

FlexTk

File Management Toolkit



FlexTk XML Format

Version 4.8

Feb 2011

Flexense Ltd.
www.flexense.com
info@flexense.com

1	General	3
1.1	Overview.....	3
1.2	Rules List	3
1.3	Nested Rules	3
2	File Matching Rules	4
2.1	File Extension	4
2.2	File Name and Path	5
2.3	File Size	6
2.4	File Time	7
2.5	File Date	8
2.6	Text Patterns	9
2.7	Binary Patterns	10
2.8	File Category	11
2.9	JPEG EXIF Tags.....	12
2.10	Windows File Attributes.....	13
2.11	UNIX File Permissions	14
2.12	User Name	15
3	File Classification Plugins.....	16
3.1	Overview.....	16
3.2	Structure.....	16
3.3	Plugin Info.....	17
3.4	Class List.....	17
3.5	Class Definition	18
3.6	Nested Rules	18
3.7	Class Groups	19
3.8	Nested Groups	19
3.9	Example File Classification Plugin	20
4	User-Defined Commands.....	21
4.1	File Search	22
4.2	File Classification.....	25
4.3	Storage Utilization Analysis.....	27
4.4	Duplicate Files Search.....	30
4.5	File Synchronization.....	33
4.6	File Organizing.....	35
4.7	File Copy	38
4.8	File Delete	40
4.9	Disk Monitoring	42

1 General

1.1 Overview

The FlexTk file management toolkit extensively uses various types of file matching rules for all the supported file management operations ranging from basic file search to advanced file classification and storage utilization analysis. In general, file matching rules define the set of files on which a file management operation should be performed. For example, in file synchronization, file copy, file delete and other operations, the user is provided with the ability to process files matching user-defined criteria. These user-defined criteria may be specified using the XML format explained in this document.

1.2 Rules List

The rules list is the most fundamental element of the XML format and it is widely used in file search, file classification and other file management operations. A rule list may contain one or more file matching rules. FlexTk supports a large number of various types of rules capable of matching files by the file extension, name, path, size, time, content, etc., which are explained in details in subsequent sections of this document. During the file matching process, file matching rules are evaluated in the same order as they are specified in the rules list.

```
<rules logic='OR'>
  <r1>Rule Data</r1>
  <r2>Rule Data</r2>
  ...
  <rX>Rule Data</rX>
</rules>
```

Rules Logic:

Logic	Description
AND (Default)	If the rules logic is set to 'AND', the rules list matches a file when all the specified rules evaluated to TRUE. Rules are evaluated until a first FALSE evaluation, after which, all the remaining file matching rules are dismissed.
OR	If the rules logic is set to 'OR', the rules list matches a file when any of the specified rules evaluated to TRUE. Rules are evaluated until a first TRUE evaluation, after which, all the remaining file matching rules are skipped.

1.3 Nested Rules

Sometimes, it may be necessary to match files according to multiple criteria with nested logical operations. For example, it may be required to organize or analyze all documents and image files that were modified during the last month. In this case, we need to define a composite rules list where the top level list will contain a rule matching files that were modified during the last month and a nested rules list containing rules matching documents and image files.

```
<rules logic='AND'>
  <r1>Rule matching file modified during this month</r1>
  ...
  <rules logic='OR'>
    <r2>Rule matching all types of document files</r2>
    <r3>Rule matching all types of image files</r3>
  </rules>
</rules>
```

2 File Matching Rules

2.1 File Extension

The '**extension**' rule allows one to match files by a specific pattern in the file extension. The rule accepts up to 8 extension patterns for each declaration. By default, the '**extension**' rule matches all files having one of the specified file extensions. If the rule operator is set to 'NE' (not equal), the rule matches all files with the file extension matching none of the specified extension patterns. In addition to the EQ/NE operators, the user is provided with numerous advanced pattern matching operators allowing one to match files with extensions beginning with, ending with or containing specific extension patterns.

Format:

`<extension op='Operator'>ext1 ext2 ... extX</extension>`

Operators:

Operator	Description
EQ (Default)	Matches files having one of the specified extensions.
NE	Matches files having none of the specified extensions.
CS	Matches files with extensions containing one of the specified strings.
NC	Matches files with extensions containing none of the specified strings.
BW	Matches files with extensions beginning with one of the specified strings.
NB	Matches files with extensions beginning with none of the specified strings.
EW	Matches files with extensions ending with one of the specified strings.
NS	Matches files with extensions ending with none of the specified strings.

Examples:

Rule	Description
<code><extension>mpg mp3 mpeg</extension></code>	Matches all files with extensions: 'MPG', 'MP3' and 'MPEG'
<code><extension op='BW'>mp</extension></code>	Matches all files with extensions beginning with: 'MP'
<code><extension op='NE'>mpg mp3</extension></code>	Matches all files except files with extensions: 'MPG' and 'MP3'.
<code><extension op='CS'>htm</extension></code>	Matches all files containing pattern 'HTM' in files' extensions including 'HTM', 'HTML', 'SHTML', etc.

