) which have to appear within the body of the data. This is a powerful feature that assists in helping you create accurate data recovery profiles. The landmark(s) can be at a specific, static, byte offset from the start of the file, or at any location (floating) within the structure.

You may also use Landmarks to identify specific keywords within the file you are searching for.



The third tab relates to the file (or record) length information. We have added a new field which relates to a multiplier for the data length marker. This allows us some additional scope for data recovery where a length marker may relate to an object size and not necessarily a marker for the length of the entire object or file.

Some examples where this could be very useful is in the recovery of SQLite databases, or individual Microsoft Internet Explorer INDEX.DAT records. The examples shown above relate to Internet Explorer URL entries within an INDEX.DAT file. At byte offset 4 for each record, there is a record length marker which indicates how many blocks are required for the full record. Internet Explorer uses a 128 byte block

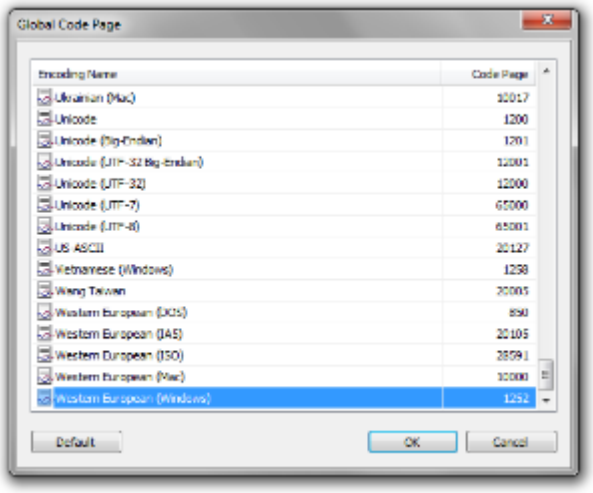
length. To establish the length of the record, you would take the length marker and multiply it by 128. Figure 3 shows the 'Data Length Marker' at offset 4, which is an unsigned 32bit integer. The "Data Length Multiplier" is a fixed length of 128 bytes. As "Use Length Multiplier" has been activated, Blade will read the length marker at offset 4 and multiply it by 128 thereby identifying the correct record length.

We will be making some further enhancements to the recovery profile options with the release of Blade v1.10 which will extend Blade's data recovery abilities.

Global Code Page

When Blade recovers a sequence of bytes that need to be interpreted as an ANSI string, the code page to use in the conversion can now be set as an option (Tools » Options » Change Code Page). This means that these strings can now be converted into a readable form using the same code page that was used by the source system when the data was originally saved to disk.

This is particularly important for modules where binary data records are found and deconstructed (such as with Microsoft Windows shortcut/lnk files or INFO2 records).



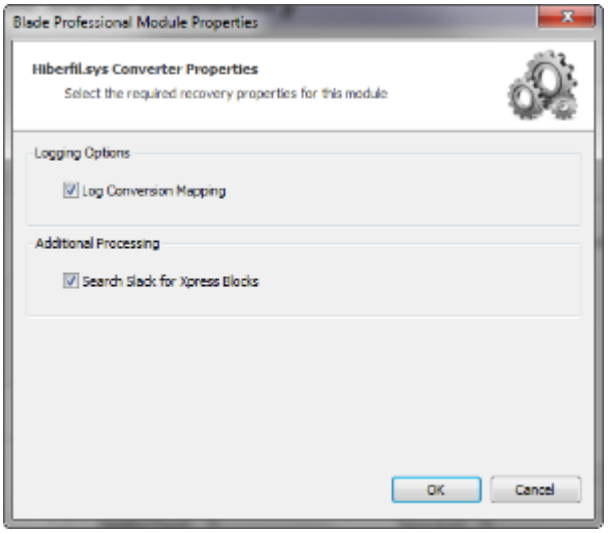
Hiberfil.sys Converter - Professional Module

This professional module processes a Windows hibernation (hiberfil.sys) file and converts it into a raw memory dump output file that can then be used for subsequent searching by Blade. A hibernation file contains blocks of data compressed using the Xpress Compression algorithm as documented on MSDN. The converter module decompresses these xpress blocks and writes out the pages of memory they contain, all assembled back into their correct page slots in the output file. The resulting output file is therefore an ordered dump of the pages of memory that were in use when the source computer entered hibernation.

