Data Digester 1.6.15

User Manual

LEISENFELS

Green Software Solutions

Leisenfels UG (haftungsbeschränkt)

Am Langenkamp 11

49143 Bissendorf

Germany

www.leisenfels.com

THINK B4
YOU PRINT!

Bissendorf, 04/15/2015

User Manual

This user manual is part of the available documentation of the Data Digester software (Digester for short). It is delivered as a printable PDF document and also as help system to be used directly from the software. It contains pieces of information required for the use of the described software. You can find detailed pieces of information about the installation of the software in the installation manual.

Most sections of this manual contain information for all available editions of the software. Special instructions for the various editions are either highlighted directly in the text or by appropriate symbols (see the symbols below).

If you find an error or just want to give feedback or suggestions concerning this documentation, please send an e-mail to:

support@leisenfels.com

Leisenfels on the Internet

If you have questions or if you need additional assistance, please visit our Internet server www.leisenfels.com which is available around the clock. Current versions of the software and documentation are available for download here.

Used Symbols

⚠	Important instruction that should be followed
ⓘ	Additional instructions which can be helpful
BE	Special instruction for the Basic Edition
SE	Special instruction for the Standard Edition
EE	Special instruction for the Enterprise Edition

Table of Contents

Digester Main Window

The main window of the Digester software is divided into <u>main menu</u> (1), tool bar (2), <u>quick check</u> (3), <u>docking windows</u> (4) and <u>status bar</u> (5).
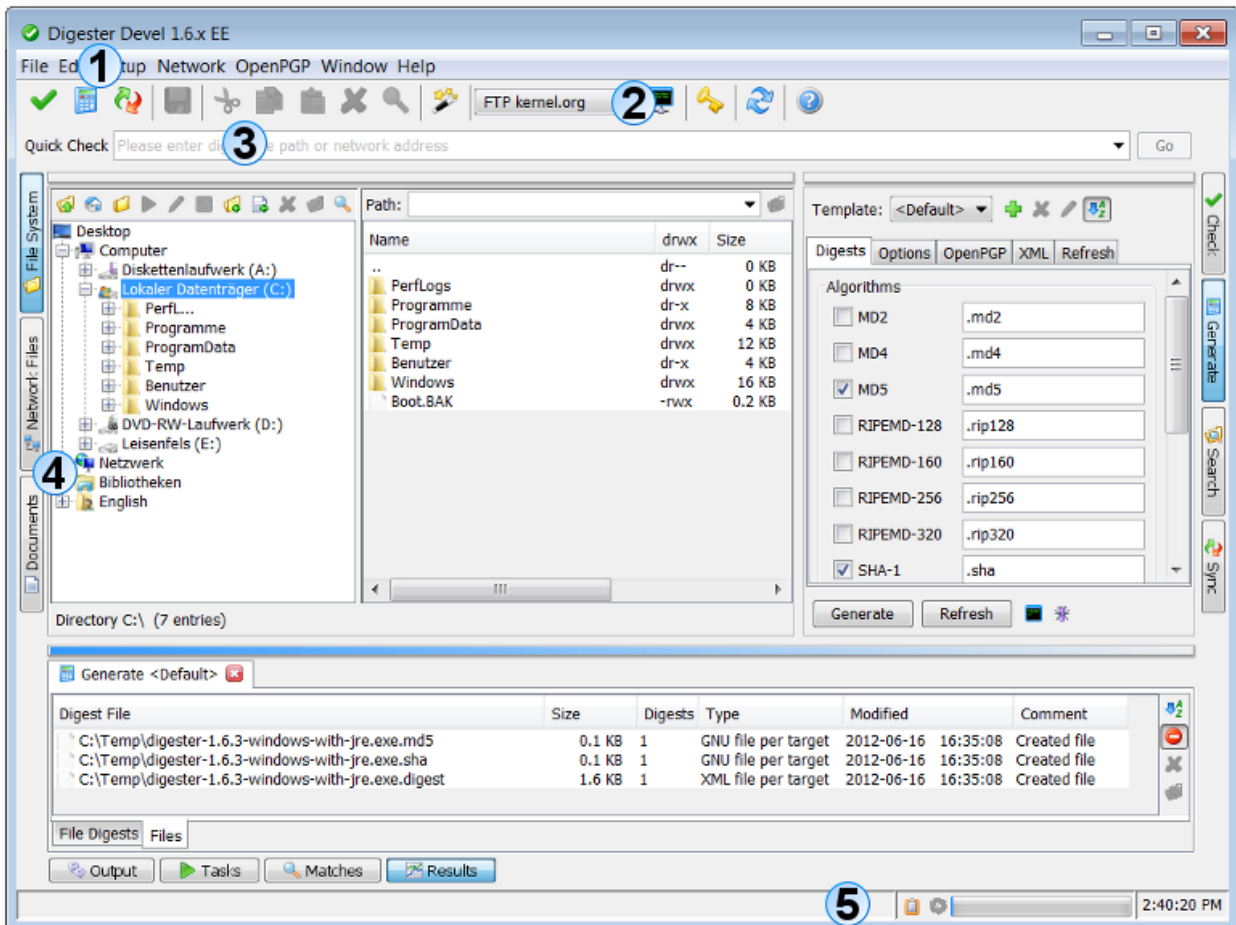
Main Menu

The main menu (a.k.a. the menu bar) you can find on top as usual. Here the keyboard <u>shortcuts</u> are implemented allowing the program functions to be activated more quickly. The main menu consists of the following areas which are explained in detail by the "Main Menu and Components" section:

- **File**: Open files in document editor, call the wizards
- **Edit**: Functions of the clipboard, delete and searching
- **Setup**: Language, look & feel, font size, and miscellaneous preferences
- **Net**work: Manage connections, open sessions
- **OpenPGP**: Create, import, and export OpenPGP keys
- **Window**: Docking windows, screen shots, general window functions
- **Help**: Help system, manual, release notes, status report, license registration

> Some of the mentioned functions are supported exclusively by the Standard Edition or the Enterprise Edition. Please visit our Internet website to read more about the functions of the available editions.

Tool Bar

The tool bar is located directly below the main menu and holds the most important symbols of the main menu for a faster access. Here you can find the wizard dialogs (**File** menu), the clipboard functions (**Edit** menu), the preferences (**Setup** menu), the network connection management (**Network** menu) as well as the key management (**OpenPGP** menu) and the help system (**Help** menu).

In addition to the symbols the tool bar can be configured the way, that the associated labels under the symbols are being displayed. You can find a detailed description of the tool bar functions in the section "Main Menu and Components".

Quick Check

The quick check input field can be used to specify a local file path on your PC like `C:\Temp\javainfo.jar`) or a network address like:

```
http://download.opensuse.org/distribution/12.1/iso/openSUSE-12.1-DVD-i586.iso.asc
```

You have the choice to enter a path for a digest or signature file or the path of an original file to be checked. By clicking the **Go** button the Digester software checks for existing digests/signatures for the given file. If this is the case then these files can be used to check the integrity of the referenced original files. The original files like bigger ISO images to burn DVDs may be downloaded and used after the checking has been completed successfully.

You can find detailed pieces of information to use the quick check functionality in section "Main Menu and Components".

Docking Windows

The docking windows are the central components of the Digester software. Here you will find all the important functions of checking, generating, and synchronizing to the special functions such as program messages (logs) and the checking results. The windows will each be represented by small icons that are integrated in the dock. Possible window positions are on top, left, right, and bottom. The docking windows are structurally designed for either horizontal or vertical use, a move from left to right or from bottom to top etc. is possible.

The docking windows can either be attached to the dock as shown above or can be displayed as separate windows. This is useful, for example, in order to deploy larger windows such as the sync

window in an optimal way. Currently, the following docking windows are delivered with the Digester software, which are described in section "Docking Windows" in detail:

- **File System Window**: Navigator for the local file system of the PC
- **Network File System Window**: Navigator for remote network file systems
- **Document Window**: Editors/viewers for texts, binary, and graphics files
- **Check Window**: Check digests and verify OpenPGP signatures
- **Generate Window**: Generate digests and OpenPGP signatures
- **Sync Window**: Synchronize files/directories with digests/signatures
- **Search Window**: Find files/directories based on their names
- **Output Window**: Program messages (logs)
- **Task Window**: Tasks which are or have been executed by the program
- **Search Results Window**: Matching files/directories
- **Results Window**: Other results like checking results

Status Bar

The status bar at the bottom of the main window is used by program to output important messages for the user. The area to the right contains other components such as to indicate the state of the clipboard (whether texts, files etc. have been copied) and to show the progress of ongoing tasks. Far right, the time of the PC clock is being displayed. Some components of the status bar can also be hidden, also the status bar can be hidden completely in order to save place in favor of other components.

For detailed information on how to configure the status bar see section "Main Menu and Components".

Additional Links

- Preferences

Data Digester Editions

In order to provide all customers with an appropriate and inexpensive program version the Data Digester software is currently published in three editions:

- Basic Edition (BE)
- Standard Edition (SE)
- Enterprise Edition (EE)

Basic Edition (BE)

The Data Digester Basic Edition (BE) is the smallest and cost-effective solution for all users searching for a single tool to check message digests and to verify OpenPGP signatures. The Basic Edition provides support for the commonly used message digest algorithms like MD5 and SHA as well as for OpenPGP signatures.

The integrated file system window allows for an easy and intuitive working with files and directories. Simply locate the archived or downloaded files on your PC, then double-click the message digest or OpenPGP signature file and the Data Digester starts working. After an instant, you are informed whether the referenced target files have been checked/verified successfully or if they were corrupted or manipulated and should not be used any further for security reasons.

The most important features of the Basic Edition are as follows:

- Check message digests like MD5 or SHA, verify OpenPGP signatures.
- Supports GNU, BSD, and the new Digester XML file format.
- Check files with multiple digests and signatures.
- Ideal solution to check Internet downloads and archived files.
- Runs on Windows, Mac OS X, Linux, and many more systems.
- Use the installers or execute directly with Java Web Start.
- Call the command-line interface from your own applications.
- Native support for Ant build scripts.

Please visit our website to read more about the features of the Data Digester Basic Edition.

Standard Edition (SE)

With the Data Digester Standard Edition (SE) you hold the perfect solution for message digests and OpenPGP signatures in your hands. The inituitive application window is easy to use, so you can generate message digests and OpenPGP signatures even for whole server file systems with only a few mouse clicks. If you are a system administrator, you may want to use the command-line interface digestercli to deploy the Digester software for your scripts and applications. The refresh mode helps to minimize the generation costs since only new or updated files are being processed.

The Data Digester Standard Edition has a built-in file system navigator where you can easily select single files or whole directories to be processed. Using the new DIgester XML file format it is possible to generate and store arbitrary message digests, intermediate digests, and OpenPGP signatures together in one file. The generation template support allows you to configure the application to generate the digests/signatures appropriate for different environments (e.g. FTP server, multimedia archive, server backups).

The most important features of the Data Digester Standard Edition are:

- Generate and check message digests like MD5 or SHA.
- Create and verify OpenPGP signatures.

- Supports GNU, BSD, and the new Digester XML file format.
- Create files with multiple digests and signatures.
- Process single files, directories, or whole file system trees efficiently and fast.
- Create digests/signatures for different environments with generate templates.
- Create, import, and export OpenPGP keys comfortably.
- Runs on Windows, Mac OS X, Linux, and many more systems.
- Use the installers or execute directly with Java Web Start.
- Call the command-line interface from your own applications.
- Native support for Ant build scripts.
- Supports all functions of the Data Digester Basic Edition.

Please visit our website to read more about the features of the Data Digester Standard Edition.

Enterprise Edition (EE)

The Data Digester Enterprise Edition (EE) is the flagship of the Digester product series. It includes all features of the Basic Edition and Standard Edition, and adds sophisticated new functions to work with network file systems connected via FTP/SFTP and to synchronize files and directories based on digests and signatures.

Message digests and OpenPGP signatures are not limited to integrity and authentication checking, but can also be deployed very efficiently for file synchronization. It's the ideal solution to transfer even large file collections from local to remote servers or vice versa while consuming only a minimum of time and bandwidth. By analyzing the available digests and signatures the Digester software ensures that only new or modified files are being transferred. So you don't have to take care about the edited files, let the Digester do the job by simply pressing the **Synchronize** button.

The following list shows the most important features of the Enterprise Edition at a glance:

- Generate and check message digests like MD5 or SHA.
- Create and verify OpenPGP signatures.
- Synchronize files, directories, or file system trees based on digests and signatures.
- Access network file systems with FTP/SFTP. Generate and check digests/signatures.
- Create, import, and export OpenPGP keys comfortably.
- Runs on Windows, Mac OS X, Linux, and many more systems.
- Call the command-line interface from your own applications.
- Native support for Ant build scripts.
- Supports all functions of the Basic Edition and the Standard Edition.

Please visit our website to read more about the features of the Data Digester Enterprise Edition.

Additional Links

- Data Digester Basic Edition (BE)↗
- Data Digester Standard Edition (SE)↗
- Data Digester Enterprise Edition (EE)↗

- Compare Data Digester Editions↗
- Digester Main Window

About Message Digests

Without diving too deep into the sometimes complex mathematical background of the algorithms the following section gives you an overview of the message digest technology. Message digests (aka checksums) are cryptographic hash functions which always return the same binary value for a fixed set of data (determinism). If two digest calculations return different results then the processed data sets must be different. The hash result is usually represented by a hexadecimal string of fixed length and can be stored e.g. based on the GNU or BSD file format.

MD5 example (hexadecimal result):

```
23176e28d47e61777f6d246e459b795d
```

The most commonly used MD5 checksums always consist of a total of 16 single binary values (bytes), the generated hexadecimal strings are therefore 32 characters long. The input data can have arbitrary lengths while the results always have a fixed length. This is especially useful when digests are stored into a database or cache because the space needed can be accurately determined. The digest calculation can be based on different techniques called algorithms. The algorithms can be classified by their characteristic calculation result lengths.

Hash functions are used for many purposes. While they are deployed to ensure data integrity on the file level, hash functions particularly can be used for the encoding of passwords for operating systems and databases. In order to avoid the storage of plain-text passwords on insecure hard disks, the hash results are stored instead. The original passwords cannot be reconstructed from the message digests if algorithms like RIPEMD-160 are used which are known to be secure.

The following table shows the algorithms supported by the current Digester version. The list also shows the lengths of the respective calculation results in bytes and the typical time consumption (in seconds) to calculate a reference file of approximately 4 gigabytes. The following sections provide guidance on which algorithm to use and how the Digester software can be effectively deployed to ensure data integrity on the file level.

Algorithm	Bytes	t/s	Comments
MD2	16	1423	Designed for 8-bit computers
MD4	16	219	Used for password hashes for Windows
MD5	16	63	The most common algorithm besides SHA-1
RIPEMD-128	16	74	Based on principles of MD4
RIPEMD-160	20	100	Used within OpenPGP
RIPEMD-256	32	74	Variant of RIPEMD-128
RIPEMD-320	40	94	Variant of RIPEMD-128
SHA-1	20	107	Very often used, developed by the NSA
SHA-224	28	136	Developed by der NSA

SHA-256	32	139	Developed by der NSA
SHA-384	48	311	Developed by der NSA
SHA-512	64	310	Developed by der NSA
Tiger-192	24	291	Designed for 64-bit computers, e.g. Gnutella file sharing
Whirlpool	64	2082	Based on principles of AES

Regarding the shown typical time consumption values please note that the Digester software is able to calculate arbitrary algorithms simultaneously by utilizing a specific multiplexing technique. The input data is only read once. This allows to calculate multiple algorithms very efficiently. Since modern computers often lack fast hard disks while the calculations do not really challenge modern CPUs, the calculation and checking of digests can be done very efficiently with the Digester software. This makes the software ideal especially for the processing of large amounts of data while the number of simultaneously used algorithms has only little impact on the overall performance.

In the following sections the terms **original files** and **digest files** are used. Original files are those files for which the digests and signatures have been generated for. These files can be found on FTP servers or are created by backup software. Message digest files contain the digests that have been calculated for the original files. When comparing the digests from the digest files with the calculated digests e.g. after the download, deviations can be identified that indicate transmission errors or manipulated data.

Additional Links

- OpenPGP Signatures
- Supported File Formats
- DTD Digester XML
- Check Window
- Generate Window
- Sync WIndow

Supported File Formats

To check single files or file collections it is necessary to store the computed digests for the original files into message digest files. Normally these digest files are stored together with the original files for example on magnetic tape which are then delivered to a safe place like a bank safe. In the emergency case such backups can be restored. On the Internet you can find digests files predominantly on FTP and other download servers. Normally the digest files are named after the original files by adding a distinctive suffix such as `.md5` for MD5 message digest files

- GNU File Format
- BSD File Format

- [OpenPGP File Format](#)
- [Digester XML Format](#)

GNU File Format

The most common file format used for message digest files on the Internet is the GNU format. The data is stored as text lines each of which holding a calculated digest for exactly one single original file. The lines are based on a fixed structure, the first element is the digest value followed by a SPACE character, followed by an asterisk * to indicate that it is binary data, and finally followed by the reference to the original file. Normally just the file name without path is used here but the full path could be used as well due to the GNU format. If full file paths such as `C:\Temp\leisenfels_48.png` are stored here such references might not be resolved on another computer with a different file system structure so that digest checking could not be performed. Therefore only the file names without paths should be stored here. The files generated with the Digester software only contain the names without paths. A sample message digest file `leisenfels_48.png.md5` based on the GNU format looks as follows:

```
497f82bf72503317c62ecf6b2645a735 *leisenfels_48.png
```

If the digests for multiple files in a directory are calculated and then stored together into a single file the GNU format allows to do so. The GNU file then contains multiple lines as described above. Also this creation of GNU files per directory is supported by the Digester software although most digest files on the Internet are dedicated to one single original file. Usually GNU digest file names are extended by a characteristic file suffix representing the used algorithm. In the example the created message digest file therefore is named `leisenfels_48.png.md5`. The Digester software automatically appends these suffixes depending on the configuration so that any name may be used for message digest files. Therefore it is not possible to store BSD and GNU files simultaneously since both formats use the same file naming scheme. The new Digester XML format provides more flexibility than the older file formats GNU/BSD do.

BSD File Format

Another widely used file format for message digest files on the Internet is the BSD file format. As with the GNU files, the digest values are stored as separate text lines for each original file while the line format is slightly different:

```
MD5 (leisenfels_48.png) = 497f82bf72503317c62ecf6b2645a735
```

The first element denotes the algorithm used for the digest calculation (here MD5), after the SPACE character follows the reference to the original file in parentheses which is followed by the equal sign and the hexadecimal digest value. Normally just the file name without path is used here but the full path could be used as well due to the BSD format. If full file paths such as `C:\Temp\leisenfels_48.png` are stored here such references might not be resolved on another computer with a different file system structure so that digest checking could not be performed.

Therefore only the file names without paths should be stored here. The files generated with the Digester software only contain the names without paths.

If digests for multiple original files shall be stored into one single message digest file the same rules apply as for the GNU files (1 line per original file). The line structure of the BSD file format is more attached to the mathematical description of hash functions. For example a typical line can be translated like this: apply the hash function MD5 to the parameter file `leisenfels_48.pn` and the result of this operation equals the value X.

Usually also BSD digest file names are extended by a characteristic file suffix representing the used algorithm. In the example the created message digest file therefore is named `leisenfels_48.png.md5`. The Digester software automatically appends these suffixes depending on the configuration so that any name may be used for message digest files. Therefore it is not possible to store BSD and GNU files simultaneously since both formats use the same file naming scheme. The new Digester XML format provides more flexibility than the older file formats GNU/BSD do.

OpenPGP File Format

OpenPGP signatures are by far the most reliable mechanisms to check files from the Internet, since this allows for both checking the integrity of a file as well as determining the origin of the file. Each signature based on the OpenPGP standard is generated with one or more secret keys, the key identifiers are included in each signature file. For the overall comparison of signatures the public key of the signature's producer is required. Public keys are usually downloaded from the website of the respective download provider and can be easily imported into the Digester software.

A typical signature file in accordance with the OpenPGP standars looks like as follows:

```
-----BEGIN PGP SIGNATURE-----
Version: Digester 1.6.3
iEYEABECAAYFAk/4NgcACgkQJFLzC/zoww/htgCfY0Denq1XxXT83PGa9LcdAkeC57AAoI7YXdmme2SrI5JrpxT/ueUwn3UT
=civA
-----END PGP SIGNATURE-----
```

Signature files may be created either in ASCII format (see listing) or in binary format. Binary files have the advantage that less space is needed for storage. Inside the Digester XML files OpenPGP signatures are always stored in ASCII format and not as binary data since this may cause some problems. Unlike the other formats presented here, each signature file contains only one signature, since signature files with multiple signatures are not possible here. The exact format of OpenPGP signatures is defined by RFC 4880.

Typical extensions for signature files are `.asc` and `.sig`, but you may configure arbitrary file extensions within the Digester software to be generated. To label generated signature files, a single-line text can be configured to be stored in the header of the ASCII signatures. Here, you can place the name of your company or organization for example.

Digester XML Format

Due to the limitations described above you should neither use the GNU nor the BSD file format to store message digests. But these older formats can sometimes be a good solution for files on FTP servers since customers are often better prepared for GNU/BSD files. Wherever possible the new XML-based file format should be preferred which can be used very efficiently together with the Digester software. Here you can also create single digest files for each original file but the digests can also be bundled together in arbitrary combinations even over drive boundaries.

Thus, with the Digester XML format it is possible to store digest information for complete directory trees of a hard drive into just a single message digest file which allows a much clearer view than thousands of digest files. In addition such files can be stored on arbitrary drives since absolute file paths can be included. Here you can see an example of a Digester message digest file in XML format `jnlp-1_5-mr-spec.pdf.digest`):

```
<?xml version="1.0" encoding="UTF-8"?>
<summary version="1.1" date="Sat Jul 07 15:13:43 CEST 2012" targets="1">
  <comment>Created with Leisenfels Digester</comment>
  <target relpath="jnlp-1_5-mr-spec.pdf" length="227496"
          modified="Tue May 29 10:56:56 CEST 2012" digests="6" pgpsigs="1">
      <digest algorithm="SHA-1" size="20"
              format="hex">9346cb18d8cf8d5066e661dc468540b14275285f</digest>
      <digest algorithm="SHA-1" size="20" pos="8192"
              format="hex">0ad56058ddd6fba142c38f4552bf4fe16f93103f</digest>
      <digest algorithm="SHA-1" size="20" pos="16384"
              format="hex">68bb5167eb62667d3f5ea9bfd529e438daed0586</digest>
      <digest algorithm="SHA-1" size="20" pos="32768"
              format="hex">36b31b4fe0a50557bf66f1976eb2c07b5228616f</digest>
      <digest algorithm="SHA-1" size="20" pos="65536"
              format="hex">57010aa5db867eecc6553b5842f04f158df09cb2</digest>
      <digest algorithm="SHA-1" size="20" pos="131072"
              format="hex">128e90d30b816addb24b1106195accdf78952a6a</digest>
      <pgpsig keyid="0x2452F30BFCE8C30F" keyname="Leisenfels Development"
              keyemail="devel@leisenfels.com"
      keyurl="http://ftp.leisenfels.com/security/pgpkeys/devel@leisenfels.com.asc"
              size="194" format="ascii">-----BEGIN PGP SIGNATURE-----
Version: Digester 1.6.3
iEYEABECAAYFAk/4NgcACgkQJFLzC/zoww/htgCfY0Denq1XxXT83PGa9LcdAkeC
57AAoI7YXdmme2SrI5JrpxT/ueUwn3UT
=civA
-----END PGP SIGNATURE-----</pgpsig>
   </target>
</summary>
```

As you can see, also these XML files are simple text files compatible with most text editors or text processing programs. The XML format has the advantage that a wide range of character encodings (character sets) can be used for the data (in this case UTF-8). For example also Japanese characters can be used for file names. Every XML file consists of so-called tags which are expressions in angle brackets, while each tag is closed by a second tag of the same name and the leading "/" character. In between the associated tags either data or other structuring tags can be found. Tags can also be closed by inserting the "/" character directly at the end of the opening tag so that the second tag can be omitted (short form of writing).

The XML files refer to the document format `summary` version 1.0 or 1.1. The time and date of writing the message digest file follows within the `date` attribute. Equipped with this data field such XML files can be easily moved or copied while the timestamp of the copied file is normally lost. For this reason the time of writing is documented separately within each digest file.

The number of original files referenced by the digest file is provided by the `targets` attribute since an arbitrary number of original files can be referenced here. The next XML tag `comment` contains a description to identify the file. This text can be configured with the Digester software so that notes for the FTP server or terms of use can be stored.

The digest values are stored with the `target` attribute which can be repeated any number of times, the example shows that only the file `jnlp-1_5-mr-spec.pdf` had been calculated. The `relpath` attribute (short for relative path) contains the plain file name without any paths for the computer as stored into GNU/BSD files. Such references may generally be resolved until the Digester XML file stays in the same directory as the original files. In addition the Digester software can be configured to store the `abspath` attribute (short for absolute path). Here the full access paths to the original files such as `C:\Temp\jnlp-1_5-mr-spec.pdf` can be found.

Since absolute references can be resolved at least on the computer on which the Digester XML file was created, the digest files can be moved to any place on the same computer so that the data can still be processed correctly. So you could for example let the Digester software scan all operating system drives in a single pass, calculate the digests as configured, and store the overall results including all digests and additional data into one single Digester XML file. This allows for a good overview because digest files are not written for each original file or for each directory as necessary for the GNU/BSD format.

An important detail about the original files is the file length in bytes. This information may be an indicator whether an examination of the contained digests should be started at all. For example you could decide that the original files on an FTP server only need to be recalculated if either the timestamp or the length of the file has changed. This additional information which is only available within Digester XML files may help to avoid unnecessary digest checks or calculations. This can imply significant speed advantages especially when checking original files on remote servers against mirrored local examinee files.

Like the `date` attribute the `modified` attribute contains a timestamp but here the timestamp of the original file at the time of calculating the digests is remembered. As already described above, such additional information can help to decide whether checking must be performed or not. By placing

this date within the digest file changes of the original files can be detected very easily. Furthermore, timestamps at the level of the operating system are usually lost when original files are copied from one drive to another.

The `digests` attribute counts the number of digests following. Compared with GNU/BSD files Digester XML files can contain digests calculated by algorithms in any combination. Thus, for example MD5 and SHA-1 digests can be stored together into a Digester XML file.

Now to the digests in the `digest` tags. The `algorithm` attribute describes the used hash algorithm (here MD5 and SHA-1). Each `digest` tag contains the information about exactly one digest, so the example shows two such sections. The next attribute `size` denotes the length of the digest value in bytes. This information is important if the data from a Digester XML file is imported into a database with fixed length fields. The `format` attribute denotes the format of the digest, currently the Digester software supports the values "hex" for hexadecimal digests and "base64" for digests in the Base64 encoding. Base64 for example is often used to encode e-mail attachments, normally digests should be stored using the hexadecimal format. Right before the closing `digest` tag follows the digest value. Due to the "hex" format setting the example shows the same value as for the GNU/BSD files which would be different if the "base64" format would have been activated.

In addition to digests such as MD5, SHA-1 etc. also OpenPGP signatures may be stored within the Digester XML format (`pgpsig` element). In addition to the signature itself which is always stored in ASCII format, a variety of information can be stored to identify the used OpenPGP keys. Within the Digester software, the following key attributes may be selected to be added:

- `keyid`: hexadecimal ID of the key
- `keyname`: name of the key
- `keyemail`: eMail address of the key owner
- `keyurl`: Internet address to download the (public) key

Another advantage of the Digester XML format is the capability to store intermediate digests which can be efficiently processed with the Digester software. While GNU/BSD files only can contain one single digest for each original file, the use of Digester XML files allows to store as many intermediate digests as desired. The above shown Digester XML file contains only the final message digest value as a result from processing all bytes of the original file. The Digester software can now be configured to create interim results at defined positions as intermediate digests which can then be stored along with the exact position within the Digester XML files.

This has the advantage that deviations of the original file and the examinee file often can be detected more quickly than if the entire examinee file has to be processed. Here you can take advantage of the effect that examinee files often differ from the original files already in the first data blocks. Especially when checking files on remote servers this can have substantial speed advantages since less data must be transferred or be processed.

The following example shows a Digester XML file with intermediate digests that have been created after 1 kilobyte (pos=1024). The mode for the intermediate digest creation can be configured within the Digester software. Currently you can create intermediate digests linearly in fixed intervals (for