2.2 File Name and Path

FlexTk provides a number of file matching rules allowing one to match files by a specific text pattern in the file name, file path, full file name (file name including full path) and file name only (without extension and path). By default, all name rules will match files having the specified text pattern in the selected part of the file name. If the rule operator is set to 'NE' (not equal), these rules will match files with file names different from the specified text pattern. In addition to the EQ/NE operators, the user is provided with numerous advanced pattern matching operators allowing one to match files with names beginning with, ending with or containing specific text patterns.

Format:

<RuleName op='Operator'>Text Pattern</RuleName>

Rules:

Rule	Description
NAME	Compares the file name with the specified text pattern
PATH	Compares the file path (path without the file name) with the specified pattern.
FULLNAME	Compares the full file name (file name and path) with the specified pattern.
DIRNAME	Compares the name of the directory (without path) with the specified pattern.
NAMEONLY	Compares the file name only (without path and extension).

Operators:

Operator	Description
EQ (Default)	Matches files with the file name identical to the specified text pattern.
NE	Matches files with the file name different from the specified text pattern.
CS	Matches files with the file name containing the specified text pattern.
NC	Matches files with the file name not containing the specified text pattern.
BW	Matches files with the file name beginning with the specified text pattern.
NB	Matches files with the file name not beginning with the specified text pattern.
EW	Matches files with the file name ending with the specified text pattern.
NS	Matches files with the file name not ending with the specified text pattern.

Examples:

Rule	Description
<name>readme.txt</name>	Matches all files with the file name: README.TXT
<name op='BW'>readme</name>	Matches all files with the file name beginning with 'README' including: README, README.TXT, README.DOC, etc.
<name op='NE'>readme</name>	Matches all files except files with the file name: README
<name op='CS'>report</name>	Matches all files containing the text pattern 'REPORT' in the file name.
<path>C:\</path>	Matches files located in the root directory of disk 'C:\'.
<fullname op='BW'>c:\readme</fullname >	Matches files with the full file name beginning with: 'C:\README'

2.3 File Size

The **'size'** rule allows one to match files by a specific file size. By default, the **'size'** rule matches all files having the specified file size. If the rule operator is set to **'NE'** (not equal), the rule matches all files with the file size different from the specified value. In addition to the basic EQ/NE operators, the user is provided with numerous advanced operators allowing one to match files with the file size less than, less or equal, more than, more or equal to the specified value. Finally, there are two range-based operators capable of matching files having the file size within or outside of a specific range.

Basic Format:

`<size op='Operator'>Size Modifier(Value)</size>`

Range Format:

`<size op='Range Operator'>Size Modifier(From Value ... To Value)</size>`

Operators:

Operator	Description
EQ (Default)	Matches files with the file size equal to the specified value.
NE	Matches files with the file size different from the specified value.
GT	Matches files with the file size greater than the specified value.
GE	Matches files with the file size greater or equal to the specified value.
LT	Matches files with the file size less than the specified value.
LE	Matches files with the file size less or equal to the specified value.

Range Operators:

Operator	Description
IN	Matches files having the file size within the specified range
NI	Matches files having the file size outside of the specified range

Size Modifiers:

Modifier	Description
Bytes	The file size value or range is specified in Bytes.
KBytes	The file size value or range is specified in Kilobytes.
MBytes	The file size value or range is specified in Megabytes.
GBytes	The file size value or range is specified in Gigabytes.

Examples:

Rule	Description
<code><size>Bytes(1024)</size></code>	Matches files with the file size 1024 bytes
<code><size op='LT'>Bytes(1024)</size></code>	Matches files with the file size less than 1024 Bytes
<code><size op='GT'>MBytes(10)</size></code>	Matches files with the file size greater than 10 MBytes
<code><size op='IN'>MBytes(1...100)</size></code>	Matches files with the file size from 1 to 100 MBytes.
<code><size op='EQ'>Bytes(0)</size></code>	Matches zero-length files.
<code><size op='NE'>Bytes(0)</size></code>	Matches non-zero-length files.

2.4 File Time

FlexTk provides a number of file matching rules allowing one to match files by a specific file creation, last modification or last access time. By default, all time rules match files having the specified time value in the selected file time. If the rule operator is set to 'NE' (not equal), these rules will match files with the selected file time different from the specified time value. In addition to the basic EQ/NE operators, the user is provided with numerous advanced operators allowing one to match files with the file time less than, less or equal, more than, more or equal to the specified time value. Finally, there are two range-based operators capable of matching files having the selected file time within or outside of a specific time range.

Basic Format:

<RuleName op='Operator'>Time Modifier(Time Value)</RuleName>

Range Format:

<RuleName op='Operator'>Time Modifier(From ... To)</RuleName>

Rules:

Rule	Description
CTIME	Compares the file creation time with the specified time value.
MTIME	Compares the file last modification time with the specified time value.
ATIME	Compares the file last access time with the specified time value.

Operators:

Operator	Description
EQ	Matches files with the file time equal to the specified value.
NE	Matches files with the file time different from the specified value.
GT	Matches files with the file time greater than the specified value.
GE	Matches files with the file time greater or equal to the specified value.
LT	Matches files with the file time less than the specified value.
LE	Matches files with the file time less or equal to the specified value.