The module supports hibernation files from Windows XP, Vista and 7, both 32-bit and 64-bit. It is able to intelligently work out the source operating system from the structure of the hibernation file. If it is an 'active' hibernation file (i.e. the hiberfil.sys file was captured while the source computer was in hibernation) it will still have its file header and information such as the system time when the hiberfil.sys was written is extracted. If the hibernation file is 'inactive' (i.e. the source computer had been successfully restored from hibernation when the file was captured) the file header is zero'd out, but the xpress block data still exists in the file and can be read and converted by the module.

To provide traceablilty back to the original data, the module has the option to log exactly how each xpress block in the original hiberfil.sys file has been mapped to the pages in the output file.

Depending on how many pages of memory were actually being used when the hibernation file was written, there may still be in the hiberfil.sys a large number of xpress blocks located in the file slack. These xpress blocks will have come from previous hibernations, they are not mapped as being currently in use and therefore cannot be written to the memory dump output file. The module provides the option to search for and decompress these 'slack' xpress blocks. They are written to a separate slack output file and for traceability their mapping is also logged.



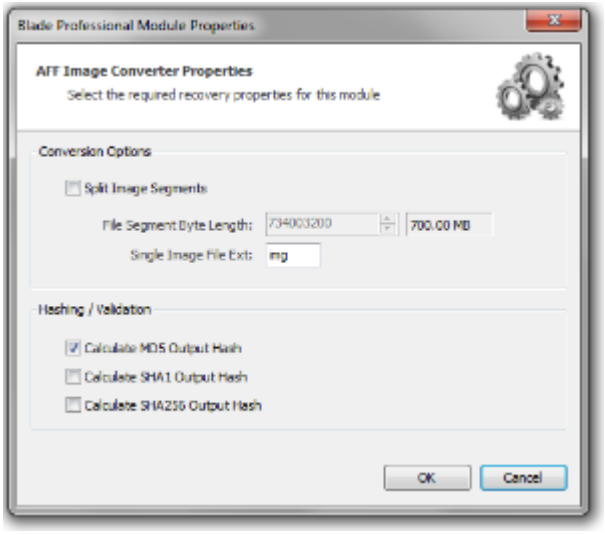
Advanced Forensics Format (AFF®) Support and AFF Image Converter

The Advanced Forensics Format (AFF®) is an extensible open format for the storage of disk images and related forensic metadata. It was developed by Simson Garfinkel and Basis Technology. Blade (and HstEx) now support the processing of AFF® image files (as well as other forensic formats). The following page lists the supported file formats: Forensic Image Formats Supported by Blade.

We have added a new Professional Module to allow you to convert AFF® images to a single (or multiple) segment binary image. As you can see from the Image Converter properties (accessible by right clicking on the 'AFF Image Converter' and selecting 'Module Properties'), there are options for splitting the image and creating a specific segment length as well as calculating MD5 and SHA1 hashes on the output data.

Lo

gic ub e For en sic Do ssi er ®



E01 Support

According to Logicube:

> *"The sixth generation of computer forensic solutions from Logicube, the Forensic Dossier® was designed and engineered exclusively to meet forensic investigators' requirements. Version 2.0.1 provides support for the E01 file format compression (hardware-based compression to maintain line-speed performance), and support for NTFS file format for support of 2TB and greater capacity hard drives and support of single, disk-wide dd image capture."*

With Blade v1.9, we have added support for the E01 files produced by the Logicube Forensic Dossier. Unfortunately, earlier versions of Blade are unable to load and read the E01 files generated by the Logicube Dossier because of an incompatibility with the metadata fields. Some of the values written to these fields are in a different format than those written by EnCase or FTK Imager. This has now been resolved.