example every 10 megabytes) or exponentially (for example starting with 16 kilobytes, then 32 kilobytes, 64 kilobytes etc.). For both intermediate digest generation types you can configure a maximum number of intermediates so that the Digester XML files cannot grow uncontrolled.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<summary version="1.0" date="Sat Feb 07 17:10:18 CET 2009" targets="1">
   <comment>Created with Leisenfels Digester</comment>
    <target relpath="leisenfels_48.png" length="1906"
          modified="Sun Jul 13 16:51:04 CEST 2008" digests="4">
       <digest algorithm="MD5" size="16"
               format="hex">497f82bf72503317c62ecf6b2645a735</digest>
       <digest algorithm="MD5" size="16" pos="1024"
               format="hex">13e5314fc870d2ec9ef84515cc34abdc</digest>
       <digest algorithm="SHA-1" size="20"
               format="hex">6e09d8f73240cefcaf15167ce3749d8e99ecba63</digest>
       <digest algorithm="SHA-1" size="20" pos="1024"
               format="hex">59877ebbf5854b8ac798b46a7ab5fd80339eacd2</digest>
   </target>
</summary>
```

Unlike for GNU/BSD files the extension of Digester XML files is not determined by the used algorithm since any algorithms can be used in arbitrary combinations. The Digester software allows to configure the default extension (defaults to `.digest`). You can also specify the `.xml` extension for instance which is normally used for XML files.

For the DTD (Document Type Definition) of the Digester XML format see section DTD Digester XML.

Additional Links

- RFC 4480: OpenPGP Message Format↗
- DTD Digester XML

File System Window

The file system window shows the local file system depending on the underlying operating system. The figure shows a typical file system under Windows with drives, the network environment etc. For other operating systems like Linux the subdirectories of the root directory are listed. The figure shows the window components file system overview (1), directory overview (2), tool bar (3), context menu (4), and status bar (5).

The navigation is intuitive by double-clicking the directories (aka folders) with the mouse. To jump a directory level back you can double-click the first list entry with the two dots "..". By selecting folders and/or files you can deploy the functions either via the context menu directly on the file list (right mouse button) or with the tool bar icons. These include well-known functions such as cut, copy, paste, delete, rename, and find.

Files and directories may be easily exchanged between the file system window, the network file system window and other docking windows. Here you can configure which drag & drop actions are allowed. Currently the Digester supports dragging files from the file system and network file system window and dropping into the sync window. Source and targets sets may be assembled very easily this way. If the drag & drop turns out to be not useful for your daily work you may disable drag & drop completely within the preferences.

File System Overview

This overview is shown on the left side of the file system window while the area width can be adjusted by dragging the separator between the two areas. This setting is saved automatically and restored the next time the software is started. The tree shows the entries of the computer's file system as passed to the Java environment by the operating system (in this case Windows 7). You can see the drives and the network environment. By double-clicking the folders or by clicking the plus symbols in front of the folders you may expand the subdirectories. The file system overview shows only directories and no files, these will be displayed by the directory overview on the right side when overview folders are selected. The details for the current selection such as selection type (file or directory) as well as the full path and the selection size (in bytes) you can find in the status bar at the bottom of the file system window.

Above the file system overview the tool bar is located which is part of the file system window and must not be confused with the [tool bar of the Digester software](#) (larger symbols). The tool bar icons provides you with known functions such as cut, copy, paste, delete, rename, and find.

Directory Overview

This overview is located on the right side of the display area. The table shows the files and directories for the selection in the file system overview where only one single entry may be selected, otherwise the directory overview is empty. As the figure shows, all entries are listed alphabetically with additional information. Along with the name of the entry in the first column the table shows the permissions in the **drwx** column (d=directory, r=read permission, w=write permission, x=execute permission). This detail is especially helpful for UNIX/Linux systems where usually more restrictive permissions are configured as for Windows.

The **Size** column shows the length/size of the entry in abbreviated form (e.g. kilobytes), the **Type** column lists the type of entry (file, directory, digest file etc.). This information is provided directly by the underlying operating system so that the shown types may vary. This is the case also for the icons in the first column **Name**. The **Modified** column finally shows the date and time of last modification in ISO notation (YYYY-MM-DD).

Within the directory overview you can also select any entries and activate the context menu (right mouse button) or click the buttons of the window tool bar.

Tool Bar

The tool bar of the file system window contains many functions for fast navigation and for the work with files and directories. The following list shows the functions of the tool bar:

- **Parent Directory**: Go one directory level up, same as double-clicking "**..**"
- **User's Home**: Go to the user's home directory
- **Reset Folders**: Close all open directories and create initial view
- **Open**: Let the application or operating system open the selected files
- **Edit**: Let the application or operating system edit the selected files
- **Hexedit**: Opens the integrated hex editor for the selected files
- **Create Directory**: Creates a new folder in the selected folder
- **Create File**: Creates an empty file in the selected folder
- **Delete**: Deletes selected entries, user is prompted to commit
- **Rename**: Gives the selected entry a new name
- **Search**: Starts searching for files/directories based on the name

The button with the up arrow symbol moves to the next directory level (**Parent Directory**). The house symbol navigates into the home directory of the current user (**User's Home**), for example `/home/axel` under Linux. The next icon with the closed folder (**Reset Folders**) causes all expanded directories within the file system overview to collapse so that it shows the initial state. This is useful when many directories are opened and the window contents are confusing.

The button with the play icon (**Open**) opens the selected entry. This function is only available for files, for selected directories it is deactivated. A click on the icon starts the default programs based on the file extension. Please note, that this strongly depends on the operating system and possibly clicking does not show any response. Similarly, the next button with the editor icon let's the operating system start the default editor (**Edit**). On Windows operating systems for example, the program WordPad may be opened if this is configured as an editor for the selected file.

The Digester software supports opening and editing some specific file types (e.g. `.jpg`, `.png`, `.gif` and `.xml`). Such files may be directly displayed by the document window. For all other file types the action is delegated to the underlying operating system. For the file types supported by Digester you have the choice to open the files either by the application or the operating system (see preferences in section "Main Menu and Components"). The next button with the dots and strokes (**Hexedit**) opens a file with the integrated hex editor of the Digester software. This function may be called for any file as long as the program is able to access it.

The next icon with the plus symbol on the folder (**Create Directory**) may be used to create new folders in the selected directory. This function is only activated when you selected a directory within the file system overview. To create a new directory, a dialog will be shown prompting the user for the name of the new folder. After successful creation of the new folder it is shown by the file system window components. You may create new entpy files with the next button showing the plus symbol on the document (**Create File**). The icon with the red X (**Delete**) can be used to delete files or directories including subdirectories.

> ⚠️ Directories and files deleted with the delete function cannot be restored programmatically, so please backup your data first. Before entries are deleted a security dialog is displayed which draws attention to this fact and where the user explicitly has to confirm the deletion.

Clicking the icon with the pen (**Rename**) allows you to rename the selected entry while both files and directories can be renamed. As for the creation of a directory, a special dialog is displayed which prompts you to enter a new name. The icon with the magnifying glass (**Search**) can be used to find entries starting from the selected directory. When you click on this icon, an input field on top of the file system overview is shown where you can enter a search pattern. The search window is shown automatically which is normally positioned on the right side of the docking window. There you can configure the search, for instance the search may be case sensitive or can be based on regular expressions. The search is currently limited on the names of files and directories, file contents cannot be searched currently.

Many of the features presented here are also available in the tool bar and can be activated also with the associated entries in the menu bar of the Digester software. For example, you can also delete files/directories by first selecting the desired entries and then clicking the DELETE button or alternatively by activating the **Edit** -> **Delete** menu. Where possible, the functions of the main window menu and tool bar are configured by the docking windows. Whether these functions are available or not is controlled by the active docking window.

Above the directory overview you can see the **Path** input field. Here, an absolute path can be specified, the corresponding entry is then displayed by the other components of the file system

window (commit with ENTER). If the clipboard contains a valid path for example then you can insert this entry and make it visible by clicking the icon to the right of the input field. Alternatively you can commit your editing with ENTER. The path input field stores the previous entries automatically as a history list so that it is available the next time the software is started. You can delete the entries of this history list by activating the **Clear History** item from the context menu of the input field (right mouse button).

Context Menu

Of course even the file system window - like many other applications - provides support for the context menu which can be opened directly on the selection. You can select entries in the file system overview or directory overview and activate the menu with the right mouse button (see illustration). Here you can find all of the above described functions of the tool bar. In addition to the editing functions such as cut/copy/paste you can open or edit the selection here (if applicable), create new directories, delete entries, and rename entries. Also file searches can directly be started from here. Besides these standard features, the context menu provides support for additional functions specific for the Digester software.

- **Check File**: Start checking digests/dignatures for configured file extensions
- **Generate Digest**: Generates digests, algorithm and file format can be selected
- **Generate Signature**: Generates OpenPGP signatures, key/format is selectable
- **Cut**: Like delete
- **Copy**: Copies the selection into the clipboard
- **Paste**: Inserts files/directories from the clipboard (creates copies)
- **Open**: Opens the selected files with program or operating system
- **Edit**: Starts editing the selection with program or operating system
- **Hexedit**: Opens the integrated hex editor for the selected files
- **Create Directory**: Creates a new directory in the selected folder
- **Create File**: Creates an empty file in the selected folder
- **Delete**: Deletes the selected entries, user is prompted to commit
- **Rename**: Gives the selected entry a new name
- **Search**: Starts searching for files/directories based on the name
- **Close Search**: Closes the search field and the search window

Status Bar

The status bar at the bottom of the file system window shows pieces of information for the selected files/directories. For single selections the absolute path of the entry and the exact size in bytes are being displayed. For multiple selections or directories the status bar shows the number of entries as well as the total size of the selection. If a path is displayed it may be copied with the context menu (right mouse button) into the clipboard to be used by other applications. If listing the current directory takes a long time then the status bar component displays a progress bar or indicator on the right side.

Additional Links

- Document Window
- Search Window
- Search Results Window
- Check Window
- Generate Window
- Preferences
- Network File System Window

Network File System Window

The network file system window shows a server file system depending on the used protocol and operating system. The figure shows a typical FTP file system. The figure shows the window components file system overview (1), directory overview (2), tool bar (3), context menu (4), status bar (5), and sessions (6).

The navigation is intuitive by double-clicking the directories (aka folders) with the mouse. To jump a directory level back you can double-click the first list entry with the two dots "..". By selecting folders and/or files you can deploy the functions either via the context menu directly on the file list (right mouse button) or with the tool bar icons. These include well-known functions such as cut, copy, paste, delete, rename, and find.

Files and directories may be easily exchanged between the file system window, the network file system window and other docking windows. Here you can configure which drag & drop actions are allowed. Currently the Digester supports dragging files from the file system and network file system window and dropping into the sync window. Source and targets sets may be assembled very easily this way. If the drag & drop turns out to be not useful for your daily work you may disable drag & drop completely within the preferences.

> **BE**
> **SE**
> Please keep in mind that the network functionality is not supported by the Basic Edition and the Standard Edition.

File System Overview

This overview is shown on the left side of the network file system window while the area width can be adjusted by dragging the separator between the two areas. This setting is saved automatically and restored the next time the software is started. The tree shows the entries of the computer's file system as passed to the Java environment by the operating system (in this case Windows 7). You can see the drives and the network environment. By double-clicking the folders or by clicking the plus symbols in front of the folders you may expand the subdirectories. The file system overview shows only directories and no files, these will be displayed by the directory overview on the right side when overview folders are selected. The details for the current selection such as selection type (file or directory) as well as the full path and the selection size (in bytes) you can find in the status bar at the bottom of the network file system window.

Above the file system overview the tool bar is located which is part of the network file system window and must not be confused with the tool bar of the Digester software (larger symbols). The tool bar icons provides you with known functions such as cut, copy, paste, delete, rename, and find.

Directory Overview

This overview is located on the right side of the display area. The table shows the files and directories for the selection in the file system overview where only one single entry may be selected, otherwise the directory overview is empty. As the figure shows, all entries are listed alphabetically with additional information. Along with the name of the entry in the first column the table shows the permissions in the **drwx** column (d=directory, r=read permission, w=write permission, x=execute permission). This detail is especially helpful for UNIX/Linux systems where usually more restrictive permissions are configured as for Windows.

The **Size** column shows the length/size of the entry in abbreviated form (e.g. kilobytes), the **Type** column lists the type of entry (file, directory, digest file etc.). This information is provided directly by the underlying operating system so that the shown types may vary. This is the case also for the icons in the first column **Name**. The **Modified** column finally shows the date and time of last modification in ISO notation (YYYY-MM-DD).

Within the directory overview you can also select any entries and activate the context menu (right mouse button) or click the buttons of the window tool bar. In the preferences you can configure how file symbols and types should be determined by the program. Here, you can choose either to show default icons (faster) or to show symbols and types of your computer (costly, since temporary files have to be created).

Tool Bar

The tool bar of the network file system window contains many functions for fast navigation and for the work with files and directories. The following list shows the functions of the tool bar:

- **Parent Directory**: Go one directory level up, same as double-clicking "**..**"
- **User's Home**: Go to the user's home directory
- **Reset Folders**: Close all open directories and create initial view
- **Open**: Let the application or operating system open the selected files
- **Edit**: Let the application or operating system edit the selected files
- **Hexedit**: Opens the integrated hex editor for the selected files
- **Create Directory**: Creates a new folder in the selected folder
- **Create File**: Creates an empty file in the selected folder
- **Delete**: Deletes selected entries, user is prompted to commit
- **Rename**: Gives the selected entry a new name
- **Search**: Starts searching for files/directories based on the name

The button with the up arrow symbol moves to the next directory level (**Parent Directory**). The house symbol navigates into the home directory of the current user (**User's Home**), for example `/home/axel` under Linux. The next icon with the closed folder (**Reset Folders**) causes all expanded directories within the file system overview to collapse so that it shows the initial state. This is useful when many directories are opened and the window contents are confusing.

The button with the play icon (**Open**) opens the selected entry. This function is only available for files, for selected directories it is deactivated. A click on the icon starts the default programs based on the file extension. Please note, that this strongly depends on the operating system and possibly clicking does not show any response. Similarly, the next button with the editor icon let's the operating system start the default editor (**Edit**). On Windows operating systems for example, the program WordPad may be opened if this is configured as an editor for the selected file.

The Digester software supports opening and editing some specific file types (e.g. `.jpg`, `.png`, `.gif` and `.xml`). Such files may be directly displayed by the document window. For all other file types the action is delegated to the underlying operating system. For the file types supported by Digester you have the choice to open the files either by the application or the operating system (see preferences in section "Main Menu and Components"). The next button with the dots and strokes (**Hexedit**) opens a file with the integrated hex editor of the Digester software. This function may be called for any file as long as the program is able to access it.

The next icon with the plus symbol on the folder (**Create Directory**) may be used to create new folders in the selected directory. This function is only activated when you selected a directory

within the file system overview. To create a new directory, a dialog will be shown prompting the user for the name of the new folder. After successful creation of the new folder it is shown by the file system window components. You may create new entpy files with the next button showing the plus symbol on the document (**Create File**). The icon with the red X (**Delete**) can be used to delete files or directories including subdirectories.

> ⚠️ Directories and files deleted with the delete function cannot be restored programmatically, so please backup your data first. Before entries are deleted a security dialog is displayed which draws attention to this fact and where the user explicitly has to confirm the deletion.

Clicking the icon with the pen (**Rename**) allows you to rename the selected entry while both files and directories can be renamed. As for the creation of a directory, a special dialog is displayed which prompts you to enter a new name. The icon with the magnifying glass (**Search**) can be used to find entries starting from the selected directory. When you click on this icon, an input field on top of the file system overview is shown where you can enter a search pattern. The search window is shown automatically which is normally positioned on the right side of the docking window. There you can configure the search, for instance the search may be case sensitive or can be based on regular expressions. The search is currently limited on the names of files and directories, file contents cannot be searched currently.

Many of the features presented here are also available in the tool bar and can be activated also with the associated entries in the menu bar of the Digester software. For example, you can also delete files/directories by first selecting the desired entries and then clicking the DELETE button or alternatively by activating the **Edit** -> **Delete** menu. Where possible, the functions of the main window menu and tool bar are configured by the docking windows. Whether these functions are available or not is controlled by the active docking window.

Above the directory overview you can see the **Path** input field. Here, an absolute path or an URL can be specified, the corresponding entry is then displayed by the other components of the network file system window (commit with ENTER). If the clipboard contains a valid path for example then you can insert this entry and make it visible by clicking the icon to the right of the input field. Alternatively you can commit your editing with ENTER. The path input field stores the previous entries automatically as a history list so that it is available the next time the software is started. You can delete the entries of this history list by activating the **Clear History** item from the context menu of the input field (right mouse button).

Context Menu

Of course even the network file system window - like many other applications - provides support for the context menu which can be opened directly on the selection. You can select entries in the file system overview or directory overview and activate the menu with the right mouse button (see illustration). Here you can find all of the above described functions of the tool bar. In addition to the editing functions such as cut/copy/paste you can open or edit the selection here (if applicable), create new directories, delete entries, and rename entries. Also file searches can directly be

started from here. Besides these standard features, the context menu provides support for additional functions specific for the Digester software.

- **Check File**: Start checking digests/dignatures for configured file extensions
- **Generate Digest**: Generates digests, algorithm and file format can be selected
- **Generate Signature**: Generates OpenPGP signatures, key/format is selectable
- **Cut**: Like delete
- **Copy**: Copies the selection into the clipboard
- **Paste**: Inserts files/directories from the clipboard (creates copies)
- **Open**: Opens the selected files with program or operating system
- **Edit**: Starts editing the selection with program or operating system
- **Hexedit**: Opens the integrated hex editor for the selected files
- **Create Directory**: Creates a new directory in the selected folder
- **Create File**: Creates an empty file in the selected folder
- **Delete**: Deletes the selected entries, user is prompted to commit
- **Rename**: Gives the selected entry a new name
- **Search**: Starts searching for files/directories based on the name
- **Close Search**: Closes the search field and the search window

Status Bar

The status bar at the bottom of the network file system window shows pieces of information for the selected files/directories. For single selections the absolute path of the entry and the exact size in bytes are being displayed. For multiple selections or directories the status bar shows the number of entries as well as the total size of the selection. If a path is displayed it may be copied with the context menu (right mouse button) into the clipboard to be used by other applications. Path URLs are always copied into clipboard without username/password for security reasons since also other programs may access the data. If listing the current directory e.g. for slow ISDN dial-up connections takes a long time then the status bar component displays a progress bar or indicator on the right side.

Sessions

For each of the connections the window displays a separate tab showing the connection name. The tabs may be closed using the red X button, active connections are then disconnected. You should wait for all tasks to be properly finished to avoid data loss. New sessions may be created either with the Network menu in the main menu, using the tool bar, or the network connection management.

Within the preferences you can configure if sessions should be created automatically on program start. You have the choice here to start with no connections or to restore the previous connections.

Additional Links

- Edit Connections
- Document Window
- Search Window
- Search Results Window
- Check Window
- Generate Window
- Preferences
- File System Window

Document Window

The figure shows the components of the document window with editing area (1), tool bar (2), status bar (3) and document tabs (4).
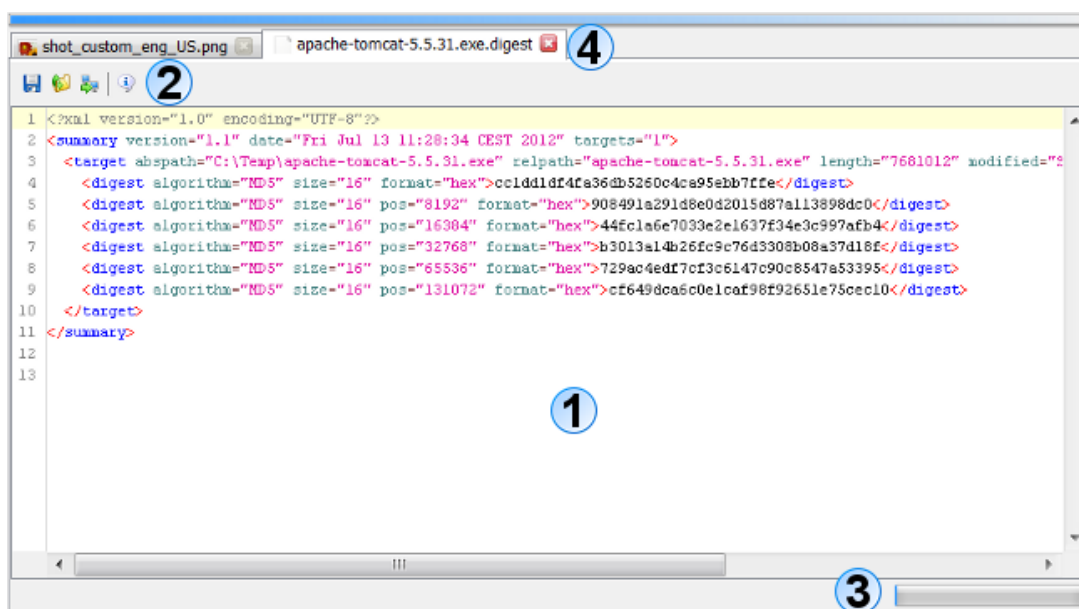
Editing Area

The center of the document window shows the contents of the files loaded from the PC or the network. The figure shows an opened graphics file `shot_custom_eng_US.png` as well as the message digest file `apache-tomcat-5.5.31.exe.digest`. Currently the following file types are supported by the window which means that such files can be viewed or edited:

- **Graphics files** `*.bmp`, `*.jpg`, `*.wbmp`, `*.png`, `*.gif` (viewing only)
- **Text files** `*.txt`, `*.log`, `*.asc`, `*.sig` (viewing/editing)
- **XML files** `*.xml` (viewing/editing)
- **Java Properties** `*.conf`, `*.properties` (viewing/editing)

For these file types arbitrary file extensions may be configured in the preferences. In addition to that you can configure if such files are opened/edited using the document window or the operating

system based on the file extension. Please note, that this strongly depends on the operating system and possibly the system does not show any response. On Windows operating systems for example, the program WordPad may be opened if this is configured as an editor for the selected file.

The open function distinguishes between opening and editing files. While editing normally causes the appropriate editor to be shown in the document window opening may cause a specific program function to be executed. So the check routine of the Digester software is started if an `*.md5` file is opened.

Besides the file types mentioned above the **Hexedit** function of the file system window and network file system window can be used to open any readable file. The editing area therefore shows a special hex editor component, where arbitrary data may be presented on the byte-level. In order to prevent from loading of very large files in the editing area, you can configure a maximum file size within the preferences. Additionally you can limit the number of currently opened files so that the editing area does not become confusing.

In the preferences you can find more ways to configure the document window. There it is possible to start the window with either no documents or with the previous documents. Also you can configure how to handle files with pending changes if tabs shall be closed.

Any supported file type is displayed individually by the document window. XML documents for example are presented using syntax highlighting where XML elements, attributes etc. are colored specially. This allows for an easy overview of the opened files.

Tool Bar

The toolbar of the document window provides simple functions for the work with the opened files. The following listing shows the tool bar functions, which may vary for the supported file types:

- **Save**: Stores the file back to the original data source (e.g. file on PC)
- **Save as file**: Stores the content into a selectable file on the PC
- **Save as network file**: Stores the content into a selectable file on the server
- **Properties**: Shows the properties of the loaded ressources

The button with the diskette dymbol (**Save**) saves the current content back into the file or network file where it was originally loaded from. If the content has been loaded from a different source (e.g. from clipboard) this function is not possible, the button is disabled then. The next button with the arrow on the folder (**Save as file**) saves the content into a local file of your choice which can be specified with a dialog. The button with the server arrow (**Save as network file**) can be used similarily to save the content into a network file. The detailed properties of the resource including size and used editor type are being displayed when the info icon is clicked (**Properties**). If possible for the file type additional symbols may be shown here e.g. for printing.

Status Bar

The status bar at the bottom of the document window shows messages of the editors for example if resources are saved. If saving a network files for instance takes a long time then the status bar component displays a progress bar or indicator on the right side.

Document Tabs

For each of the opened files and network files (resource in general) the window displays a separate tab showing the resource name. The tab tooltips display the full names or URLs of the resources. The tabs may be closed using the red X button, the contents are being removed from main memory. You should always save changed files properly to avoid data loss.

Within the preferences you can configure how to proceed if changed documents are closed:

- **Ask me what to do with changed files**
- **Save changed files automatically when closed**
- **Close changed files without saving (changes are lost)**

Within the preferences you can configure if documents should be reopened automatically on program start. You have the choice here to start with no documents (empty window) or to restore the previous documents.

Additional Links

- File System Window
- Network File System Window
- Preferences

Check Window

The comprehensive checking functionality is provided by the docking window named **Check**. In the default configuration of the Digester software this window is located on the right side of the main window. As you can see the window consists of two tabs **Files** and **Options** where the checking process may be controlled. If you like entire directories, directory trees, or drives being checked, please use the check window.
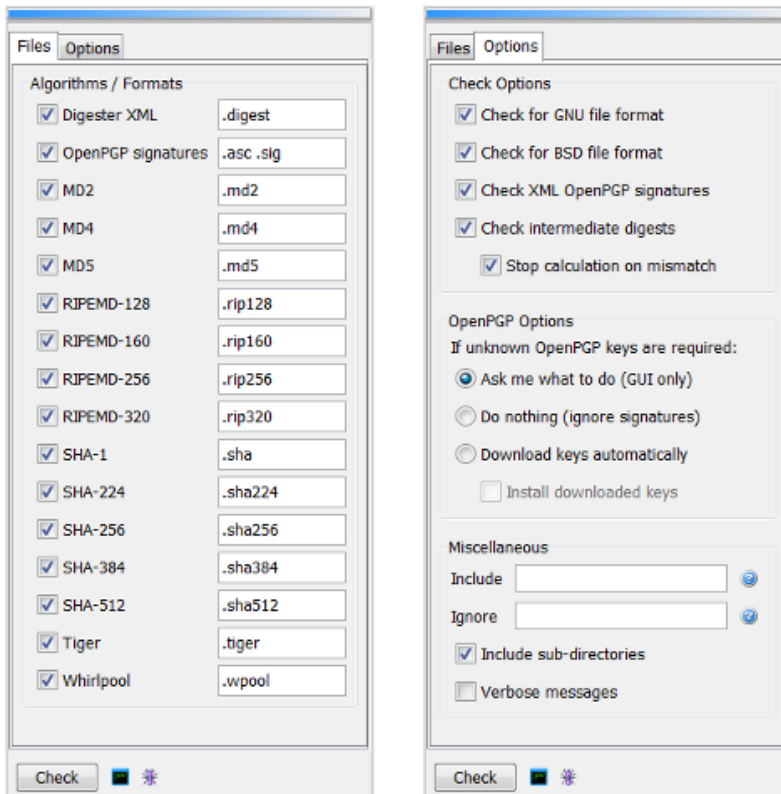
Tab Files

As the figure shows, the **Files** tab provides the configuration of the digest algorithms and their associated file extensions used for searching. By selecting the checkboxes for the various algorithms, the Digester software is advised to recognize these algorithms when determining digest files to be checked. You can configure any combination of algorithms, but at least one algorithm needs to be selected. If you like to select all algorithms in one step, please activate the context menu (right mouse button) directly on the algorithm checkboxes. In addition to the standard file extensions `.md5`, `.sha` etc. you can let the application especially check files in Digester XML Format as well as OpenPGP signatures.

In the input fields to the right of the algorithms you can enter the file extensions to be recognized. The fields are already pre-configured with widely used file extensions. If you for instance selected an examinee file and selected the MD5 option, then all files with the .md5 extension are expected to contain valid digests to be checked. The file extensions can be configured in addition to the settings of the generate window. Thus, also rarely used file extensions can be configured for checking, while the extensions used for generating can be configured separately. Please note, that you can also enter multiple extensions separated by SPACE characters into the input fields, if you like the software to recognize multiple extensions for searching.

Tab Options

This tab contains additional configuration settings with an impact on the checking process:

- **Check for GNU file format**: Include GNU message digest files for checking
- **Check for BSD file format**: Include BSD message digest files for checking
- **Check XML OpenPGP signatures**: Check signatures in Digester XML files
- **Check intermediate digests**: Check intermediate digests in Digester XML files
- **Stop calculation on mismatch**: Abort checking if intermediates are not matched

If GNU or BSD files shall be checked, then please select the appropriate options (**Check for GNU file format** and/or **Check for BSD file format**). If one of these options is not selected, the software ignores the corresponding file formats when digests are loaded.

If you like intermediate digests being checked (**Check intermediate digests**) then please make sure to also activate the **Digester XML** option on the **Files** tab, since intermediate digests can only be stored into Digester XML files. The use of intermediate digests allows to quickly detect deviations of files since these often differ already in the first data blocks. Usually it makes sense to let the Digester software cancel the checking operation, when a mismatch of an intermediate digest has been detected by selecting the **Stop calculation on mismatch** option.

Additional options exist to control the program behavior especially if OpenPGP signatures are being checked. If unknown keys have been detected in the signature files you have the following options:

- **Ask me what to do (GUI only)**: A dialog is displayed to specify the next steps
- **Do nothing (ignore signatures)**: The signatures will be ignored
- **Download keys automatically**: Download if URL is available (Digester XML only)
- **Install downloaded keys**: Store downloaded keys permanently

If only certain digest files shall be checked you may enter a regular expression in the **Include** input field. If the field is empty then all recognized files will be checked by default. If certain digest files shall *not* be checked then you can specify a regular expression with the **Ignore** input field. These options are particularly useful, if directories or entire directory trees including subdirectories are processed. You can activate recursive scanning of subdirectories by selecting the **Include subdirectories** option. In order to get detailed messages on the checking process please select the **Verbose messages** option.

Start Checking

Once all the settings within the checking window are as desired, checking can be started by clicking on the **Check** button at the bottom of the window. If the settings make sense and appropriate files and/or directories have been selected within the file system window, this button is enabled. The checking process is monitored by a special dialog where the current state is shown. Upon completion of the checking this progress dialog shows some statistics for the operation and the checking results. Besides the time consumption of the checking (top left corner) the dialog shows the remaining time (ETA = Estimated Time Access).

Please note, that it is possible to select directories within the file system window to be processed recursively including subdirectories, while this is not possible if you start checking from the context menu.

When all digests have been checked and the progress dialog was closed, the results are displayed by the results window which is automatically opened. The checking results are divided into two tabs **Files** and **File Digests** or **URLs** and **URL Digests**, if the checking has been started from the network file system window.

Command-line Arguments

In order to make checking from the command-line with `digestercli` as easy as possible the necessary arguments may be assembled by the Digester software. Please select the files/directories to be checked within the file system window in the first step. Then configure the checking options with the check window as if the process would be started directly from here. By clicking the button with the console symbol a dialog is displayed showing the corresponding `digestercli` call. This code can be easily copied into the clipboard to be used for shell or batch scripts. As you can see programming is not required here.

Ant Build Scripts

Like the command-line arguments also XML fragments for Ant build scripts may be generated using the button with the ant symbol. This code can be used for own shell or batch scripts. Please select the files/directories to be checked within the file system window in the first step. Then configure the checking options with the check window as if the process would be started directly from here. By clicking the button a dialog is displayed showing the corresponding XML code. This code can be easily copied into the clipboard to be inserted into a text editor for example. As you can see programming is not required here.

Additional Links

- File System Window
- Network File System Window
- Digester XML Format
- OpenPGP Signatures
- Generate Window
- Regular Expressions
- Results Window
- Checking with `digestercli` and Ant

Generate Window

The digest calculation functionality is provided by the docking window named **Generate**. For the default configuration of the Digester software, the button for this docking window is located on the right edge of the main window. Here you can see five tabs **Formats**, **Options**, **OpenPGP**, **XML**, and **Refresh** to control the calculation process. The generate window provides support for the recursive processing of directories including subdirectories.

BE   Please note that the generate functionality is not available in the Basic Edition.
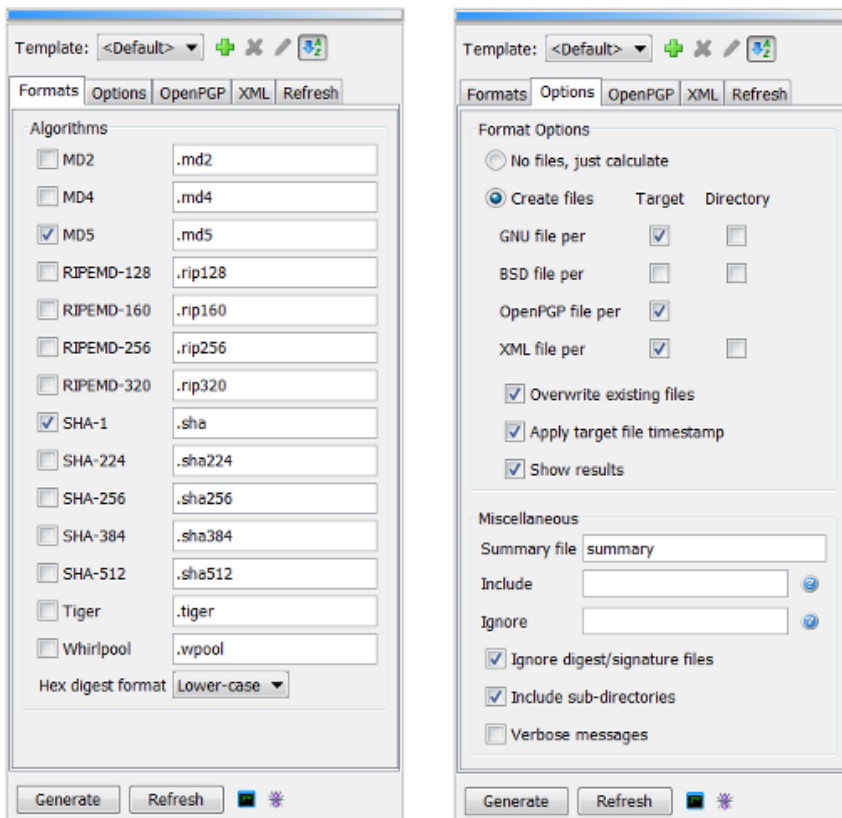
The requirements for the use of digests and signatures may be very different: for FTP servers often SHA and signature files are used, while the Digester XML format is appropriate for archiving backups. In order to support arbitrary environments appropriately, templates may be defined. You can create a new template using the button with plus symbol on top of the generate window. You will be prompted to specify a unique template name. The current settings of the generate window are used to create the new template. You may adjust the settings at any time: please select the desired template and adjust settings as desired. The templates are automatically saved if the program gets terminated and restored when restarted. Templates can be deleted with the red X button after user commit and be renamed with the button showing the pen symbol. The templates can be displayed in alphabetical order or in the order the items were created.

Tab Formats

As the figure shows, the **Formats** tab can be used to configure the digest algorithms and their associated file extensions. By selecting the various algorithms, you configure the Digester software to use these algorithms for calculations. You can specify any combination of algorithms but at least

one algorithm needs to be selected here. If you like to select all algorithms in one step, please activate the context menu (right mouse button) directly on the algorithm checkboxes.

In the fields to the right of the algorithms you can enter the file extensions to be used. The fields are already pre-configured with widely used extensions. If you calculate a hash with the MD5 algorithm for example, the software creates a digest file with the name of the original file plus the extension specified here. The configured file extensions are not only used by the generate window but for all calculation variants including the context menu (right mouse button) and the generate wizard.

Please note, that changes to the file extensions within the generate window do not affect the file associations of the installer variants of the software. The file associations to open files on the operating system level are the same as the default settings of the generate window and can be adjusted only with the appropriate programs (e.g. Windows Explorer).

The option **Hex digest format** configures the case of the calculated hexadecimal digests. It defaults to "Lower-case", the values may alternatively be generated with upper-case letters. This option does not influence OpenPGP signatures.

Tab Options

This tab provides several additional options to control the created file formats:

- **No files, just calculate**: Create no fies, just do the plain calculations
- **Create files**: Files are created in the desired formats
- **GNU file per target/directory**: Create files in GNU format for each original file or folder
- **BSD file per target/directory**: Create files in BSD format for each original file or folder
- **OpenPGP file per target**: Create signature files for each original file
- **XML file per target/directory**: Files in Digester XML format for each original file or folder
- **Overwrite existing files**: Existing digest/signature files are overwritten
- **Apply target file timestamp**: Original file timestamps are applied to created files
- **Show results**: Show results, create files silently otherwise

If files are created (**Create files**), then you can precisely control how this will be done. You can choose between the supported file formats GNU format, BSD format, the OpenPGP signatures, and the Digester XML format. You can use either the GNU or BSD format since these formats are based on the same naming scheme. The XML format can always be configured in addition to GNU/BSD since this file format produces unique filenames due to a separate file extension which does not represent the digest algorithm. The generate window helps you to find an appropriate combination here by rejecting unallowed option combinations.

If you like GNU files being created then you can specify whether GNU files shall be created on a per-target basis (per original file) or if multiple digests for a whole directory shall be saved into a single GNU file (per directory). You can also use both options in parallel. While for the GNU files per target the name of the original files plus a file extension for each algorithm is used, the GNU files per directory are named based on the setting in the **Summary file** input field (defaults to

summary). While the GNU files per target only contain one digest for the used algorithm, the GNU files per directory possibly contain multiple digest values for the algorithm. The digests are stored per line according to the GNU format. A typical GNU file per target for the franz.txt original file and the MD5 algorithm is named franz.txt.md5 for example, a typical GNU file per directory is named summary.md5 based on the default settings.

The settings for BSD files are the same as for GNU files, only that the format of the generated files is slightly different. OpenPGP signature files are limited to the per-target creation. If multiple signatures e.g. for multiple keys shall be stored together please use the Digester XML format.

If you like to create Digester XML files which may contain any number of final digest values plus intermediate digests calculated by arbitrary algorithms, please select the **XML file per target** or **XML file per directory** option or both. While for the XML files per target the name of the original files plus a configurable file extension is used (see the **XML extension** input field on the XML tab), the XML files per directory are named based on the setting in the illustrated input field (defaults to summary). While the XML files per target only contain one digest for each original file and algorithm, the XML files per directory possibly contain multiple digest values for each of the original files and algorithms. A typical XML file per target for the franz.txt original file is named franz.txt.digest for example, a typical XML file per directory is named summary.digest based on the default settings.

By setting the **Overwrite existing files** option, the software is instructed to overwrite existing files, otherwise the calculated digests will not be saved. The **Apply target file timestamp** option should always be selected if the timestamp of the processed original file should be set for the created message digest file (only possible when files are created per target). If digests are calculated for large amounts of data such as backups, then the software can be configured not to produce or display any results by deselecting the **Show results** option.

> ⚠️ Overwritten digest/signature files cannot be restored programmatically, so please backup your data first. A dialog is displayed if this option is selected.

The **Miscellaneous** section of this tab provides additional generate options:

- **Summary file**: Name of files on a per-directory basis (default: summary)
- **Include**: Regular expression to include certain original files (empty=all)
- **Ignore**: Regular expression to ignore certain original files (empty=all)
- **Ignore digest/signature files**: Do not calculate for digest/signature files
- **Include subdirectories**: Subdirectories are also searched for original files
- **Verbose messages**: Print detailed messages during generation

In order to generate only certain files for certain original files you may enter a regular expression in the **Include** input field. If the field is empty then all recognized original files will be processed by default. If certain original files shall *not* be processed then you can specify a regular expression with the **Ignore** input field. These options are particularly useful, if directories or entire directory trees including subdirectories are processed. You can activate recursive scanning of

subdirectories by selecting the **Include subdirectories** option. In order to get detailed messages on the generation process please select the **Verbose messages** option.

Tab OpenPGP

In addition to message digests the Digester software is capable of calculating OpenPGP signatures and signature files. Signature files may be created either int the **ASCII signature file format** (default) or in the **Binary signature file format**. Binary files have the advantage that a little less space in required for storage. Within Digester XML files OpenPGP signatures are always stored in the ASCII format, since binary data cannot be stored here directly. In contrast to the other file formats described here each signature file may contain only one single signature so that signature files for whole directories is not possible here.
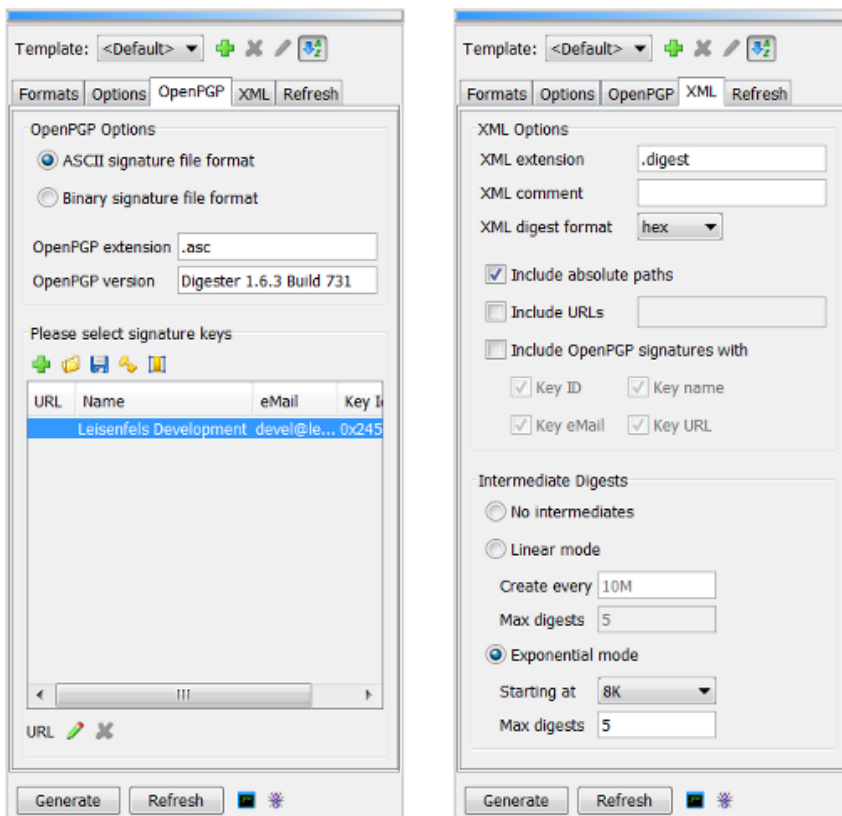
Typical **OpenPGP extensions** for signature files are `.asc` and `.sig`, but the Digester software may be configured to create arbitrary file extensions. In order to label created signature files one can specify an arbitrary single-line text to be stored in the signature file headers (**OpenPGP version**). For instance, you can add your company's or your organization's name here.

If you already have created secret OpenPGP keys using the key management they will be displayed by the list at the bottom. In addition to the optional **URL** (network address), the **Name** column shows the key name. The **eMail** column shows the e-mail address of the key owner and the hexadecimal key identifier is displayed by the **Key Id** column. In order to generate signatures according to the OpenPGP standard please select the desired key entry in the list. If a password was deposited for the key this password will be automatically used for signing, the user is

prompted by a dialog otherwise just before the signing. The key URL may be configured (optional) to store the key address within Digester XML files. You can set a key URL by clicking the button showing the pen symbol, a key may be deleted with the red X button. Only valid URLs should be entered here since the information is used to download public keys. On top of the key list you can see the main features of the key management whcih may be called conveniently directly from here.

> The OpenPGP functions can be used only if the unlimited key strength has been activated. You can find detailed pieces of information about the legal background for the deployment of cryptography on our website. With the [Check Key Strength](#) functionality you can determine the currently supported key strength of your Java environment.

Tab XML

This tab provides settings in connection with the [Digester XML format](#). Besides the above mentioned input field to specify an XML file extension, you can set a comment for XML files (optional). This is stored in the header of each Digester XML file and can for example be used to transport copyright information or notes for the FTP server. The default format of digests in Digester XML files is the hexadecimal format, you may also choose Base64 encoding.

- **XML extension**: Extension for XML files (default: `.digest`)
- **XML comment**: Optional text for the XML file headers
- **XML digest format**: Message digest format (default: `hex`)
- **Include absolute paths**: Store absolute paths for the original files
- **Include URLs**: Store fixed network address or SSI pattern
- **Include OpenPGP signatures with**: Store signatures into XML files
- **Key ID**: Save signing key identifier
- **Key name**: Save signing key name
- **Key eMail**: Save signing key owner's eMail address
- **Key URL**: Save public signing key Internet address

With the **Include absolute paths** option, all digest values are stored along with the absolute file reference for the computer. Under Windows for example this could be the path `C:\Temp\franz.txt.` Even after having copied such XML files to another drive of the same computer, the software can correctly resolve such paths and properly find the files associated with the digests. However, Digester XML files get bigger when absolute paths are added.

The **Include URLs** input field may contain a fixed network address or a SSI pattern. Here you can specify a static URL pattern e.g. for SSI (*Server Side Includes*) which can be processed by web servers like the Apache HTTP server. Such URLs can be translated dynamically into the appropriate URLs for the original files on the server. If Digester XML files are being downloaded with the software, then the referenced files can be identified using the URL entries to be downloaded, checked, and processed in other ways.

If Digester XML files are being created and OpenPGP keys have been specified to calcuate signatures then the signature values may be added to the Digester XML files (**Include OpenPGP**

signatures with). Signatures are represented by separate XML elements which are stored on the level together with digests (see <u>DTD Digester XML</u>). The associated flags **Key ID**, **Key name**, **Key eMail**, and **Key URL** may be configured to include the key identifier, key name, key owner's eMail addresse, and a static network address to download the public key (all optional).

A special feature of the Digester software is the capability to create interim results - the so-called intermediate digests - in addition to the final digest values. While GNU/BSD files can only contain one single digest per original file, the use of Digester XML files allows to store an arbitrary number of intermediate digest values. Message digest files based on the GNU/BSD format may only contain the final digest value which is created when all bytes of the original files have been processed. The Digester software can now be configured to create interim results at defined positions which are then stored into the Digester XML files along with a position indication. The use of intermediate digests allows to quickly detect deviations of examinee files and original files since these often differ already in the first data blocks. In such cases the calculated intermediate digests for examinee files are different from the intermediate digests found in the digest files. Especially when checking files on remote servers, this effect may lead to significant time savings since less data needs to be transferred and processed.

- **No intermediates**: Switch this function off
- **Linear mode**: Extract intermediates after fixed block lengths
- **Create every**: Block length for linear mode like 10M=every 10 megabytes
- **Max digests**: Limit maximum number of intermediates (linear mode)
- **Exponential mode**: Extract intermediates after 2 kilobytes, 4 kilobytes, 8 kilobytes etc.
- **Starting at**: First position to start extraction (e.g. 8K=8192 bytes)
- **Max digests**: Limit maximum number of intermediates (exponential mode)

If you do not like intermediate digests be produced then select the **No intermediates** option. Select **Linear mode** to have intermediate digests be created at fixed intervals. The interval can be entered into the Create every input field where the default value is 10M, causing an intermediate digest being created every 10 megabytes. In order to limit the size of the Digester XML files even for large original files, you can set the total number of intermediate digests per original file to a maximum value (see the first **Max digests** input field).

The option **Exponential mode** allows the creation of intermediate digests beginning at a certain position (see the **Starting at** input field, default is 8K = 8 kilobytes). The next intermediate digests will be produced after the next 16 kilobytes, 32 kilobytes, etc. While in this mode a relatively large number of intermediate digests is produced for the first data blocks of the original files and less at the end. This allows to quickly detect deviations of examinee files and the original files since these often differ already in the first kilobytes of data. In order to limit thesize of the Digester XML files even for large original files, you can set the total number of intermediate digests per original file to a maximum (see the second **Max digests** input field).

Tab Refresh

In generate mode all message digests are calculated and files are created always. If only a few files on an FTP server for example have been added or modified this kind of behavior is not appropriate since too many calculations must be done unnecessarily. Here the refresh function of the Digester software can help. On the **Refresh** tab you can configure under which conditions existing digest files are recognized as being still correctly so that the calculations can be skipped.
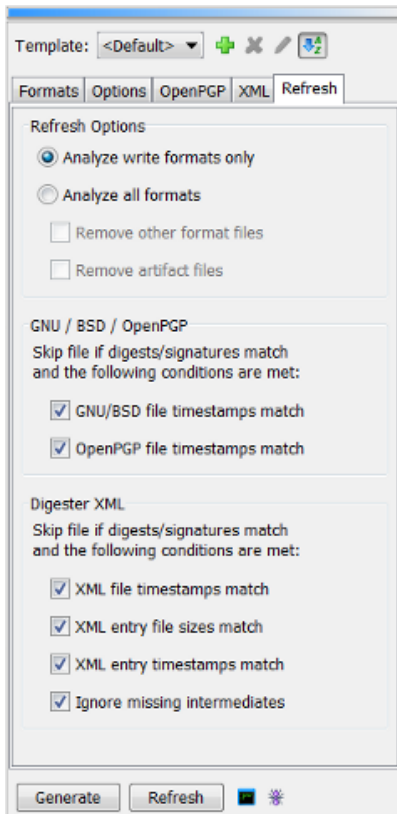
- **Analyze write formats only**: Ignore digest files in other formats
- **Analyze all formats**: Include all digest files for the analysis
- **Remove other format files**: Delete digest files in other formats
- **Remove artifact files**: Delete digest files without original files
- **GNU/BSD file timestamps match**: File timestamps must match original files
- **OpenPGP file timestamps match**: File timestamps must match original files
- **XML file timestamps match**: File timestamps must match original files (per target)
- **XML entry file sizes match**: Sizes in XML must match original files
- **XML entry timestamps match**: Timestamps in XML must match original files
- **Ignore missing intermediates**: OK also if intermediates are missing

For the refreshing, all detected digest/signature files are being used if the **Analyze all formats** option is selected. If the **Analyze write formats only** is selected instead then only those formats are being analyzed which are configured to be written and deemed relevant for the refreshing.

If all formats are being analyzed then it is possible to switch from GNU files over to XML files for example and let the software delete the previous GNU files programmatically. This way only those files are kept which are deemed relevant for the synchronization (write formats). Activate the **Remove other format files** option to achieve this. By setting the **Remove artifact files** option

so-called artifact files may be automatically removed. Artifacts are files with digests/signatures where no original files exist at all. This way useless files can be cleared automatically.

The section titled **GNU / BSD / OpenPGP** contains options to control the program behavior for such files. Such files are skipped (not calculated again) if digests match and if the file timestamps match (**GNU/BSD file timestamps match**). OpenPGP files are skipped if signatures match and if the file timestamps match (**OpenPGP file timestamps match**).

The section titled **Digester XML** contains options to control the program behavior for Digester XML files. Such files are skipped (not calculated again) if digests/signatures match and if the file timestamps match (**XML file timestamps match**). Such files are skipped if digests/signatures match and if the file sizes within the XML entries match (**XML entry file sizes match**). Such files are skipped if digests/signatures match and if the file timestamps within the XML entries match (**XML entry timestamps match**). Last but not least missing intermediates may be ignored with the option **Ignore missing intermediates**. Maybe intermediates have not been generated the last time so this can be compensated here as long as the overall digests match for the files.

Please keep in mind that the options **XML entry file sizes match**, **XML entry timestamps match**, and **Ignore missing intermediates** refer to the element contents within the XML files, not the XML files themselves. The XML files especially contain information about the original files at the time the XML files were created (sizes, timestamps) which may be evaluated here.

If file timestamps should be analyzed please select the option **Apply target file timestamp** as well (see [options](#) tab). The timestamps of the original files are then applied to the generated files so that this information can be used for the refresh functionality.

Start Generating

Once all the settings within the generate window are as desired, the calculation can be started by clicking on the **Generate** button at the bottom of the window. If the settings make sense and appropriate files and/or directories have been selected within the file system window, this button is enabled. The checking process is monitored by a special dialog where the current state is shown. Upon completion of the calculations this progress dialog shows some statistics for the operation and the generate results. Besides the time consumption of the calculations (top left corner) the dialog shows the remaining time (ETA = Estimated Time Access).

Please note, that it is possible to select directories within the file system window to be processed recursively including subdirectories, while this is not possible if you start generating from the [context menu](#). The **Refresh** button is disabled if the option **No files, just calculate** is selected. The refreshing functionality is executed the same way as the generating process.

When all digests have been calculated and the progress dialog was closed, the results are displayed by the [results window](#) which is automatically opened. The generate results are divided into two tabs **Files** and **File Digests** or **URLs** and **URL Digests**, if the generating has been started from the network file system window.

Command-line Arguments

In order to make generating from the command-line with `digestercli` as easy as possible the necessary arguments may be assembled by the Digester software. Please select the files/directories to be processed within the file system window in the first step. Then configure the generate/refresh options with the generate window as if the process would be started directly from here. By clicking the button with the console symbol a dialog is displayed showing the corresponding `digestercli` call. Either the generate or refresh mode has to be selected first if the button is clicked. This code can be easily copied into the clipboard to be used for shell or batch scripts. As you can see programming is not required here.

Ant Build Scripts

Like the command-line arguments also XML fragments for Ant build scripts may be generated using the button with the ant symbol. This code can be used for own shell or batch scripts. Please select the files/directories to be processed within the file system window in the first step. Then configure the generate/refresh options with the generate window as if the process would be started directly from here. By clicking the button a dialog is displayed showing the corresponding XML code. Either the generate or refresh mode has to be selected first if the button is clicked. This code can be easily copied into the clipboard to be inserted into a text editor for example. As you can see programming is not required here.

Additional Links

- File System Window
- Network File System Window
- Digester XML Format
- OpenPGP Signatures
- Key Management
- Check Key Strength
- JCE Unlimited Strength Jurisdiction Policy Files↗
- Regular Expressions
- Results Window
- Generate with digestercli and Ant

Sync Window

The functionality to synchronize files and directories with digests is provided by the docking window named **Sync**. For the default configuration of the Digester software, the button for this docking window is located on the right edge of the main window. Here you can see four tabs **Entries**, **Formats**, **Sync**, and **Options**.

> BE Please note that the synchronize functionality is not available in the Basic Edition and
> SE Standard Edition.
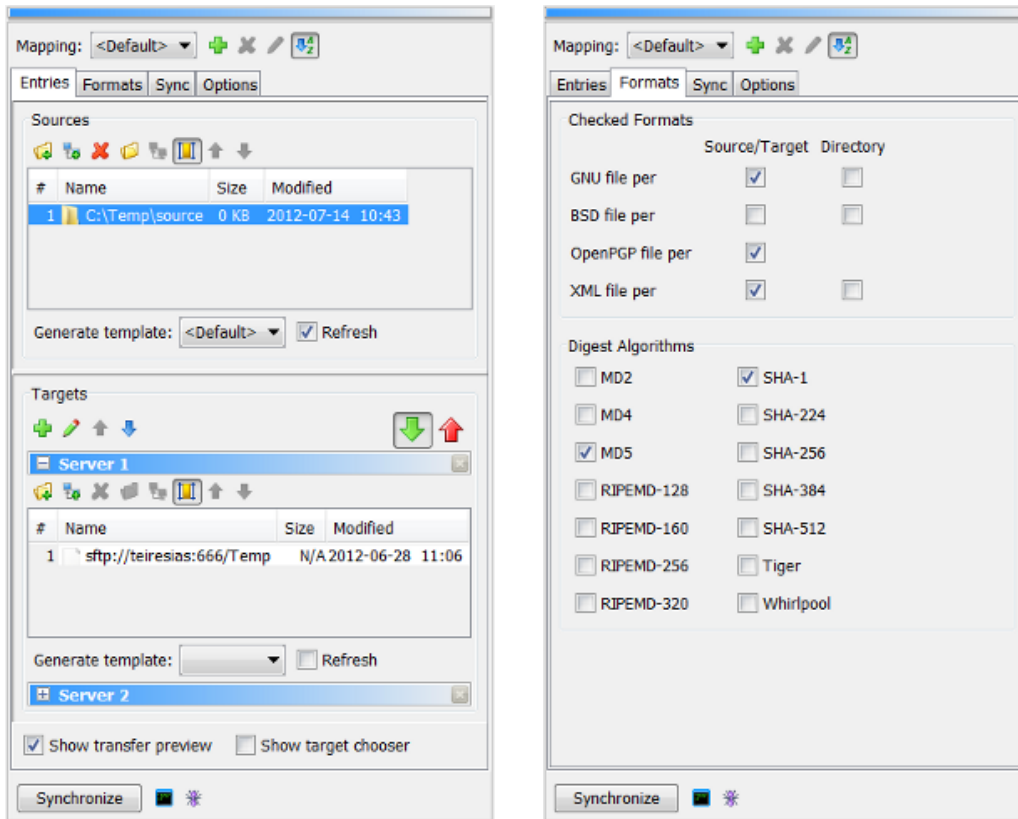
The requirements for the use of digests and signatures may be very different: for FTP servers often SHA and signature files are used, while the Digester XML format is appropriate for archiving backups. In order to support arbitrary environments appropriately, mappings may be defined. You can create a new mapping using the button with plus symbol on top of the sync window. You will be prompted to specify a unique mapping name. The current settings of the sync window are used to create the new mapping. You may adjust the settings at any time: please select the desired mapping and adjust settings as desired. The mappings are automatically saved if the program gets terminated and restored when restarted. Mappings can be deleted with the red X button after user commit and be renamed with the button showing the pen symbol. The mappings can be displayed in alphabetical order or in the order the items were created.

Tab Entries

When files/directories and possibly subdirectories are being synchronized files are transferred from a *source* to one or multiple *targets*. If the source entries must be transferred at all (new or modified) may be evaluated based on existing digests and signatures. Like the refresh functionality provided by the generate window also this function analyzes the candidate directories for existing digests/signatures. Some files may be recognized as being unmodified and will not be tranferred under certain coniditions which may be configured with the sync window. This saves time and bandwidth without the need to take care for each single file's state.

On top of the tab you can configure the sources. The easiest way here is to use the drag & drop functions of the file system window and network file system window. Please select the desired files/directories and drag the selection into the sources area or into the expanded targets area at

the bottom. With the two buttons showing the plus symbols files, directories, or URLs may be specified with a dialog to be used as sources or targets. With the red X button list entries may be deleted. The two buttons with the folder/network icons the selected list entry may be displayed by either the file system window or the network file system window.

The other buttons on top of the lists provide functions to set the columns widths automatically as well as to sort the list entries. Below the lists for sources and targets a **Generate template** can be specified (optional). Often it makes sense to update at least the source digests before the synchronization is started. The specified template of the generate window is then called before the sync. Set the **Refresh** flag here in order to optimize the digest/signature calculations.

For each source exactly one target entry of the same type must be specified for the synchronization to take place. The list entries must contain either files *or* directories. However, each source file must be mapped to a target file and each source directory must be mapped to a target directory. You may specify multiple target sets here in order to synchronize multiple FTP server directories with a local directory in one single step for example. The origin of the sources and targets is not limited here, the sources may be provided by a server, the targets may be local directories and so on. New target sets can be created using the plus symbol on top of the **Targets** area where you are prompted for a name. The target sets may be sorted manually. A **Generate template** can be selected for each target set (optional) to be executed before the sync. Target sets may be deleted using the red X button or from the context menu (right mouse button), the set sets can be renamed with the pen symbol button. One single target set may be expanded, this set becomes the active target set here. However, all configured target sets are being synchronized, also if the sets are not currently expanded.