Range Operators:

Operator	Description
IN	Matches files having the selected file time within the specified range
NI	Matches files having the selected file time outside of the specified range

Time Modifiers:

Modifier	Description
Days	The file time value or range is specified in days.
Weeks	The file time value or range is specified in weeks.
Months	The file time value or range is specified in months.
Years	The file time value or range is specified in years.

Examples:

Rule	Description
<atime op='GT'>Weeks(2)</atime>	Matches files last accessed more than 2 weeks
<ctime op='LE'>Months(3)</ctime>	Matches files created during the last 3 months
<mtime op='IN'>Years(1...5)</mtime>	Matches files last modified from 1 to 5 years

2.5 File Date

In addition to the relative time-based file matching rules, FlexTk provides a set of rules allowing one to match files by absolute creation, last modification or access dates. By default, all date rules match files having the specified date value in the selected file date. If the rule operator is set to 'NE' (not equal), these rules will match files with the selected file date different from the specified date value. In addition to the basic EQ/NE operators, the user is provided with numerous advanced operators allowing one to match files with the file date less than, less or equal, more than, more or equal to the specified date value. Finally, there are two range-based operators capable of matching files having the selected file date within or outside of a specific date range.

Basic Format:

`<RuleName op='Operator'>Absolute Date</RuleName>`

Range Format:

`<RuleName op='Operator'>From Absolute Date ... To Absolute Date</RuleName>`

Rules:

Rule	Description
CDATE	Compares the file creation date with the specified date value.
MDATE	Compares the file last modification date with the specified date value.
ADATE	Compares the file last access date with the specified date value.

Operators:

Operator	Description
EQ	Matches files with the file date equal to the specified value.
NE	Matches files with the file date different from the specified value.
GT	Matches files with the file date greater than the specified value.
GE	Matches files with the file date greater or equal to the specified value.
LT	Matches files with the file date less than the specified value.
LE	Matches files with the file date less or equal to the specified value.

Range Operators:

Operator	Description
IN	Matches files having the selected file date within the specified range
NI	Matches files having the selected file date outside of the specified range

Absolute Date Format:

Absolute dates should be specified in the following format:

YYYY/MM/DD

Examples:

Rule	Description
<code><adate>2009/03/01</adate></code>	Matches files last accessed on 1-Mar-2009
<code><cdate op='LT'>2009/03/01</cdate></code>	Matches files created before 1-Mar-2009
<code><mdate op='IN'>2009/02/01...2009/03/01</mdate></code>	Matches files last modified from 1-Feb-2009 to 1-Mar-2009

2.6 Text Patterns

The **'text'** rule allows one to match files containing or not containing a specific text pattern. By default, the **'text'** rule matches all files containing the specified text pattern. If the rule operator is set to **'NC'** (not contain), the rule matches files not containing the specified text pattern.

Format:

`<text op='Operator'>Text Pattern</text>`

Operators:

Operator	Description
CS (Default)	Matches files containing the specified text pattern.
NC	Matches files not containing the specified text pattern.

Examples:

Rule	Description
<code><text>Have a Good Day</text></code>	Matches file containing the specified text.
<code><text op='NC'>Have a Good Day</text></code>	Matches files not containing the specified text.
<pre> <rules logic='OR'> <text>Pattern 1</text> <text>Pattern 2</text> </rules> </pre>	Matches files containing 'Pattern 1' or 'Pattern 2'.
<pre> <rules logic='AND'> <text op='CS'>Pattern 1</text> <text op='NC'>Pattern 2</text> </rules> </pre>	Matches files containing 'Pattern 1', but not containing 'Pattern 2'.

Special Cases:

1. If the operator is set to **'CS'** and the file size is shorter than the specified text pattern, the rule evaluates to **FALSE** without checking the text pattern.
2. If the operator is set to **'NC'** and the file size is shorter than the specified text pattern, the rule evaluates to **TRUE** without checking the text pattern.

2.7 Binary Patterns

The **'pattern'** rule allows one to match files containing or not containing a specific binary pattern at a user-specified position. By default, the **'pattern'** rule matches all files having the specified binary pattern at the specified position. If the rule operator is set to 'NE' (not equal), the rule matches files not containing the specified binary pattern at the specified position.

Format:

```
<pattern op='Operator' pos='Position'>Binary Pattern</pattern>
```

Operators:

Operator	Description
EQ (Default)	Matches files containing the specified pattern at the specified position.
NE	Matches files not containing the specified pattern at the specified position.

Position:

The position attribute specifies the absolute position from the beginning of the file where the specified binary pattern should be found. If not specified, FlexTk will search for the specified binary pattern at the position 0.

Binary Pattern:

The binary pattern should be specified as one or more two-symbol HEX tokens delimited by the space character with each token representing a single byte of data. The length of the binary pattern is determined by the amount of the specified HEX tokens. For example, 'FF FE FD FE' represents a four-byte binary pattern. The **'pattern'** rule is capable of matching binary patterns of up to 255 bytes long.