Blade Options (Tools » ns)

As we have added a number of new options to Blade v1.9, we have created a new Options windows. This can be found by selecting Tools » (

Moving the options to this location makes it easier to configure the software, and assists in removing additional clutter from the main user inte

The following options have been removed from the main user interface window and moved to the Options window:

- Block Size (Sectors) - This is the size of the block Blade will read in with each cycle as it searches for data; the default is 512 Sectors
- Files per Folder - This is the number of files Blade will write out to any single folder during the recovery process.
- Log Type - This changes the type of log written during the recovery process. You may be asked to set the log type to 'debug' by our s

A new option added to Blade is to set the Code Page. This is used when converting byte streams into ANSI characters.

Added handling of OMA DRM Content Format for Discrete Media Profile (DCF) to ISO Base Media Format Validation (Intelli-Carve®)

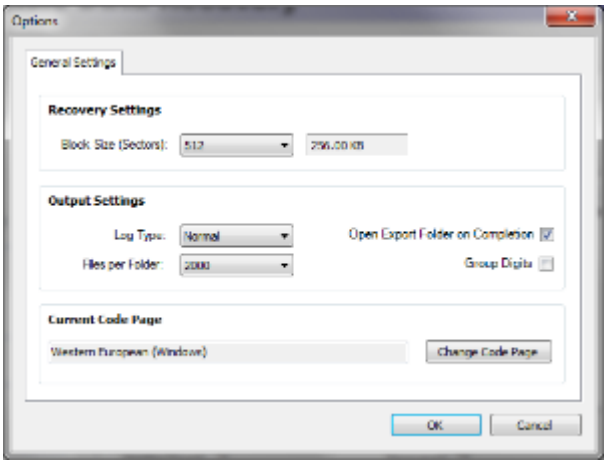
DRM content format (DCF) is used to package and protect discrete media, such as ring tones, applications, and images. The content in DRM content format is encrypted using a symmetric encryption key. The content is then placed as a single content object in the DCF internal structure and layout. The MIME type for objects conforming to the DRM content format is `application/vnd.oma.drm.content`.

Packetized DRM content format (PDCF) is used to protect continuous media, such as audio and video. Continuous media is protected as a

separate profile because it consists of data packets.

- Digital Rights Management

Blade now supports the recovery of OMA DRM Content Format for Discrete Media Profile (DCF) for ISO Base Media Format files. We have also enhanced the Intell-Carve® profile to make it more robust when handling potentially damaged and corrupt video files.



Other Changes to Blade v1.9 Explained

In addition to the changes outlined above, we have made some other changes which will be briefly explained as follows:

SQLite Database Recovery

With the addition of the length multiplier option in the recovery profile, we have added a recovery profile which utilises this feature. The main database file consists of one or more pages. The size of a page is a power of two between 512 and 65536 inclusive. All pages within the same database are the same size. The page size for a database file is determined by the 2-byte integer located in the header of the file. Also located in the header is a 4-byte integer recording the size of the database file in pages. The multiplier feature can read this value and work out the correct size of the database.

UInt8 Length Option

We have also added an option for identifying 1-byte length markers.

VALIDATION_ERROR Log Renamed

We have renamed the VALIDATION_ERROR log to DATA_VALIDATION_LOG for clarification purposes as some users were confusing the purpose of the log. The log is used by Intelli-Carve® profiles to record instances were identified data has failed recovery validation. Validation may fail for a number of reasons, such as non-contiguous clusters, or overwritten/corrupted data. The log identifies any instances where a file or record header has been found and the recovery process has been halted because the data failed the Intelli-Carve® validation. This is not an error log, but an audit of data validation.

Waveform Audio File Format Recovery

The WAV file recovery profile has been updated to include the WAV length marker. This will allow for more accurate recovery of WAV files.

Blade Slowing Down with the Recovery of Large Numbers of Files

When recovering large numbers of files, as the number increased, the searching speed would decrease. This was caused by the increasing time it would take to check for, and remove, duplicate offset entries from the recovery list. Duplicate entry removal is now completed at the end of the searching process. As a result, the searching phase is considerably faster than with Blade v1.8.

Byte Order Marker missing from TSV/CSV Export Files

The Byte Order Marker (BOM) is missing from TSV/CSV data exported in Blade v1.8. This has been added back for Blade v1.9 so that Unicode data can be correctly displayed by applications supporting Byte Order Marks.