Synchronization normally is done from source to target, which means that the entries on top are transferred to the targets at the bottom (the button with the green down arrow is selected). In some cases it may be necessary to transfer the entries in the opposite direction from targets to sources. Please click on the button with the red up arrow then. Here, the currently expanded (active) target set is transferred exclusively. The collapsed target sets are ignored for the reverse mode. If sources and targets should be flipped permanently please use the context menu (right mouse button) of the target sets. The entries are then moved from top to bottom and vice versa.

- **Show transfer preview**: Show the transfer list before the transfer starts
- **Show target chooser**: Show the list of targets before the sync for pre-selection

Tab Formats

As for the generate functionality you can choose which file formats to use for the synchronization with digests/signatures:

- **GNU file per source/target/directory**: Analyze files in GNU format
- **BSD file per source/target/directory**: Analyze files in BSD format
- **OpenPGP file per source/target**: Analyze files with OpenPGP signatures
- **XML file per source/target/directory**: Analyze files in Digester XML format

OpenPGP signature files are limited to the per-target creation, storing signatures for whole directories are not possible here. In addition to the formats the analzed algorithms may be specified on this tab. All options should be in harmony with the options of the used generate template since checking XML files would be useless if the template only creates GNU files for example.

Tab Sync

The synchronization process may be controlled in detail with the options found on the **Sync** and **Options** tabs. One of the most important sync features is the optimization where only the required sources should be transferred (new or modified).
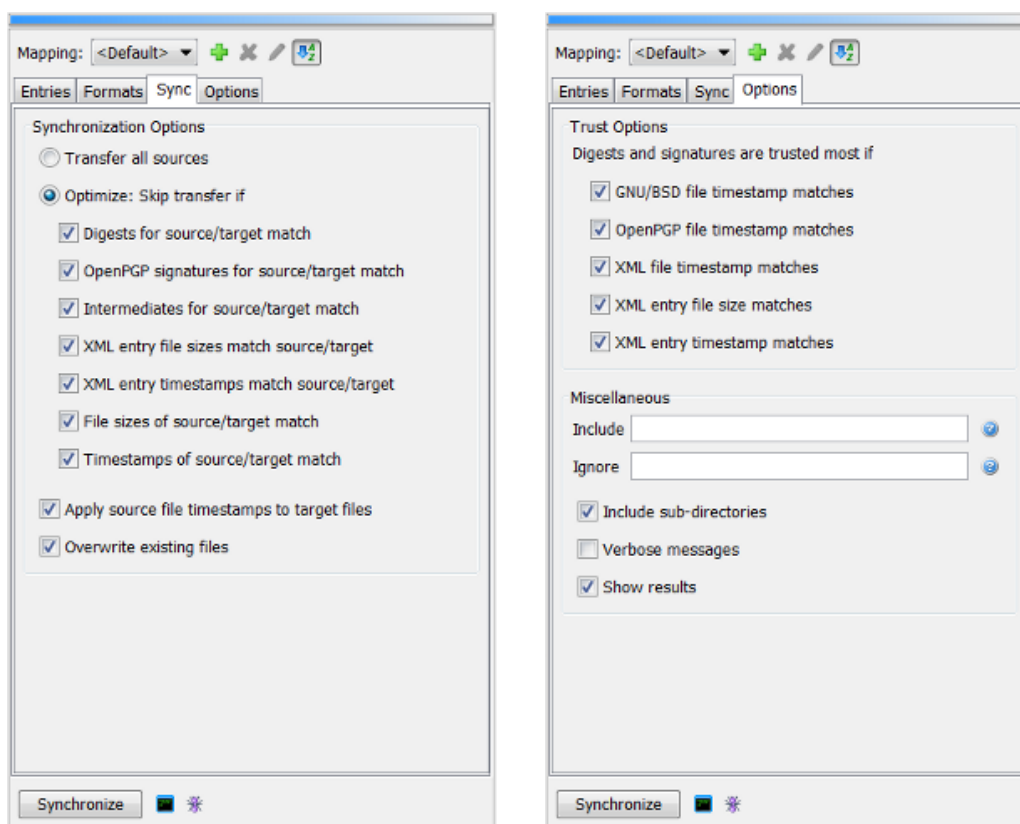
- **Transfer all sources**: Do not compare digests, all sources are transferred
- **Optimize: Skip transfer if**: Analyze digests/signatures
- **Digests for source/target match**: Digests must match, transferred otherwise
- **OpenPGP signatures for source/target match**: Signatures must match
- **Intermediates for source/target match**: Intermediates must match
- **XML entry file sizes match source/target**: XML files sizes must match
- **XML entry timestamps match source/target** : XML file timestamps must match
- **File sizes of source/target match**: File sizes must match, transferred otherwise
- **Timestamps of source/target match**: Timestamps must match, transferred otherwise
- **Apply source file timestamps to target files**: Timestamps are applied to targets
- **Overwrite existing files**: Existing target files are overwritten

If you want to make sure that sources and targets are equal then please choose the **Transfer all sources** option. In order to transfer only the relevant files/directories please choose the **Optimize: Skip transfer if** option. The next seven associated flags configure the criteria to decide whether a source should be transferred or not. If existing digest/signature files are analyzed at all may be controlled with the **Digests for source/target match** and **OpenPGP signatures for source/target match** options. In order to evaluate intermediate digests (XML only) select the **Intermediates for source/target match** option. This setting is recommended for Digester XML files, since deviations of source and target files may be detected possibly already after reading the first few blocks of data which allows for a faster comparison.

The **XML entry file sizes match source/target** and **XML entry timestamps match source/target** options are also only applicable for existing Digester XML files. The XML files especially contain information about the original files at the time the XML files were created (sizes, timestamps) which may be evaluated here to decide whether a source should be transferred or not.

In addition to the digests/signatures also simple characteristics of the sources/targets may be checked. Files of different sizes are never equal and should be transferred. Also the timestamps of sources and targets can be compared although this seems to be an unsafe way at least in the case that the computer with the sources has a different time base than the computer with the targets. If the **Timestamps of source/target match** option is selected also the **Apply source file timestamps to target files** should be activated in order to be able to compare timestamps at all.

By setting the **Overwrite existing files** option, the software is instructed to overwrite existing files. This option applies to the target files as well as to the files with digest/signatures on the respective target system. Such files are also transferred to be able to use the contained information for the next synchronization.

> ⚠️ Overwritten digest/signature files cannot be restored programmatically, so please backup your data first. A dialog is displayed if this option is selected.

Tab Options

In the **Trust Options** area of this tab you' ll find the options to define the conditions when existing digests and signatures are trusted most and if they should be used to compare sources and digests. The main goal of these options is to identify the best digests/signatures for the comparison. If both GNU and XML files are available then the data contained in the XML files should be preferred. An existing digest/signature is used (regarded as being the best value) if most defined conditions can be met. A digest can be trusted most if all XML criteria can be fulfilled (file timestamps, XML file sizes, and XML timestamps). The first three timestamp options are only applicable for those files which have been created on a per-target basis. Files for whole directories are not unfluenced by these options.

- **GNU/BSD file timestamp matches**: Timestamps of GNU/BSD and original file match
- **OpenPGP file timestamp matches**: Timestamps of signature and original file match
- **XML file timestamp matches**: Timestamps of XML and original file match

- **XML entry file size matches**: File size in XML matched original file
- **XML entry timestamp matches** : File timestamp in XML matched original file

In order to analyze only certain files you may enter a regular expression in the **Include** input field. If the field is empty then all recognized files will be processed by default. If certain files shall *not* be processed then you can specify a regular expression with the **Ignore** input field. These options are particularly useful, if directories or entire directory trees including subdirectories are synchronized. You can activate recursive scanning of subdirectories by selecting the **Include subdirectories** option. In order to get detailed messages on the generation process please select the **Verbose messages** option. If the sync results shall not be displayed after completion the **Show results** option should be disabled.

Start Synchronizing

Once all the settings within the sync window are as desired, the calculation can be started by clicking on the **Synchronize** button at the bottom of the window. If the settings make sense and appropriate files and/or directories have been specified as sources and targets, this button is enabled. The sync process is monitored by a special dialog where the current state is shown. Upon completion of the synchronization this progress dialog shows some statistics for the operation and the sync results. Besides the time consumption of the synchronization (top left corner) the dialog shows the remaining time (ETA = Estimated Time Access).

If the synchronization has been completed and the progress dialog was closed, the results are displayed by the results window which is automatically opened. For each of the processed target sets a separate tab showing the target set name is displayed. The tabs show the sources and targets including file sizes and comments.

Command-line Arguments

In order to make synchronizing from the command-line with `digestercli` as easy as possible the necessary arguments may be assembled by the Digester software. Please configure the sync options with the sync window as if the process would be started directly from here. By clicking the button with the console symbol a dialog is displayed showing the corresponding `digestercli` call. This code can be easily copied into the clipboard to be used for shell or batch scripts. As you can see programming is not required here.

Ant Build Scripts

Like the command-line arguments also XML fragments for Ant build scripts may be generated using the button with the ant symbol. This code can be used for own shell or batch scripts. Please configure the sync options with the sync window as if the process would be started directly from here. By clicking the button a dialog is displayed showing the corresponding XML code. This code can be easily copied into the clipboard to be inserted into a text editor for example. As you can see programming is not required here.

Additional Links

- [Generate Window](#)
- [File System Window](#)
- [Network File System Window](#)
- [Digester XML Format](#)
- [OpenPGP Signatures](#)
- [Regular Expressions](#)
- [Results Window](#)
- [Synchronizing with digestercli and Ant](#)

Search Window

This window allows for searching files/directories based on the name (not the content). The search window is usually located on the right side of the screen and is opened by the [file system window](#) or [network file system window](#) automatically when the search function is activated (also the search pettern is specified there).

If this behavior is not desired, you can disable the **Automatic** flag in the context menu (right mouse button) of the docking window button **Search**.
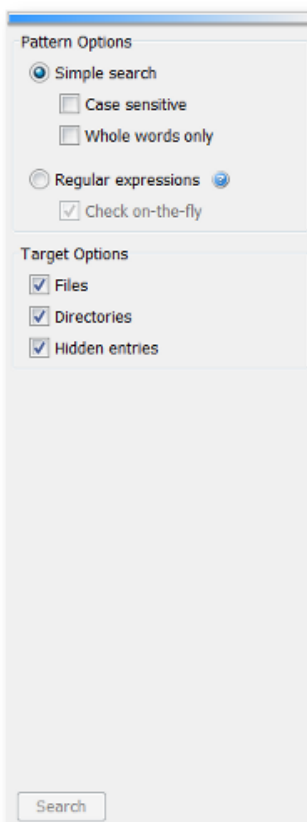
- **Simple search**: Direct pattern search (wildcards like * are not supported)

- **Case sensitive**: The case of the pattern matters (lower/upper)
- **Whole words only**: If a single word is matched then the file/directory is matched
- **Regular expressions**: Flexible search e.g. for files starting with numbers
- **Check on-the-fly**: Errors in regular expressions are detected early
- **Files**: Search for regular files
- **Directories**: Search for regular directories
- **Hidden entries**: Search for hidden entries, depends on the operating system

The search window lets you configure the search in many ways. You can choose the simple search here where all files/directories are searched based on the given search pattern. If you like the search being case sensitive, please select the **Case sensitive** option for the simple search. The option **Whole words only** configures the search engine to search for words within the entry names. While in this mode, only one single part of the entry names needs to match the pattern to be shown by the hit list. Thus, you can search for OR-combined sub-patterns here.

In some cases, the simple search is inadequate, for example if all file names shall be listed that start with a digit. In these cases, you can use regular expressions by selecting the **Regular expressions** option, where the search pattern specified with the input field of the file system window must be a valid regular expression. To detect errors as early as possible, the software can be configured to check the pattern input already while you are typing. Please activate the **Check on-the-fly** option to do so. If you want to search for files and/or directories and/or hidden entries then you can select the options **Files**, **Directories**, and **Hidden entries** accordingly.

Start Searching

The search can also be started directly from the search window once all settings are appropriate. Click the **Search** button at the bottom of the window. If the search is not possible for example due to a faulty regular expression, this button will be disabled. If the auto-show/auto-hide feature is enabled when the search is started, then the search window is hidden automatically and a progress dialog is opened which informs you about the current state of the search operation. This dialog can also be configured to perform the search in the background so that you can continue working with the application.

After completing the search, the results are presented by the search results window.

Additional Links

- File System Window
- Network File System Window
- Regular Expressions
- Search Results Window

Output Window

The output window of the Digester software shows the log messages generated by the running program which are also written to the log file in the configuration directory. The window is divided with tabs, currently only the mentioned log file `digester.log` is managed. The log messages are interesting if you plan to contact the Leisenfels support since all relevant pieces of information concerning the running application are documented here precisely. For the daily work with the software this output is usually not important.
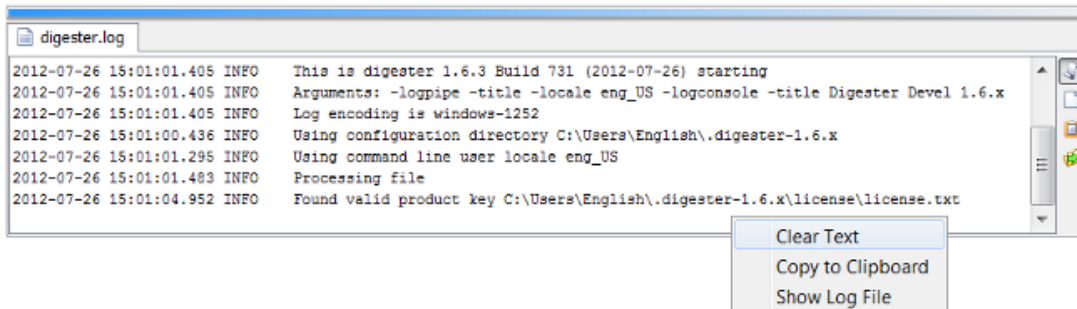
- **Latest Entry**: Auto-scroll to end of file (latest log entries)
- **Clear Text**: Clear text area (not log file)
- **Copy to Clipboard**: Copy contents to clipboard for other programs
- **Show Log File**: Show log file in file system window

The tool bar at the right side of this window allows you to control the display. The scroll symbol (**Latest Entry**) can be selected so that the visible text area is automatically scrolled to the last row. So you can always see the latest log messages. You have to turn off this feature if you want to see the log messages at the beginning of the log file. This setting is stored when the application is terminated and restored the next time it is started.

The symbol with the empty sheet (**Clear Text**) clears the text in the text area but does not touch the log file contents. Use this function if a lot of log messages have been generated by the program and the scrolling becomes too imprecise. Click the clipboard icon (**Copy to Clipboard**) to copy the complete text into the clipboard of the operating system. This can be useful when forwarding log files to the Leisenfels support without the log file being attached to an e-mail. The button with the green arrow (**Show Log File**) shows the log file in the file system window.

As an alternative to the buttons, the corresponding functions are also provided by the context menus which you can activate with the right mouse button on the text area (see figure).

Additional Links

- File System Window

Task Window

The task window is used by the Digester software to list all tasks that have been executed in the background, by the progress dialog, or that are currently running. The figure shows a completed checking operation, the successful search for .exe files, and a currently running generate operation.

If a task has not yet completed, its status is usually shown as a progress indicator with the remaining time shown by the **ETA** column (Estimated Time Access, here 1 minute and 15 seconds). The **Message** column presents the last message generated by the task, while the final status is indicated by a special icon in front of the message in most cases. The final status will also be displayed as text by the **Status** column. The **Duration** column contains the time period as hours:minutes:seconds that the task execution took so far or that the task execution took finally. The **Prio** column shows a number to indicate which priority the task has been executed with.
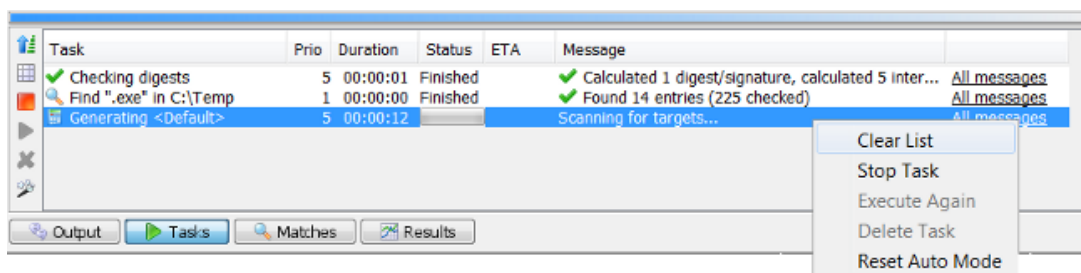
- **Order**: Show latest task on top, as started otherwise
- **Clear List**: Delete all completed tasks from the list
- **Stop Task**: Cancel the selected tasks if possible
- **Execute Again**: Restart the selected tasks if supported
- **Delete Task**: Delete the selected tasks from the list
- **Reset Auto Mode**: Reset auto backgrounding mode (ask with dialog)

The left tool bar can be used to manage the list of tasks. The button with the blue arrow (**Order**) can be clicked to show the latest task on top of the list or in the order as they were started otherwise. The activated magic wand icon (**Reset Auto Mode**) shows the window automatically whenever a new task is passed to be managed. This may be done easily from the progress dialog by clicking the **Background** button. The icon with the empty table (**Clear List**) can be used to delete the entire list of tasks with one single click. This feature is available only if all the tasks have been completed. If the docking window is configured to be **Automatic**, then removing the entire list items causes the task window to be hidden, too.

The gear icon with the red square (**Stop Task**) is enabled when a running task is selected which can be terminated. A click on this button causes the task being immediately canceled. This is useful for example to abort a long running search operation. The task is assigned the **Canceled** status here. If you like a task to be executed again (if supported by the task), then please select the task in the list and click the gear icon with the green triangle (**Execute Again**). The task is then restarted based on the previous settings. This is also possible for canceled tasks, unless the task declares itself not to be restartable. If you like to remove tasks from the list, please use the gear icon with the red X (**Delete Task**).

As an alternative to the buttons, the corresponding functions are also provided by the context menus which you can activate with the right mouse button on the selected list entry (see figure).

Additional Links

- Search Window
- File System Window
- Network File System Window

Search Results Window

After the search has been initiated from the search window and the results are available, these are listed by the search results window (see illustration). Here all hits are listed while the list is sorted alphabetically by default. You can select the hit list entries as in the file system window and also activate the context menu (right mouse button). This menu usually provides all features of the file system window. Especially you can also start the special functions for digest calculation or file checking from here. So you can search for specific digest files and then check these files against examinees. The details for the current selection such as the selection type (file or directory) as well as the full path and the selection size (in bytes) is displayed by the status bar at the bottom of the docking window.

The search results window is structured with tabs with each tab assigned to a search operation. This way, virtually any number of operations can be displayed here without the need for multiple result windows. The buttons on the left side allow to automatically show/hide the window like the search window (magic wand symbol). The icon with the empty window can be used to close all displayed result tabs. This function is useful if you like to remove all search results with one single click. If the docking window is configured to be **Automatic** then the docking window is also automatically hidden.
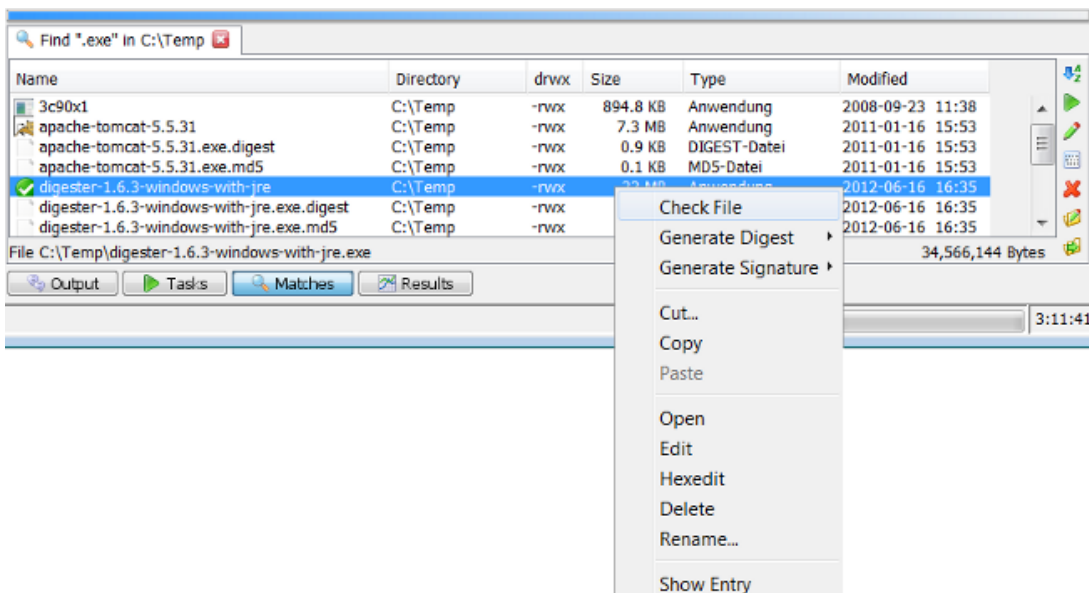
- **Sorting**: Sort matches alphabetically or in order of finding
- **Open**: Opens the selected files with program or operating system
- **Edit**: Starts editing the selection with program or operating system
- **Hexedit**: Opens the integrated hex editor for the selected files
- **Delete**: Deletes the selected entries, user is prompted to commit
- **Rename**: Gives the selected entry a new name
- **Show Entry**: Show entry in <u>file system window</u> or <u>network file system window</u>

The right tool bar contains various functions to control the display of the hit list. With the AZ icon (**Sorting**) you can either sort the list alphabetically (default) or show the items in the order that the files have been found.

The button with the play icon (**Open**) lets the program or operating system open this entry. This function is only available for files, this button is deactivated for directories. A click on this icon starts the standard programs associated with the selected file extension or one of the integrated editors. Please note, however, that this is strongly dependent on the operating system and there may be no response at all. This is also true for the next button with the editor icon (**Edit**) which can be used to let the program or operating system start the standard editor for the selected file. Under Windows operating systems for example the program WordPad is opened if this is the default editor for the selected file.

The Digester software supports opening and editing some specific file types (e.g. `.jpg`, `.png`, `.gif` and `.xml`). Such files may be directly displayed by the document window. For all other file types the action is delegated to the underlying operating system. For the file types supported by Digester you have the choice to open the files either by the application or the operating system (see <u>preferences</u> in section "Main Menu and Components"). The next button with the dots and strokes (**Hexedit**) opens a file with the integrated hex editor of the Digester software. This function may be called for any file as long as the program is able to access it.

The icon with the red X (**Delete**) can be used to delete the selected files and directories including subdirectories. A click on the icon with the pen (**Rename**) allows you to rename the selected entry which may be both files or directories. Similar to the creation of a directory, a special dialog is displayed that prompts you to enter the new name. The button with the green arrow (**Show Entry**) shows the selected hit list entry in the file system window. This symbol varies a little if the search has been started from the network file system window.

> ⚠ Directories and files deleted with the delete function cannot be restored programmatically, so please backup your data first. Before entries are deleted a security dialog is displayed which draws attention to this fact and where the user explicitly has to confirm the deletion.

As an alternative to the buttons, the corresponding functions for the search results window are also provided by the context menus which you can activate with the right mouse button on the entries of the hit list. The context menu items are almost identical to the items provided by the file system window extended by special items such as **Show Entry**. The context menu in the figure shows well-known items such as **Check File** and **Generate Digest** for the specific functions of the other docking windows.

Additional Links

- [Search Window](#)
- [File System Window](#)
- [Network File System Window](#)

Results Window

The results window presents the results generated by tasks. For search results, the special [search results window](#) is available since it provides some special functions related to the hit list. A wide variety of task results exist. The figure for example shows some digest checking results generated by the [check window](#). The checking results are divided into two tabs **Files** and **File Digests**. The results for checked URLs are presented slightly different with the two tabs **URLs** and **URL Digests**.
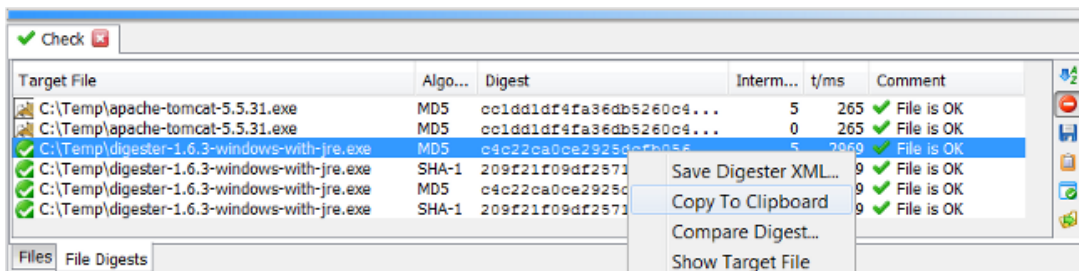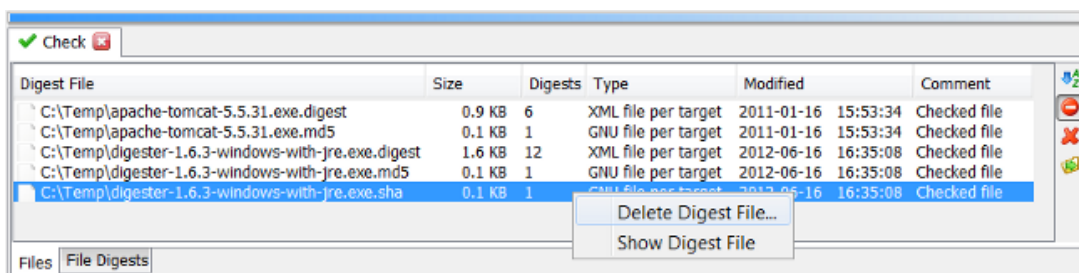
The **File Digests** tab shows the calculated digests and the status for the particular examinee file, the **Files** tab shows an overview of the processed digest files. These tabs are also created by the [generate window](#) after generating digests/signatures.

The results window is structured with tabs with each tab assigned to a task operation. This way, virtually any number of operations can be displayed here without the need for multiple result windows. The icon with the red X can be used to close a single result tab. Result tabs are rejected when the program is closed and cannot be restored.

Tab Files

- **Sorting**: Sort digest files alphabetically or in order of finding

- **Ignored Digest Files**: Show or hide the ignored files
- **Delete Digest File**: Deletes the selected entries, user is prompted to commit
- **Show Digest File**: Show file in <u>file system window</u> or <u>network file system window</u>

The right tool bar on the **Files** tab contains various functions to control the display of the result list. With the AZ icon (**Sorting**) you can either sort the list alphabetically (default) or show the items in the order that the files have been found. Digest files may have been ignored, if for example a regular expression has been specified for the check window. Such files may be shown or hidden here, please click on the icon with the stop symbol (**Ignored Digest Files**).

The icon with the red X (**Delete Digest File**) can be used to delete the selected files. The button with the green arrow (**Show Digest File**) shows the selected entry in the file system window. This symbol varies a little if the operation has been started from the network file system window.

⚠ Directories and files deleted with the delete function cannot be restored programmatically, so please backup your data first. Before entries are deleted a security dialog is displayed which draws attention to this fact and where the user explicitly has to confirm the deletion.

As an alternative to the buttons, the corresponding functions for the results window are also provided by the context menus which you can activate with the right mouse button on the entries of the list.

Tab File Digests

- **Sorting**: Sort target files alphabetically or in order of finding
- **Ignored Target Files**: Show or hide the ignored files
- **Save Digester XML**: Save the selection together as Digester XML
- **Copy to Clipboard**: Make digest available for other programs
- **Compare Digest**: Compare selected digest directly or with clipboard content
- **Show Target File**: Show file in <u>file system window</u> or <u>network file system window</u>

The right tool bar on the **File Digests** tab contains various functions to control the display of the result list (see illustration). With the AZ icon (**Sorting**) you can either sort the list alphabetically (default) or show the items in the order that the files have been found. Target files may have been ignored, if for example a regular expression has been specified for the check window. Such files may be shown or hidden here, please click on the icon with the stop symbol (**Ignored Target Files**).

Here, you may save the generated digests and signatures into Digester XML files also when the calculations have already been completed. Please select the desired digests on the tab and click the icon with the diskette symbol (**Save Digester XML**). A dialog appears which prompts you to choose a target file for saving. The original file references will be automatically stored correctly within the Digester XML files. This functionality may be also called if digests have been calculated for whole drives or file system trees which shall be stored into one single file (e.g. for backups).

You can click the clipboard icon (**Copy to Clipboard**) to load a selected digest value into the operating system clipboard to be shared with other applications. The next button with the flag

Compare Digest) opens a dialog where you can compare the selected digest directly with a typed value which may also be inserted from the clipboard. The button with the green arrow (**Show Target File**) shows the selected target file in the file system window. This symbol varies a little if the operation has been started from the network file system window.

As an alternative to the buttons also the results window provides the corresponding functions within the context menu (right mouse button) on a selected message digest.

Additional Links

- Search Results Window
- Check Window
- Generate Window
- File System Window
- Network File System Window

File Menu

- Check Files..
- Generate Wizard...
- Sync Wizard...
- Recently Opened
- Edit Files...
- Edit Network Files...
- Save
- Save As...
- Save As Network File...
- Preferences
- Restart
- Exit

Check Files

Opens a file selection dialog that allows you to define one or more digest files for examination. Possible file types are those types defined by the check window, which you can choose with the **Selection** component. Also original files can be selected here (like `tomcat.zip`), the existing digest/signature files such as `tomcat.zip.md5` are automatically detected based on the configured extensions and then loaded.

The checking process is represented by a special dialog where the current state can be monitored. After completion this progress dialog shows some statistics about the operations performed as well as the checking results. In addition to the elapsed period of time on the left, the anticipated time is being displayed here.

After the checking has been completed and the progress dialog was closed, the results are automatically displayed by the resultswindow. The checking results are divided into the two tabs **Files** and **File Digests** or **URLs** and **URL Digests**, if the checking has been called from the network file system window.

This function can also be accessed from the tool bar using a button.

Generate Wizard

The generate wizard in the first step prompts you to select original files to be calculate digests for. By holding down the control key multiple files can be selected at once. If you accidentally selected digest or signature files while the option **Ignore digest/signature files** is activated in the generate window there will be no digest files created (like MD5 for MD5 file).

In the second step you can select the message digest algorithms to be used for calculation. In the third step you can configure the mode of operation:

- **No files, just calculate**: Do not create files, just do the calculations
- **Create files**: Create files in the desired formats

These are the same settings of the generate window. If files are being generated, then you can control the exact mode in the next step. One option is the file format, where you can generally select the GNU format, the BSD format, as well as the Digester XML format. You can either choose between the GNU or BSD format because these formats use the same file names. The XML format can be generated always in addition to the GNU/BSD formats, because here a separately configurable file extension is used.

- **GNU file per target/directory**: Files in GNU format per target or directory
- **BSD file per target/directory**: Files in BSD format per target or directory
- **XML file per target/directory**: Digester XML files per target or directory
- **Overwrite existing files**: Overwrite existing digest/signature files
- **Apply target file timestamp**: Set original file timestamps for the created files

If you would like to create GNU files, you can now specify whether a GNU file per target (original file) shall be created or if all the digests for the directory should be saved into one single GNU file. You can also use both methods in parallel. While the GNU files per target carry the name of the original files plus the extension for each of the used algorithms, the GNU files per directory are named based on the contents of the **Summary file** input field (defaults to `summary`). While the GNU files per target each contains only one single digest per algorithm, the GNU files per directory may contain multiple digests per algorithm. A typical GNU file per target for the original file `franz.txt` and the MD5 algorithm is named `franz.txt.md5` for example, a typical GNU file per directory by defaul is named `summary.md5`.

The settings for BSD files correspond to those for GNU files, except that the format of the generated files is slightly different.

If you want to create Digester XML files that contains any number of digests for any algorithm constellation including intermediate digests, please select the option **XML file per target** or **XML file per directory** or both. While the XML files per target carry the name of the original files plus the extension in the **XML extension** field (see <u>XML tab</u>, defaults to `.digest`), the XML files per directory are named based on the contents of the **Summary file** input field (defaults to `summary`). While the XML files per target each contains only one single digest per algorithm, the XML files per directory may contain multiple digests per algorithm. A typical XML file per target for the original file `franz.txt` is named `franz.txt.digest` for example, a typical XML file per directory by defaul is named `summary.digest`.

By setting the option **Overwrite existing files** you allow the Digester software to overwrite existing files. It can be useful to apply the original file timestamps to the created digest files (only possible for digest files per target). Then please set the **Apply target file timestamp** as well.

> ⚠️ You will not be able to restore overwritten digest/signature files with the program, please backup your data first.

If you selected the **No files, just calculate** option or if you like to create Digester XML files then an additional step is shown by the wizard where you are prompted to configure the intermediate digests:

- **No intermediates**: Switch this function off
- **Linear mode**: Extract intermediates after fixed block lengths
- **Create every**: Block length for linear mode like 10M=every 10 megabytes
- **Max digests**: Limit maximum number of intermediates (linear mode)
- **Exponential mode**: Extract intermediates after 2 kilobytes, 4 kilobytes, 8 kilobytes etc.
- **Starting at**: First position to start extraction (e.g. 8K=8192 bytes)
- **Max digests**: Limit maximum number of intermediates (exponential mode)

If you like no intermediate digests being created please select the **No intermediates** option. Select the **Linear mode** in order to extract intermediates in fixed data intervals. The used clock length can be specified with the value in the **Create every** input field which defaults to 10M where an intermdiate digest is extracted every 10 megabytes. In order to keep the Digester XML files as small as possible even for bigger original files you can limit the total number of intermediates per original file (input field **Max digests**).

The option **Exponential mode** allows the extraction of intermediate digests starting at a definable data position (field **Starting at**, defauls to 8K = 8 kilobytes). The next intermediates are extracted after 16 kilobytes, then 32 kilobytes and so on. While in this mode the software extracts a relatively large number of intermediates for the first blocks of an original file and less to the end of the file. This technique makes it possible to detect deviations of original files and message digests very early after processing the first data blocks of a file. In order to keep the Digester XML files as small as possible even for bigger original files you can limit the total number of intermediates per original file (second input field **Max digests**).

Now all settings for the digest generation are as desired, so that the calculation may be started with a click on the **Finish** button.

This wizard can also be started from the <u>tool bar</u> using a button. The current settings are restored if you start the wizard the next time.

Sync Wizard

This wizard can be used to create a new mapping for the <u>sync window</u>. This mapping can then be used permanently to synchronize local and remote files and directories. The steps are optimized the way that you can get the new mapping with only a few necessary settings.

In the first step, the wizard will prompt you to select a set of source entries (files or directories). In addition to local files (**FileSystem** tab) you may choose to use network resources (**Network Files** tab). You can select either local or remote files/folders here. The source entries may be expanded or modified later with the sync window.

In the second step, you can select target entries the same way. This wizard supports to setup up exactly one target set, additional target sets can be set up later with the sync window. If entries from a network file system are being used, the corresponding connection may first be established. You can use the **Network Files** tab to do this where you can find several settings. For a synchronization to take place, for each source entry exactly one target entry of the same type must be configured (for each of the targets sets). The list rows of the sync window therefore must contain either files or directories. A source file must be synchronized with a target file, a source directory must be synchronized with a target directory.

In the next step to setup the generate options please enter a unique name for the newly created mapping to be used by the sync window. Often is is useful to update the source digests before the synchronization. The respective generate template of the generate window is then executed before the sync. Alternatively you can enable the **Refresh** option in order to optimize the calculation of digests and signatures. Setting a generate template is optional. If another template is selected then the template settings are directly activated within the wizard window so that a minimum of input is required. This also ensures that the new mapping is created with a consistent configuration.

As for the generate function also here you can choose from the list of file formats to be used for the synchronization of existing digests/signatures:

- **GNU file per source/target/directory**: Analyze files in GNU format
- **BSD file per source/target/directory**: Analyze files in BSD format
- **XML file per source/target**: Analyze OpenPGP signature files
- **XML file per source/target/directory**: Analyze Digester XML files

OpenPGP files can only be created per target and not on a directory basis. All settings should be consistent with the settings of the configured generate template since checking XML files would be useless for example if the template creates only GNU files. If you like subdirectories be analyzed as well please set the **Include subdirectories** option. The option **Intermediate digests** can be enabled to generate intermediate digests for <u>Digester XML files</u>. This option is enabled only if XML files are generated.

In the last step of the wizard the digest options can be adjusted. Besides the algorithms MD5, SHA-1 etc. you can select the (secret) OpenPGPP keys here to use for signature creation. Depending on the previous selections not all components may be enabled here.

If OpenPGP secret keys have been already configured using the <u>key management</u>, they will appear in the list below. Besides the optionally configured **URL** (network address), here appears the **Name** as well as some pieces of information like the **eMail** and **Key Id**. In order to generate signatures based on the OpenPGP standard, please simply select the desired key entry in the list. If a passphrase is available for the key, it will be automatically used for signing, otherwise the passphrase will be requested shortly before signing with a dialog. The key URL can optionally be stored for each key in order to be added to Digester XML files. A key URL can be specified by clicking on the pen icon button and deleted again with the red X button. URLs should be existing Internet addresses since they are used for automatic downloads of public keys. On top of the key list you can see the most important functions of the key management which can be started comfortably from here.

> (i)   The OpenPGP functions can be used only if the unlimited key strength has been activated. You can find detailed pieces of information about the legal background for the deployment of cryptography on our website. With the <u>Check Key Strength</u> functionality you can determine the currently supported key strength of your Java environment.

This wizard can also be started from the <u>tool bar</u> using a button. After the wizard has been closed using the **Finish** button the <u>sync window</u> shows the new mapping. You can adjust your wizard settings later using the window, for example you can add additional targets sets to be synchronized.

Recently Opened

This submenu shows the files and network files which have been opened by the <u>document window</u> for viewing or editing purposes. Here, a maximum of 10 entries are being displayed. You can open these files again by selecting the desired submenu entry. In order to clear the list from time to time or due to data security concerns please select the last entry **Reset**.

Edit Files

Opens the file chooser dialog where you can select one or multiple files to be viewed or edited by the <u>document window</u>. Currently the following file types are supported by this window which means viewing or editing is possible for these types:

- **Graphics files** in the formats `*.bmp`, `*.jpg`, `*.wbmp`, `*.png`, `*.gif` (viewing only)
- **Text files** `*.txt`, `*.log`, `*.asc`, `*.sig` (viewing/editing)
- **XML files** `*.xml` (viewing/editing)
- **Java properties** `*.conf`, `*.properties` (viewing/editing)

For these file types arbitrary extensions may be configured in the <u>preferences</u>, an appropriate editor is then opened automatically. In the <u>file system window</u> it is possible to open arbitrary file

formats with the integrated hex editor. Please select the **Hexedit** context menu (right mouse button).

This function can also be called with the shortcut **Ctrl+E** (Windows) or **Cmd+E** (Mac).

Edit Network Files

Opens the network file chooser dialog where you can select one or multiple files to be viewed or edited by the document window. Currently the following file types are supported by this window which means viewing or editing is possible for these types:

- **Graphics files** in the formats `*.bmp`, `*.jpg`, `*.wbmp`, `*.png`, `*.gif` (viewing only)
- **Text files** `*.txt`, `*.log`, `*.asc`, `*.sig` (viewing/editing)
- **XML files** `*.xml` (viewing/editing)
- **Java properties** `*.conf`, `*.properties` (viewing/editing)

For these file types arbitrary extensions may be configured in the preferences, an appropriate editor is then opened automatically. In the network file system window it is possible to open arbitrary file formats with the integrated hex editor. Please select the **Hexedit** context menu (right mouse button).

Save

The currently displayed file or network file within the documentwindow can be saved with this function, if the content has been changed. This is always indicated by an asterisk (*) after the file name on the tab. If a file name is not known right now (e.g. after inserting content from the clipboard), then a file selection dialog appears to be used for storage.

This function can also be called with the shortcut **Ctrl+S** (Windows) or **Cmd+S** (Mac) as well as from the tool bar using a button.

Save As

For contents of the documentwindow to be saved into local files, please use the **SaveAs** function. A file chooser dialog is displayed to specify a target file to save the data. If the file already exists and the content has not already been saved into this file, a security dialog appears, where you can cancel the saving.

Save As Network File

For contents of the documentwindow to be saved into network files, please use the **SaveAs Network File** function. A file chooser dialog is displayed to specify a target file to save the data. If the file already exists and the content has not already been saved into this file, a security dialog appears, where you can cancel the saving.

If network files are being accessed the respective network connection must have been established. In order to do so you can use the functions in the header area of the dialog. You can create new connections with the Edit Connections menu.

Preferences

The program settings are managed with the Preferences menu. All file operations around the preferences are located in the **Preferences** menu of the **File** menu together with the other file functions.

- **Save Preferences:** Save preferences into configuration file(s)
- **Import Preferences:** Load previously exported preferences from a file
- **Export Preferences:** Save current preferences into a file of choice

With **Save Preferences** the current preferences of the program may be saved explicitely e.g. after changes were applied. Each time the program is terminated the preferences are automatically saved in the configuration file of the application. This file is located in the PC user's home directory in `.digester-1.6.x/config/digester.conf`. Depending on the program this directory contains additional configuration files like OpenPGP keystores which are saved automatically as well when the program terminates. If the program has to be stopped using the Windows Task Manager then the preferences are not *saved*, so that your settings at the time of interruption may be lost. After bigger changes to the configuration it makes sense to save the preferences explicitely once with this menu function.

This function can also be called with the shortcut **Ctrl+Alt+S** (Windows) or **Cmd+Alt+S** (Mac).

With **Import Preferences** you can choose a file to load preferences which have stored using the **Export Preferences** functionality previously. After these two functions have been called a file chooser dialog is displayed where to select a configuration archive in ZIP format. If the file does already exist a security dialog is shown where the action can be cancelled. Please keep in mind, that you can only import such configuration archives that have been exported with exactly the same program version. If this is not possible the program shows a special message.

> The export functionality should be used regularly to create backups of the program preferences. The exported archives should then be saved on secure media and carried to secure locations. Import/export could also be used to transfer the preferences from one PC to another or to restore the preferences for example if the operating system has been re-installed.

Restart

The program can be properly terminated and restarted with the **Restart** menu. This is possible for both the installer versions as well as the Web Start versions of the software. When restarted the current program state is being analyzed. Dialogs may be created e.g. if tasks are still being executed and where the appropriate steps can be configured. Finally the program shows a dialog

where you must confirm the restart of the program (not the PC!). This dialog may be opted out by setting the **Do not show again** flag so that the next restart can be executed directly without being asked.

Program restarts may be initiated by the program itself e.g. after the memory size has been adjusted.

Exit

The program should be properly terminated after use, since certain data and preferences are saved for the next run. If the program is stopped by the Windows Task Manager for example your manuall applied settings may be lost and must be applied again the next time. When restarted the current program state is being analyzed. Dialogs may be created e.g. if tasks are still being executed and where the appropriate steps can be configured.

This function can also be called with the shortcut **Ctrl+Q** (Windows) or **Cmd+Q** (Mac).

Additional Links

- Check Window
- Results Window
- Generate Window
- Sync Window
- Digester XML Files
- Document Window
- File System Window
- Network File System Window
- Preferences

Edit Menu

- **Cut (Ctrl+X/Cmd+X):** Delete files/directories, list entries etc.
- **Copy (Ctrl+C/Cmd+C):** Copy text, files, or network files in clipboard
- **Paste (Ctrl+V/Cmd+V):** Paste contents from clipboard
- **Delete (Delt/Delt):** Like Cut
- **Rename:** Give files/directories or network files new names
- **Find (Ctrl+F/Cmd+F):** Call the search functions e.g. of the file system window

This menu contains standard functions for the work with files, directories, network files, texts, lists etc. The functions may vary depending on the active docking window. The cut function of the file system window for example causes the selected files or directories to be deleted while this function deletes the selected text in the active editor of the document window. These menu entries are enabled if the functions are supported by the active docking window, deactivated otherwise.

Some of the functions can also be called with shortcuts (Windows/Mac) or by clicking tool bar buttons.

Additional Links

- File System Window
- Network File System Window
- Document Window
- Tool Bar

Setup Menu

- Language
- Look & Feel
- Fonts
- Memory
- View
- Web Start
- Preferences

Language

The user language can be very easily configured during installation via installer or using Web Start so that the Digester software already displays the desired language initially. Especially on computer systems with multiple users it may be necessary that the software supports multiple languages. While one user wants the English version, another user prefers the German version. This is no problem because the language setting within the Digester software is stored on a per-user basis. You may also switch the language at runtime, only a restart of the application is necessary in such cases.

- **English (US):** English language, United States (American English)
- **German (DE):** German language, Germany

To do so, please open the **Language** submenu and select the desired language. The current language is grayed out and not selectable (here English). If you like to use English as your language then select the English (US) entry. Please note, that the languages are selectable here only if the relevant language packs have been installed. If you want to make all supported languages available for the computer users, then please select the **Full Installation**. When installing the Digester software via compressed archives then all available languages are installed by default. If you use the Digester software via Web Start, then you can choose your favorite language by simply selecting the appropriate link on the Leisenfels homepage. The menu for the Web Start version (see above) always shows all available languages on the server. The required language pack will be automatically downloaded when the language is changed.

Once you have selected a different language, the program prompts you to restart the application (not the whole computer). A restart is possible with all program variants directly from the menus. By clicking on Restart Now the application is terminated and then restarted. If it does not work, please start the application as usual e.g. by double-clicking the desktop icon or from the program group. You can use the new language when the software has been restarted.

Look & Feel

The Digester software uses a standard look & feel based on the operating system and therefore looks differently under Windows and let's say Linux. This makes sense since the application integrates flawlessly into the current environment. If you do not like the default look & feel then you can choose from an extensive list of additional look & feels. The Leisenfels website shows s small set of available look & feels.

To change the look & feel please activate the **Look & Feel** menu. The current look & feel is grayed out and not selectable. Please note, that the look & feels are provided only if the components have been installed. If you want to make all supported look & feels available for the computer users, then please select the **Full Installation**. When installing the Digester software via compressed archives then all available look & feels are installed by default. This is also the case for the Web Start variants of the software.

Changing the look & feel does not require the application to be restarted, the new look & feel is displayed immediately. When switching between different look & feels, it is possible that single components are displayed unexpectedly. Then please restart the application (not the whole computer) and the freshly initialized application components are displayed properly.

By activating the menu **Show Overview...** a dialog is opened. On the **Look & Feels** tab you can see the list of currently installed look & feels with their respective short names. Right next to the list you can find the available look & feel themes as well as the individual characteristics of the selection. If you select a look & feel with the mouse then the available themes are determined and displayed. The bottom right area shows the main properties of the selected look & feel including short name, description, and full class name. This status dialog can also be opened with the help menu <u>About Digester</u>.

Fonts

In order to improve the readability of the displayed fonts you can adjust the font sizes:

- **Larger Fonts (Ctrl+Plus/Cmd+Plus):** Increase font sizes by 10%
- **Smaller Fonts (Ctrl+Minus/Cmd+Minus):** Decrease font sizes by 10%
- **Reset Font Size:** Set font sizes back to default value (100%)

Some of the functions can also be called with <u>shortcuts</u> (Windows/Mac). In order to configure a precise font size percentage please adjust the input field within the <u>preferences</u>.

Memory

Java programs are often deployed in professional environments and are subject to some (reasonable) limitations on resource consumption. Thus, the amount of used memory cannot grow infinitely, but the maximum must be specified before the application is started. This has the advantage that the running computer programs cannot crash the system (important for 24/7 servers). This precaution, however, makes it necessary to determine how much memory may be consumed (memory here always means the RAM). How much memory is currently consumed by the application and how much more is possible, these values can be controlled with the memory controller component on the right side of the [main window status bar](#).

- **Release Caches:** Clear internal caches in order to increase available memory
- **Memory Size:** Adjust the maximum memory size with a dialog
- **Collect Garbage:** Free memory that is not longer needed

With the menu function **Release Caches** the Digester software clears the currently managed internal caches, if possible. For example this includes many of the displayed icons and dialogs which are kept in memory for a certain period of time (normally 10 minutes). This is efficient since the resources do not have to be reloaded again and again. After the configured timeout the caches are automatically released then the total amount of available memory for the program can be increased. You can use this function before complex calculations to free as much memory as possible.

By default, the Digester software is configured to consume a maximum of 128 megabytes (MB) of memory. Here the amount of memory that the Java environment itself consumes must be added (under Windows about 35 megabytes). This amount of memory should be sufficient for average checking and generating operations, however, you should configure more memory if you plan to process entire disks or directory trees. While the algorithmic calculations require little memory, the result structures may require many hundreds of thousands of objects being managed simultaneously.

In order to adjust the memory size for the programm in general please click on the **Memory Size** item, a dialog is shown to adjust the memory. You now have the choice of a fixed value from the list (e.g. 64, 128, 256 megabytes) or you can enter the desired value manually (e.g. 384). Values of less than 64 megabytes will not be accepted by the dialog. If you like to increase the default value of 128 megabytes, please select one of the values 256 or 512. It must not exceed the half of the physical memory since this can cause the operating system to crash. If your computer for example is equipped with 1 gigabyte of memory (1024 megabytes) then you should enter a maximum value of 512 megabytes.

Now you can confirm the dialog with the **OK** button to commit the maximum memory size. Then another dialog is opened that prompts you to restart the program (not the whole computer). If you select **Cancel** here then the previous memory size is not changed. Click **Restart Now** to terminate the application which is then immediately restarted. If it does not work, please start the application as usual e.g. with the desktop icon or from the program group.

⚠ If the amount of memory is too large, the application may not start! To reset the previous value for installer versions please open the digester.vmoptions file in the application installation directory with a text editor (e.g. WordPad under Windows) and then set the value back to -Xmx128M (default). Start the application as usual and select a suitable amount of memory as described above. For Web Start versions you can simply go to the Leisenfels homepage and reinstall the software by clicking the appropriate hyperlink. The application will then use the default memory size of 128MB.

Often large amounts of text or image data are loaded into the clipboard which may require much memory. The clipboard can be cleared at any time by clicking the clipboard icon next to the memory controller in the status bar and then by clicking the symbol with the blue recycle icon. The amount of consumed memory shown by the memory controller will then be reduced. The **Collect Garbage** function is automatically called in the background either at irregular intervals and if the Java environment runs out of memory.

This function can also be called with the shortcut **Ctrl+G** (Windows) or **Cmd+G** (Mac).

View

- **Tool Tips:** Show little automatic windows with help texts for components
- **Tool Bar:** Show the tool bar at the top
- **Quick Check:** Show the input field for quick checking at the top
- **Status Bar:** Show the bar for messages, progress, and memory at the bottom

With the menu **View** you can turn certain GUI components on or off. The so-called tool tips are the small help windows being shown after a certain time period when placing the mouse pointer over a window component. Tool tips are enabled by default because new users can find useful hints here. If you do not like tool tips to be shown just deselect the checkbox item named **Tool Tips**.

If you like to save some place on the main window you can hide the tool bar (the row of icon buttons on top of the main window). With the menu **Tool Bar** you can reactivate this component at any time. The tool bar can also be hidden by using the context menu (right mouse button) on the right side of the tool bar component next to the icons. Many of the menu functions can be called by their corresponding context menus provided directly by the GUI components.

The quick check field is located between the tool bar and the central docking area of the Digester main window. This component can also be hidden by selecting the context menu **Hide Quick Check** (right mouse button) directly on the quick check. Detailed information about this component can be found in the Quick Check section.

Like the tool bar also the status bar can be hidden. The status bar can be found at the bottom of the main window where the above mentioned memory controller is positioned. With the menu function **Status Bar** you can reactivate this component at any time. The status bar can also be hidden by using the context menu (right mouse button) directly on the status bar component

(select the **Hide Status Bar** menu item). This context menu provides two additional menu items to hide the memory controller and the time component separately.

Web Start

- **Java Control Panel:** Web Start application management of the Java environment
- **Uninstall:** Uninstall Web Start version, keep local configuration files
- **Clean Uninstall:** Uninstall and also delete local configuration files

The Digester software can be installed locally via Installer or downloaded and started from the Internet via Web Start. To support the work with the Web Start version the software provides various menu functions. You can find these functions in the submenu **Web Start**. Please note that all menu items shown here are only available within the Web Start versions and disabled for the installer versions.

The Java Control Panel can be opened with the **Java Control Panel** menu item. The Java executable `javaws` must by accessible from the execution path which is normally adjusted by the Java installer automatically. Within the Java Control Panel you can list and remove currently installed applications, view JNLP files, and much more.

With the menu function **Uninstall** you can remove the Web Start application from the Web Start environment including resources from the local disk. If desktop icons have been installed during Web Start installation these will also be removed here. The application of course cannot be started after uninstallation and must then be installed again from the Leisenfels Internet server. If you also want the configuration directory of the Digester software be deleted on uninstall please activate the menu **Clean Uninstall**. This allows the software to be quickly installed e.g. on a customer's computer and uninstalled after the work has been done without leaving any files on the system (data security).

> Please verify that the configuration directory has been successfully removed when using the **Clean Uninstall** function since especially Windows systems permit applications to remove files which are currently opened. For instance this is the case for the Digester log file.

Preferences

Here you' ll find the options to configure the program. For a detailed listing of the configurable values please see the [Preferences](#) section.

The import/export of preferences as well as the function to explicitly save the settings is implemented by the functions of the [File](#) menu.

Additional Links

- [Preferences](#)

- Tool Bar
- Quick Check
- Status Bar

Network Menu

- Edit Connections
- New Session
- Connect
- Disconnect
- Clear Pool

| BE | Please keep in mind that the network functionality is not supported by the Basic Edition and the Standard Edition. |
| SE | |
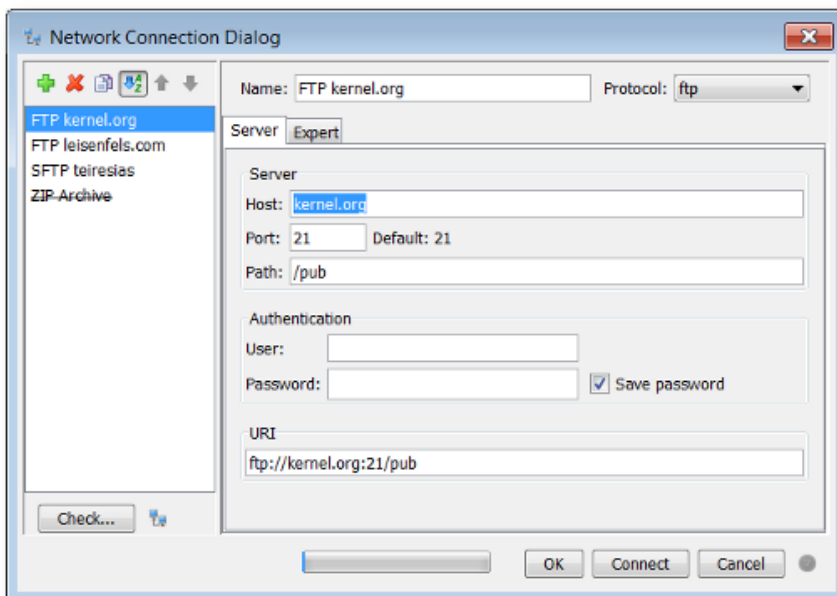
The dialog for the management of network connections allows to create and maintain connections with local or remote servers. On the left side of the dialog window new connections can be created (plus symbol), existing connections can be deleted (red X), and selected connections can be copied (symbol with double document). The order of the connection items can be alphabetically or in the order of creation. If the connections are shown as created (unordered) then the connections can be sorted manually by moving the items up or down. Invalid connections due to invalid configurations (e.g. because the used protocol is not supported) are shown with a stroke. The tool tip shows a message.

In order to adjust the existing connections please select the entry in the list. The settings for this connection are then displayed on the right side of the dialog. To check the current settings, please click the **Check** button at the bottom. A special dialog is opened where a connection can be established using the current settings. Please let the details be shown to see all messages. Is the

connection test successful, the dialog lists the capabilities of the connected file system. If the connection could not be established e.g. due to a missing password an appropriate message is displayed. Sometimes the log file of the software contains a hint why problems have occurred during connection tests (blocked/wrong ports, wrong password etc.). In addition to the connection check you can open the network file chooser dialog in order to test the selected connection.

> ⓘ The network protocols must have been installed in order to be listed here. When in doubt please choose the **Full Installation** for the installer.

Each connection can be given an appropriate name on the right side which can be adjusted later. At the top you can select the network **Protocol** to be used for the data exchange with the server. The combobox shows the available protocols like FTP or SFTP. The network protocols must have been installed in order to be listed here. This can be done for the installation with installer or with Web Start. A separate installation of specialized operating system drivers or similar is not required. Depending on the selected protocol the settings below may vary from connection to connection.

The connection settings are devided into **Server** and **Expert** options. The server options include:

- **Server:** Name or IP address of the server
- **Port:** Network port for the service like 22 für SFTP, default is displayed
- **Path:** Access path on the server
- **User:** Name of the user for login (authentication)
- **Password:** Password of the user for login
- **URI:** Overall address for the server access consisting of the other fields

You can set a directory on the server as path which can be accessed by the configured user. If the user `bobby` logs in to a FTP server and the home directory of the user is `/home/bobby`, then set `/` to access the directory directly or `/temp` to list the subdirectory `/home/bobby/temp`. Create links on the server i norder to access the root directory of the system if this is allowed for the user. Depending on the server system the access rules configured on this system are valid also for the network connection so that the access to files and directories is secure.

If user name and password are required please edit the data in the text fields. If the password may be stored permanently so that is available for the next connects please enable the **Save password** flag. If the password shall be used for the next connect only please disable the flag. The password is not remembered then and has to be typed again the next time. If you like to edit user name and password each time a connection is established (not remembered) then use the expert option **Display settings before connection is established**.

The input field for the URI (*Uniform Resource Identifier*) combines all entries of the other components for this connection. You may either fill out the single fields or alternatively edit the overall URI, the other fields are updated appropriately.

A URI must apply to the follwing scheme:

```
[protocol]://[username[:password]@]hostname[:port][absolute-path]
```

Examples of valid URIs are as follows:

```
ftp://kernel.org/pub/
```

```
ftp://root@ftp.mycompany.com/media/hdd/movie
```

```
sftp://bobby@mycompany.com :22/Temp
```

> ⓘ The configured password is not shown as part of the URI but used only internally in order to avoid misuse and spying.
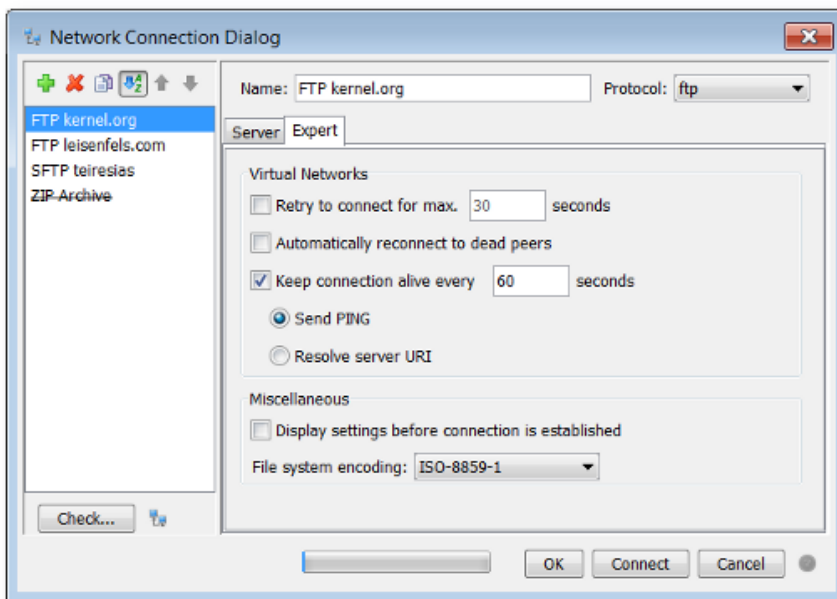
The **Expert** options include:

- **Retry to connect for max x seconds:** Wait for slow dialin connections
- **Automatically reconnect to dead peers:** Re-establish connections when interrupted
- **Keep connection alive every x seconds:** Keep connection (ISDN/VPN)
- **Send PING:** Contact server, normally by sending ICMP PING
- **Resolve server URI:** Contact server by listing the access path
- **Display settings before connection is established:** Show dialog always
- **File system encoding:** Charset for the file names on the server-side

Once all settings shown by the dialog are as desired you can connect to the selected servers by clicking the **Connect** button at the bottom of the dialog. If the selected connections are valid this button is enabled, disabled otherwise. For each connection a separate tab with the name of the connection will be displayed by the network file system window. With a click on the **OK** button the current settings will only be saved without creating new connections.

The configured connections are used program-wide, which means that these connections are provided by the network file chooser dialog and by the sync wizard for example. If files/directories

on remote servers are synchronized it makes sense to deposit user name and password for the connections, if these functions are called from the command-line or from build scripts.

The network connection management may be called also from the tool bar by clicking a button. The preferences provide further settings for network connections, for example if connections of the network file system window shall be restored or not when the program is restarted.

New Session

With the **New Session** menu one can establish connections with configured server systems. The currently selected connection within the tool bar is used for this. This selection is automatically set to the current list selection each time the management dialog is closed so that one can work directly with the connection that has been edited last. For each connection a separate tab with the name of the connection will be displayed by the network file system window. If configured for the connection the management dialog is presented here where you can edit user name and password to login.

The function to create a new session may be called also from the tool bar by clicking a button.

Connect

The **Connect** menu is enabled if the network file system window is visible and the active connection tab is disabled which means that currently there is not connection to the server. Together with the **Disconnect** menu it is easily possible to keep connection tabs (sessions) permanently on the screen (see preferences) while the connection is established on demand and disconnected after use.

Disconnect

The **Disconnect** menu is enabled if the network file system window is visible and the active connection tab is enabled which means that the tab is currently connected to the server. Together with the **Connect** menu it is easily possible to keep connection tabs (sessions) permanently on the screen (see preferences) while the connection is established on demand and disconnected after use.

Clear Pool

The **Clear Pool** menu may be called to free memory consumed by re-used network connections. Connection recycling is an optimization to limit the costs produced by new network connections. Connection pools are managed for example for stateful protocols like FTP and SFTP.

Additional Links

- Preferences

- Tool Bar
- Output Window
- Network File System Window

OpenPGP Menu

- Create New Key
- Import Keys
- Export Keys
- Check Key Strength
- OpenPGP Keys

> ⓘ The OpenPGP functions can be used only if the unlimited key strength has been activated. You can find detailed pieces of information about the legal background for the deployment of cryptography on our website. With the Check Key Strength functionality you can determine the currently supported key strength of your Java environment.

Create New Key

If you do not have generated own OpenPGP keys the Digester software provides functions to create some using a wizard dialog. The generated keys may also be imported by other programs with encryption functionality as long as OpenPGP-compatible keys are supported for import.

After the **Create New Key** menu has been selected a wizard dialog appears where you are prompted to setup the settings of the new key. In the first step the following pieces of information can be specified:

- **Real name:** Arbitrary input of your name or the organization's name
- **Comment:** Optional extension in brackets behind the real name
- **E-mail address:** E-mail address of the key owner
- **Key type & length:** Choose algorithm and strength of the new key
- **Expire date:** Date when the key becomes invalid (optional, unlimited otherwise)

Name, comment, and e-mail address will be displayed by OpenPGP-compatible programs in the form "Real name (Comment) <E-Mail address>", for example "Leisenfels Development (temporary) <devel@leisenfels.com>" where name and e-mail is required and the comment may be omitted. In the key parameter section you may choose between the algorithms RSA or DH/DSS (DH=Diffie/Hellman). The key length must be specified in the range between 512 and 4096 bits while values less than 2048 should not be used any longer due to security reasons.

The input fields for name, comment, and e-mail remember their values in separate histories so that the values can be re-used for the next key generations. You can clear the histories with the context menues (right mouse button). Here you can also paste values from the clipboard or copy into clipboard for other programs.

In the second step of the wizard you can set the passphrase (password) for the new key:

- **Enter passphrase:** Passphrase to decrypt the secret key part
- **Confirm passphrase:** Enter passphrase again to avoid input errors
- **Show cleartext:** The passphrases are displayed as clear text (directly readable)
- **Remember passphrase:** Passphrase is stored permanently e.g. to check signatures

After clicking the **Finish** button, the new OpenPGP key is generated based on your settings. This may take some seconds depending on the settings and depending on the computer speed. The button is only clickable if all values are consistent. The fields containing missing or faulty entries are highlighted as an indication. After the key generation has been completed a message is displayed by the [status bar](#) of the program. The new key may be used from this point by the program functions. If the passphrase has not been remembered permanently, you can store the passphrase at any time later using the key management. This is especially useful if signatures are generated from the command-line or from build scripts since no dialogs may be displayed to prompt for passphrases here.

Import Keys

If you already have OpenPGP keys then these may be imported with the Digester software. If you plan to generate signatures for FTP servers for example then you must import the secret keys. In order to do this the passphrase is required, too, which you can edit after the successful import of your key using the key management. The import of public keys is sufficient if you only plan to check signatures from the Internet. Here, you do not need a passphrase.

After the **Import Keys** menu has been selected a wizard dialog appears where you are prompted to select one or multiple keystore files. Usually OpenPGP keys are saved to files with the `.asc` or `.sig` extensions, other extensions may be possible. If confirmed with **Next** the selected files are being analyzed for usable keys which are presented by a list. Please select the desired keys for import and confirm with the **Finish** button. When the import has been completed an appropriate message is displayed by the [status bar](#) of the program. The loaded keys may be used from this point by the program functions. You can store the passphrases at any time later using the key management. This is especially useful if signatures are generated from the command-line or from build scripts since no dialogs may be displayed to prompt for passphrases here.

Export Keys

In order to re-use OpenPGP keys with other programs with encryption functionality, you may export keys with the Digester software into files. Please consult the documentation of the software, if OpenPGP keys can be imported there.

After the **Export Keys** menu has been selected a wizard dialog appears where you are prompted to make some settings. Please select the keys you wish to export and confirm with the **Next** button. In the next step you have the following options:

- **Export public keys only:** The secret key is *not* exported separately
- **Export secret keys only:** The public key is *not* exported separately

- **Export full keys (public and secret):** Save all key parts
- **Write ASCII format:** Keystore file is created as a text file
- **Write binary format:** Keystore file is created as a binary file

In the next step you can choose the keystore file for the export. Usually OpenPGP keys are saved to files with the `.asc` or `.sig` extensions, other extensions may be possible. After confirmation with **Finish** the export takes place. If the keystore file already exists a dialog is displayed. When the export has been completed an appropriate message is displayed by the status bar of the program.

Check Key Strength

Since some countries have special laws and regulations to limit the import/export of cryptographic technologies the Digester software may be deployed only for strong encryption (key lengths up to 128 bits) by default. This behavior is managed by the Java environment and may be modified appropriately if allowed by the laws. For key lengths up to 128 bits the OpenPGP functionality is switched off, since longer keys lengths are required here.

You may check the current crypto settings of your Java envirnoment easily by calling the **Check Key Strength** menu. If the key length is not limited then an appropriate message is displayed by the status bar of the program. The OpenPGP functions found in the **OpenPGP** menu can be used without restrictions here. If your Java environment is currently limited to key lengths up to 128 bits then a dialog with further information is being displayed. With the **Install & Restart** button you can let the software make the required modifications for the Java environment automatically. You are advised to consult your lawyer or local authorities in questions of cryptographic restrictions in your country. Please enable the unlimited key strength only if you have checked the legal situation in advance.

If the Java environment has been successfully configured to use unlimited key strength then the software is restarted. Afterwards you can use the OpenPGP functionality without restrictions. If the modifications cannot be executed due to missing permissions on your computer the dialog must be closed with the **Cancel** button. Please ask your system administrator to implement the required modifications which are explained on our website↗ for the recent Java versions. Two files have to be replaced by appropriate files from a ZIP archive which can be downloaded directly from the Java manufacturer's website.

OpenPGP Keys
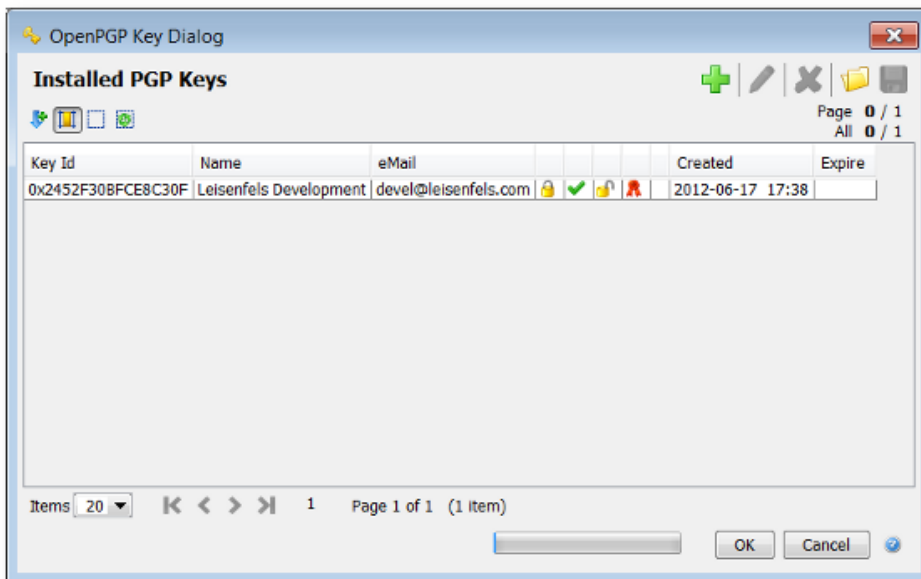
Currently OpenPGP keys are used internally by the Digester software e.g. by the generate window to calculate signatures based on the OpenPGP standard. The OpenPGP keys can be managed comfortably with a special dialog of the Digester software. In the center of the dialog you can see a table with available OpenPGP keys showing the **Key Id**, **Name**, **eMail**, **Created** (date and time), and **Expire** (date). In addition to that the table shows which functions may be used for the keys. If a secret key exists, then signatures can be generated and encrypted data can be decrypted. Of only the public key is available then signatures can be verified and data can be encrypted. Data encryption and decryption is currently not suported by the Digester software. The **Created** column may show a pen symbol if a passphrase has been specified for the key which is only possible for secret keys.

Above the table you can find some buttons on the left side which can used to control the table display. The symbol with the down arrow allows to display an index column. The second symbol sets the column widths automatically so that the texts in the columns will be readable (without abbreviations). The symbol with the empty box deselects all entries in the table, the symbol with the green arrow in the box selects all table entries automatically.

The symbols on the right side above the table call the corresponding functions of the **OpenPGP** menu in the main menu. The plus symbol creates new OpenPGP keys. A detailed description can be found in the Create New Key section.

The symbol with the pen may be used to set passphrases for the keys. Please select the OpenPGP key in the table and then click the pen symbol. Alternatively you can double-click the table entry. A dialog is then opened where you are prompted to enter or modify the passphrase. Set the **Show cleartext** option to be able to read your input directly. Please confirm your settings with the **OK** button, an empty input field resets the passphrase.

> Passphrases may be shown as cleartext only if entered the first time due to security reasons.

With the red X symbol you can delete the selected keys from the list. If you accidentally deleted keys then please abort the dialog using the **Cancel** button and call the dialog again. The previous keys will be visible again. It makes sense to create key backups from time to time. You can use the export function for this. In order to transfer keys from one PC to another you can use the import function. Both functions may be called alternatively from the key management dialog (folder and diskette symbols).

> ⚠️ OpenPGP keys represent valuable and critical data which has to be handled with care and should be kept in a safe place. Create backups of your keys from time to time and keep the keystore files at a safe location like a bank vault. Do never write down passphrases and allow access to keys and passphrases only for entitled people.

The OpenPGP key management may also be called from the tool bar with a button. The button functions on the right side above the table may alternatively be called from the context menu (right mouse button) where the selected keys are being processed.

Additional Links

- Status Bar
- OpenPGP Signatures
- Generate Window
- Tool Bar

Window Menu

- Docking Windows
- Screen Shot
- Resolutions
- Reset Windows
- Close Tab
- Hide
- Refresh

Docking Windows

The docking windows in the main display area are the key components of the Digester software. Using the buttons at the top, bottom, left, and right sides along the edges of the main window, the docking windows can be made visible or invisible. Which docking windows are displayed at all can be configured with the **Docking Windows** menu. Depending on the edition, single or multiple windows can be missing, for example the generate window is not available in the Basic Edition.

Each docking window can be configured individually so that you can easily make those windows visible that are relevant for your daily work. In order to display a docking window, the checkbox menu entry for the docking window must be selected. If the checkbox is disabled, the window is

not visible within the side bars. Some of the docking windows are shown automatically by the application when needed. For example, this is performed for the search results docking window.

Screen Shot

The Digester software provides various functions to take screen shots to be saved as files or to be printed. This functionality is useful e.g. when the Leisenfels support is contacted, since certain behavior of the software or error conditions can be documented effectively. Any support request to Leisenfels should be as precise as possible and in some cases a screen shot attached can be much more informative than a textual description. With the **Screen Shot** menu you can either open the capture tool dialog or take screen shots with the available sub-menus directly.

- **This Window:** Print or save main window or copy to clipboard
- **Full Screen:** Print or save full screen or copy to clipboard
- **Capture Tool:** Show dialog and capture windows after timeout

> ⚠️ Please note, that screen shots can contain sensitive information! Check the visible data carefully before giving away to other persons. In some environments like hospitals a higher security level may be required.

For the print function a printer is required which must be supported by the operating system (e.g. Windows print service). The available printers of the operating system are being presented by the dialog. The page format is configured automatically for the screen shot to fit on the page. Alternatively you can save the screen shots as graphics files. You will be prompted to specify a target file when the dialog is opened. If the file already exists a dialog is shown. Currently supported graphics formats are BMP, JPG, WBMP, PNG, and GIF.
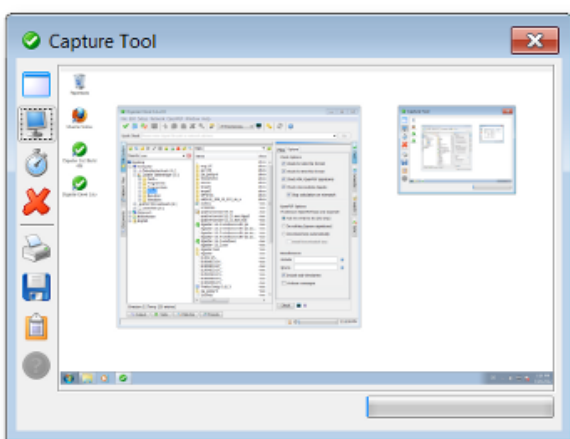
By activating the **Capture Tool** sub-menu a special dialog is opened. This special tool is independent of the Digester main window and can thus be positioned anywhere on the screen. On the left side of the dialog you can see numerous buttons. The first three buttons can be used to produce a screen shot of a single window of the Digester software, to capture the entire screen, or

to take a full screen shot after a timeout. To capture a single window just click the top button and then activate the desired window of the Digester application. Please note, that this only works for the windows of the Digester application while the windows of other programs cannot be captured currently. The captured image data is then displayed by the preview area of the capture dialog (see illustration). The **Hide Dialog** flag in the context menu of the timeout button can be set if the tool dialog itself shall not be visible on the photo.

If you like to take a screen shot of the entire screen so that all visible windows are captured, please click the second button from the top (monitor icon). Alternatively you can take a full screen shot after a certain timeout (e.g. 5 seconds). This function is useful to capture context menues which are opened right before the screen shot to be on the photo. The timeout value (default: 5 seconds) can be adjusted using the context menu (right mouse button) directly on the stop watch symbol (1 to 60 seconds). With the X button you can remove the image data from the preview since this saves significant amounts of memory. This could be done whenever you finished working with the capture tool.

The captured images can now be processed in various ways. The Digester software can directly print the image on paper or save the data into files. Click on the printer icon to print the image on a local printer. This function uses the default print service of the operating system. Under Windows, the well-known Windows print dialog is opened. Usually the default printer is already selected so that the image can be directly printed on paper. It depends on your printer configuration whether the screen shot is printed in color or just black and white for example.

With the floppy disk icon you can save the captured image data into files. Currently the file formats BMP, JPG, WBMP, PNG, and GIF are available. Of course you can store the images in various formats in multiple steps. The generated files can then be used for e-mail attachments or opened with image editing programs such as Gimp. The following figure shows the file selection dialog which is displayed after clicking on the floppy disk icon and where you can specify the filename and format. If you like the image data be available for other applications with access to the system clipboard, you can click the clipboard button. This loads the image data into the clipboard of the operating system if supported (e.g. under Windows). The popular image editing programs are able to process such data.

The capture tool can also be opened with a [shortcut](#) (Windows/Mac).

Resolutions

In order to set standardized dimensions of the [main window](#) several submenues of the **Resolutions** menu exist. For example you can set the size 1024 x 768 pixels (XGA) here.

Reset Windows

You can call this function to get the docking windows back to their default positions when the program was started the first time. This may be required if a windowshas been moved accidentally from lieft to right and alike. Before the resetting takes place a dialog is displayed which can still be

canceled. The reset includes the window sizes, positions in the dock, as well as the window modes (pinnded, unpinned etc.) which are reset to the initial values. The current settings will be lost. The options of the docking windows (e.g. the templates of the generate window) are not influenced by the reset.

During reset the docking windows are closed once and then reopened where positions, sizes etc. are then adjusted with the values as if the program is started the first time.

Close Tab

The **Close Tab** menu can be called in order to close the active tab within a docking window. This is only possible for some docking windows like the document window. The function is executed for the active tab of the window (e.g. the shown document). The other tabs are not influenced by this function, but may be closed also by calling this function multiple times. Depending on the docking window special dialogs or messages may be displayed if for example a modified document should be saved before it is closed.

This function can also be called with a shortcut (Windows/Mac) if the active docking window supports it. If not, the **Close Tab** menu is not clickable.

Hide

Docking windows can be hidden always if a docking windows has been activated to receive input. The docking window then shows a special header as an indication (e.g. different color).

This function can also be called with the corresponding shortcut (Windows/Mac).

This function is useful in order to make displayed docking windows invisible quickly for instance if critical data like passphrases are currently being shown which should not be seen by other persons. If a docking window is hidden then the previously active window becomes visible again and so on. By calling this function multiple times all docking windows are made invisible. If a docking window is being shown exlusively it will be hidden and the exclusive display of this window ends.

Refresh

The **Refresh** menu allows to update the active docking window. The exact function may vary depending on the type of the docking window. In the case of the file system window and the network file system window the current directory will be read and listed again.

This function can also be called with the corresponding shortcut (Windows/Mac) if supported by the active docking window. If not the **Refresh** menu is nicht clickable.

Additional Links

- [Docking Windows](#)
- [Tool Bar](#)
- [Document Window](#)
- [File System Window](#)
- [Network File System Window](#)

Help Menu

- [Help](#)
- [Help Topics](#)
- [Manual (PDF)](#)
- [Release Notes](#)
- [Check for Updates](#)
- [Send Status Report](#)
- [Product Homepage](#)
- [Buy License](#)
- [Register](#)
- [About Data Digester](#)

Help

This menu opens the [help system](#) appropriately for the active docking window or the current program state. The possible entry positions are depending on the active docking window. If the help is not possible for the current context then the menu is disabled. Then please call the **Help Topics** menu instead and search for a pattern of interest.

The context-sensitive help can also be called with the corresponding [shortcut](#) (Windows/Mac) or from the [tool bar](#) with a button.

Help Topics

The help system in JavaHelp format is part of the Digester software and can be opened directly from the running application. It is particularly suitable for fast lookup. The convenient search function allows to search the contents for keywords or even parts of keywords. The help system can be displayed by activating the **Help Topics** menu. Alternatively you can press the F1 key which opens a special dialog with the following tabs:

- **Table of Contents**: List of chapters and sections of the help system
- **Find**: Search for pattern and sub-patterns in the whole text
- **Favorites**: Bookmarks with frequently used text positions

The first tab **Table of Contents** lists the chapters and sections of the help system. Normally, there is already a start page selected, so that the viewing area initially shows some text. If the help

system is called from certain positions of the program, the display will navigate directly to the associated section (context help).

To search the full text of the help system, go to the Find tab (magnifying glass icon) and enter a search term where also parts of terms are allowed. When committed with ENTER, the search is started and the find results are listed. By selecting a matching list entry with the mouse, you can view the text in the text area to the right.

The **Favorites** tab finally allows to store help page references permanently, so that such references are directly available the next time the help system is opened. To store a favorite text page from the help system, please select a page from the table of contents or from the find tab so that the text content is displayed by the text area. Then click on the **Add to Favorites** button with the same symbol as the **Favorites** tab. You can find this button in the tool bar of the help system window (the slightly larger symbols). The added text page then appears on the **Favorites** tab and is also available the next time the help system is opened.

Within the tool bar there are some more buttons to navigate **Back** and **Next** as you probably already know from your Internet browser. You can click the printer icon to print the currently displayed text page on paper.

Please note, that a printer is needed here which must be configured on the operating system level. In order to adjust the size of the help system window for your computer system, you can drag the divider between the list area on the left and the text area with the mouse.

In addition to the help system a printable manual in PDF format is shipped with the Digester software.

Manual (PDF)

In addition to the help system, the Digester software also provides a printable manual in PDF format. You can view the PDF manual by activating the menu **Manual (PDF)**. If there is a PDF viewer such as Acrobat Reader installed on the computer, it will be automatically executed. If no window shows up at this point, the system probably lacks a suitable viewer for PDF files. Please install either an appropriate program to display PDF files and repeat the above steps, or use the help system instead which can be directly opened from the Digester software. The Digester software does not contain any viewer for PDF files.

The PDF manual is available for the Web Start versions of the software as well as for the installer versions, provided that this component has been installed. More information about installing the PDF manual can be found in section Installer for Windows. You can view the PDF manual for an installed version also by clicking the special symbols within the program group for the Digester software, provided that the program links have been created during the installation process.

From the Web Start versions, the PDF manual can be exclusively viewed by activating the **Manual (PDF)** menu which first of all downloads the PDF file from the Leisenfels web server to the

configuration directory within the user's home directory. The next time you call the PDF manual, this file is already available and does not need to be downloaded again.

Release Notes

To find out about the features of the available Digester versions, the Leisenfels web server is available around the clock. The software itself also provides so-called release notes. While the information for the currently used version is always available chances are good that you can find the release notes of older versions here as well. Please activate the **Release Notes** menu, so that a special dialog is displayed.

You can find the list of included program versions at the top left corner with the currently running version being selected by default. If you like to view the release notes for an older version, please select the corresponding entry in the list. The available notes for the selected program version are displayed by the list on the right side. The entry symbols represent the nature of the applied changes to the program version (e.g. new functionality, performance enhancement etc.). In addition to the title of the note the table shows the date when the entry has been deposited by the author. If you select a certain note from the list, the details for the note are being displayed by the text area at the bottom of the dialog.

The top right corner of the text area may show hyperlinks to be clicked directly with the mouse. For example, the links let you browse further information from the Internet or from the application help system. The target addresses of the links are displayed as tooltips if the mouse is crossing for a while. You can filter the available notes by type, category, and edition. Possible change types are New feature, Enhancedfunctionality , Fixed bug, Security issue solved, and Miscellaneous change. The categories are program specific, currently there are entries such as Graphical User Interface and Digest generation available. Notes may also belong to multiple categories. The edition filter allows you to display only the notes specific for one of the available editions.

Check for Updates

In order to keep the Digester software up-to-date, the software provides a special dialog. Here you can explicitly check for newer versions as well as configure automatic update checks performed in regular intervals. The figure shows the **Check for Updates** menu which opens the dialog shown below. This is the well-known information dialog providing relevant status data for the running application. The update information can be found on the **Updates** tab.

> ⓘ   Please note, that an Internet connection is required to load update information from the Leisenfels download server.

When you call the status dialog via the **Check for Updates** menu it will immediately attempt to connect to the Leisenfels Internet server. If you have no working internet connection then the update connection will fail and no update information is displayed. If the update server can be reached, a request is sent including the current program version as well as the unique license ID. The server responds based on the request by transmitting a data record which may contain a list

of available newer program versions. A special hint message is displayed by the dialog if the currently used version has not yet been replaced by a newer version or if the license is invalid (e.g. expired).

In the example, the request is generated by the installed program version 1.6.2 and the newer version 1.6.3 Build 229 is available on the server. In addition to the exact program version, the update list of the dialog shows the date of publishing for the update versions. The Update column indicates the type of update, if it is a newer major version, a newer Java version, a newer minor version, or a newer build version. The table shows examples for the different types of updates for the Digester software based on the following scheme:

```
[Hauptversion].[Java-Version].[Mini-Version] Build [Build-Version]
```

The major version starts with 1 and is incremented for each further major version. New major versions are published when the software has changed fundamentally. Commercial licenses are usually valid for a specific major version so that updating to another major version may not be possible.

The Java version in the second position of a typical program version denotes the minimum required Java version. The first Digester versions are based on Java 1.6, so the program versions are 1.6.2, 1.6.3, and so on. By adding the Java version to the scheme it is possible to publish multiple versions within a major version for example for Java 1.5 and Java 1.6. Please keep in mind that a suitable Java runtime environment is required for such updates at least for the Web Start versions. Some of the installer-based installation archives already include an appropriate Java environment already.

Update Type	Short	Example for Version Change
Major version	Major	**1**.6.3 to **2**.6.1
Java version	Java	1.**5**.4 to 1.**6**.1
Minor version	Minor	1.6.**2** to 1.6.**3**
Build version	Build	1.6.2 Build **184** to 1.6.2 Build **223**

The most common update type is the minor version change, since updates usually replace older versions of the same major version including minor enhancements and bug fixes. Such update versions can usually be used with an existing license. Both the installer and the Web Start variants may be installed simultaneously so that you can safely testdrive a new version prior to production use. Build version updates are only relevant for the development of the software and will not be available for end-users.

In order to avoid problems during the update process, please always check the latest release notes for the respective version at the Leisenfels website first. You can do this comfortably by clicking the hyperlinks shown by the update list of the software. These links are provided directly by the update server depending on the program and allows you to directly navigate to the appropriate area of the Leisenfels homepage. By clicking the links, your default web browser is

opened. The above mentioned dialog requests update information from the server while the update packages themselves are not downloaded here. Thus you always have the full control over the update process for your software.

If you are using an installer-based variant the installed version should be uninstalled prior to the installation of the update version. If you are using the Web Start version of the Digester software, it is possible to let Web Start care for the automatic updating or you can update your software manually by installing a fixed program version. For detailed information about the available Web Start versions of the software and the update techniques see section Web Start.

By clicking the **Check Now** button you can always check explicitly for available updates. If you like to query the update server automatically on a regular basis, then you should activate the **Auto-check** option shown by the dialog. Here you can configure the time intervals between the update requests. Currently, queries can be generated **On every start**, **Daily**, **Weekly**, or **Monthly**. The Digester software will automatically contact the update server and keep you informed about available updates by displaying a special icon in the status bar. By clicking this icon, the dialog is opened where you can view the list of available updates.

Send Status Report

The **Send Status Report** menu is only enabled if your license includes a commercially available Premium Support option. This option is transported within the license key and activated by the software when such a license is registered. The menu **Send Status Report** is then enabled and can be selected.
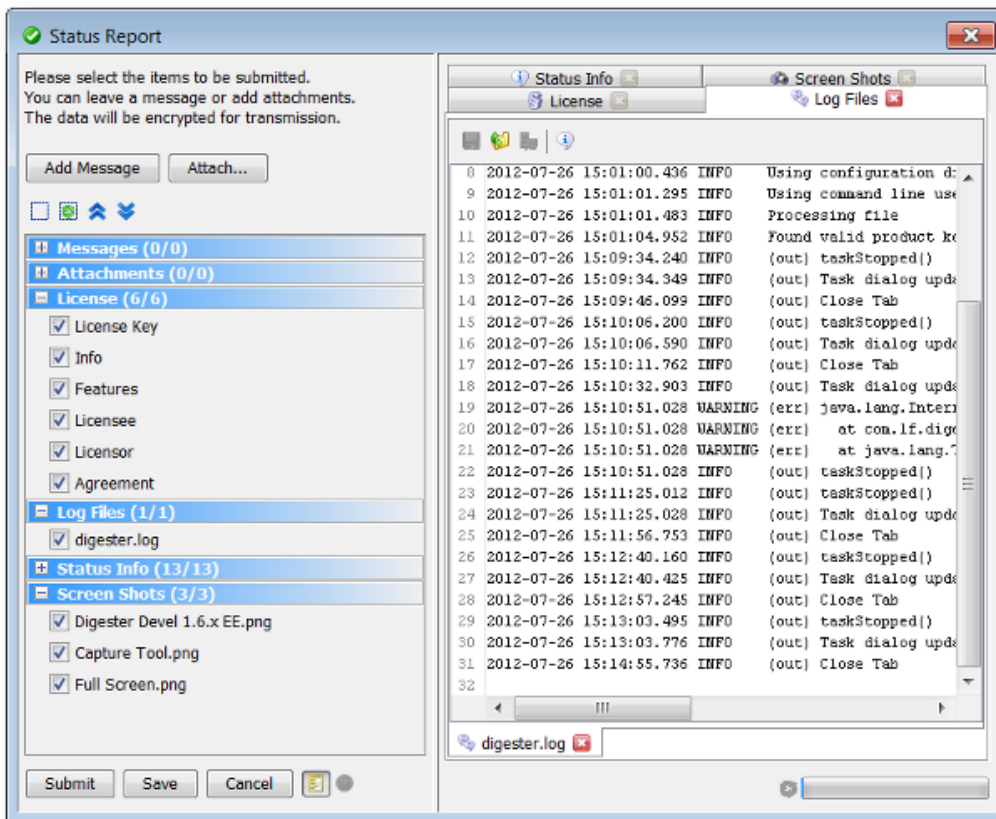
If the menu was called it may take a few seconds without any reaction by the program. Please wait until the dialog is opened or a status message is displayed by the program. If only a minimum of free memory is available please deactivate memory-consuming elements of the status reports like screen shots from within the [preferences](#) before you call the status report function.

> ⓘ The status report functionality currently is reserved for customers with Premium Support⬈ option.

A status report consists of different pieces of information collected by the running program and is transferred encrypted either directly to the Leisenfels support team or saved as a local file. Saving as file becomes necessary if no Internet connection is available. The export file may then be sent for example as an e-mail attachment from a PC with Internet connection to the Leisenfels support for further analysis. The contained data is handled confidentially and only in the way it is described in the license agreement (e.g. for analyzing errors).

Currently the following information can be sent with a status report:

- **Messages**: Program messages or own remarks as text
- **Attachments**: Program resources or own additional files
- **License:** Identification data about the current license
- **Log Files**: File with program output to be analyzed for errors
- **Status Info**: General information about the running program
- **Screen Shots**: Photos of the main window and the whole screen

The status report parts to be finally transferred may be configured in detail. If you like elements like screen shots not being transferred you may opt out this feature within the <u>preferences</u>. In addition to that any data elements presented by the dialog may be deselected or closed so that these data report parts are not transferred or stored. The transferred or save status reports are encrypted to avoid unwanted access.

The dialog to send status report on the left side shows the button **Add Message** to create a new message, where the text could be typed like an e-mail. Please write messages either in German or English, other languages are currently not supported. If you like additional files be attached to the status report (like sketches, ZIP archives etc.) then please click the **Attach** button and choose one or multiple files. Please keep in mind that the overall size of a status report is limited, the program warns if this maximum size will be exceeded.

Below the buttons on the left you can see the collapsible areas representing the report parts each. With the little buttons above the collapsibles you can deselect all, select all, expand all, or collapse all parts. With the context menu of the collapsibles (right mouse button) you may select or deselect the elements within the parts.

In the preview area on the right you can see the previews for the contained information as it is seen later by the support team. The upper tabs each represent the collapsible elements on the left side where at least one element must be selected in order to see the tab on the right. With the red X you may close the tabs on the right so that that the associated data is not transferred. The tabs at the bottom each represent the flags as shown by the expanded collapsibles on the left side. Also these tabs may be closed in order to ignore the associated data for the final transfer.

In order to send the status report please click on the **Submit** button. The program then tries to establish a connection to the Leisenfels server on the Internet. During transfer the progress is shown by a special dialog, you may do the transfer in the background like other tasks or cancel the transfer. If the transfer was successful then an appropriate message is displayed.

If the report could not be sent another dialog is displayed where you can store the status report as a local file. You can sent the stored report attached to an e-mail to our support team from a PC with Internet connection or at a later time. If you generally like the status report sent via e-mail then click the **Save** button where you are prompted to specify an export file. If this file does already exist a message is displayed.

In order to save memory, the preview for the scheduled data may be switched off (golden symbol next to **Cancel**).

Product Homepage

After selecting the **Product Homepage** menu you will be navigated to the Digester pages on the Leisenfels Internet site. To do this your default Internet browser will be started and the page will be preselected. If nothing happens probably no default Internet browser is configured on your PC.

> Please note, that an Internet connection is required to access the Leisenfels homepage.

Buy License

After selecting the **Buy License** menu you will be navigated to the Leisenfels shop on the Internet. To do this your default Internet browser will be started and the page will be preselected. If nothing happens probably no default Internet browser is configured on your PC.

> Please note, that an Internet connection is required to access the Leisenfels homepage.

Register

To deploy the Digester software, a commercial license must be obtained. You can find the available product licenses and terms of use on the Leisenfels website. The software can be licensed as a *Single-User License* where the Digester software can be used by a single person also on multiple computers. On first start or after selecting the **Register** menu the Digester software shows a dialog where you are prompted to specify a valid license key. Here you can see the following tabs:

- **License Key**: Text with encrypted license key from file
- **Info**: Information about the license like type, ID, validities
- **Features**: Active options for the software like Premium Support
- **Licensee**: Data about the licensee (customer)
- **Licensor**: Date about the licensor (Leisenfels)
- **Agreement**: Binding license agreement text for customer and Leisenfels

You can now specify a license key in several ways. If you have already purchased a license or started an evaluation, you have received an e-mail with the associated license key in the form of a text fragment or as a file attachment.

License files usually are named like `license-DIGEST16BESU-LF-00001267.txt`. Please save this file on your hard disk or copy the text fragment from the e-mail to the clipboard of your operating system. Under Windows this can be done by selecting the text with the mouse and pressing the key combination **Ctrl+C**. Then switch to the registration dialog and either insert the copied text (under Windows by pressing the key combination **Ctrl+V**) or select the license file by clicking the **File** button so that the license key is inserted into the text area of the dialog. Once the **Register** button becomes active, please click on this button.

The license will then be decrypted and verified. If the license is valid for the current version, the license details are displayed by the other tabs **Info**, **Licensee**, **Licensor**, and **Agreement**. Please read the license details carefully. In particular you should read the license agreement carefully because this provides the legal framework for the use of the software. If you confirm the dialog with the **OK** button, another dialog will be opened prompting you to accept the license agreement.

If you do not accept, the dialog is closed and the Digester software is terminated because no valid license key is configured. If you do accept the agreement, the dialog is closed and you can use the software from now on. The license will be deposited within the configuration directory and automatically loaded and checked the next time the application is started. If the deposited license

key is invalid e.g. because the license has expired you will have to set a valid license as described above.

> ⓘ If you want the Digester software to be improved in the future, we ask you to obtain valid licenses for each deployment of the software and to make sure the license agreement is respected. Thank you very much!

About Data Digester
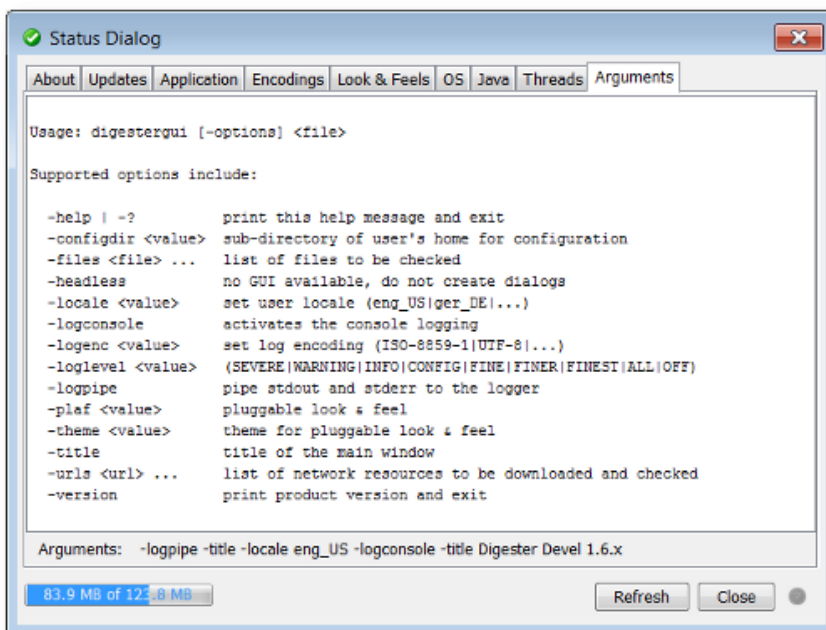
To view detailed information about the internal state of the Digester software you can activate the **About Data Digester** menu which opens the dialog described here. Alternatively, this status dialog can be displayed using the corresponding tool bar icon. This dialog provides various tabs with detailed information about the current program, the Java runtime environment as well as the operating system. Some of the tabs can be refreshed by clicking the **Refresh** button. The **Updates** tab can also be called directly with a dedicated menu item, please refer to the section Check for Updates.

- **About**: Symbol and info about license, Java environment, and program version
- **Updates**: Access to information about newer program versions
- **Application**: Characteristics about the running program and the computer
- **Encodings**: System charsets e.g. for the `-logenc` command-line argument
- **Look & Feels**: Info about the available look & feels
- **OS**: Environment variables of the operating system, important for support
- **Java**: Environment variables of the Java environment, important for support
- **Threads**: Characteristics about the currently executed tasks
- **Arguments**: Command-line arguments the software has been started with

The **Application** tab lists the main internal parameters of the running application. These include the computer name, the installed printers, the IP address of the computer, the presence of an Internet connection, the local configuration directory within the user's home directory, the license file, the log file as well as many other details, which can be useful when contacting the Leisenfels support. If the individual values are not readable, then you can adjust the column widths by moving the boundary line between the header cells accordingly.

The **Arguments** tab displays the command-line parameters supported by the current application version. Below the text area you can see the parameters actually in use. For more information on using command-line arguments please refer to the section <u>Arguments for `digestergui`</u>.

Additional Links

- Premium Support↗
- <u>Tool Bar</u>
- <u>Shortcuts</u>
- <u>Arguments for `digestergui`</u>
- <u>Preferences</u>

Preferences

- <u>Files</u>
- <u>File Types</u>
- <u>Data Limits</u>
- <u>Download</u>
- <u>Docking Windows</u>
- <u>File System Window</u>
- <u>Network File System Window</u>
- <u>Document Window</u>
- <u>Dialogs</u>
- <u>Disabled Dialogs</u>
- <u>Drag & Drop</u>
- <u>Miscellaneous</u>
- <u>Support</u>
- <u>Energy Saving</u>

The dialog for the preferences of the Digester software shows an overview of the supported units on the left side. The viewing mode may be set to either **Options Tree** or **Symbol List** (see figure). Both the symbols and tree nodes are clickable, the settings of the selected units are presented on the right side of the dialog.

By clicking the **Cancel** button, the current changes may be rejected. You can enter a search pattern in the input field to search for. The matching units are then displayed on the left side, the matches are highlighted on the right side to be easily identified. The search can be stopped with the red X button, then all units are being shown again.
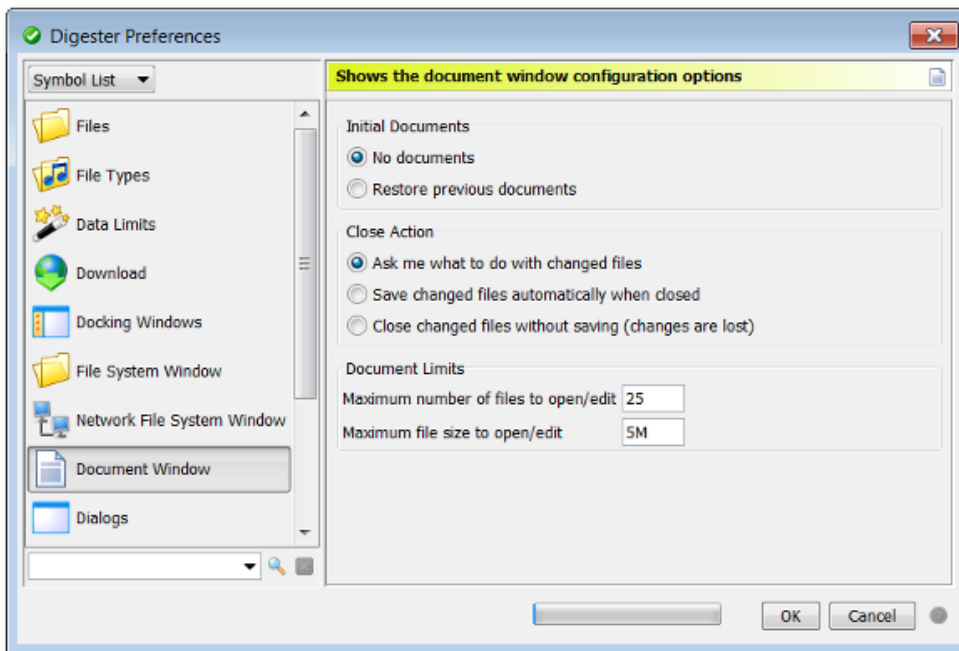
With the arrow buttons above the list you can expand or collapse the tree (tree mode only). These functions can be called also from the context menu of the units (right mouse button). Here you can find the additional menues **Restore Unit Defaults** and **Restore All Units**, if you like to reset the settings for a single or multiple units.

> ⚠️ When option units are reset your pending changes will be lost. Please backup your preferences prior to resetting, you can use the export preferences function to achieve this. Backups may be imported again with the import preferences function.

The preferences can be displayed also with shortcuts (Windows/Mac) or from the tool bar by clicking a button.

Files

- **Version:** The header text for created OpenPGP files, leave emty otherwise
- **Use default icons and types**: Display network files with default icons (faster)
- **Use local icons and types (costly)**: Display network files like local files

Here you can find the settings for processing files with the program. If OpenPGP files are written, the header may contain a single-line text (**Version**, optional). This text is arbitrary and may contain information like copyrights or your name or your company's name. The signature files headers created by the generate window are not influenced by this option.

The options **Use default icons and types** is used to display network files which are listed for instance by the network filesystem window. Here, a file symbol and the file type is being displayed where this pieces of information are provided by the underlying operating system (see file system window). It is not possible for network files so that either default symbols (document) and default types (file, folder etc.) may be shown or as an alternative you may choose the **Use local icons**

and types (costly) option. Here, a temporary local file is created in order to determine the network file type. This may be very costly and may slow down the program's speed significantly. This effect may be reduced a little if a fast cache is used where previously determined file symbols and types may be used again. Please activate the **Use fast cache** option for this.

File Types

Several file types are already supported by the software. Graphics files in the formats PNG and GIF may be displayed within the document window, XML files may also be edited there etc. The list shows the configured file types with the associated file extensions. If selected then the input field above may be used to edit the extensions. Multiple extensions must be separated with space characters in order to support arbitrary extensions. Changes are being displayed immediately by the list.

You can configure for each file extension how the type should be determined by the program. Normally the option **Simple extensions** should be sufficient which means that the file types must end with one of the configured extensions. For example the files `test.zip`, `test.ZIP`, `test.Zip` are recognized with a configured `zip` extension. If the case of writing (lower/upper) should matter then please choose the **Case sensitive** option.

Regular expressions are much more flexible and also file names beginning with a number or special characters may be recognized. Please choose the **Regular expression** option for this. You'll find the details on how regular expressions are used in section Regular Expressions.

For some of the listed extensions it can be configured how files should be opened or edited (**Open** and **Edit**). This is always possible if the program has an internal feature implemented to view the contained data (document window). You may choose for open or edit action separately if it is executed by the **Application** or by the **System** (operating system).

Please keep in mind that the system execution depends heavily on the used operating system and it may show no reaction at all. Under Window operating systems for example the WordPad is opened if this program has been configured as the default viewer for the selected file.

Data Limits

- **Maximum number of files to check/generate/synchronize**
- **Maximum amount of data to check/generate/synchronize**

The **Maximum number of files to check/generate/synchronize** option controls how many files are allowed to check, generate, and synchronize digests/signatures. This option helps to avoid unnecessary program errors (e.g. due to a large memory consumption). If it becomes clear right before the checking that too many files are going to be processed then the GUI version of the program shows a dialog window where you must explicitly confirm to continue. The command-line version of the software will abort at an early stage. Please enter a plausible numeric value into the field (default is 1000 files).

> ⓘ   Data limits should be adjusted appropriately also to compensate inout errors to some
> extent. For example a digest calculation may be started accidentally for a whole drive,
> then this process may be cancelled before data is unnecessarily processed.

Like the number of files also the amount of processed data may be limited. The total size of the processed files is calculated and compared with the value **Maximum amount of data to check/generate/synchronize** before digests/signatures are being checked, generated, or synchronized. If the maximum amount of data is exceeded, then the GUI version of the program shows a dialog window where you must explicitly confirm to continue. The command-line version of the software will abort at an early stage.

Please enter a plausible numeric value into the field (default is 1G = 1 gigabyte). You may can use the commonly used abbreviations (K)ilobytes, (M)egabytes, (G)igabytes and (T)erabytes here, where a postive number must be specified in the first place. Faulty input is highlighted here.

Download

Here you can setup the download options of the program like the download directory:

- **Application download directory**: subdirectory of the configuration directory
- **Fixed directory**: Always use the same directory
- **Ask me before a resource is downloaded**: Ask with dialog

Within the configuration directory of the program (like `.digester-1.6.x`) which is always located in the user's home directory, a special directory is being created for downloads. Here all downloads of the Digester software may be downloaded (**Application download directory**). If you like to use a different directory please select the **Fixed directory** option and click the button with the dots in order to choose a directory. If you like to decide from file to file please select the **Ask me before a resource is downloaded** option.

Besides the specific limitations for check, generate, and synchronize (see section [Data Limits](#)), also downloads can be limited to a certain extent:

- **Maximum number of files to download**
- **Maximum amount of data to download**

The **Maximum number of files to download** option controls how many files are allowed to be downloaded. This option helps to avoid unnecessary program errors (e.g. due to a large memory consumption). If it becomes clear right before the download that too many files are going to be processed then the GUI version of the program shows a dialog window where you must explicitly confirm to continue. The command-line version of the software will abort at an early stage. Please enter a plausible numeric value into the field (default is 100 files).

Like the number of files also the amount of processed data may be limited. The total size of the processed files is calculated and compared with the value **Maximum amount of data to**

download. If the maximum amount of data is exceeded, then the GUI version of the program shows a dialog window where you must explicitly confirm to continue. The command-line version of the software will abort at an early stage.

Please enter a plausible numeric value into the field (default is 1G = 1 gigabyte). You may can use the commonly used abbreviations (K)ilobytes, (M)egabytes, (G)igabytes and (T)erabytes here, where a postive number must be specified in the first place. Faulty input is highlighted here.

Docking Windows

The docking windows of the Digester software are the central components of the program and allow for several options to be configured here:

- **Enable ESCAPE to hide windows:** Fast hide
- **Move window to dock and show:** Show window after close
- **Move window to dock and hide:** Hide window after close
- **Close window completely:** Disable window after close
- **Automatically iconify separate windows:** Minimize windows together
- **Automatically deiconify (restore) separate windows:** Restore windows together

Docking windows may be shown/hidden using either the dock button or the separate window decoration. If a window must be hidden quickly without grabbing the mouse (for example because other persons could read critical data) then the ESCAPE button can be configured for this (option **Enable ESCAPE to hide windows**).

Some docking windows are often displayed in separate windows (e.g. the sync window). If such a window is being closed different kinds of behavior are possible which may be configured here. In the normal case the window is moved back to the dock in order to show it again at the new place (active window, option **Move window to dock and show**). A slightly different behavior is the option **Move window to dock and hide** where the window is not displayed by the dock afterwards. The variant **Close window completely** finally does not move to the dock but disabled the windows completely. It may be restored at any time with the menu **Window** -> **Docking Windows**.

If separate windows exist you can control the behavior here slighty. If the main window is being minimized then also the separate windows can be minimized, too (**Automatically iconify separate windows**) so that all program windows are in the same state. If the main window is being restored (displayed on the screen again) then also the separate windows can be restored. Please select the **Automatically deiconify (restore) separate windows** option here.

File System Window

This unit provides settings for the file system window:

- **Root directory:** View local main directory initially
- **User's home directory:** View user's home directory initially

- **Restore previous directory:** View previous directory initially
- **Fixed directory:** Always view a fixed directory initially

If the program gets started or the file system window is opened then it shows an initial directory which you can configure here. The **Root directory** is determined by the underlying operating system, under UNIX this directory is the root `/`. The **User's home directory** is determined by the underlying operating system, too, under UNIX this is `/home/bobby` for example.

If you work frequently with the file system window and it is useful to always start with the previous settings again, then please choose the **Restore previous directory** option. If you like to use a different directory instead, then the option **Fixed directory** is your friend. You can choose a directory with the button with the dots.

Network File System Window

Provides the settings of the <u>network file system window</u>:

- **No connections:** Start without any sessions/connections
- **Restore previous sessions:** Start with previous sessions/connections

If the program gets started or the network file system window is opened with the menu then it shows an initial set of sessions each listing the files and directories. You can configure this behaviour here.

If you like to start with an empty network file system window then choose the **No connections** option. If you work frequently with the network file system window and it is useful to always start with the previous settings again, then please choose the **Restore previous sessions** option.

Please keep in mind that all required connection parameters like user names and passwords must be present in order to successfully reestablish previous sessions. If the password has not been stored the last time then it must be entered now again. Please select the **Display settings before connection is established** option in <u>Edit Connections</u>. You will be prompted to enter your credentials if the connection is going to be established.

> BE SE   Please keep in mind that the network functionality is not supported by the <u>Basic Edition</u> and the <u>Standard Edition</u>.

Document Window

Shows the configuration options of the <u>document window</u>:

- **No documents:** Start without any documents
- **Restore previous documents:** Start with previous documents
- **Ask me what to do with changed files:** Show dialog
- **Save changed files automatically when closed**
- **Close changed files without saving (changes are lost)**

- **Maximum number of files to open/edit**
- **Maximum file size to open/edit**

If the program gets started ot the document window is opened using the menu then it starts showing local or network resources. Which documents are automatically loaded can be configured here.

If you like to start with an empty document window please choose the **No documents** option. If you work frequently with the document window and it is useful to always start with the previous settings again, then please choose the **Restore previous documents** option. Please remember that the resources must still exist and must be accessible in order to be restored (e.g. the network connection must be available).

When a document or the document window ot the whole application are closed then the question is how to proceed with pending document changes. The option **Ask me what to do with changed files** shows a dialog where you can choose on how to continue (save or reject). The option **Save changed files automatically when closed** allows one to automatically save changes when a document becomes closed. In order to reject changes by default please choose the **Close changed files without saving (changes are lost)** option.

⚠️ When edited documents are closed without saving your pending changes will be lost and may not be restored with the program.

The **Maximum number of files to open/edit** option controls how many files are allowed to be opened or edited. This option helps to avoid unnecessary program errors (e.g. due to a large memory consumption). If it becomes clear right that too many files are going to be processed then the GUI version of the program shows a dialog window where you must explicitly confirm to continue. The command-line version of the software will abort at an early stage. Please enter a plausible numeric value into the field (default is 25 files).

Like the number of files also the amount of opened data may be limited. The size of the files is compared with the value **Maximum file size to open/edit** before a file is opened. If the maximum file size is exceeded, then the GUI version of the program shows a dialog window where you must explicitly confirm to continue.

Please enter a plausible numeric value into the field (default is 10M = 10 megabyte). You may can use the commonly used abbreviations (K)ilobytes, (M)egabytes, (G)igabytes and (T)erabytes here, where a postive number must be specified in the first place. Faulty input is highlighted here.

Dialogs

This unit provides the following dialog options:

- **Java style (default):** Display file choosers independently from OS
- **Operating system style (native):** Display file choosers like OS
- **Auto-close timeout x milliseconds:** Used by task dialog

If tasks like file downloads or digest checks are processed the task dialog is presented by the application by default. Here you can find the messages of the program, the progress etc. If a task has been successfully finished and the **Close when finished** flag has been selected then the dialog automatically disappears after a certain period of time. Please enter a plausible numeric value into the **Auto-close timeout x milliseconds** field (default is 1500 milliseconds = 1.5 seconds). Faulty input is highlighted here.

Disabled Dialogs

Here, dialogs can be reactivated (shown again) which have been disabled before. Many of the program dialogs provide support for the **Do not show again** flag on the left side (text may vary), if the dialog should not be displayed again. This is useful for dialogs that appear again and again showing well-known messages after input errors or due to special program states.

The list shows the dialog type, the description, and the current state of the dialog (enabled/disabled). If the **Do not show again** flag has been selected for a dialog then the flag in the **Enabled** column is not selected here. You may reactivate disabled dialogs by setting the **Enabled** flag so that the dialog will be shown again the next time.

If you like to enabled or disable all dialogs at once please click on the **Enable All** or **Disable All** buttons.

Drag & Drop

Files and directories may be easily exchanged between the [file system window](#), the [network file system window](#) and other docking windows. Here you can configure which drag & drop actions are allowed. Currently the Digester supports dragging files from the file system and network file system window and dropping into the [sync window](#). Source and targets sets may be assembled very easily this way.

- **File system window drag**
- **Network file system window drag**
- **Sync window drop**

The list shows the drag & drop type, the description, and the current state of the action (enabled/disabled). You may reactivate disabled drag & drop actions by setting the **Enabled** flag so that the action is possible again.

If the drag & drop turns out to be not useful for your daily work you may disable drag & drop completely with the **Enable Drag & Drop** flag at the top. If you like to enabled or disable all actions at once please click on the **Enable All** or **Disable All** buttons.

Miscellaneous

Here you can find the settings that do not fit into the other categories:

- **Show controller at x %:** Memory controller in <u>status bar</u> is shown for threshold value
- **Detect and unlock hanging GUI after x seconds**
- **Scale fonts:** Set zoom factor for program texts, 100% = default size
- **Data Lists:** Show top of list or restore previous scroll position
- **Logging:** Adjust log level (output)

Typically the memory controller is displayed if a certain threshold value is reached (default: 75%). Then memory should be released, dialogs should be closed, or the available memory size should be increased by settings a parameter. For extremely high memory consumption near 100% the application should be terminated in order to avoid data loss. Please enter the value 0 here to always display the memory controller, the value 100 never displays this component.

The program may "hang" in certain situations, the GUI does not respond here. You can easily detect such a state with a look at the integrated digital watch. During a hang the time value is not being updated and the value remains. This behavior may be triggered for example if network connections break down or if the program waits for network drives. Many hang situations may be properly detected and unlocked successfully so that working with the program is possible again.

Please enter a plausible numeric value into the **Detect and unlock hanging GUI after x seconds** field (default is 15 seconds). Faulty input is highlighted here. If selected the program creates a dialog in such situations. Customers with Premium Support option may send a <u>status report</u> for the error to be analyzed by the Leisenfels support.

Please enter a plausible numeric value into the **Scale fonts** field (default is 100%). Faulty input is highlighted here. You can use e.g. 125% in order to display larger fonts or e.g. 80% in order to decrease the font size. You can calibrate the font size also with the <u>Fonts</u> menu or with the keyboard shortcuts.

In **Data Lists** you can configure if updated lists should display the top entry or if the previous scroll position should be restored (default). In the **Logging** area the current program log level may be adjusted. This option controls which log messages are being generated: OFF turns logging completely off, ALL generates all possible messages. The log messages are directly displayed by the <u>output window</u>. The log level is primarily controlled by a <u>command line argument</u> so changing this setting takes effect for the current runtime of the program only. The next time the software is started your changes will be reset to the value specified as command line argument.

⚠️ **Caution:** The finer log levels may produce a lot of messages! The levels FINE/FINER/FINEST/ALL should be handled with care.

Support

Here you can configure how <u>status reports</u> are being collected. The list shows the available categories for the status report, the description, as well as the current state, whether the category is to be transferred or not (**Enabled** column). You can reactivate disabled categories by selecting the flag in the **Enabled** column so that data portions of that category are being collected again. If

you like to enabled or disable all categories in the list at once please click on the **Enable All** or **Disable All** buttons.

Currently the following types of data can be transmitted within a status report:

- **Messages**: Messages of the program or own remarks as text
- **Attachments**: Resources of the program or own additional files
- **License Info:** Characteristics about the current license
- **Log Files**: File with messages of the program for error analysis
- **Status Info**: General information about the running program
- **Screen Shots**: Photos of the program windows and the whole screen

> The status report functionality currently is reserved for customers with Premium Support↗ option.

Energy Saving

This section contains the functions to save energy (auto-off). If long running tasks are being executed by the application like the search for files then the program may be configured so that the computer is turned off after all tasks have finished. This makes sense for tasks which are started before closing time and that take some time to finish unattendedly. It would be a waste of energy to keep the computer running the whole night through.

- **Standby Command**: System command to move the computer to standby mode
- **Log Off Command**: System command to log the user off the computer
- **Reboot Command**: System command to restart the computer
- **Shutdown Command**: System command to turn the computer off completely
- **Allow auto-off be cancelled within n seconds**

The commands to be configured here are deposited as far as possible. For single operating systems it may be necessary to adjust the commands for specialties (like being run as administrator). For the standby mode the **Exit** option can be set so that the program is terminated before the computer goes to standby mode.

In order to prevent from accidentally auto-off the program shows a dialog right before the auto-off where the action can be cancelled for a certain period of time. This time to wait in seconds may be adjusted here.

The auto-off function can be started by using the button found in the status bar of the program once tasks are executed in the background.

Additional Links

- Export Preferences
- Import Preferences

- [Save Preferences](#)
- [Tool Bar](#)
- [Shortcuts](#)
- [Generate Window](#)
- [Network File System Window](#)
- [File System Window](#)
- [Regular Expressions](#)
- [Document Window](#)
- [Docking Window](#)
- [Sync Window](#)
- [Edit Connections](#)
- Premium Support↗
- [Send Status Report](#)

Tool Bar
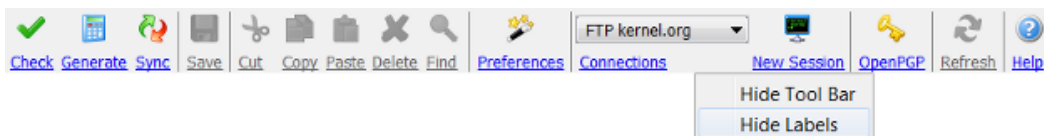
The tool bar is located between the main menu and the [quick check](#) of the [main window](#). There you can find several functions of the main menu which are represented by graphical symbols. Optionally the tool bar displays labels for the buttons. Buttons and labels may be clicked directly in order to call the associated function. Some of the functions are context-sensitive which means they may be disabled depending on the active docking window or program state and may not be clicked then.

The labels below the buttons may be hidden or shown with the context menu of the tool bar (right mouse button). If you like the tool bar not being displayed at all then you could configure this with the main menu **Setup** -> **View** -> **Tool Bar**.

Currently the tool bar provides the following functions:

- **Check:** Open a [file chooser](#) to check digests/signatures
- **Generate:** Start the [wizard](#) to generate digests
- **Sync:** Start the [wizard](#) for the synchronization (create new mapping)
- **Save**: Save current document in [document window](#)
- **Cut**: Delete files/directories, list entries etc.
- **Copy**: Copy text, files, or network files in clipboard depending on the context
- **Paste**: Paste contents from clipboard (texts, files etc.)
- **Delete:** Delete files/directories, list entries etc.
- **Find:** Search for [files](#)/[network files](#), search documents etc.
- **Preferences:** Edit the program [preferences](#)
- **Connections:** Start the [network connection management](#)

- **New Session:** Open a new session tab in the network file system window
- **OpenPGP**: Start the OpenPGP key management
- **Refresh:** Update lists, directories etc. depending on the context
- **Help**: Open the help system at a certain page (context-sensitive)

Many of the functions can also be called with shortcuts (Windows/Mac).

Additional Links

- Main Window
- Quick Check
- Document Window
- Preferences
- Network File System Window
- Shortcuts

Quick Check
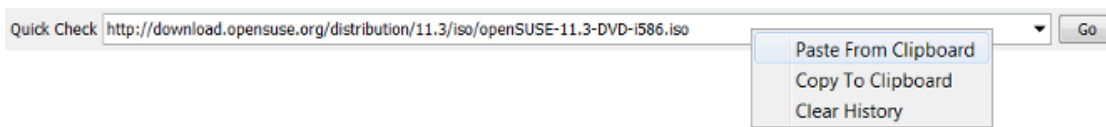
> If you directly click on a digest file in your Internet browser like Firefox the browser downloads this file (e.g. MD5) as a temporary file on your computer and then starts the configured program like Digester. Most digest files do not carry any Internet address information, so the checking must fail with an error message here. This is not the fault of the Digester software but due to the digest files.
>
> **In such cases please use the quick check as follows:**
>
> 1. Open your Internet browser again.
> 2. Copy the link address of the file with the right mouse button.
> 3. Paste the address into the quick check field and click on the **Go** button.

You can find the quick check between tool bar and the docking window area of the Digester main window. This component extends the check window, where files and network files must be selected first from within the file system window or the network file system window. You have to establish network connections first etc. so this mechanism is not the best solution to download and check files from the Internet quickly.

You can enter either a path for a local file on your PC (like `C:\Temp\javainfo.jar`) or a network address like:

```
http://download.opensuse.org/distribution/12.1/iso/openSUSE-12.1-DVD-i586.iso.asc
```

You have the choice to enter a path for a digest or signature file or the path of an original file to be checked. By clicking the **Go** button the Digester software checks for existing digests/signatures for the given file. If this is the case then these files can be used to check the integrity of the referenced original files. The original files like bigger ISO images to burn DVDs may be downloaded and used after the checking has been completed successfully.

After the **Go** button has been clicked (if not disabled) then a download dialog with a list of identified files is being shown (original files like the ISO image in the figure and digests/signatures). By setting the flags in the **Get** column of the list you may schedule the files to be downloaded. If you like to select or deselect all files in the list please click on the **Select All** or **Deselect All** buttons at the bottom. By default all recognized files are automatically downloaded here. If you accept this auto-selection you may disable this dialog by setting the **Do not show again flag**. The files are then directly downloaded and checked.

For the download a download directory must have been specified where to save the files. This directory can be configured in the preferences and set to the following values:

- **Application download directory**
- **Fixed directory**
- **Ask me before a resource is downloaded**

After the download of the files has been completed the digests/signatures are being checked if appropriate digests or signatures have also been downloaded. The checking results are presented then. The downloaded files can be found in the desired directory if the download has been successful.

The context menu of the quick check (right mouse button) makes it possible to insert a path from the clipboard which has been copied from an Internet browser like Firefox (Firefox function **Copy Link Location**). The path within the quick check input field may be copied also into the clipboard to be accessed by other programs, use the context menu for this. The previously edited paths are remembered by the input field so that entries may be used directly the next time the quick check is used. If you like to clear the list please use the context menu **Clear History**.

Additional Links

- Main Window
- Tool Bar
- File System Window
- Network File System Window
- Check Window

Status Bar
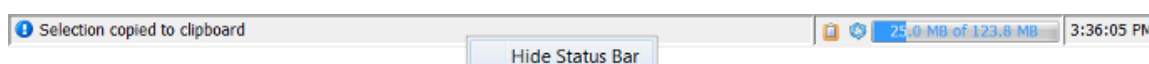
You can find the status bar at the bottom of the [Digester main window](#).

This is the place where messages of the program are displayed on the left. Depending on the type of issue the message symbols vary, thus simple hints and critical warnings can be easily distinguished.

To the right of the message area you can see several icon buttons to release the clipboard and to collect garbage. The memory controller right next to it serves to control the current memory usage of the program. When generating very many message digests e.g. for all system drives, memory usage can grow very rapidly so that the calculation can not be executed. Based on the memory controller display this can be easily detected and the memory size for the program can then be adjusted directly from the software. Please use the **Setup** -> **Memory** -> **Memory Size** menu to do this.

The software starts with a maximum memory size of 128 megabytes, although this size does not exactly correspond with the actual memory size of the program at the level of the operating system which is shown for example by the Windows Task Manager. In the [preferences](#) one can configure if and at which threshold value the memory controller is displayed automatically.

Far right in the status bar there is a digital clock arranged displaying the current date when passing with the mouse.

If you like the status bar not being displayed at all then you can configure this with the **Setup** -> **View** -> **Status Bar** menu item. Alternatively you can use the context menu (right mouse button) to show or hide the status bar (see figure). The memory controller and digital clock each have an own context menu (right mouse button) where you can control these components separately so that the status bar can be easily customized.

Some of the functions can also be called with [shortcuts](#) (Windows/Mac).

Additional Links

- [Main Window](#)
- [Preferences](#)
- [Shortcuts](#)

Arguments for `digestergui`

The Digester software supports numerous arguments when started from the command line (e.g. from MS-DOS prompt). The following list shows an overview of the supported arguments, which can be viewed also directly in the program using the menu **Help** -> **About Digester...** (see **Arguments** tab). Here, also the currently used arguments are shown in the row below the argument list.

All supported options can be passed in any order, but the corresponding values must be in place. All arguments are optional, input errors are detected as far as possible and compensated

automatically. Either the program reports incorrect arguments directly or creates appropriate entries in the log file.

- [Limitations of the Installer](#)
- [Parameters for Web Start](#)

```
Usage: digestergui [-options] <file>
```

```
Supported options include:
```

```
-help | -?          print this help message and exit
-configdir <value>  subdirectory of user's home for configuration
-files <file> ...   list of files to be checked
-headless           no GUI available, do not create dialogs
-locale <value>     set user locale (eng_US|ger_DE|...)
-logconsole         activates the console logging
-logenc <value>     set log encoding (ISO-8859-1|UTF-8|...)
-loglevel <value>   (SEVERE|WARNING|INFO|CONFIG|FINE|FINER|FINEST|ALL|OFF)
-logpipe            pipe stdout and stderr to the logger
-plaf <value>       pluggable look & feel
-theme <value>      theme for pluggable look & feel
-title              title of the main window
-urls <url> ...     list of network resources to be downloaded and checked
-version            print product version and exit
```

-help | -?

Print the list of supported arguments and terminates the application directly. You can use either `-help` or `-?`.

-configdir <value>

Sets the directory to use for storing the configuration of the Digester software. By default, the configuration files are stored within the directory `.digester-1.6.x` in the user's home directory. On Windows, for example `C:\Documents and Settings\User\.digester-1.6.x`

The directory will always be created within the user's home directory, the argument `-configdir .digester` for example forces storage in the directory `C:\Documents and Settings\User\.digester`. Specifying an absolute path to another drive is not possible.

-files <file> ...

Passes one or more files from the local filesystem to be checked by the Digester software. Here you can pass either message digest files or examinee files. The program recognizes the

candidates for checking and automatically starts processing the available digests. The results can be viewed as usual within the Digester software.

Besides this parameter one single file can also be passed directly as a parameter without option. This feature is required to open message digest files directly from the operating system such as Windows. Please note that file associations are not available for all operating systems.

-headless

Starts the application in the command-line mode without calling any functions of the graphical user interface (GUI). This argument can be used under Linux/UNIX for example, if starting the program from a SSH shell causes X server messages like the following being created (e.g. if the arguments should only be listed):

```
X connection to localhost:10.0 broken (explicit kill or server shutdown).
```

-locale <value>

Sets the user language based on ISO 639-1/639-2 language codes and ISO 3166-1 country codes. Possible values are **eng_US** (language = English, country = U.S.A.), **ger_DE** (language = German, country = Germany) etc. If the specified language is not supported, the application falls back to the previous language. If this is not possible, **eng_US** is used since this is the default language for the Digester software.

-logconsole

Enables the logging using the command line. By default only the log file is written. If desired, the log messages can also be printed via the command line using this parameter.

-logenc <value>

Sets the character set for logging. Possible values are ISO-8859-1 (default) or UTF-8. You can use any of the supported encodings here. The list of available charsets can be viewed directly from the program by activating the **Help** -> **About Digester...** menu (see the **Encodings** tab).

-loglevel <value>

Activates the log level in the sense of importance for the created log messages. Possible values are SEVERE, WARNING, INFO, CONFIG, FINE, FINER, FINEST, ALL, and OFF. By default, the Digester software logs with the INFO level, but it may be useful to have a finer log level being enabled (e.g. CONFIG).

-logpipe

Enables forwarding the program output to the logging engine. Thus, all messages generated by the program are written to the log file. This setting usually makes sense, because important messages could be overlooked if the program is executed directly from the command line.

`-plaf <value>`

Sets the pluggable look & feel. The parameter value must be passed in full class notation like

```
com.sun.java.swing.plaf.nimbus.NimbusLookAndFeel
```

for the new Nimbus look & feel of Java (version 1.6.0_10 or later). The list of available look & feels can be viewed directly from the program by activating the menu **Help** -> **About Digester...** (see tab **Look & Feels**). See also the -theme parameter.

`-theme <value>`

Sets the theme for the look & feel. Some look and feels provide support for so-called themes (a.k.a. *skin*). This parameter is only interpreted, if the -plaf parameter is passed as well (see above). Possible values for the look & feel

```
javax.swing.plaf.metal.MetalLookAndFeel
```

to be passed as themes are for example the **Default Metal** and **Ocean** themes. The available themes for the supported look & feels can be viewed directly from the application by activating the menu **Help** -> **About Digester...** (see the **Look & Feels** tab). See also the -plaf parameter.

`-title <value>`

Sets the title of the Digester main window to the desired value. This may be useful if the Digester software is deployed using both Web Start and the installer variants simultaneously. Both applications are titled the same, so that a distinction becomes difficult. With this parameter, you can easily add terms such as "Web Start" to be able to distinguish the application versions.

`-urls <url> ...`

Passes one or more network files to be downloaded and checked by the Digester software. Here you can pass either digest files or alternatively original files. The program detects potential candidates for checking automatically and starts checking the contained digests. The results can be viewed as usual with the Digester software.

A directory is required here, where the files can be downloaded. You can setup this directory in the preferences with the following alternatives:

- **Application download directory**
- **Fixed directory**

- **Ask me before a resource is downloaded**

> BE
> SE
> Please keep in mind that the network functionality is not supported by the Basic Edition and the Standard Edition.

-version

Prints the current program version to standard output and terminates the program immediately.

Limitations of the Installer

The installer software version currently used for the Digester software does not support dynamic command line arguments, so that the default arguments need to be statically compiled and cannot be changed by the user. Nevertheless, it is possible to specify additional parameters.

Please start the Digester software and activate the menu **Help** -> **About Digester...** and see the **Arguments** tab for the parameters currently in use. These parameters cannot be changed at the moment, but the other parameters can be added to the parameter list as described above.

Please note, that the parameters `-help/-?` and `-version` causes the program to print some information to standard output which then terminates. Unfortunately, this output does not become visible for the current installer versions of the software. We are in contact with the manufacturer of the installer software so that the mentioned limitations may be fixed in the near future. Please consider not to use the parameters `-help/-?` and `-version` if not necessary.

Parameters for Web Start

Most of the above mentioned parameters can also be used for the Web Start installations of the software. All arguments can be used as HTTP parameters to be processed by the server logic which can then forward the desired configuration to your local installation (install additional JAR files etc.). A typical Web Start call using HTTP parameters looks as follows:

http://jws.leisenfels.com:8080/digester-1.6.x/install.jnlp?logconsole&locale=ger_DE

In the example, the application is started with command line logging and German language enabled.

Please note, that the parameters `-help/-?` and `-version` causes the program to print some information to standard output which then terminates. Unfortunately, this output does not become visible unless the Web Start console is being activated. And even then, the application will be closed immediately, so that the messages are shown only for a few milliseconds. Please consider not to use the parameters `-help/-?` and `-version` when using the Web Start release of the Digester software.

Additional Links

- Digester Main Window
- Preferences
- Arguments for `digestercli`
- Checking with `digestercli` and Ant
- Generating with `digestercli` and Ant
- Refreshing with `digestercli` and Ant
- Synchonizing with `digestercli` and Ant

Arguments for `digestercli`

The command-line version of the Digester software supports numerous arguments to be passed. The following list shows an overview of the supported arguments. By passing one of the `-check`, `-generate`, `-refresh`, or `-sync` arguments the software is called in the according operational mode. Each mode has its own list of supported parameters which are discussed by the following acticles. This page describes the parameters supported by the `digestercli` application in general (all modes).

- Checking with `digestercli` and Ant
- Generating with `digestercli` and Ant
- Refreshing with `digestercli` and Ant
- Synchronizing with `digestercli` and Ant

All supported options can be passed in any order, but the corresponding values must be in place. All arguments are optional, input errors are detected as far as possible and compensated automatically. Either the program reports incorrect arguments directly or creates appropriate entries in the log file. You can pass files, directories, or URLs to be processed as a space-separated list using the `-targets` argument.

```
Usage: digestercli [-options] -targets <file|directory|url> ...

Supported options include (type 'digestercli -? sync' to see the sync mode options):
```

-help | -? <mode>   print help and exit, modes: check, generate, refresh, sync
-bsdperdirectory   process BSD-compliant file per directory
-bsdpertarget   process BSD-compliant file per target
-calculateonly   no files, just calculate
-check   check message digest files
-checkbsd   check for BSD files
-checkgnu   check for GNU files
-checkintermediates   check intermediate message digests
-checkpgp   check for PGP signature files
-checkxml   check for Digester XML files
-checkxmlpgp   check for Digester XML file signatures
-configdir <value>   subdirectory of user's home for configuration
-download   download URLs and check, checked directly otherwise

-downloaddir <value>   directory to store downloads
-exponentialstart <value>   start creating XML intermediates at byte index
-extmd2 <value>   file extension for MD2
-extmd4 <value>   file extension for MD4
-extmd5 <value>   file extension for MD5
-extpgp <value>   file extension for PGP signatures
-extripemd128 <value>   file extension for RIPEMD-128
-extripemd160 <value>   file extension for RIPEMD-160
-extripemd256 <value>   file extension for RIPEMD-256
-extripemd320 <value>   file extension for RIPEMD-320
-extsha1 <value>   file extension for SHA-1
-extsha224 <value>   file extension for SHA-224
-extsha256 <value>   file extension for SHA-256
-extsha384 <value>   file extension for SHA-384
-extsha512 <value>   file extension for SHA-512
-exttiger <value>   file extension for Tiger
-extwhirlpool <value>   file extension for Whirlpool
-extxml <value>   file extension for Digester XML
-failonerror   exit if an error occurs
-flipsourcetarget <set>   reverse mode: transfer from target set to source
-generate   generate message digests
-gensrctemplate <template>   execute generate template before sync (source set)
-gensrctemplaterefresh   execute template in refresh mode (source set)
-gentemplate <set> <template> execute generate template before sync (named target set)
-gentemplaterefresh <set>   execute template in refresh mode (named target set)
-gnuperdirectory   process MD5SUM-compliant file per directory
-gnupertarget   process MD5SUM-compliant file per target
-headless   no GUI available, do not create dialogs
-hexcase <value>   format of hexadecimal digests (lower|upper)
-ignoredigests   do not generate digests for digest files
-ignoreregex <value>   regular expression for files to be ignored
-includeregex <value>   regular expression for files to be included
-locale <value>   set user locale (eng_US|ger_DE|...)
-maxintermediates <value>   limit the number of XML intermediates
-md2   process MD2 message digests
-md4   process MD4 message digests
-md5   process MD5 message digests
-optdigest   digests for source and target match
-optfilesize   source/target file sizes match
-optintermediate   intermediates for source and target match
-optpgpsig   PGP signatures for source and target match
-opttimestamp   source/target timestamps match
-optxmlentryfilesize   XML entry file sizes match source and target
-optxmlentrytimestamp   XML entry timestamps match source and target
-overwrite   overwrite existing files
-pgpformat <value>   write binary or ASCII PGP signature files (ascii|binary)
-pgpkey <ID> <URL>   use PGP key with ID (required) and URL (optional)
-pgpkeydownload   automatically download unknown PGP keys
-pgpkeyignore   ignore signatures for unknown PGP keys
-pgpkeyinstall   install downloaded PGP keys permanently
-pgppertarget   process PGP signature file per target
-pgpversion <value>   version header for PGP signatures
-refresh   refresh message digest files
-refreshallformats   search files in all formats for existing digests
-refreshgnubsdtimestamp   recycle digest if GNU/BSD file timestamp matches

```
-refreshignoremissing        recycle digest although intermediates are missing
-refreshpgptimestamp         recycle digest if PGP signature file timestamp matches
-refreshremoveartifacts      delete message digest files without targets
-refreshremoveother          delete message digest files in other formats
-refreshxmltimestamp         recycle digest if XML file timestamp matches
-refreshxmlentryfilesize     recycle digest if target XML entry file size matches
-refreshxmlentrytimestamp    recycle digest if target XML entry timestamp matches
-ripemd128           process RIPEMD-128 hash files
-ripemd160           process RIPEMD-160 hash files
-ripemd256           process RIPEMD-256 hash files
-ripemd320           process RIPEMD-320 hash files
-sha1 | sha          process SHA-1 hash files
-sha224              process SHA-224 hash files
-sha256              process SHA-256 hash files
-sha384              process SHA-384 hash files
-sha512              process SHA-512 hash files
-sourcetimestamp             apply source file timestamps to target files
-stoponmismatch              stop calculation on intermediate mismatch
-subdirectories              include subdirectories
-summaryfile <value>         name of the directory files without extension
-sync                synchronize files and directories with message digests
-syncgnubsdtimestamp         trust digests if file timestamp matches (GNU/BSD)
-syncmapping <value>         use sync mapping options as defaults
-syncpgptimestamp            trust signatures if file timestamp matches (PGP)
-syncxmlentryfilesize        trust digests if XML entry file size matches
-syncxmlentrytimestamp       trust digests if XML entry timestamp matches
-syncxmltimestamp            trust digests if file timestamp matches (XML)
-template <value>            use template options as defaults
-tiger               process Tiger hash files
-timestamp           apply target file timestamps for generated files
-transferall         transfer all sources without optimization
-verbose             print additional messages
-whirlpool           process Whirlpool hash files
-version             print product version and exit
-xmlabsolutepaths            add absolute paths for Digester XML files
-xmladdurls          add URL elements for Digester XML files
-xmlcomment <value>          comment for Digester XML files
-xmldigestformat <value>     format of XML file digests (hex|base64)
-xmlintermediates <value>    create XML intermediates (linear|exponential)
-xmllinearevery <value>      create XML intermediates every n bytes
-xmlperdirectory             process Digester XML file per directory
-xmlpertarget                process Digester XML file per target
-xmlpgp              add PGP signatures for Digester XML files
-xmlpgpkeyid         add PGP key IDs for Digester XML files
-xmlpgpkeyname               add PGP key names for Digester XML files
-xmlpgpkeyemail              add PGP key eMail addresses for Digester XML files
-xmlpgpkeyurl                add PGP key network addresses for Digester XML files
-xmlurl <value>              URL address to be added, see xmladdurls
```

-help | -? <Modus>

Print the list of supported arguments and terminates the application directly. You can use either `-help` or `-?`. Please add one of the `check`, `generate`, `refresh,` or `sync` arguments to get the list of supported arguments for the check, generate, refresh, or sync mode of the software.

`-configdir <Wert>`

Sets the directory to use for storing the configuration of the Digester software. By default, the configuration files are stored within the directory `.digester-1.6.x` in the user's home directory. On Windows, for example `C:\Documents and Settings\User\.digester-1.6.x`

The directory will always be created within the user's home directory, the argument `-configdir .digester` for example forces storage in the directory `C:\Documents and Settings\User\.digester`. Specifying an absolute path to another drive is not possible.

`-failonerror`

If an error occurs during processing then the program terminates automatically. This may take a while depending on the application state or processed task. This parameter is useful especially when the software is being executed from Ant build scripts.

`-headless`

Starts the application in the command-line mode without calling any functions of the graphical user interface (GUI). This argument can be used under Linux/UNIX for example, if starting the program from a SSH shell causes X server messages like the following being created (e.g. if the arguments should only be listed):

```
X connection to localhost:10.0 broken (explicit kill or server shutdown).
```

`-locale <Wert>`

Sets the user language based on ISO 639-1/639-2 language codes and ISO 3166-1 country codes. Possible values are **eng_US** (language = English, country = U.S.A.), **ger_DE** (language = German, country = Germany) etc. If the specified language is not supported, the application falls back to the previous language. If this is not possible, **eng_US** is used since this is the default language for the Digester software.

`-version`

Prints the current program version to standard output and terminates the program immediately.

Additional Links

- [Generating with `digestercli` and Ant]

- Generating with `digestercli` and Ant
- Refreshing with `digestercli` and Ant
- Synchronizing with `digestercli` and Ant

Checking with `digestercli` and Ant

- Checking with `digestercli`
- Checking with Ant

Checking with `digestercli`

The Digester software also supports checking from the command-line. You can pass files, directories, or network files to be processed. In theory, all checking operations of the graphical program version can also be done using `digestercli`, so the software can be called from shell scripts for example.

You can either create your program calls manually or let the Digester GUI version assemble the necessary parameters for you. In the latter case, please configure a checking operation with the check window as if you would do some checking from there. Once all options are as desired, let the window assemble an appropriate `digestercli` call by pressing the console symbol at the bottom. This snippet can be easily copied into your own scripts.

All supported options can be passed in any order, but the corresponding values must be in place. All arguments are optional, input errors are detected as far as possible and compensated automatically. Either the program reports incorrect arguments directly or creates appropriate entries in the log file. You can pass files, directories, or URLs to be processed as a space-separated list using the `-targets` argument.

The command-line version of the Digester software currently provides support for the following arguments and values, which may be printed by calling the application using `digestercli -?` `check`:

```
Usage: digestercli [-options] -targets <file|directory|url> ...

Supported check mode options include:

-check                   check message digest files
-checkbsd                check for BSD files
-checkgnu                check for GNU files
-checkintermediates      check intermediate message digests
-checkpgp                check for PGP signature files
-checkxml                check for Digester XML files
-checkxmlpgp             check for Digester XML file signatures
-download                download URLs and check, checked directly otherwise
```

```
-downloaddir <value>          directory to store downloads
-extmd2 <value>               file extension for MD2
-extmd4 <value>               file extension for MD4
-extmd5 <value>               file extension for MD5
-extpgp <value>               file extension for PGP signatures
-extripemd128 <value>         file extension for RIPEMD-128
-extripemd160 <value>         file extension for RIPEMD-160
-extripemd256 <value>         file extension for RIPEMD-256
-extripemd320 <value>         file extension for RIPEMD-320
-extsha1 <value>              file extension for SHA-1
-extsha224 <value>            file extension for SHA-224
-extsha256 <value>            file extension for SHA-256
-extsha384 <value>            file extension for SHA-384
-extsha512 <value>            file extension for SHA-512
-exttiger <value>             file extension for Tiger
-extwhirlpool <value>         file extension for Whirlpool
-extxml <value>               file extension for Digester XML
-ignoreregex <value>          regular expression for files to be ignored
-includeregex <value>         regular expression for files to be included
-md2                          process MD2 message digests
-md4                          process MD4 message digests
-md5                          process MD5 message digests
-overwrite                    overwrite existing files
-pgpkeydownload               automatically download unknown PGP keys
-pgpkeyignore                 ignore signatures for unknown PGP keys
-pgpkeyinstall                install downloaded PGP keys permanently
-ripemd128                    process RIPEMD-128 hash files
-ripemd160                    process RIPEMD-160 hash files
-ripemd256                    process RIPEMD-256 hash files
-ripemd320                    process RIPEMD-320 hash files
-sha1 | sha                   process SHA-1 hash files
-sha224                       process SHA-224 hash files
-sha256                       process SHA-256 hash files
-sha384                       process SHA-384 hash files
-sha512                       process SHA-512 hash files
-stoponmismatch               stop calculation on intermediate mismatch
-subdirectories               include subdirectories
-tiger                        process Tiger hash files
-verbose                      print additional messages
-whirlpool                    process Whirlpool hash files
```

-check

Activate the operation mode to check digests and OpenPGP signatures. The arguments basically are the same as the options shown by the check window of the graphical version of the Digester software.

If the parameters described below are missing, then the check window defaults will be used where possible.You can list the currently used arguments and values with the `-verbose` parameter.

-checkbsd

Activates the checking of digest files in BSD format where the files are recognized based on the configured file extensions (see parameters -`extmd2`, -`extmd4` , -`extmd5` etc.).

-checkgnu

Activates the checking of digest files in GNU format where the files are recognized based on the configured file extensions (see parameter -`extmd2`, -`extmd4` , -`extmd5` etc.).

-checkintermediates

Intermediate digests are checked if this parameter is specified. Optionally the program can be automatically terminated when a mismatch of the calculated digests and the file digests has been detected (see parameter -`stoponmismatch`).

-checkpgp

Activates the checking of OpenPGP signature files for directory based checking where the files are recognized based on the configured file extensions (see parameter -`extpgp`).

-checkxml

Activates the checking of Digester XML files containing digests/signatures for directory based checking where the files are recognized based on the configured file extensions (see parameter -`extxml`).

-checkxmlpgp

Activates the checking of OpenPGP signatures found in Digester XML files. Here, the recognition of XML files must be activated as well (see parameter -`checkxml`) and at least one XML file extension needs to be configured (see parameter -`extxml`).

-download

Activates the download of network resources which have been passed on the command-line and are to be checked (see parameter -<u>downloaddir</u> and -<u>overwrite</u>). The URLs will be downloaded into the configured directory in the first step and then checked in the second step. If this parameter is missing, then URLs are checked directly on the target system, which can be also done using the check functionality from within the <u>network file system window</u>.

The download of OpenPGP keys can be controlled with the -<u>pgpkeydownload</u>, -<u>pgpkeyignore</u> and -<u>pgpkeyinstall</u> parameters.

-downloaddir <value>

Sets the directory to save downloaded files. This directory is used for downloaded OpenPGP keystore files or URLs which have been downloaded for checking automatically (see parameter -<u>download</u> and -<u>overwrite</u>).

-extmd2 <value>

Sets the file extensions for GNU/BSD files containing MD2 message digests (default: `.md2`). You can setup multiple file extensions with a space-separated list here. Please do not use the following characters: \ / : * ? " < > |

-extmd4 <value>

Sets the file extensions for GNU/BSD files containing MD4 message digests (default: `.md4`). You can setup multiple file extensions with a space-separated list here. Please do not use the following characters: \ / : * ? " < > |

-extmd5 <value>

Sets the file extensions for GNU/BSD files containing MD5 message digests (default: `.md5`). You can setup multiple file extensions with a space-separated list here. Please do not use the following characters: \ / : * ? " < > |

-extpgp <value>

Sets the file extensions for signature files containing OpenPGP signatures (default: `.asc .sig`). You can setup multiple file extensions with a space-separated list here. Please do not use the following characters: \ / : * ? " < > |

-extripemd128 <value>

Sets the file extensions for GNU/BSD files containing RIPEMD-128 message digests (default: `.rip128`). You can setup multiple file extensions with a space-separated list here. Please do not use the following characters: \ / : * ? " < > |

-extripemd160 <value>

Sets the file extensions for GNU/BSD files containing RIPEMD-160 message digests (default: `.rip160`). You can setup multiple file extensions with a space-separated list here. Please do not use the following characters: \ / : * ? " < > |

-extripemd256 <value>

Sets the file extensions for GNU/BSD files containing RIPEMD-256 message digests (default: `.rip256`). You can setup multiple file extensions with a space-separated list here. Please do not use the following characters: \ / : * ? " < > |

-extripemd320 <value>

Sets the file extensions for GNU/BSD files containing RIPEMD-320 message digests (default: `.rip320`). You can setup multiple file extensions with a space-separated list here. Please do not use the following characters: \ / : * ? " < > |

-extsha1 <value>

Sets the file extensions for GNU/BSD files containing SHA-1 message digests (default: `.sha` `-sha1`). You can setup multiple file extensions with a space-separated list here. Please do not use the following characters: \ / : * ? " < > |

-extsha224 <value>

Sets the file extensions for GNU/BSD files containing SHA-224 message digests (default: `.sha224`). You can setup multiple file extensions with a space-separated list here. Please do not use the following characters: \ / : * ? " < > |

-extsha256 <value>

Sets the file extensions for GNU/BSD files containing SHA-256 message digests (default: `.sha256`). You can setup multiple file extensions with a space-separated list here. Please do not use the following characters: \ / : * ? " < > |

-extsha384 <value>

Sets the file extensions for GNU/BSD files containing SHA-384 message digests (default: `.sha384`). You can setup multiple file extensions with a space-separated list here. Please do not use the following characters: \ / : * ? " < > |

-extsha512 <value>

Sets the file extensions for GNU/BSD files containing SHA-512 message digests (default: `.sha512`). You can setup multiple file extensions with a space-separated list here. Please do not use the following characters: \ / : * ? " < > |

-exttiger <value>

Sets the file extensions for GNU/BSD files containing Tiger message digests (default: `.tiger`). You can setup multiple file extensions with a space-separated list here. Please do not use the following characters: \ / : * ? " < > |

-extwhirlpool <value>

Sets the file extensions for GNU/BSD files containing Whirlpool message digests (default: `.wpool`). You can setup multiple file extensions with a space-separated list here. Please do not use the following characters: \ / : * ? " < > |

-extxml <value>

Sets the file extensions for Digester XML files containing digests/signatures (default: `.digest`). You can setup multiple file extensions with a space-separated list here. Please do not use the following characters: \ / : * ? " < > |

-ignoreregex <value>

Sets a [regular expression](#) to ignore certain files when searching for digests/signatures (not processed). Alternatively you can include certain files using the parameter -`includeregex`.

-includeregex <value>

Sets a [regular expression](#) to include certain files when searching for digests/signatures (are being processed). Alternatively you can ignore certain files using the parameter -`ignoreregex`.

-md2

Activates the recognition of GNU/BSD files containing MD2 digests for directory-based checking where the files are recognized based on the configured file extensions (see parameter -`extmd2`).

–md4

Activates the recognition of GNU/BSD files containing MD4 digests for directory-based checking where the files are recognized based on the configured file extensions (see parameter -extmd4).

–md5

Activates the recognition of GNU/BSD files containing MD5 digests for directory-based checking where the files are recognized based on the configured file extensions (see parameter -extmd5).

-overwrite

Existing files are overwritten, if downloaded files are to be stored into the configured download directory (see parameters -download and -downloaddir).

> ⚠️   You will not be able to restore overwritten download files with the program, please backup your data first.

-pgpkeydownload

Activates the automatic download of OpenPGP keys into the download directory (see parameters -downloaddir and -overwrite). Network addresses for the download of keystore files are only supported by the Digester XML format.

OpenPGP keys are downloaded when required keys are not yet available within the key management of the Digester software (unknown). If downloaded keys shall be imported permanently you can additionally use the -pgpkeyinstall parameter. In order to ignore unknown keys please set the -pgpkeyignore argument.

The download of network resources for checking can be controlled with the -download parameter.

-pgpkeyignore

Unknown OpenPGP keys are simply ignored (no download). Signatures cannot be calculated and checked then. In this case, the required keys must be installed manually for example using the key management functionality of the Digester software. In order to download unknown keys automatically, please set the -pgpkeydownload instead and consider specifying the -pgpkeyinstall parameter as well.

-pgpkeyinstall

Downloaded keys are being installed permanently into the key management functionality of the Digester software. Those keys become directly available for the next checks (no downloads). In order to download unknown keys automatically, the -pgpkeydownload argument must be set, too.

-ripemd128

Activates the recognition of GNU/BSD files containing RIPEMD-128 digests for directory-based checking where the files are recognized based on the configured file extensions (see parameter -<ins>extripemd128</ins>).

-ripemd160

Activates the recognition of GNU/BSD files containing RIPEMD-160 digests for directory-based checking where the files are recognized based on the configured file extensions (see parameter -<ins>extripemd160</ins>).

-ripemd256

Activates the recognition of GNU/BSD files containing RIPEMD-256 digests for directory-based checking where the files are recognized based on the configured file extensions (see parameter -<ins>extripemd256</ins>).

-ripemd320

Activates the recognition of GNU/BSD files containing RIPEMD-320 digests for directory-based checking where the files are recognized based on the configured file extensions (see parameter -<ins>extripemd320</ins>).

-sha1 | sha

Activates the recognition of GNU/BSD files containing SHA-1 digests for directory-based checking where the files are recognized based on the configured file extensions (see parameter -<ins>extsha1</ins>).

-sha224

Activates the recognition of GNU/BSD files containing SHA-224 digests for directory-based checking where the files are recognized based on the configured file extensions (see parameter -<ins>extsha224</ins>).

-sha256

Activates the recognition of GNU/BSD files containing SHA-256 digests for directory-based checking where the files are recognized based on the configured file extensions (see parameter -<ins>extsha256</ins>).

-sha384

Activates the recognition of GNU/BSD files containing SHA-384 digests for directory-based checking where the files are recognized based on the configured file extensions (see parameter -extsha384).

-sha512

Activates the recognition of GNU/BSD files containing SHA-512 digests for directory-based checking where the files are recognized based on the configured file extensions (see parameter -extsha512).

-stoponmismatch

Terminates the checking if a mismatch of the calculated intermediate digests and the file intermediates is detected (see parameter -checkintermediates). The other algorithms will then not be calculated as well.

-subdirectories

Activates the processing of subdirectories for directory-based checking. Otherwise only the specified directories themselves (without subdirectories) are being processed.

-tiger

Activates the recognition of GNU/BSD files containing Tiger digests for directory-based checking where the files are recognized based on the configured file extensions (see parameter -exttiger).

-verbose

Create more detailed messages. Especially all passed arguments and values are presented as a list for control purposes. For many program settings defaults are being used if a parameter is not specified. This functionality is useful if templates are used as default values where the current settings are to be controlled right before the processing starts.

-whirlpool

Activates the recognition of GNU/BSD files containing Whirlpool digests for directory-based checking where the files are recognized based on the configured file extensions (see parameter -extwhirlpool).

Checking with Ant

The Digester software also supports checking from Ant build scripts. You can pass files, directories, or network files to be processed. In theory, all check operations of the graphical program version can also be done using Apache Ant, so the software can be called from own builds scripts for example.

You can either create your XML tags manually or let the Digester GUI version assemble the necessary elements and attributes for you. In the latter case, please configure a check operation with the [check window](#) as if you would do some checking from there. Once all options are as desired, let the window assemble an appropriate Ant build script code by pressing the ant symbol at the bottom. This snippet can be easily copied into your own build scripts.

All supported options can be passed in any order, but the corresponding values must be in place. All arguments are optional, input errors are detected as far as possible and compensated automatically. The program reports incorrect arguments directly and terminates if you have specified the `failonerror` tag.

For the checking with Ant all arguments of the command-line version of the Digester software `digestercli` can be used which are represented by separate XML elements each. In the header of your build script the task `digestercli` must be declared. Please adjust the `classpath` attribute by the value appropriate for your installation. The Web Start version cannot be used for checking from Ant build scripts.