Examples:

Rule	Description
<code><pattern pos='12'>FF FE FD FC</pattern></code>	Matches files containing the pattern 'FF FE FD FE' at the position 12
<code><pattern>4D 5A</pattern></code>	Matches files containing the pattern '4D 5A' at the position 0
<code><pattern op='NE'>4D 5A</pattern></code>	Matches files not containing the pattern '4D 5A' at the position 0
<pre><rules logic='AND'> <pattern pos='1'>4D 5A</pattern> <pattern pos='6'>5A 4D</pattern> </rules></pre>	Matches files containing the pattern '4D 5A' at the position 1 and the pattern '5A 4D' at the position 6.

Special Cases:

1. If the operator is set to 'EQ' and after setting the position the remaining file size is shorter than the length of the specified binary pattern, the rule evaluates to FALSE without checking the binary pattern.
2. If the operator is set to 'NE' and after setting the position the remaining file size is shorter than the length of the specified binary pattern, the rule evaluates to TRUE without checking the binary pattern.

2.8 File Category

The **'category'** rule matches files related to a specific file category or file class in a file classification plugin. If the user has specified an individual file class, the **'category'** rule will match all files related to the specified file class. If the user has specified a class group, the **'category'** rule will match all files related to any file class in the specified class group including all classes in all subgroups. If the rule operator is set to 'NE' (not equal), the rule matches files not related to the specified class group or file class.

Format:

```
<category op='Operator'>
```

```

  <plugin>File Classification Plugin Name</plugin>
  <group>Full Group Name</group>                (Optional)
  <class>Optional Class Name</class>            (Optional)
  <display>Optional Display String</display>    (Optional)

```

```
</category>
```

Operators:

Operator	Description
EQ (Default)	Matches files related to the specified file category.
NE	Matches files not related to the specified file category.

Elements:

Element	Description
Plugin	The 'plugin' element specifies the name of the file classification plugin that should be used during the file matching process. The specified file classification plugin should be registered in the product configuration. If the specified file classification plugin does not exist or not valid, the whole operation will be aborted.
Group	The 'group' element specifies the name of the class group that should be used during the file matching process. If the user has specified a class group and not specified a file class, the 'category' rule will match all files related to any file class in the specified class group including all classes in all subgroups. Subgroups should be specified including full names of all parent groups delimited by the '/' character.
Class	The 'class' element is an optional element allowing one to specify the name of an individual file class that should be used during the file matching process. If the user has specified a file class, the 'category' rule will match files related to this specific class only.
Display	The 'display' element is an optional element allowing one to specify a short textual description of the category to be displayed in the FlexTk GUI application instead of the full group/class name. If not specified, FlexTk will construct a default display name automatically.

Examples:

Rule	Description
<pre><category> <plugin>Default File Hierarchy</plugin> <group>Music and Audio Files</group> </category></pre>	Matches all files related to the 'Music and Audio Files' group in the 'Default File Hierarchy' file classification plugin.

2.9 JPEG EXIF Tags

FlexTk provides the user with the ability to perform operations on JPEG images by one or more standard or custom EXIF tags. By default, the EXIF-based file matching rule match JPEG images having the specified text pattern in the selected EXIF tag. If the rule operator is set to 'NE' (not equal), the rule will match JPEG images with the selected EXIF tag different from the specified text pattern. In addition to the EQ/NE operators, the user is provided with numerous advanced pattern matching operators allowing one to match JPEG images with EXIF tags beginning with, ending with or containing specific text patterns.

Format:

```
<exif op='Operator' tag='EXIF Tag ID'>Text Pattern</exif>
```

EXIF Tag ID:

The EXIF tag ID should be specified as a HEX number without the leading '0x' prefix. FlexTk allows one to match files by any standard or custom EXIF tag ID.

Operators:

Operator	Description
EQ (Default)	Matches files with the EXIF tag identical to the specified text pattern.
NE	Matches files with the EXIF tag different from the specified text pattern.
CS	Matches files with the EXIF tag containing the specified text pattern.
NC	Matches files with the EXIF tag not containing the specified text pattern.
BW	Matches files with the EXIF tag beginning with the specified text pattern.
NB	Matches files with the EXIF tag not beginning with the specified text pattern.
EW	Matches files with the EXIF tag ending with the specified text pattern.
NS	Matches files with the EXIF tag not ending with the specified text pattern.

Examples:

Rule	Description
<code><exif tag='10F'>NIKON CORPORATION</exif></code>	Matches JPEG images with the EXIF tag '10F' (Camera Vendor) set to 'NIKON CORPORATION'.
<code><exif op='BW' tag='110'>NIKON</exif></code>	Matches JPEG images with the EXIF tag '110' (Camera Model) beginning with 'NIKON'.
<code><exif op='NE' tag='A002'>1024</exif></code>	Matches JPEG images with the EXIF tag 'A002' (Image Width) not set to 1024.

2.10 Windows File Attributes

The `'attr'` rule allows one to match files by one or more of Windows file attributes. By default, the `'attr'` rule matches files having all of the specified file attributes. If the rule operator is set to `'NE'` (not equal), the rule matches files with none of the specified Windows file attributes.

Format:

```
<attr op='Operator'>Attribute List</attr>
```

Operators: EQ, NE

Operator	Description
EQ (Default)	Matches files having all the specified Windows file attributes.
NE	Matches files having none of the specified Windows file attributes.

Attribute List:

The attribute list may include one or more of the following Windows file attributes: ARCHIVE, COMPRESSED, ENCRYPTED, HIDDEN, NORMAL, OFFLINE, READONLY, REPARSE-POINT, SPARSE-FILE, SYSTEM, TEMPORARY. For detailed information about meaning of each specific file attribute refer to the Windows help.

Examples:

Rule	Description
<code><attr>SYSTEM HIDDEN</attr></code>	Matches system and hidden files.
<code><attr op='NE'>SYSTEM</attr></code>	Matches non-system files.
<pre><rules logic='AND'> <attr op='EQ'>HIDDEN</attr> <attr op='NE'>SYSTEM</attr> </rules></pre>	Matches hidden, but not system files.

2.11 UNIX File Permissions

The `'perm'` rule allows one to match files by UNIX file access permissions. By default, the `'perm'` rule matches files having all the specified file permissions. If the rule operator is set to `'NE'` (not equal), the rule matches files with not matching file permissions.

Format:

```
<perm op='Operator'>Permissions</perm>
```

Operators: EQ, NE

Operator	Description
EQ (Default)	Matches files having the specified UNIX file permissions.
NE	Matches files not having the specified UNIX file permissions.

Permissions:

File permissions should be specified as three octal (0-7) digits specifying user, group and other permissions respectively. For more detailed information about UNIX file permissions refer to UNIX or Linux operating system manuals.

Examples:

Rule	Description
<code><perm>777</perm></code>	Matches files with read, write and execute permissions enabled for the user, group and other.
<code><perm op='NE'>777</perm></code>	Matches files with file permissions different from 777.
<pre><rules logic='OR'> <perm>777</perm> <perm>666</perm> </rules></pre>	Matches files with file permissions set to 777 or 666.

2.12 User Name

The **'user'** rule allows one to match files by the user name. By default, the **'user'** rule matches all files having the specified user name. If the rule operator is set to **'NE'** (not equal), the rule matches all files with the user name different from the specified value. In addition to the EQ/NE operators, the user is provided with numerous advanced pattern matching operators allowing one to match files with the user name beginning with, ending with or containing specific text patterns.

Format:

```
<user op='Operator'>User Name</user>
```

Operators:

Operator	Description
EQ (Default)	Matches files owned by the specified user.
NE	Matches files with the user name different from the specified pattern.
CS	Matches files with the user name containing the specified pattern.
NC	Matches files with the user name containing none of the specified patterns.
BW	Matches files with the user name beginning with the specified pattern.
NB	Matches files with the user name not beginning with the specified pattern.
EW	Matches files with the user name ending with the specified pattern.
NS	Matches files with the user name not ending with the specified pattern.

Examples:

Rule	Description
<code><user>Admisistrator</user></code>	Matches all files owned by user 'Administrator'
<code><user op='BW'>Admin</user></code>	Matches all files related to user names beginning with 'Admin'
<code><user op='NE'>root</user></code>	Matches all files not owned by 'root'.
<code><user op='CS'>Admin</user></code>	Matches all files with the user name containing pattern 'Admin'

3 File Classification Plugins

3.1 Overview

The FlexTk file management toolkit is capable of scanning disks, network shares and enterprise storage systems, classifying all the existing files and organizing them into a categorized file hierarchy. File classification and categorization operations are performed using file classification plugins. By default, FlexTk comes with the 'Default File Hierarchy' file classification plugin and provides numerous additional file classification plugins that may be installed from the FlexTk plugins server.

In addition to file classification plugins provided by Flexense, FlexTk allows one to add custom file classification plugins especially designed for user-specific needs and requirements. This section explains how to write custom FlexTk file classification plugins using an XML-Based format. Properly written XML classification plugins may be imported into the FlexTk product configuration and used in various file management operations including file classification, file search, file synchronization, storage utilization analysis, etc.

3.2 Structure

A file classification plugin consists of three distinct sections: the plugin info section, the class list section and the group list section. The plugin info section includes information about the plugin version, the required FlexTk version, the plugin protection status and an optional author name.

Format:

```
<plugin name='Plugin Name'>
```

```
  <!-- Plugin Info -->
```

```
    <version>Plugin Version</version>
    <flexTk>Required FlexTk Version</flexTk>
    <protected>Protection Status</protected>           (Optional)
    <author>Module Author</author>                     (Optional)
    <match>Maximum Match Count</match>                 (Optional)
```

```
  <!-- Class List -->
```

```
    <class1>Class Data</class1>
    ...
    <classX>Class Data</classX>
```

```
  <!-- Group List -->
```

```
    <group1>Group Data</group1>
    ...
    <groupX>Group Data</groupX>
```

```
</plugin>
```

The class list should include declarations for one or more file classes specifying file matching rules that should be used in order to detect each specific file class. The group list should include declarations for one or more class groups specifying references to all file classes related to each specific class group. In order to be registered in the FlexTk product configuration, each file classification plugin should have a unique plugin name. Use the 'name' attribute in the 'plugin' element to specify a unique file classification plugin name.