```
<taskdef name="digestercli"
   classname="com.lf.digester.ant.DigesterCLITask"
   classpath="C:\Programme\digester-1.6.3\ant\digester-ant.jar"/>
```

An example XML code for Ant build scripts looks like the following where a single Digester XML file is being checked:

```
<digestercli>
   <check />
   <checkxml />
   <extxml>.digest</extxml>
   <checkxmlpgp />
   <checkintermediates />
   <stoponmismatch />
   <pgpkeyignore />
   <fileset dir="D:\checksum_test">
       <include name="jnlp-1_5-mr-spec.pdf.digest" />
   </fileset>
</digestercli>
```

Additional Links

- [Arguments for `digestercli`](#)

Generating with `digestercli` and Ant

- [Generating with `digestercli`](#)
- [Generating with Ant](#)

Generating with `digestercli`

The Digester software also supports generating from the command-line. You can pass files, directories, or network files to be processed. In theory, all generating operations of the graphical program version can also be done using `digestercli`, so the software can be called from shell scripts for example.

You can either create your program calls manually or let the Digester GUI version assemble the necessary parameters for you. In the latter case, please configure a checking operation with the [generate window](#) as if you would do some generating from there. Once all options are as desired, let the window assemble an appropriate `digestercli` call by pressing the console symbol at the bottom. This snippet can be easily copied into your own scripts.

All supported options can be passed in any order, but the corresponding values must be in place. All arguments are optional, input errors are detected as far as possible and compensated automatically. Either the program reports incorrect arguments directly or creates appropriate entries in the log file. You can pass files, directories, or URLs to be processed as a space-separated list using the `-targets` argument.

The command-line version of the Digester software currently provides support for the following arguments and values, which may be printed by calling the application using `digestercli -? generate`:

```
Usage: digestercli [-options] -targets <file|directory|url> ...

Supported generate mode options include:

-generate                     generate message digests

-bsdperdirectory              process BSD-compliant file per directory
```

```
-bsdpertarget              process BSD-compliant file per target
-calculateonly             no files, just calculate
-exponentialstart <value>  start creating XML intermediates at byte index
-extmd2 <value>            file extension for MD2
-extmd4 <value>            file extension for MD4
-extmd5 <value>            file extension for MD5
-extpgp <value>            file extension for PGP signatures
-extripemd128 <value>      file extension for RIPEMD-128
-extripemd160 <value>      file extension for RIPEMD-160
-extripemd256 <value>      file extension for RIPEMD-256
-extripemd320 <value>      file extension for RIPEMD-320
-extsha1 <value>           file extension for SHA-1
-extsha224 <value>         file extension for SHA-224
-extsha256 <value>         file extension for SHA-256
-extsha384 <value>         file extension for SHA-384
-extsha512 <value>         file extension for SHA-512
-exttiger <value>          file extension for Tiger
-extwhirlpool <value>      file extension for Whirlpool
-extxml <value>            file extension for Digester XML
-gnuperdirectory           process MD5SUM-compliant file per directory
-gnupertarget              process MD5SUM-compliant file per target
-hexcase <value>           format of hexadecimal digests (lower|upper)
-ignoredigests             do not generate digests for digest files
-ignoreregex <value>       regular expression for files to be ignored
-includeregex <value>      regular expression for files to be included
-maxintermediates <value>  limit the number of XML intermediates
-md2                       process MD2 message digests
-md4                       process MD4 message digests
-md5                       process MD5 message digests
-overwrite                 overwrite existing files
-pgpformat <value>         write binary or ASCII PGP signature files (ascii|binary)
-pgpkey <id> <url>         use PGP key with ID (required) and URL (optional)
-pgppertarget              process PGP signature file per target
-pgpversion <value>        version header for PGP signatures
-ripemd128                 process RIPEMD-128 hash files
-ripemd160                 process RIPEMD-160 hash files
-ripemd256                 process RIPEMD-256 hash files
-ripemd320                 process RIPEMD-320 hash files
-sha1 | sha                process SHA-1 hash files
-sha224                    process SHA-224 hash files
-sha256                    process SHA-256 hash files
-sha384                    process SHA-384 hash files
-sha512                    process SHA-512 hash files
-subdirectories            include subdirectories
```

```
-summaryfile <value>          name of the directory files without extension
-template <value>             use template options as defaults
-tiger                        process Tiger hash files
-timestamp                    apply target file timestamps for generated files
-verbose                      print additional messages
-whirlpool                    process Whirlpool hash files
-xmlabsolutepaths             add absolute paths for Digester XML files
-xmladdurls                   add URL elements for Digester XML files
-xmlcomment <value>           comment for Digester XML files
-xmldigestformat <value>      format of XML file digests (hex|base64)
-xmlintermediates <value>     create XML intermediates (linear|exponential)
-xmllinearevery <value>       create XML intermediates every n bytes
-xmlperdirectory              process Digester XML file per directory
-xmlpertarget                 process Digester XML file per target
-xmlpgp                       add PGP signatures for Digester XML files
-xmlpgpkeyid                  add PGP key IDs for Digester XML files
-xmlpgpkeyname                add PGP key names for Digester XML files
-xmlpgpkeyemail               add PGP key eMail addresses for Digester XML files
-xmlpgpkeyurl                 add PGP key network addresses for Digester XML files
-xmlurl <value>               URL address to be added, see xmladdurls
```

-generate

Activate the operation mode to generate digests and OpenPGP signatures. The arguments basically are the same as the options shown by the generate window of the graphical version of the Digester software.

If the parameters described below are missing, then the generate window defaults will be used where possible.You can list the currently used arguments and values with the `-verbose` parameter.

-bsdperdirectory

Create digest files in BSD format for each directory. All digests generated for the original files in a single directory are stored into one single digest file. The algorithms can be specified using the parameter -md5 for MD5 digests etc. The files are named with the extension specified by the -extmd5 parameter etc. and the file name specified with the -summaryfile parameter. Alternatively you can use the -gnuperdirectory argument but not both since the same files are being created.

In addition you can let the software create BSD files for each original file (see parameter -bsdpertarget).

-bsdpertarget

Create digest files in BSD format for each original file. For each of the calculated original files one single digest file is being created per algorithm (e.g. -md5 for MD5). The files are named with the extension specified by the -extmd5 parameter etc. Alternatively you can use the -gnupertarget argument but not both since the same files are being created.

In addition you can let the software create BSD files on a directory basis (see parameter -bsdperdirectory).

-calculateonly

No digest/signature files are being created. Instead, the values are only calculated and presented after calculation. Set the -verbose argument as well, if you like more detailed messages being created.

-exponentialstart <Wert>

This option allows intermediate digests to be created starting at a certain position (e.g. 8K = 8 kilobytes). The next intermediates will then be extracted in steps after the next 16 kilobytes, 32 kilobytes etc. You can use numeric values or abbreviations like (K)ilobytes, (M)egabytes, (G)igabytes and (T)erabytes.

This argument is evaluated only if intermediate digests for Digester XML are being calculated (see parameter -xmlintermediates) and if the exponential mode has been specified.

-extmd2 <value>

Sets the file extension for GNU/BSD files containing MD2 message digests (default: .md2). Please do not use the following characters: \ / : * ? " < > |

-extmd4 <value>

Sets the file extension for GNU/BSD files containing MD4 message digests (default: .md4). Please do not use the following characters: \ / : * ? " < > |

-extmd5 <value>

Sets the file extension for GNU/BSD files containing MD5 message digests (default: .md5). Please do not use the following characters: \ / : * ? " < > |

-extpgp <value>

Sets the file extension for signature files containing OpenPGP signatures (default: .asc). Please do not use the following characters: \ / : * ? " < > |

-extripemd128 <value>

Sets the file extension for GNU/BSD files containing RIPEMD-128 message digests (default: `.rip128`). Please do not use the following characters: \ / : * ? " < > |

-extripemd160 <value>

Sets the file extension for GNU/BSD files containing RIPEMD-160 message digests (default: `.rip160`). Please do not use the following characters: \ / : * ? " < > |

-extripemd256 <value>

Sets the file extension for GNU/BSD files containing RIPEMD-256 message digests (default: `.rip256`). Please do not use the following characters: \ / : * ? " < > |

-extripemd320 <value>

Sets the file extension for GNU/BSD files containing RIPEMD-320 message digests (default: `.rip320`). Please do not use the following characters: \ / : * ? " < > |

-extsha1 <value>

Sets the file extension for GNU/BSD files containing SHA-1 message digests (default: `.sha`). Please do not use the following characters: \ / : * ? " < > |

-extsha224 <value>

Sets the file extension for GNU/BSD files containing SHA-224 message digests (default: `.sha224`). Please do not use the following characters: \ / : * ? " < > |

-extsha256 <value>

Sets the file extension for GNU/BSD files containing SHA-256 message digests (default: `.sha256`). Please do not use the following characters: \ / : * ? " < > |

-extsha384 <value>

Sets the file extension for GNU/BSD files containing SHA-384 message digests (default: `.sha384`). Please do not use the following characters: \ / : * ? " < > |

-extsha512 <value>

Sets the file extension for GNU/BSD files containing SHA-512 message digests (default: `.sha512`). Please do not use the following characters: \ / : * ? " < > |

-exttiger <value>

Sets the file extension for GNU/BSD files containing Tiger message digests (default: `.tiger`). Please do not use the following characters: \ / : * ? " < > |

-extwhirlpool <value>

Sets the file extension for GNU/BSD files containing Whirlpool message digests (default: `.wpool`). Please do not use the following characters: \ / : * ? " < > |

-extxml <value>

Sets the file extension for Digester XML files containing digests/signatures (default: `.digest`). Please do not use the following characters: \ / : * ? " < > |

-gnuperdirectory

Create digest files in GNU format for each directory. All digests generated for the original files in a single directory are stored into one single digest file. The algorithms can be specified using the parameter -md5 for MD5 digests etc. The files are named with the extension specified by the -extmd5 parameter etc. and the file name specified with the -summaryfile parameter. Alternatively you can use the -bsdperdirectory argument but not both since the same files are being created.

In addition you can let the software create GNU files for each original file (see parameter -gnupertarget).

-gnupertarget

Create digest files in GNU format for each original file. For each of the calculated original files one single digest file is being created per algorithm (e.g. -md5 for MD5). The files are named with the extension specified by the -extmd5 parameter etc. Alternatively you can use the -bsdpertarget argument but not both since the same files are being created.

In addition you can let the software create GNU files on a directory basis (see parameter -gnuperdirectory).

-hexcase <value>

Sets the case of writing of hexadecimal digest values (`lower` = lower-case, `upper` = upper-case).

-ignoredigests

Files containing digests/signatures are not used to calculate additional digests/signatures. This means that no MD5 files are being created for existing MD5 files etc. This argument should be used by default.

-ignoreregex <value>

Sets a [regular expression](#) to ignore certain files when searching for digests/signatures (not processed). Alternatively you can include certain files using the parameter -includeregex.

-includeregex <value>

Sets a [regular expression](#) to include certain files when searching for digests/signatures (are being processed). Alternatively you can ignore certain files using the parameter -ignoreregex.

-maxintermediates <value>

Use this option to limit the number of calculated intermediate digests.

This argument is evaluated only if intermediate digests for Digester XML are being calculated (see parameter -xmlintermediates) and for both the exponential and linear mode.

-md2

Activates the generation of MD2 message digests. The files are created based on the configured file extensions (see parameter -extmd2).

-md4

Activates the generation of MD4 message digests. The files are created based on the configured file extensions (see parameter -extmd4).

-md5

Activates the generation of MD5 message digests. The files are created based on the configured file extensions (see parameter -extmd5).

-overwrite

Existing digest/signature files are overwritten when the calculation has been finished.

> ⚠️ You will not be able to restore overwritten digest/signature files with the program, please backup your data first.

-pgpformat <value>

Specify `ascii` here to create OpenPGP files in ASCII format (text) or `binary` for the binary format.

This argument is evaluated only if OpenPGP signatures are being calculated.

-pgpkey <id> <url>

Use the OpenPGP key with the given ID (required) and the key URL (optional) to calculate signatures. The ID must be a full hexadecimal identifier like `0x2452F30BFCE8C30F` in order to identify the key. The URL value is optional and used if the additional -<ins>xmlpgpkeyurl</ins> parameter has been specified (only for Digester XML).

This argument is evaluated only if OpenPGP signatures are being calculated.

-pgppertarget

Create OpenPGP signature files for each original file. For each of the calculated original files one single signature file is being created containing a value for each key. The files are named with the extension specified by the -<ins>extpgp</ins> parameter.

Creating signature files on a directory basis is not possible for OpenPGP signatures.

-pgpversion <value>

In order to label OpenPGP signatures files one can specify an arbitrary single-line text to be stored in the signature file headers. Here, you can set your company's or organization's name for example.

-ripemd128

Activates the generation of RIPEMD-128 message digests. The files are created based on the configured file extensions (see parameter -<ins>extripemd128</ins>).

-ripemd160

Activates the generation of RIPEMD-160 message digests. The files are created based on the configured file extensions (see parameter -<ins>extripemd160</ins>).

-ripemd256

Activates the generation of RIPEMD-256 message digests. The files are created based on the configured file extensions (see parameter -<ins>extripemd256</ins>).

-ripemd320

Activates the generation of RIPEMD-320 message digests. The files are created based on the configured file extensions (see parameter -extripemd320).

-sha1 | sha

Activates the generation of SHA-1 message digests. The files are created based on the configured file extensions (see parameter -extsha1).

-sha224

Activates the generation of SHA-224 message digests. The files are created based on the configured file extensions (see parameter -extsha224).

-sha256

Activates the generation of SHA-256 message digests. The files are created based on the configured file extensions (see parameter -extsha256).

-sha384

Activates the generation of SHA-384 message digests. The files are created based on the configured file extensions (see parameter -extsha384).

-sha512

Activates the generation of SHA-512 message digests. The files are created based on the configured file extensions (see parameter -extsha512).

-subdirectories

Activates the processing of subdirectories for directory-based generating. Otherwise only the specified directories themselves (without subdirectories) are being processed.

-summaryfile <value>

File names for digest files on the directory basis (default: summary). The file extensions are then chosen according to the format of the digest files (for example see parameter -extxml).

-template <value>

The values of the given template will be used as defaults and overwritten by the specified parameters. By default the options of the current template selected in the generate window will be used as defaults.

-tiger

Activates the generation of Tiger message digests. The files are created based on the configured file extensions (see parameter -exttiger).

-timestamp

Timestamps of the processed original files are applied to the created digest/signature files. This is only possible for GNU/BSD/OpenPGP/XML files per target, not for files created on the directory basis (per directory).

-verbose

Create more detailed messages. Especially all passed arguments and values are presented as a list for control purposes. For many program settings defaults are being used if a parameter is not specified. This functionality is useful if templates are used as default values where the current settings are to be controlled right before the processing starts.

-whirlpool

Activates the generation of Whirlpool message digests. The files are created based on the configured file extensions (see parameter -extwhirlpool).

-xmlabsolutepaths

Special option for Digester XML files: add absolute paths of the processed original files. With this option all digests hold the absolute reference on the computer, e.g. `C:\Temp\franz.txt` under Window. Even after copying such XML files to another drive of the same computer these paths may be resolved properly allowing to find the original files for the digests. By using this option the Digester XML files become slightly bigger.

-xmladdurls

Special option for Digester XML files: add fixed network address or SSI pattern for each original file. The value is passed using the -xmlurl parameter.

-xmlcomment <value>

Optional text for the XML file header. This argument is evaluated only if Digester XML files are being generated. Use the comment to deposit your own name or your company's name in order to label the XML files.

-xmldigestformat <value>

Set the encoding format of hexadecimal digests in Digester XML files (`hex` = hexadecimal, `base64` = Base64 encoding). See the -`hexcase` parameter to control the case of writing (lower/upper) of hexadecimal values.

-xmlintermediates <value>

Create intermediate message digests, deactivated otherwise. Use the `linear` value to extract intermediates in fixed intervals or `exponential` to extract intermediates dynamically after blocks like 8 kilobytes, then 16 kilobytes etc.

Additional parameters to control the intermediate digest generation are -`xmllinearevery`, -`maxintermediates` and -`exponentialstart`.

-xmllinearevery <value>

This option allows intermediate digests to be extracted after fixed intervals. The interval size can be specified with values like 10M where intermediates are extracted every 10 megabytes. You can use numeric values or abbreviations like (K)ilobytes, (M)egabytes, (G)igabytes and (T)erabytes.

This argument is evaluated only if intermediate digests for Digester XML are being calculated (see parameter -`xmlintermediates`) and if the linear mode has been specified.

-xmlperdirectory

Create digest/signature files in Digester XML format for each directory. All digests generated for the original files in a single directory are stored into one single digest file. The algorithms can be specified using the parameter -`md5` for MD5 digests etc. The files are named with the extension specified by the -`extxml` parameter and the file name specified with the -`summaryfile` parameter.

In addition you can let the software create Digester XML files for each original file (see parameter -`xmlpertarget`).

-xmlpertarget

Create digest/signature files in Digester XML format for each original file. For each of the calculated original files one single XML file is being created containing the values for the chosen algorithms (e.g. -`md5` for MD5) as well as the OpenPGP signatures. The files are named with the extension specified by the -`extxml` parameter.

In addition you can let the software create XML files on a directory basis (see parameter -<u>xmlperdirectory</u>).

-xmlpgp

If Digester XML files are being created and OpenPGP keys have been specified to calcuate signatures then the signature values may be added to the Digester XML files. Signatures are represented by separate XML elements which are stored on the level together with digests (see <u>DTD Digester XML</u>).

-xmlpgpkeyid

In addition to the OpenPGP signatures themselves Digester XML files can contain the IDs of the OpenPGP keys used for signature generation. The IDs can help importing software to identify or download appropriate public keys to check the signatures (siehe Parameter -<u>xmlpgp</u>).

-xmlpgpkeyname

In addition to the OpenPGP signatures themselves Digester XML files can contain the names of the OpenPGP keys used for signature generation (siehe Parameter -<u>xmlpgp</u>).

-xmlpgpkeyemail

In addition to the OpenPGP signatures themselves Digester XML files can contain the eMail addresses of the OpenPGP keys used for signature generation (siehe Parameter -<u>xmlpgp</u>).

-xmlpgpkeyurl

In addition to the OpenPGP signatures themselves Digester XML files can contain the network addresses of the OpenPGP keys used for signature generation. The URLs can help importing software to identify or download appropriate public keys to check the signatures (siehe Parameter -<u>xmlpgp</u>).

-xmlurl <value>

Special option for Digester XML files: add fixed network address or SSI pattern. Here you can specify a static URL pattern e.g. for SSI (*Server Side Includes*) which can be processed by web servers like the Apache HTTP server. Such URLs can be translated dynamically into the appropriate URLs for the original files on the server. If Digester XML files are being downloaded with the software, then the referenced files can be identified using the URL entries to be downloaded, checked, and processed in other ways.

This argument is evaluated only if also -<u>xmladdurls</u> is passed.

Generating with Ant

The Digester software also supports generating from Ant build scripts. You can pass files, directories, or network files to be processed. In theory, all generate operations of the graphical program version can also be done using Apache Ant, so the software can be called from own builds scripts for example.

You can either create your XML tags manually or let the Digester GUI version assemble the necessary elements and attributes for you. In the latter case, please configure a generate operation with the generate window as if you would do some generating from there. Once all options are as desired, let the window assemble an appropriate Ant build script code by pressing the ant symbol at the bottom. This snippet can be easily copied into your own build scripts.

All supported options can be passed in any order, but the corresponding values must be in place. All arguments are optional, input errors are detected as far as possible and compensated automatically. The program reports incorrect arguments directly and terminates if you have specified the `failonerror` tag.

For the generating with Ant all arguments of the command-line version of the Digester software `digestercli` can be used which are represented by separate XML elements each. In the header of your build script the task `digestercli` must be declared. Please adjust the `classpath` attribute by the value appropriate for your installation. The Web Start version cannot be used for checking from Ant build scripts.

```
<taskdef name="digestercli"
    classname="com.lf.digester.ant.DigesterCLITask"
    classpath="C:\Programme\digester-1.6.3\ant\digester-ant.jar"/>
```

An example XML code for Ant build scripts looks like the following where a single Digester XML file is being created:

```
<digestercli>
    <generate />
    <md5 />
    <extmd5>.md5</extmd5>
    <sha1 />
    <extsha1>.sha</extsha1>
    <xmlpertarget />
    <overwrite />
    <timestamp />
    <ignoredigests />
    <subdirectories />
    <extxml>.digest</extxml>
    <xmldigestformat>hex</xmldigestformat>
    <xmlabsolutepaths />
    <xmlintermediates>exponential</xmlintermediates>
```

```
    <exponentialstart>8K</exponentialstart>

    <maxintermediates>5</maxintermediates>

    <fileset dir="D:\checksum_test">

        <include name="franz.txt" />

    </fileset>

</digestercli>
```

Additional Links

- Arguments for `digestercli`
- Checking with `digestercli` and Ant
- Refreshing with `digestercli` and Ant
- Synchronizing with `digestercli` and Ant
- Arguments for `digestergui`
- Generate Window
- Supported File Formats
- DTD Digester XML

Refreshing with `digestercli` and Ant

- Refreshing with `digestercli`
- Refreshing with Ant

Refreshing with `digestercli`

The Digester software also supports refreshing from the command-line. You can pass files, directories, or network files to be processed. In theory, all refresh operations of the graphical program version can also be done using `digestercli`, so the software can be called from shell scripts for example.

You can either create your program calls manually or let the Digester GUI version assemble the necessary parameters for you. In the latter case, please configure a refresh operation with the generate window as if you would do some refreshing from there. Once all options are as desired, let the window assemble an appropriate `digestercli` call by pressing the console symbol at the bottom. This snippet can be easily copied into your own scripts.

All supported options can be passed in any order, but the corresponding values must be in place. All arguments are optional, input errors are detected as far as possible and compensated automatically. Either the program reports incorrect arguments directly or creates appropriate entries in the log file. You can pass files, directories, or URLs to be processed as a space-separated list using the `-targets` argument.

The command-line version of the Digester software currently provides support for the following arguments and values, which may be printed by calling the application using `digestercli -?` `refresh`:

```
Usage: digestercli [-options] -targets <file|directory|url> ...

Supported refresh mode options include:
```

-refresh   refresh message digest files
-bsdperdirectory   process BSD-compliant file per directory
-bsdpertarget   process BSD-compliant file per target
-exponentialstart <value>   start creating XML intermediates at byte index
-extmd2 <value>   file extension for MD2
-extmd4 <value>   file extension for MD4
-extmd5 <value>   file extension for MD5
-extpgp <value>   file extension for PGP signatures
-extripemd128 <value>   file extension for RIPEMD-128
-extripemd160 <value>   file extension for RIPEMD-160
-extripemd256 <value>   file extension for RIPEMD-256
-extripemd320 <value>   file extension for RIPEMD-320
-extsha1 <value>   file extension for SHA-1
-extsha224 <value>   file extension for SHA-224
-extsha256 <value>   file extension for SHA-256
-extsha384 <value>   file extension for SHA-384
-extsha512 <value>   file extension for SHA-512
-exttiger <value>   file extension for Tiger
-extwhirlpool <value>   file extension for Whirlpool
-extxml <value>   file extension for Digester XML
-gnuperdirectory   process MD5SUM-compliant file per directory
-gnupertarget   process MD5SUM-compliant file per target
-hexcase <value>   format of hexadecimal digests (lower|upper)
-ignoredigests   do not generate digests for digest files
-ignoreregex <value>   regular expression for files to be ignored
-includeregex <value>   regular expression for files to be included
-maxintermediates <value>   limit the number of XML intermediates
-md2   process MD2 message digests
-md4   process MD4 message digests
-md5   process MD5 message digests
-overwrite   overwrite existing files
-pgpformat <value>   write binary or ASCII PGP signature files (ascii|binary)
-pgpkey <id> <url>   use PGP key with ID (required) and URL (optional)
-pgppertarget   process PGP signature file per target
-pgpversion <value>   version header for PGP signatures
-refreshallformats   search files in all formats for existing digests
-refreshremoveother   delete message digest files in other formats
-refreshremoveartifacts   delete message digest files without targets
-refreshignoremissing   recycle digest although intermediates are missing
-refreshgnubsdtimestamp   recycle digest if GNU/BSD file timestamp matches
-refreshpgptimestamp   recycle digest if PGP signature file timestamp matches
-refreshxmltimestamp   recycle digest if XML file timestamp matches
-refreshxmlentryfilesize recycle digest if target XML entry file size matches
-refreshxmlentrytimestamp recycle digest if target XML entry timestamp matches
-ripemd128   process RIPEMD-128 hash files
-ripemd160   process RIPEMD-160 hash files

-ripemd256   process RIPEMD-256 hash files
-ripemd320   process RIPEMD-320 hash files
-sha1 | sha   process SHA-1 hash files
-sha224   process SHA-224 hash files
-sha256   process SHA-256 hash files
-sha384   process SHA-384 hash files
-sha512   process SHA-512 hash files
-subdirectories   include subdirectories
-summaryfile <value>   name of the directory files without extension
-template <value>   use template options as defaults
-tiger   process Tiger hash files
-timestamp   apply target file timestamps for generated files
-verbose   print additional messages
-whirlpool   process Whirlpool hash files
-xmlabsolutepaths   add absolute paths for Digester XML files
-xmladdurls   add URL elements for Digester XML files
-xmlcomment <value>   comment for Digester XML files
-xmldigestformat <value>   format of XML file digests (hex|base64)
-xmlintermediates <value>   create XML intermediates (linear|exponential)
-xmllinearevery <value>   create XML intermediates every n bytes
-xmlperdirectory   process Digester XML file per directory
-xmlpertarget   process Digester XML file per target
-xmlpgp   add PGP signatures for Digester XML files
-xmlpgpkeyid   add PGP key IDs for Digester XML files
-xmlpgpkeyname   add PGP key names for Digester XML files
-xmlpgpkeyemail   add PGP key eMail addresses for Digester XML files
-xmlpgpkeyurl   add PGP key network addresses for Digester XML files
-xmlurl <value>   URL address to be added, see xmladdurls

-refresh

Activate the operation mode to refresh digests and OpenPGP signatures. The arguments basically are the same as the options shown by the generate window of the graphical version of the Digester software.

If the parameters described below are missing, then the generate window defaults will be used where possible.You can list the currently used arguments and values with the -verbose parameter.

-bsdperdirectory

Create digest files in BSD format for each directory. All digests generated for the original files in a single directory are stored into one single digest file. The algorithms can be specified using the parameter -md5 for MD5 digests etc. The files are named with the extension specified by the -extmd5 parameter etc. and the file name specified with the -summaryfile parameter. Alternatively you can use the -gnuperdirectory argument but not both since the same files are being created.

In addition you can let the software create BSD files for each original file (see parameter -bsdpertarget).

`-bsdpertarget`

Create digest files in BSD format for each original file. For each of the calculated original files one single digest file is being created per algorithm (e.g. -md5 for MD5). The files are named with the extension specified by the -extmd5 parameter etc. Alternatively you can use the -gnupertarget argument but not both since the same files are being created.

In addition you can let the software create BSD files on a directory basis (see parameter -bsdperdirectory).

`-exponentialstart <value>`

This option allows intermediate digests to be created starting at a certain position (e.g. 8K = 8 kilobytes). The next intermediates will then be extracted in steps after the next 16 kilobytes, 32 kilobytes etc. You can use numeric values or abbreviations like (K)ilobytes, (M)egabytes, (G)igabytes and (T)erabytes.

This argument is evaluated only if intermediate digests for Digester XML are being calculated (see parameter -xmlintermediates) and if the exponential mode has been specified.

`-extmd2 <value>`

Sets the file extension for GNU/BSD files containing MD2 message digests (default: .md2). Please do not use the following characters: \ / : * ? " < > |

`-extmd4 <value>`

Sets the file extension for GNU/BSD files containing MD4 message digests (default: .md4). Please do not use the following characters: \ / : * ? " < > |

`-extmd5 <value>`

Sets the file extension for GNU/BSD files containing MD5 message digests (default: .md5). Please do not use the following characters: \ / : * ? " < > |

`-extpgp <value>`

Sets the file extension for signature files containing OpenPGP signatures (default: .asc). Please do not use the following characters: \ / : * ? " < > |

`-extripemd128 <value>`

Sets the file extension for GNU/BSD files containing RIPEMD-128 message digests (default: `.rip128`). Please do not use the following characters: \ / : * ? " < > |

-extripemd160 <value>

Sets the file extension for GNU/BSD files containing RIPEMD-160 message digests (default: `.rip160`). Please do not use the following characters: \ / : * ? " < > |

-extripemd256 <value>

Sets the file extension for GNU/BSD files containing RIPEMD-256 message digests (default: `.rip256`). Please do not use the following characters: \ / : * ? " < > |

-extripemd320 <value>

Sets the file extension for GNU/BSD files containing RIPEMD-320 message digests (default: `.rip320`). Please do not use the following characters: \ / : * ? " < > |

-extsha1 <value>

Sets the file extension for GNU/BSD files containing SHA-1 message digests (default: `.sha`). Please do not use the following characters: \ / : * ? " < > |

-extsha224 <value>

Sets the file extension for GNU/BSD files containing SHA-224 message digests (default: `.sha224`). Please do not use the following characters: \ / : * ? " < > |

-extsha256 <value>

Sets the file extension for GNU/BSD files containing SHA-256 message digests (default: `.sha256`). Please do not use the following characters: \ / : * ? " < > |

-extsha384 <value>

Sets the file extension for GNU/BSD files containing SHA-384 message digests (default: `.sha384`). Please do not use the following characters: \ / : * ? " < > |

-extsha512 <value>

Sets the file extension for GNU/BSD files containing SHA-512 message digests (default: `.sha512`). Please do not use the following characters: \ / : * ? " < > |

-exttiger <value>

Sets the file extension for GNU/BSD files containing Tiger message digests (default: `.tiger`). Please do not use the following characters: \ / : * ? " < > |

-extwhirlpool <value>

Sets the file extension for GNU/BSD files containing Whirlpool message digests (default: `.wpool`). Please do not use the following characters: \ / : * ? " < > |

-extxml <value>

Sets the file extension for Digester XML files containing digests/signatures (default: `.digest`). Please do not use the following characters: \ / : * ? " < > |

-gnuperdirectory

Create digest files in GNU format for each directory. All digests generated for the original files in a single directory are stored into one single digest file. The algorithms can be specified using the parameter -md5 for MD5 digests etc. The files are named with the extension specified by the -extmd5 parameter etc. and the file name specified with the -summaryfile parameter. Alternatively you can use the -bsdperdirectory argument but not both since the same files are being created.

In addition you can let the software create GNU files for each original file (see parameter -gnupertarget).

-gnupertarget

Create digest files in GNU format for each original file. For each of the calculated original files one single digest file is being created per algorithm (e.g. -md5 for MD5). The files are named with the extension specified by the -extmd5 parameter etc. Alternatively you can use the -bsdpertarget argument but not both since the same files are being created.

In addition you can let the software create GNU files on a directory basis (see parameter -gnuperdirectory).

-hexcase <value>

Sets the case of writing of hexadecimal digest values (`lower` = lower-case, `upper` = upper-case).

-ignoredigests

Files containing digests/signatures are not used to calculate additional digests/signatures. This means that no MD5 files are being created for existing MD5 files etc. This argument should be used by default.

-ignoreregex <value>

Sets a [regular expression](#) to ignore certain files when searching for digests/signatures (not processed). Alternatively you can include certain files using the parameter -[includeregex](#).

-includeregex <value>

Sets a [regular expression](#) to include certain files when searching for digests/signatures (are being processed). Alternatively you can ignore certain files using the parameter -[ignoreregex](#).

-maxintermediates <value>

Use this option to limit the number of calculated intermediate digests.

This argument is evaluated only if intermediate digests for Digester XML are being calculated (see parameter -[xmlintermediates](#)) and for both the exponential and linear mode.

-md2

Activates the generation of MD2 message digests. The files are created based on the configured file extensions (see parameter -[extmd2](#)).

-md4

Activates the generation of MD4 message digests. The files are created based on the configured file extensions (see parameter -[extmd4](#)).

-md5

Activates the generation of MD5 message digests. The files are created based on the configured file extensions (see parameter -[extmd5](#)).

-overwrite

Existing digest/signature files are overwritten when the calculation has been finished.

> ⚠ You will not be able to restore overwritten digest/signature files with the program, please backup your data first.

-pgpformat <value>

Specify `ascii` here to create OpenPGP files in ASCII format (text) or `binary` for the binary format.

This argument is evaluated only if OpenPGP signatures are being calculated.

-pgpkey <id> <url>

Use the OpenPGP key with the given ID (required) and the key URL (optional) to calculate signatures. The ID must be a full hexadecimal identifier like `0x2452F30BFCE8C30F` in order to identify the key. The URL value is optional and used if the additional -xmlpgpkeyurl parameter has been specified (only for Digester XML).

This argument is evaluated only if OpenPGP signatures are being calculated.

-pgppertarget

Create OpenPGP signature files for each original file. For each of the calculated original files one single signature file is being created containing a value for each key. The files are named with the extension specified by the -extpgp parameter.

Creating signature files on a directory basis is not possible for OpenPGP signatures.

-pgpversion <value >

In order to label OpenPGP signatures files one can specify an arbitrary single-line text to be stored in the signature file headers. Here, you can set your company's or organization's name for example.

-refreshallformats

For the refreshing, all detected digest/signature files are being used. If this parameter is omitted then only those formats are being analyzed which are configured to be written and deemed relevant for the synchronization.

This option can be extended with the -refreshremoveother und -refreshremoveartifacts parameters.

-refreshremoveother

If all formats are being analyzed (see parameter -refreshallformats) then it is possible to switch from GNU files over to XML files for example and let the software delete the previous GNU files programmatically. This way only those files are kept which are deemed relevant for the synchronization (write formats).

> ⚠️ You will not be able to restore overwritten digest/signature files with the program, please backup your data first.

-refreshremoveartifacts

If all formats are being analyzed (see parameter -refreshallformats) then it is also possible to delete artifact files automatically. Artifacts are files with digests/signatures where no original files exist at all. This way useless files can be cleared automatically.

> ⚠️ You will not be able to restore overwritten digest/signature files with the program, please backup your data first.

-refreshignoremissing

Skip file and do not re-calculate if digests match but intermediate digests are missing. Maybe intermediates have not been generated the last time so this can be compensated here as long as the overall digests match for the files.

-refreshgnubsdtimestamp

Skip file and do not re-calculate if digests match and if the file timestamps match.

-refreshpgptimestamp

Skip file and do not re-calculate if signatures match and if the file timestamps match.

-refreshxmltimestamp

Skip file and do not re-calculate if digests/signatures match and if the XML file timestamps match. The entries from within the XML files may be controlled separately with the -refreshxmlentryfilesize and -refreshxmlentrytimestamp arguments.

-refreshxmlentryfilesize

Skip file and do not re-calculate if digests/signatures match and if the file sizes within the XML entries match. The timestamp of the XML file itself may be controlled separately with the -refreshxmltimestamp argument.

-refreshxmlentrytimestamp

Skip file and do not re-calculate if digests/signatures match and if the timestamps within the XML entries match. The timestamp of the XML file itself may be controlled separately with the -refreshxmltimestamp argument.

-ripemd128

Activates the generation of RIPEMD-128 message digests. The files are created based on the configured file extensions (see parameter -extripemd128).

-ripemd160

Activates the generation of RIPEMD-160 message digests. The files are created based on the configured file extensions (see parameter -extripemd160).

-ripemd256

Activates the generation of RIPEMD-256 message digests. The files are created based on the configured file extensions (see parameter -extripemd256).

-ripemd320

Activates the generation of RIPEMD-320 message digests. The files are created based on the configured file extensions (see parameter -extripemd320).

-sha1 | sha

Activates the generation of SHA-1 message digests. The files are created based on the configured file extensions (see parameter -extsha1).

-sha224

Activates the generation of SHA-224 message digests. The files are created based on the configured file extensions (see parameter -extsha224).

-sha256

Activates the generation of SHA-256 message digests. The files are created based on the configured file extensions (see parameter -extsha256).

-sha384

Activates the generation of SHA-384 message digests. The files are created based on the configured file extensions (see parameter -extsha384).

-sha512

Activates the generation of SHA-512 message digests. The files are created based on the configured file extensions (see parameter -extsha512).

-subdirectories

Activates the processing of subdirectories for directory-based generating. Otherwise only the specified directories themselves (without subdirectories) are being processed.

-summaryfile <value>

File names for digest files on the directory basis (default: `summary`). The file extensions are then chosen according to the format of the digest files (for example see parameter -`extxml`).

-template <value>

The values of the given template will be used as defaults and overwritten by the specified parameters. By default the options of the current template selected in the generate window will be used as defaults.

-tiger

Activates the generation of Tiger message digests. The files are created based on the configured file extensions (see parameter -`exttiger`).

-timestamp

Timestamps of the processed original files are applied to the created digest/signature files. This is only possible for GNU/BSD/OpenPGP/XML files per target, not for files created on the directory basis (per directory).

-verbose

Create more detailed messages. Especially all passed arguments and values are presented as a list for control purposes. For many program settings defaults are being used if a parameter is not specified. This functionality is useful if templates are used as default values where the current settings are to be controlled right before the processing starts.

-whirlpool

Activates the generation of Whirlpool message digests. The files are created based on the configured file extensions (see parameter -`extwhirlpool`).

-xmlabsolutepaths

Special option for Digester XML files: add absolute paths of the processed original files. With this option all digests hold the absolute reference on the computer, e.g. `C:\Temp\franz.txt` under

Window. Even after copying such XML files to another drive of the same computer these paths may be resolved properly allowing to find the original files for the digests. By using this option the Digester XML files become slightly bigger.

-xmladdurls

Spezielle Option für Digester XML-Dateien: Feste Netzwerk-Adresse oder SSI-Ausdruck pro Originaldatei speichern. Der Wert wird mit dem Parameter -<u>xmlurl</u> übergeben.

-xmlcomment <value>

Optional text for the XML file header. This argument is evaluated only if Digester XML files are being generated. Use the comment to deposit your own name or your company's name in order to label the XML files.

-xmldigestformat <value>

Set the encoding format of hexadecimal digests in Digester XML files (`hex` = hexadecimal, `base64` = Base64 encoding). See the -<u>hexcase</u> parameter to control the case of writing (lower/upper) of hexadecimal values.

-xmlintermediates <value>

Create intermediate message digests, deactivated otherwise. Use the `linear` value to extract intermediates in fixed intervals or `exponential` to extract intermediates dynamically after blocks like 8 kilobytes, then 16 kilobytes etc.

Additional parameters to control the intermediate digest generation are -<u>xmllinearevery</u>, -<u>maxintermediates</u> and -<u>exponentialstart</u>.

-xmllinearevery <value>

This option allows intermediate digests to be extracted after fixed intervals. The interval size can be specified with values like 10M where intermediates are extracted every 10 megabytes. You can use numeric values or abbreviations like (K)ilobytes, (M)egabytes, (G)igabytes and (T)erabytes.

This argument is evaluated only if intermediate digests for Digester XML are being calculated (see parameter -<u>xmlintermediates</u>) and if the linear mode has been specified.

-xmlperdirectory

Create digest/signature files in Digester XML format for each directory. All digests generated for the original files in a single directory are stored into one single digest file. The algorithms can be

specified using the parameter -md5 for MD5 digests etc. The files are named with the extension specified by the -extxml parameter and the file name specified with the -summaryfile parameter.

In addition you can let the software create Digester XML files for each original file (see parameter -xmlpertarget).

-xmlpertarget

Create digest/signature files in Digester XML format for each original file. For each of the calculated original files one single XML file is being created containing the values for the chosen algorithms (e.g. -md5 for MD5) as well as the OpenPGP signatures. The files are named with the extension specified by the -extxml parameter.

In addition you can let the software create XML files on a directory basis (see parameter -xmlperdirectory).

-xmlpgp

If Digester XML files are being created and OpenPGP keys have been specified to calcuate signatures then the signature values may be added to the Digester XML files. Signatures are represented by separate XML elements which are stored on the level together with digests (siehe DTD Digester XML).

-xmlpgpkeyid

In addition to the OpenPGP signatures themselves Digester XML files can contain the IDs of the OpenPGP keys used for signature generation. The IDs can help importing software to identify or download appropriate public keys to check the signatures (siehe Parameter -xmlpgp).

-xmlpgpkeyname

In addition to the OpenPGP signatures themselves Digester XML files can contain the names of the OpenPGP keys used for signature generation (siehe Parameter -xmlpgp).

-xmlpgpkeyemail

In addition to the OpenPGP signatures themselves Digester XML files can contain the eMail addresses of the OpenPGP keys used for signature generation (siehe Parameter -xmlpgp).

-xmlpgpkeyurl

In addition to the OpenPGP signatures themselves Digester XML files can contain the network addresses of the OpenPGP keys used for signature generation. The URLs can help importing

software to identify or download appropriate public keys to check the signatures (siehe Parameter -<u>xmlpgp</u>).

-xmlurl <value>

Special option for Digester XML files: add fixed network address or SSI pattern. Here you can specify a static URL pattern e.g. for SSI (*Server Side Includes*) which can be processed by web servers like the Apache HTTP server. Such URLs can be translated dynamically into the appropriate URLs for the original files on the server. If Digester XML files are being downloaded with the software, then the referenced files can be identified using the URL entries to be downloaded, checked, and processed in other ways.

This argument is evaluated only if also -<u>xmladdurls</u> is passed.

Refreshing with Ant

The Digester software also supports refreshing from Ant build scripts. You can pass files, directories, or network files to be processed. In theory, all refresh operations of the graphical program version can also be done using Apache Ant, so the software can be called from own builds scripts for example.

You can either create your XML tags manually or let the Digester GUI version assemble the necessary elements and attributes for you. In the latter case, please configure a refresh operation with the **generate window** as if you would do some refreshing from there. Once all options are as desired, let the window assemble an appropriate Ant build script code by pressing the ant symbol at the bottom. This snippet can be easily copied into your own build scripts.

All supported options can be passed in any order, but the corresponding values must be in place. All arguments are optional, input errors are detected as far as possible and compensated automatically. The program reports incorrect arguments directly and terminates if you have specified the `failonerror` tag.

For the refreshing with Ant all arguments of the command-line version of the Digester software `digestercli` can be used which are represented by separate XML elements each. In the header of your build script the task `digestercli` must be declared. Please adjust the `classpath` attribute by the value appropriate for your installation. The Web Start version cannot be used for checking from Ant build scripts.