3.3 Plugin Info

Operator	Description
Version	The ' version ' element specifies the version of the file classification plugin. FlexTk uses the plugin version to show the plugin status in the plugins management dialog. The plugin version should be specified in the format 'X.Y.Z' (tree decimal numbers delimited by the '.' Character) where X represents the major version, Y represents the minor version and Z represents the revision of the file classification plugin.
FlexTk	The ' flextk ' element specifies the minimal version of the FlexTk file management toolkit required for this specific file classification plugin. The FlexTk version should be specified in the format 'X.Y.Z' (tree decimal numbers delimited by the '.' Character) where X represents the major version, Y represents the minor version and Z represents the revision of the required version of the FlexTk file management toolkit.
Protected	The ' protected ' element specifies the plugin protection status. The protection status may be set to 'YES' (default is 'YES') or 'NO'. If the protection status is set to 'YES', FlexTk will prevent the file classification plugin from being exported to any of the supported file formats. If the protection status is set to 'NO', FlexTk will allow export of the plugin to all the supported file formats.
Author	The ' author ' element is an optional element allowing one to specify the plugin's author name (default is 'Flexense'). The plugin's author may be specified as a free-format text string up to 255 characters long.
Match	The ' match ' element is an optional element (default value is 1) allowing one to specify the maximum number of matches for each file during the file classification process. Sometimes, it may be useful to categorize each file to more than one file class. This parameter controls the maximum number of classes each file may be related to.

3.4 Class List

Each file classification plugin should define one or more file classes specifying file matching rules that will be used to detect each specific file class. Each file class in the class list should have a unique class name. During the file classification process, FlexTk will pass each file through the class list and evaluate file matching rules specified in each specific file class.

Format:

```
<class name='Class Name 1'>
    <rules>
        <!-- File Matching Rules -->
    </rules>
</class>
...
<class name='Class Name X'>
    <rules>
        <!-- File Matching Rules -->
    </rules>
</class>
```

File classes are evaluated in the same order as they are defined in the class list. The maximum match count parameter (described in the plugin info section) controls the maximum number of matches allowed for each file. If the parameter is set to 1 (default), FlexTk will process each file until a first matching class and skip all remaining classes in the class list.

3.5 Class Definition

Each file class should specify one or more file matching rules that will be used during the file classification process. All the specified file matching rules will be evaluated in the same order as they are specified in the class definition. FlexTk supports a large number of various types of rules capable of matching files by the file extension, name, path, size, time, content, etc. For detailed information about all the supported types of file matching rules refer to the section 1 in this document.

Format:

```
<class name='Class Name'>
  <rules logic='or'>
    <r1>Rule Data</r1>
    <r2>Rule Data</r2>
    ...
    <rX>Rule Data</rX>
  </rules>
</class>
```

3.6 Nested Rules

Sometimes, it may be necessary to match files according to multiple criteria with nested logical operations. For example, it may be required to organize or analyze all documents and image files that were modified during the last month. In this case, we need to define a composite rule list where the top level list will contain a rule matching files that were modified during the last month and the nested rule list will contain rules matching documents or image files.

Example:

```
<class name='Class Name'>
  <rules logic='AND'>
    <r1>Rule matching file modified during this month</r1>
    ...
    <rules logic='OR'>
      <r2>Rule matching all types of document files</r2>
      <r3>Rule matching all types of image files</r3>
    </rules>
  </rules>
</class>
```

The FlexTk file management toolkit allows one to physically reorganize data on the disk according to the logical hierarchy defined by a file classification plugin. During this operation, FlexTk builds a hierarchy of directories according to the hierarchy of class groups and file classes as it is defined in the selected file classification plugin. In order to use this specific capability, the character set used for group and class names should be limited to characters allowed for file names on the target file system. Due to this reason the following characters are not allowed to be used in group and class names: '/', '\', ':', '*', '?', '>', '<', '|'.

3.7 Class Groups

Each file classification plugin should define one or more class groups with references to one or more file classes related to each specific class group. Each class group should have a unique group name in the group list.

Format:

```
<group name='Group Name 1'>
    <class>Class Name 1</class>
    ...
    <class>Class Name 2</class>
</group>
...
<group name='Group Name X'>
    <class>Class Name 3</class>
    ...
    <class>Class Name 4</class>
</group>
```

3.8 Nested Groups

The group list may be organized into a logical hierarchy with an unlimited number of nested class groups. The hierarchy of class groups is used by the FlexTk file management toolkit to display file classification results and perform file management operations on file classification results.

Format:

```
<group name='Top Level Group'>
    <group name='Nested Group 1'>
        <class>Class Name 1</class>
        <class>Class Name 2</class>
        ...
        <class>Class Name X</class>
    </group>
    <group name='Nested Group 2'>
        <class>Class Name 1</class>
        <class>Class Name 2</class>
        ...
        <class>Class Name Y</class>
    </group>
</group>
```

The FlexTk file management toolkit allows one to physically reorganize data on the disk according to the logical hierarchy defined by a file classification plugin. During this operation, FlexTk builds a hierarchy of directories according to the hierarchy of class groups and file classes as it is defined in the selected file classification plugin. In order to use this specific capability, the character set used for group and class names should be limited to characters allowed for file names on the target file system. Due to this reason the following characters are not allowed to be used in group and class names: `'/','\',':', '*','?', '>', '<', '|'`.