```
<taskdef name="digestercli"
   classname="com.lf.digester.ant.DigesterCLITask"
   classpath="C:\Programme\digester-1.6.3\ant\digester-ant.jar"/>
```

An example XML code for Ant build scripts looks like the following where a directory including subdirectories is being refreshed:

```
<digestercli>
    <refresh />
    <md5 />
    <extmd5>.md5</extmd5>
    <sha1 />
    <extsha1>.sha1</extsha1>
    <xmlperdirectory />
    <overwrite />
    <timestamp />
    <summaryfile>summary</summaryfile>
    <ignoredigests />
    <subdirectories />
    <pgpversion>Digester 1.6.3</pgpversion>
    <extxml>.digest</extxml>
    <xmldigestformat>hex</xmldigestformat>
    <xmlabsolutepaths />
    <xmlpgp />
    <xmlpgpkeyid />
    <xmlpgpkeyname />
    <xmlpgpkeyemail />
    <xmlintermediates>exponential</xmlintermediates>
    <exponentialstart>8K</exponentialstart>
    <maxintermediates>5</maxintermediates>
    <refreshignoremissing />
    <refreshxmlentryfilesize />
    <refreshxmlentrytimestamp />
    <pgpkey>
        <id>0x7EB2B10924E58CDB</id>
    </pgpkey>
    <dirset dir="D:\">
        <include name="checksum_test" />
    </dirset>
</digestercli>
```

Additional Links

- [Arguments for `digestercli`](#)
- [Checking with `digestercli` and Ant](#)
- [Generating with `digestercli` and Ant](#)
- [Synchronizing with `digestercli` and Ant](#)
- [Arguments for `digestergui`](#)
- [Generate Window](#)
- [DTD Digester XML](#)

Synchronizing with `digestercli` and Ant

- [Synchronizing with `digestercli`](#)
- [Synchronizing with Ant](#)

Synchronizing with `digestercli`

The Digester software also supports synchronizing from the command-line. You can pass files, directories, or network files to be processed. In theory, all sync operations of the graphical program version can also be done using `digestercli`, so the software can be called from shell scripts for example.

You can either create your program calls manually or let the Digester GUI version assemble the necessary parameters for you. In the latter case, please configure a sync operation with the [sync window](#) as if you would do some synchronizing from there. Once all options are as desired, let the window assemble an appropriate `digestercli` call by pressing the console symbol at the bottom. This snippet can be easily copied into your own scripts.

All supported options can be passed in any order, but the corresponding values must be in place. All arguments are optional, input errors are detected as far as possible and compensated automatically. Either the program reports incorrect arguments directly or creates appropriate entries in the log file. You can pass files, directories, or URLs in the source and target sets with the `-sourceset` and `-targetset` (plus name) arguments.

The command-line version of the Digester software currently provides support for the following arguments and values, which may be printed by calling the application using `digestercli -? sync`:

```
Usage: digestercli [-options] -sourceset <file|directory|url> ... -targetset <name>
<file|directory|url> ...

Supported sync mode options include:
```

-[sync](#)	synchronize files and directories with message digests
-[bsdperdirectory](#)	process BSD-compliant file per directory
-[bsdpertarget](#)	process BSD-compliant file per target
-[flipsourcetarget](#) \<set\>	reverse mode: transfer from target set to source
-[gensrctemplate](#) \<template\>	execute generate template before sync (source set)
-[gensrctemplaterefresh](#)	execute template in refresh mode (source set)
-[gentemplate](#) \<set\> \<template\>	execute generate template before sync (named target set)
-[gentemplaterefresh](#) \<set\>	execute template in refresh mode (named target set)
-[gnuperdirectory](#)	process MD5SUM-compliant file per directory
-[gnupertarget](#)	process MD5SUM-compliant file per target
-[ignoreregex](#) \<value\>	regular expression for files to be ignored
-[includeregex](#) \<value\>	regular expression for files to be included
-[md2](#)	process MD2 message digests

```
-md4                        process MD4 message digests
-md5                        process MD5 message digests
-optdigest                  digests for source and target match
-optfilesize                source/target file sizes match
-optintermediate            intermediates for source and target match
-optpgpsig                  PGP signatures for source and target match
-opttimestamp               source/target timestamps match
-optxmlentryfilesize        XML entry file sizes match source and target
-optxmlentrytimestamp       XML entry timestamps match source and target
-overwrite                  overwrite existing files
-pgppertarget               process PGP signature file per target
-ripemd128                  process RIPEMD-128 hash files
-ripemd160                  process RIPEMD-160 hash files
-ripemd256                  process RIPEMD-256 hash files
-ripemd320                  process RIPEMD-320 hash files
-sha1 | sha                 process SHA-1 hash files
-sha224                     process SHA-224 hash files
-sha256                     process SHA-256 hash files
-sha384                     process SHA-384 hash files
-sha512                     process SHA-512 hash files
-sourcetimestamp            apply source file timestamps to target files
-subdirectories             include subdirectories
-syncgnubsdtimestamp        trust digests if file timestamp matches (GNU/BSD)
-syncmapping <value>        use sync mapping options as defaults
-syncpgptimestamp           trust signatures if file timestamp matches (PGP)
-syncxmlentryfilesize       trust digests if XML entry file size matches
-syncxmlentrytimestamp      trust digests if XML entry timestamp matches
-syncxmltimestamp           trust digests if file timestamp matches (XML)
-tiger                      process Tiger hash files
-transferall                transfer all sources without optimization
-verbose                    print additional messages
-whirlpool                  process Whirlpool hash files
-xmlperdirectory            process Digester XML file per directory
-xmlpertarget               process Digester XML file per target
```

-sync

Activate the operation mode to synchronize files and directories with digests and OpenPGP signatures. The arguments basically are the same as the options shown by the sync window of the graphical version of the Digester software.

If the parameters described below are missing, then the sync window defaults will be used where possible.You can list the currently used arguments and values with the `-verbose` parameter.

-bsdperdirectory

Check digest files in BSD format for each directory. All digests generated for the original files in a single directory are stored into one single digest file. The algorithms can be specified using the parameter -md5 for MD5 digests etc. The files are named with the extension specified by the -extmd5 parameter etc. and the file name specified with the -summaryfile parameter.

In addition you can let the software check BSD files for each original file (see parameter -bsdpertarget).

-bsdpertarget

Check digest files in BSD format for each original file. For each of the calculated original files one single digest file is being created per algorithm (e.g. -md5 for MD5). The files are named with the extension specified by the -extmd5 parameter etc.

In addition you can let the software check BSD files on a directory basis (see parameter -bsdperdirectory).

-flipsourcetarget <set>

The synchronization normally takes place from source to target, which means that the sources are transferred over to the targets. It may become necessary to transfer files in reverse direction from targets to sources. The target set specified with this parameter ist then used exclusively for transfer. The other targets sets are ignored in reverse mode.

-gensrctemplate <template>

Sets the generate template to be executed for the sources. This template will then be executed right before the sychronization in order to update the digests/signatures for the sources. Optionally the refreshing may be executed (see parameter -gensrctemplaterefresh), if only the new or modified files should be updated.

-gensrctemplaterefresh

Refresh sources instead of calculating them completely in order to update only the new or modified files. This argument is evaluated only if the -gensrctemplate parameter is also specified.

-gentemplate <set> <template>

Sets the generate template (2nd value) to be executed for the named target set (1st value). This template will then be executed right before the sychronization in order to update the digests/signatures for the targets. Optionally the refreshing may be executed (see parameter -gentemplaterefresh), if only the new or modified files should be updated.

-gentemplaterefresh

Refresh targets instead of calculating them completely in order to update only the new or modified files. This argument is evaluated only if the -gentemplate parameter is also specified.

-gnuperdirectory

Check digest files in GNU format for each directory. All digests generated for the original files in a single directory are stored into one single digest file. The algorithms can be specified using the parameter -md5 for MD5 digests etc. The files are named with the extension specified by the -extmd5 parameter etc. and the file name specified with the -summaryfile parameter.

In addition you can let the software check GNU files for each original file (see parameter -gnupertarget).

-gnupertarget

Check digest files in GNU format for each original file. For each of the calculated original files one single digest file is being created per algorithm (e.g. -md5 for MD5). The files are named with the extension specified by the -extmd5 parameter etc.

In addition you can let the software check GNU files on a directory basis (see parameter -gnuperdirectory).

-ignoreregex <value>

Sets a regular expression to ignore certain files for synchronization (not processed). Alternatively you can include certain files using the parameter -includeregex.

-includeregex <value>

Sets a regular expression to include certain files for synchronization (are being processed). Alternatively you can ignore certain files using the parameter -ignoreregex.

–md2

Activates the recognition of MD2 digests for synchronization.

–md4

Activates the recognition of MD4 digests for synchronization.

–md5

Activates the recognition of MD5 digests for synchronization.

-optdigest

Optimization: do not transfer files if digests match for source and target. Evaluated only if the -transferall argument is not specified.

-optfilesize

Optimization: do not transfer files if the file sizes match for source and target. Evaluated only if the -transferall argument is not specified.

-optintermediate

Optimization: Do not transfer files if the intermediate digests match for source and target (DIgester XML only). Evaluated only if the -transferall argument is not specified.

-optpgpsig

Optimization: do not transfer files if the OpenPGP signatures match for source and target. Evaluated only if the -transferall argument is not specified.

-opttimestamp

Optimization: do not transfer files if the file timestamps match for source and target. Evaluated only if the -transferall argument is not specified.

-optxmlentryfilesize

Optimization: do not transfer files if the XML entry file sizes match for source and target (XML element values). Evaluated only if the -transferall argument is not specified.

-optxmlentrytimestamp

Optimization: do not transfer files if the XML entry timestamps match for source and target (XML element values). Evaluated only if the -transferall argument is not specified.

-overwrite

Existing digest/signature files on the target system are overwritten during synchronization.

> ⚠️ You will not be able to restore overwritten digest/signature files with the program, please backup your data first.

-pgppertarget

Check OpenPGP signature files for each original file. For each of the calculated original files one single signature file is being created containing a value for each key. The files are named with the extension specified by the -extpgp parameter.

Creating/checking signature files on a directory basis is not possible for OpenPGP signatures.

-ripemd128

Activates the recognition of RIPEMD-128 digests for synchronization.

-ripemd160

Activates the recognition of RIPEMD-160 digests for synchronization.

-ripemd256

Activates the recognition of RIPEMD-256 digests for synchronization.

-ripemd320

Activates the recognition of RIPEMD-320 digests for synchronization.

-sha1 | sha

Activates the recognition of SHA-1 digests for synchronization.

-sha224

Activates the recognition of SHA-224 digests for synchronization.

-sha256

Activates the recognition of SHA-256 digests for synchronization.

-sha384

Activates the recognition of SHA-384 digests for synchronization.

-sha512

Activates the recognition of SHA-512 digests for synchronization.

-sourcetimestamp

Timestamps of the processed source files are applied to the target files. This is only possible for GNU/BSD/OpenPGP/XML files per target, not for files created on the directory basis (per directory).

-subdirectories

Activates the processing of subdirectories for directory-based synchronization. Otherwise only the specified directories themselves (without subdirectories) are being processed.

-syncgnubsdtimestamp

Trust option: digests is trusted most if GNU/BSD file timestamps match. Those digests/signatures are selected for synchronization of sources and targets which fulfill the maximum number of trust options (is trusted most).

-syncmapping <value>

The values of the given mapping will be used as defaults and overwritten by the specified parameters. By default the options of the current mapping selected in the sync window will be used as defaults.

-syncpgptimestamp

Trust option: signatures is trusted most if OpenPGP file timestamps match. Those digests/signatures are selected for synchronization of sources and targets which fulfill the maximum number of trust options (is trusted most).

-syncxmlentryfilesize

Trust option: digests/signatures is trusted most if the file sizes from within the XML elements match. Those digests/signatures are selected for synchronization of sources and targets which fulfill the maximum number of trust options (is trusted most).

-syncxmlentrytimestamp

Trust option: digests/signatures is trusted most if the file timestamps from within the XML elements match. Those digests/signatures are selected for synchronization of sources and targets which fulfill the maximum number of trust options (is trusted most).

-syncxmltimestamp

Trust option: digests/signatures is trusted most if XML file timestamps match. Those digests/signatures are selected for synchronization of sources and targets which fulfill the maximum number of trust options (is trusted most).

-tiger

Activates the recognition of Tiger digests for synchronization.

-transferall

All sources are transferred without analyzing the targets. If you want to make sure that sources and targets are equal then please choose this option. If not set you can set one or multiple optimization parameters: -optdigest, -optfilesize, -optintermediate, -optpgpsig, -opttimestamp, -optxmlentryfilesize or -optxmlentrytimestamp in order to transfer only a minimum amount of data.

-verbose

Create more detailed messages. Especially all passed arguments and values are presented as a list for control purposes. For many program settings defaults are being used if a parameter is not specified. This functionality is useful if templates are used as default values where the current settings are to be controlled right before the processing starts.

-whirlpool

Activates the recognition of Whirlpool digests for synchronization.

-xmlperdirectory

Check digest/signature files in Digester XML format for each directory. All digests generated for the original files in a single directory are stored into one single digest file. The algorithms can be specified using the parameter -md5 for MD5 digests etc. The files are named with the extension specified by the -extxml parameter and the file name specified with the -summaryfile parameter.

In addition you can let the software check Digester XML files for each original file (see parameter -xmlpertarget).

-xmlpertarget

Check digest/signature files in Digester XML format for each original file. For each of the calculated original files one single XML file is being created containing the values for the

chosen algorithms (e.g. -md5 for MD5) as well as the OpenPGP signatures. The files are named with the extension specified by the -extxml parameter.

In addition you can let the software check XML files on a directory basis (see parameter -xmlperdirectory).

Synchronizing with Ant

The Digester software also supports synchronizing from Ant build scripts. You can pass files, directories, or network files to be processed. In theory, all sync operations of the graphical program version can also be done using Apache Ant, so the software can be called from own builds scripts for example.

You can either create your XML tags manually or let the Digester GUI version assemble the necessary elements and attributes for you. In the latter case, please configure a sync operation with the sync window as if you would do some synchronizing from there. Once all options are as desired, let the window assemble an appropriate Ant build script code by pressing the ant symbol at the bottom. This snippet can be easily copied into your own build scripts.

All supported options can be passed in any order, but the corresponding values must be in place. All arguments are optional, input errors are detected as far as possible and compensated automatically. The program reports incorrect arguments directly and terminates if you have specified the `failonerror` tag.

For the synchronizing with Ant all arguments of the command-line version of the Digester software `digestercli` can be used which are represented by separate XML elements each. In the header of your build script the task `digestercli` must be declared. Please adjust the `classpath` attribute by the value appropriate for your installation. The Web Start version cannot be used for checking from Ant build scripts.

```
<taskdef name="digestercli"
    classname="com.lf.digester.ant.DigesterCLITask"
    classpath="C:\Programme\digester-1.6.3\ant\digester-ant.jar"/>
```

An example XML code for Ant build scripts looks like the following where a server directory is being synchronized with a local directory:

```
<digestercli>
    <sync />
    <optfilesize />
    <optxmlentryfilesize />
    <optxmlentrytimestamp />
    <optdigest />
    <optintermediate />
    <sourcetimestamp />
```

```
    <overwrite />
    <subdirectories />
    <syncgnubsdtimestamp />
    <syncpgptimestamp />
    <syncxmltimestamp />
    <syncxmlentryfilesize />
    <syncxmlentrytimestamp />
    <xmlperdirectory />
    <md5 />
    <sha1 />
    <sourceset>
        <gentemplate>My Sync4</gentemplate>
        <gentemplaterefresh />
        <dirset dir="C:\Temp\digester">
            <include name="cms" />
        </dirset>
    </sourceset>
    <targetset>
        <name>Targets</name>
        <url>sftp://mydomain.com:22/home/user1/public_html/.cms</url>
    </targetset>
</digestercli>
```

Additional Links

- [Arguments for `digestercli`](#)
- [Checking with `digestercli` and Ant](#)
- [Generating with `digestercli` and Ant](#)
- [Refreshing with `digestercli` and Ant](#)
- [Arguments for `digesterqui`](#)
- [Sync Window](#)

Shortcuts

Here you can find an overview of the implemented keyboard shortcuts for the Digester software main window to be used alternatively for the various menu items. The listed shortcuts are recommended for experienced users.

In Mac OS X the default modifier key is the Command key (Cmd). Mac users have to use the Command key instead of the Control modifier key (Ctrl) as shown by the table.

Nr.	Shortcut Win	Shortcut Mac	Menu	Description
1	**Ctrl+E**	**Cmd+E**	File -> Edit Files	Open file(s) in document window
2	**Ctrl+S**	**Cmd+S**	File -> Save	Save currently edited file
3	**Ctrl+Alt+S**	**Cmd+Alt+S**	File -> Save Preferences	Save current preferences explicitely
4	**Ctrl+Q**	**Cmd+Q**	File -> Exit	Quit the Digester software
5	**Ctrl+X**	**Cmd+X**	Edit -> Cut	Delete texts/files
6	**Ctrl+C**	**Cmd+C**	Edit -> Copy	Copy texts/files to clipboard
7	**Ctrl+V**	**Cmd+V**	Edit -> Paste	Insert texts/files from clipboard
8	**Delete**	**Delete**	Edit -> Delete	Delete files
9	**Ctrl+F**	**Cmd+F**	Edit -> Find	Search for files
10	**Ctrl+Plus**	**Cmd+Plus**	Setup -> Fonts -> Larger Fonts	Show slightly larger fonts
11	**Ctrl+Minus**	**Cmd+Minus**	Setup -> Fonts -> Smaller Fonts	Show slightly smaller fonts
12	**Ctrl+G**	**Cmd+G**	Setup -> Memory -> Collect Garbage	Make RAM available again
13	**Alt+S**	**Alt+S**	Setup -> Preferences	Open the preferences dialog
14	**Ctrl+Alt+C**	**Cmd+Alt+C**	Window -> Screen Shot -> Capture Tool	Take screen shots of the running application
15	**Ctrl+F4**	**Cmd+F4**	Window -> Close Tab	Close the active tab
16	**Escape**	**Escape**	Window -> Hide	Hide the active docking window
17	**F5**	**F5**	Window -> Refresh	Update the active docking window

18	**F1**	**F1**	Help -> Help (context-sensitive)	Open the help system dialog

Third Party Components

The Digester software is based on many components from other manufacturers which are listed here as far as this is due to the licensing conditions. The installer-based deployments of the Digester software contain the files with the detailed license texts. You can find these license files in the `license` subdirectory of the installation directory on your computer.

- Alloy Look&Feel
- Apache Commons Codec
- Apache HttpComponents Client
- Apache HttpComponents Core
- Apache Commons Logging
- Apache Commons Net
- Apache Jakarta Commons VFS
- AppleJavaExtensions
- AWS SDK for Java
- Bouncy Castle
- Dropbox SDK
- Google Annotations JSR305
- Google Drive API
- HexEditor
- Install4j Multi-Platform Edition
- Jackson JSON Processor
- Java API for Servlets
- JavaMail
- Java Runtime Environment
- JavaHelp
- Jaxen
- JDOM
- JGoodies Forms
- JSch
- JTattoo
- OpenPGP Library for Java
- RSyntaxTextArea
- TableLayout

The Web Start installations include the license texts not as locally installed files, but the license texts can be downloaded as a compressed ZIP archive from the Leisenfels Internet server. If the Digester software has been installed via Web Start for example by using the following link (the address may vary), so the ZIP archive named `3rdparty.zip` can be downloaded as shown (just enter the address in your Web browser, an Internet connection is required):

Installation: http://jws.leisenfels.com:8080/digester-1.6.x/install.jnlp

ZIP archive: http://jws.leisenfels.com:8080/digester-1.6.x/license/3rdparty.zip

The Digester software is shipped with the following components developed by other manufacturers. Besides the copyright information and the name of the license text file you can find additional remarks, if this is part of the licensing conditions. The websites of the respective licensors are listed if known at the time of writing of this documentation.

Alloy Look&Feel

Copyright © INCORS GmbH

http://www.incors.com↗

```
alloy_license.txt
```

Apache Commons Codec

Copyright © The Apache Software Foundation

http://www.apache.org

```
apache_license.txt
```

Apache HttpComponents Client

Copyright © 1999-2012 The Apache Software Foundation

http://www.apache.org

```
apache_license.txt, apache_notice_httpcompclient.txt
```

Apache HttpComponents Client
Copyright 1999-2012 The Apache Software Foundation
This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

Apache HttpComponents Core

Copyright © 2005-2012 The Apache Software Foundation

http://www.apache.org

```
apache_license.txt, apache_notice_httpcompcore.txt
```

Apache HttpComponents Core
Copyright 2005-2012 The Apache Software Foundation
This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).
This project contains annotations derived from JCIP-ANNOTATIONS
Copyright (c) 2005 Brian Goetz and Tim Peierls. See http://www.jcip.net

Apache Commons Logging

Copyright © The Apache Software Foundation
http://www.apache.org

`apache_license.txt, apache_notice_logging.txt`

This product includes software developed by

The Apache Software Foundation (http://www.apache.org/).

Apache Commons Net

Copyright © 2001-2011 The Apache Software Foundation
http://www.apache.org

`apache_license.txt, apache_notice_net.txt`

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

Apache Jakarta Commons VFS

Copyright © 2002-2011 The Apache Software Foundation
http://www.apache.org

`apache_license.txt, apache_notice_vfs.txt`

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

AppleJavaExtensions

Copyright © 2003-2010 Apple Inc.

http://www.apple.com

`apple_license.txt`

AWS SDK for Java

Copyright © 2010-2012 Amazon.com, Inc. or its affiliates

http://www.amazon.com

`amazon_s3_license.txt, amazon_s3_notice.txt`

AWS SDK for Java
Copyright 2010-2012 Amazon.com, Inc. or its affiliates. All Rights Reserved.
This product includes software developed by
Amazon Technologies, Inc (http://www.amazon.com/).
****
THIRD PARTY COMPONENTS
****

This software includes third party software subject to the following copyrights:
- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- JSON parsing and utility functions from JSON.org - Copyright 2002 JSON.org.
The licenses for these third party components are included in LICENSE.txt

Bouncy Castle

Copyright © 2000-2008 The Legion Of The Bouncy Castle

http://www.bouncycastle.org

`bouncycastle_license.txt`

Dropbox SDK

Copyright © 2013 Dropbox Inc.

http://www.dropbox.com

`dropbox_license.txt`

Google Annotations JSR305

Copyright (c) 2012, William Pugh

https://developers.google.com/drive

`jsr305_license.txt`

Google Drive API

Copyright © 2013 Google Inc.

https://developers.google.com/drive

`google_drive_license.txt`

HexEditor

Copyright © 2008 Robert Futrell

http://fifesoft.com/hexeditor

`hexeditor_license.txt`

Install4j Multi-Platform Edition

Copyright © 2001-2010 ej-technologies GmbH

http://www.ej-technologies.com

`install4j_license.txt`

Jackson JSON Processor

Copyright © 2009-2011 FasterXML, LLC.

http://www.fasterxml.com

`jackson_license.txt`

Java API for Servlets

Copyright © The Apache Software Foundation

http://www.apache.org

`apache_license.txt, apache_notice_tomcat.txt`

This product includes software developed by

The Apache Software Foundation (http://www.apache.org/).

Java Management Extensions (JMX) support is provided by

the MX4J package, which is open source software. The

original software and related information is available

at http://mx4j.sourceforge.net.

The Windows Installer is built with the Nullsoft

Scriptable Install Sysem (NSIS), which is

open source software. The original software and

related information is available at

http://nsis.sourceforge.net.

Java compilation software for JSP pages is provided by Eclipse,

which is open source software. The orginal software and

related infomation is available at

http://www.eclipse.org.

JavaMail

Copyright © Oracle Corporation
http://www.oracle.com

```
javamail_license.txt
```

Java Runtime Environment

Copyright © 2006, 2012, Oracle and/or its affiliates
http://www.oracle.com

```
java_license.txt
```

JavaHelp

Copyright © Sun Microsystems, Inc.

http://java.sun.com

`javahelp_license.html`

Jaxen

Copyright © 2003-2006 The Werken Company

http://jaxen.codehaus.org

`jaxen_license.txt`

JDOM

Copyright © 2000-2004 Jason Hunter & Brett McLaughlin

http://www.jdom.org

`jdom_license.txt`

This product includes software developed by the

JDOM Project (http://www.jdom.org/).

JGoodies Forms

Copyright © 2001-2007 JGoodies Karsten Lentzsch

http://www.jgoodies.com

`jgoodies_license.txt`

JSch

Copyright © 2002,2003,2004,2005,2006 Atsuhiko Yamanaka, JCraft,Inc.

http://www.jcraft.com/

`jsch_license.txt`

JTattoo

Copyright © 2002-2007 MH Software-Entwicklung

http://www.jtattoo.de

`jtattoo_license.txt`

OpenPGP Library for Java

Copyright © DidiSoft Ltd.

http://www.didisoft.com

`pgplib_license.rtf`

RSyntaxTextArea

Copyright © 2012, Robert Futrell

http://fifesoft.com/rsyntaxtextarea

`rsyntaxtextarea_license.txt`

TableLayout

Copyright © 2001 Daniel Barbalace

https://tablelayout.dev.java.net

`tablelayout_license.txt`

Regular Expressions

Regular expressions can be used to ignore certain files for example and are implemented by the Java class `java.util.regex.Pattern`. This class is part of the Java environment by Oracle since version 1.4. The following section is an extract of the API documentation (Javadoc) which is currently `onlyavailable in English`. Copyright © 2003 Sun Microsystems, Inc. All rights reserved.

Summary of regular-expression constructs

Construct	Matches
Characters	
x	The character *x*
\\	The backslash character
\0*n*	The character with octal value 0*n* (0 <= *n* <= 7)
\0*nn*	The character with octal value 0*nn* (0 <= *n* <= 7)
\0*mnn*	The character with octal value 0*mnn* (0 <= *m* <= 3, 0 <= *n* <= 7)
\x*hh*	The character with hexadecimal value 0x*hh*
\u*hhhh*	The character with hexadecimal value 0x*hhhh*
\t	The tab character ('\u0009')
\n	The newline (line feed) character ('\u000A')
\r	The carriage-return character ('\u000D')
\f	The form-feed character ('\u000C')
\a	The alert (bell) character ('\u0007')
\e	The escape character ('\u001B')
\c*x*	The control character corresponding to *x*
Character classes	
[abc]	a, b, or c (simple class)
[^abc]	Any character except a, b, or c (negation)
[a-zA-Z]	a through z or A through Z, inclusive (range)
[a-d[m-p]]	a through d, or m through p: [a-dm-p] (union)
[a-z&&[def]]	d, e, or f (intersection)
[a-z&&[^bc]]	a through z, except for b and c: [ad-z] (subtraction)
[a-z&&[^m-p]]	a through z, and not m through p: [a-lq-z] (subtraction)
Predefined character classes	
.	Any character (may or may not match line terminators)
\d	A digit: [0-9]
\D	A non-digit: [^0-9]
\s	A whitespace character: [\t\n\x0B\f\r]
\S	A non-whitespace character: [^\s]
\w	A word character: [a-zA-Z_0-9]
\W	A non-word character: [^\w]
POSIX character classes (US-ASCII only)	
\p{Lower}	A lower-case alphabetic character: [a-z]
\p{Upper}	An upper-case alphabetic character:[A-Z]
\p{ASCII}	All ASCII:[\x00-\x7F]
\p{Alpha}	An alphabetic character:[\p{Lower}\p{Upper}]
\p{Digit}	A decimal digit: [0-9]
\p{Alnum}	An alphanumeric character:[\p{Alpha}\p{Digit}]

`\p{Punct}`	Punctuation: One of `!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~`
`\p{Graph}`	A visible character: `[\p{Alnum}\p{Punct}]`
`\p{Print}`	A printable character: `[\p{Graph}]`
`\p{Blank}`	A space or a tab: `[ \t]`
`\p{Cntrl}`	A control character: `[\x00-\x1F\x7F]`
`\p{XDigit}`	A hexadecimal digit: `[0-9a-fA-F]`
`\p{Space}`	A whitespace character: `[ \t\n\x0B\f\r]`

Classes for Unicode blocks and categories

`\p{InGreek}`	A character in the Greek block (simple block)
`\p{Lu}`	An uppercase letter (simple category)
`\p{Sc}`	A currency symbol
`\P{InGreek}`	Any character except one in the Greek block (negation)
`[\p{L}&&[^\p{Lu}]]`	Any letter except an uppercase letter (subtraction)

Boundary matchers

`^`	The beginning of a line
`$`	The end of a line
`\b`	A word boundary
`\B`	A non-word boundary
`\A`	The beginning of the input
`\G`	The end of the previous match
`\Z`	The end of the input but for the final terminator, if any
`\z`	The end of the input

Greedy quantifiers

X`?`	*X*, once or not at all
X`*`	*X*, zero or more times
X`+`	*X*, one or more times
X`{`*n*`}`	*X*, exactly *n* times
X`{`*n*`,}`	*X*, at least *n* times
X`{`*n*`,`*m*`}`	*X*, at least *n* but not more than *m* times

Reluctant quantifiers

X`??`	*X*, once or not at all
X`*?`	*X*, zero or more times
X`+?`	*X*, one or more times
X`{`*n*`}?`	*X*, exactly *n* times
X`{`*n*`,}?`	*X*, at least *n* times
X`{`*n*`,`*m*`}?`	*X*, at least *n* but not more than *m* times

Possessive quantifiers

X`?+`	*X*, once or not at all
X`*+`	*X*, zero or more times

X++	*X*, one or more times
X{*n*}+	*X*, exactly *n* times
X{*n*,}+	*X*, at least *n* times
X{*n*,*m*}+	*X*, at least *n* but not more than *m* times
Logical operators	
XY	*X* followed by *Y*
X|*Y*	Either *X* or *Y*
(*X*)	X, as a capturing group
Back references	
n	Whatever the *n*th capturing group matched
Quotation	
\\	Nothing, but quotes the following character
\\Q	Nothing, but quotes all characters until \\E
\\E	Nothing, but ends quoting started by \\Q
Special constructs (non-capturing)	
(?:*X*)	*X*, as a non-capturing group
(?idmsux-idmsux)	Nothing, but turns match flags on - off
(?idmsux-idmsux:*X*)	*X*, as a non-capturing group with the given flags on - off
(?=*X*)	*X*, via zero-width positive lookahead
(?!*X*)	*X*, via zero-width negative lookahead
(?<=*X*)	*X*, via zero-width positive lookbehind
(?<!*X*)	*X*, via zero-width negative lookbehind
(?>*X*)	*X*, as an independent, non-capturing group

Backslashes, escapes, and quoting

The backslash character ('\\') serves to introduce escaped constructs, as defined in the table above, as well as to quote characters that otherwise would be interpreted as unescaped constructs. Thus the expression \\\\ matches a single backslash and \\{ matches a left brace.

It is an error to use a backslash prior to any alphabetic character that does not denote an escaped construct; these are reserved for future extensions to the regular-expression language. A backslash may be used prior to a non-alphabetic character regardless of whether that character is part of an unescaped construct.

Backslashes within string literals in Java source code are interpreted as required by the *Java Language Specification* as either Unicode escapes or other character escapes. It is therefore necessary to double backslashes in string literals that represent regular expressions to protect them from interpretation by the Java bytecode compiler. The string literal "\\b", for example, matches a single backspace character when interpreted as a regular expression, while "\\\\b" matches a word boundary. The string literal "\\(hello\\)" is illegal and leads to a compile-time error; in order to match the string (hello) the string literal "\\\\(hello\\\\)" must be used.

Character Classes

Character classes may appear within other character classes, and may be composed by the union operator (implicit) and the intersection operator (`&&`). The union operator denotes a class that contains every character that is in at least one of its operand classes. The intersection operator denotes a class that contains every character that is in both of its operand classes.

The precedence of character-class operators is as follows, from highest to lowest:

1	Literal escape	`\x`
2	Grouping	`[...]`
3	Range	`a-z`
4	Union	`[a-e][i-u]`
5	Intersection	`[a-z&&[aeiou]]`

Note that a different set of metacharacters are in effect inside a character class than outside a character class. For instance, the regular expression `.` loses its special meaning inside a character class, while the expression `-` becomes a range forming metacharacter.

Line terminators

A *line terminator* is a one- or two-character sequence that marks the end of a line of the input character sequence. The following are recognized as line terminators:

- A newline (line feed) character (`'\n'`),
- A carriage-return character followed immediately by a newline character (`"\r\n"`),
- A standalone carriage-return character (`'\r'`),
- A next-line character (`'\u0085'`),
- A line-separator character (`'\u2028'`), or
- A paragraph-separator character (`'\u2029`).

`UNIX_LINES` mode is activated, then the only line terminators recognized are newline characters.

The regular expression `.` matches any character except a line terminator unless the `DOTALL` flag is specified.

By default, the regular expressions `^` and `$` ignore line terminators and only match at the beginning and the end, respectively, of the entire input sequence. If `MULTILINE` mode is activated then `^` matches at the beginning of input and after any line terminator except at the end of input. When in `MULTILINE` mode `$` matches just before a line terminator or the end of the input sequence.

Groups and capturing

Capturing groups are numbered by counting their opening parentheses from left to right. In the expression `((A)(B(C)))`, for example, there are four such groups:

```
1   ((A)(B(C)))
2   (A)
3   (B(C))
4   (C)
```

Group zero always stands for the entire expression.

Capturing groups are so named because, during a match, each subsequence of the input sequence that matches such a group is saved. The captured subsequence may be used later in the expression, via a back reference, and may also be retrieved from the matcher once the match operation is complete.

The captured input associated with a group is always the subsequence that the group most recently matched. If a group is evaluated a second time because of quantification then its previously-captured value, if any, will be retained if the second evaluation fails. Matching the string `"aba"` against the expression `(a(b)?)+`, for example, leaves group two set to `"b"`. All captured input is discarded at the beginning of each match.

Groups beginning with `(?` are pure, *non-capturing* groups that do not capture text and do not count towards the group total.

Unicode support

This class follows *Unicode Technical Report #18: Unicode Regular Expression Guidelines*, implementing its second level of support though with a slightly different concrete syntax.

Unicode escape sequences such as `\u2014` in Java source code are processed as described in ?3.3 of the *Java Language Specification*. Such escape sequences are also implemented directly by the regular-expression parser so that Unicode escapes can be used in expressions that are read from files or from the keyboard. Thus the strings `"\u2014"` and `"\\u2014"`, while not equal, compile into the same pattern, which matches the character with hexadecimal value `0x2014`.

Unicode blocks and categories are written with the `\p` and `\P` constructs as in Perl. `\p{`*prop*`}` matches if the input has the property *prop*, while \P{*prop*} does not match if the input has that property. Blocks are specified with the prefix `In`, as in `InMongolian`. Categories may be specified with the optional prefix `Is`: Both `\p{L}` and `\p{IsL}` denote the category of Unicode letters. Blocks and categories can be used both inside and outside of a character class.

The supported blocks and categories are those of *The Unicode Standard, Version 3.0*. The block names are those defined in Chapter 14 and in the file Blocks-3.txt of the Unicode Character Database except that the spaces are removed; `"Basic Latin"`, for example, becomes `"BasicLatin"`. The category names are those defined in table 4-5 of the Standard (p. 88), both normative and informative.

Comparison to Perl 5

Perl constructs not supported by this class:

- The conditional constructs (`?{`*X*`}`) and (`?(`*condition*`)`*X*`|`*Y*),

- The embedded code constructs (`?{`*code*`}`) and (`??{`*code*`}`),

- The embedded comment syntax (`?#comment`), and

- The preprocessing operations `\l\u` , `\L`, and `\U`.

Constructs supported by this class but not by Perl:

- Possessive quantifiers, which greedily match as much as they can and do not back off, even when doing so would allow the overall match to succeed.

- Character-class union and intersection as described above.

Notable differences from Perl:

- In Perl, `\1` through `\9` are always interpreted as back references; a backslash-escaped number greater than `9` is treated as a back reference if at least that many subexpressions exist, otherwise it is interpreted, if possible, as an octal escape. In this class octal escapes must always begin with a zero. In this class, `\1` through `\9` are always interpreted as back references, and a larger number is accepted as a back reference if at least that many subexpressions exist at that point in the regular expression, otherwise the parser will drop digits until the number is smaller or equal to the existing number of groups or it is one digit.

- Perl uses the `g` flag to request a match that resumes where the last match left off. This functionality is provided implicitly by the `Matcher` class: Repeated invocations of the `find` method will resume where the last match left off, unless the matcher is reset.

- In Perl, embedded flags at the top level of an expression affect the whole expression. In this class, embedded flags always take effect at the point at which they appear, whether they are at the top level or within a group; in the latter case, flags are restored at the end of the group just as in Perl.

- Perl is forgiving about malformed matching constructs, as in the expression `*a`, as well as dangling brackets, as in the expression `abc]`, and treats them as literals. This class also accepts dangling brackets but is strict about dangling metacharacters like +, ? and *, and will throw a `PatternSyntaxException` if it encounters them.

For a more precise description of the behavior of regular expression constructs, please see *Mastering Regular Expressions, 2nd Edition*, Jeffrey E. F. Friedl, O'Reilly and Associates, 2002.

DTD Digester XML

The following DTD (Document Type Definition) `summary.dtd` models the Digester XML format. Each Digester XML file needs to fully comply with the shown structure to enable applications to process

the contained data properly. A detailed description of the Digester XML format can be found in section [Supported File Formats](). You can reference this file with the following Internet address:

```
http://ftp.leisenfels.com/dtd/summary-1.1.dtd
```

```
<!--
    Copyright (c) 2005-2015 Leisenfels. All rights reserved.
    Use is subject to license terms.
-->
<!ELEMENT summary ( comment?, target* )>
<!ATTLIST summary
          version (1.1) #REQUIRED
          date       CDATA #IMPLIED
          targets    CDATA #REQUIRED>
<!ELEMENT comment ( #PCDATA )>
<!ELEMENT target ( url*, digest*, pgpsig* )>
<!ATTLIST target
          abspath    CDATA #IMPLIED
          relpath    CDATA #REQUIRED
          length     CDATA #REQUIRED
          modified   CDATA #REQUIRED
          digests    CDATA #REQUIRED
          pgpsigs    CDATA #REQUIRED>
<!ELEMENT url ( #PCDATA )>
<!ELEMENT digest ( #PCDATA )>
<!ATTLIST digest
          algorithm  CDATA #REQUIRED
          size       CDATA #IMPLIED
          pos        CDATA #IMPLIED
          format (hex | base64) #REQUIRED>
<!ELEMENT pgpsig ( #PCDATA )>
<!ATTLIST pgpsig
          keyid      CDATA #IMPLIED
          keyname    CDATA #IMPLIED
          keyemail   CDATA #IMPLIED
          keyurl     CDATA #IMPLIED
          size       CDATA #IMPLIED
          format (ascii) #REQUIRED>
```

Additional Links

- [Supported File Formats]()