3.9 Example File Classification Plugin

Finally, let's write an example file classification plugin capable of categorizing Microsoft office documents, text files, image files and Photoshop graphics files into a simple, two-group file hierarchy.

Example Plugin:

```
<plugin name='My File Classification Plugin'>
  <!-- Plugin Info -->
  <version>1.0.0</version>
  <flextk>3.0.0</flextk>
  <!-- Class List -->
  <class name='Text Files'>
    <rules>
      <extension>TXT</extension>
    </rules>
  </class>
  <class name='Microsoft Office Files'>
    <rules>
      <extension>DOC PPT XLS</extension>
    </rules>
  </class>
  <class name='Image Files'>
    <rules>
      <extension>JPG BMP PNG</extension>
    </rules>
  </class>
  <class name='Photoshop Graphic Files'>
    <rules>
      <extension>PSD PDD</extension>
    </rules>
  </class>
  <!-- Group List -->
  <group name='Text Files and Documents'>
    <class>Text Files</class>
    <class>Microsoft Office Files</class>
  </group>
  <group name='Graphic Files'>
    <class>Image Files</class>
    <class> Photoshop Graphic Files</class>
  </group>
</plugin>
```

The plugin info section defines the plugin version and the minimum required FlexTk version. The class list defines four file classes with file matching rules detecting each specific file type by the file extension. The group list organizes all the defined file classes into two class groups.

4 User-Defined Commands

The FlexTk file management toolkit is a powerful and flexible product providing a large number of features, capabilities and configuration options allowing one to perform many different file and storage management operations. In order to simplify management and execution of frequently used operations, FlexTk provides the user with the ability to add user-defined commands. All major product features such as file search, classification, organizing, synchronization, copy, move, delete, storage utilization analysis, duplicate files search and disk monitoring may be added as user-defined commands and pre-configured for user-specific needs and custom hardware configurations.

In addition to the user-defined commands management capabilities available in the FlexTk main GUI application, the user is provided with the ability to write user-defined commands in an XML-Based format. The XML-Based format, which is explained in this section, provides the user with full control over all the features available in the FlexTk main GUI application. Properly written XML user-defined commands may be imported to the FlexTk's product configuration or executed on-the-fly using the FlexTk's command line tool (FlexTk Ultimate).

Example Command #1:

```
<search name='Search Command Name'>
  <inputs>
    <dir>D:\</dir>
    <dir>F:\</dir>
  </inputs>
  <rules logic='AND'>
    <extension>JPG BMP PNG </extension>
    <size op='GT'>MBytes(10)</size>
  </rules>
</search>
```

The first example command will search for JPG, BMP and PNG image files with the file size greater than 10 MBytes on disks 'D:\' and 'F:\'.

Example Command #2:

```
<classify name='Classify Command Name'>
  <inputs>
    <dir>D:\</dir>
    <dir>F:\</dir>
  </inputs>
  <plugin>Default File Hierarchy</plugin>
  <report type='HTML'>
    <title>My Classification Report</title>
    <save-to>C:\classification_report.html</save-to>
  </report>
</classify>
```

The second example command will classify disks 'D:\' and 'F:\' and save a file classification report to the file 'C:\classification_report.html'. Each specific command type has its own set of parameters and options, which are explained in details in subsequent sections of this document.

4.1 File Search

FlexTk provides powerful and flexible file search capabilities allowing one to search files using multiple search criteria including the file name, file type, file size, access/modification/creation times, text or binary patterns, etc. The user is provided with the ability to define composite search queries using multiple search rules and apply the AND/OR logical operators. Frequently used queries may be saved as user-defined commands and executed in a single mouse click.

In addition to the ability to define search queries using the FlexTk GUI application, FlexTk allows one to specify search queries using an XML-Based format. The XML format provides support for all the search features available in the FlexTk main GUI application including all types of file matching rules, nested rules, logical operators, etc. File search queries defined in the XML format may be imported into the FlexTk product configuration as user-defined commands or executed on-the-fly using the FlexTk command line tool (FlexTk Ultimate).

Format:

```

<search name='Command Name'>

  <inputs>
    <dir>Directory Name 1</dir>
    ...
    <dir>Directory Name X</dir>

    <file>File Name 1</file>
    ...
    <file>File Name X</file>
  </inputs>

  <rules>
    <r1>Rule Data</r1>
    ...
    <rX>Rule Data</rX>
  </rules>

  <file-name>File Name Filter</file-name> (Optional)
  <max-result>Max Search Result</max-result> (Optional)
  <scanning>File Scanning Mode</scanning> (Optional)
  <shortcut>Keyboard Shortcut</shortcut> (Optional)

  <process> (Optional)
    <streams>Number of Process Streams</streams> (Optional)
    <speed file-delay='X' block-delay='Y'>Mode</speed> (Optional)
  </process>

  <report type='Report Type'> (Optional)
    <title>Custom Report Title</title> (Optional)
    <save-to>Report File Name</save-to> (Optional)
  </report>

  <actions results='Number of Search Results'> (Optional)
    <user-command>User Command</user-command> (Optional)
    <custom-command> Command</custom-command> (Optional)
    <email>E-Mail Address</email> (Optional)
  </actions>

  <exclude> (Optional)
    <dir>Exclude Directory 1</dir>
    ...
  </exclude>

</search>

```

Command Inputs

The inputs list should specify all the directories and files that should be processed. Each directory or file in the input list should include a full path name. Depending on the file scanning mode, all the specified directories will be processed sequentially (one after one) or in parallel.

Special Input Modes

In addition to the ability to process explicitly specified inputs, the FlexTk GUI application provides two special input modes allowing one to process external parameters or the current directory instead of the inputs list. User-defined commands with the inputs mode set to the external-parameters mode are accepting drag-and-drop operations into the command item in the commands tool pane. In order to set a user-defined command to the external-parameters mode, specify `<inputs>USER-PARAMS</inputs>` instead of the regular input list.

User-defined commands with the input mode set to the current-directory mode are processing the file navigator's current directory. In order to set a user-defined command to the current-directory inputs mode, specify `<inputs>CURRENT-DIR</inputs>` instead of the input list.

File Matching Rules

The list of file matching rules should specify one or more rules that will be used during the file search process. All the specified file matching rules will be evaluated in the same order as they are specified in the user-defined command. FlexTk supports a large number of various types of rules capable of matching files by the file extension, name, path, size, time, content, etc. For detailed information about all the supported types of file matching rules refer to the section 1 in this document.

Miscellaneous Parameters (Optional)

Parameter	Description
file-name	The ' file-name ' element allows one to specify a file name pattern to search for instead of specifying file matching rules. For example, set the file name pattern to <code>'*.pdf'</code> to find all PDF documents.
max-result	The ' max-result ' element controls the maximum number of search results for the search command. By default, FlexTk finishes the search process after first 1000 search results.
scanning	FlexTk is capable of scanning multiple disks or directories in parallel thus improving the speed of the file search process on multi-CPU/multi-core systems. The ' scanning ' element allows one to control the file scanning mode. Set the scanning mode to ' PARALLEL ' if you wish to scan all input disks and directories simultaneously.
shortcut	The FlexTk GUI application provides the user with the ability to execute user-defined commands using keyboard shortcuts. Use the ' shortcut ' element to specify a keyboard shortcut for the command.

Process Options (Optional)

Parameter	Description
streams	FlexTk is optimized for multi-core/multi-CPU computers and advanced RAID storage systems and capable of scanning multiple file systems in parallel. In order to speed up the file search process, use multiple processing streams when scanning input directories located on multiple physical hard disks or a RAID disk array.
speed	In order to minimize potential performance impact on running production systems, FlexTk allows one to intentionally slow down the file search process. According to your specific needs, set the process speed mode to ' FULL ', ' MEDIUM ', ' LOW ' or ' MANUAL '. The ' MANUAL ' performance mode requires the ' file-delay ' and ' block-delay ' parameters to be specified (ms.).

Automatic Report Generation (Optional)

The **'report'** element instructs FlexTk to automatically save results to a report file. The **'type'** attribute may be set to **'HTML'**, **'TEXT'** or **'CSV'**. An optional **'title'** element may be used to set a custom report title. If the **'save-to'** element is not specified, FlexTk will automatically generate a file name and save the report into the user's home directory. If the specified **'save-to'** value represents an existing directory, FlexTk will automatically generate a file name and save report into the specified directory. In all other cases, FlexTk will save the report using the **'save-to'** value as an absolute file name.

Search Actions

The **'actions'** element allows one to execute various types of actions if the search operation reaches the specified number of search results. The user is provided with the ability to execute a user-defined command, a custom command or send an e-mail notification.

Parameter	Description
results	This parameter specifies the number of search results that will trigger execution of the specified search actions.
user-command	This element specifies the name of a user-defined command that should be executed if the search operation reaches the specified number of search results.
custom-command	This element specifies the name of a custom command, shell script or batch file that should be executed if the search operation reaches the specified number of search results.
email	This element specifies the e-mail address that should be used to send an E-Mail notification if the search operation reaches the specified number of search results.

4.2 File Classification

The FlexTk file management toolkit is capable of scanning disks, network shares and enterprise storage systems, classifying all the existing files and organizing them into a categorized file hierarchy. FlexTk automatically recognizes more than 2000 file types and allows one to browse or export the categorized file hierarchy, perform various actions on specific files or file groups and physically reorganize existing files according to a user-defined list of rules and policies.

In addition to the ability to define file classification commands using the FlexTk GUI application, FlexTk allows one to specify file classification commands using an XML-Based format. The XML format provides support for all the file classification features and capabilities available in the FlexTk main GUI application. File classification commands defined in the XML format may be imported into the FlexTk product configuration as user-defined commands or executed on-the-fly using the FlexTk command line tool (FlexTk Ultimate).

Format:

```
<classify name='Command Name'>
  <plugin>File Classification Plugin Name</plugin>
  <inputs>
    <dir>Directory Name 1</dir>
    ...
    <file>File Name 1</file>
    ...
  </inputs>
  <action type='Action Type'>Action Value</action> (Optional)
  <filter>File Scanning Filter</filter> (Optional)
  <scanning>File Scanning Mode</scanning> (Optional)
  <shortcut>Keyboard Shortcut</shortcut> (Optional)
  <rules> (Optional)
    <r1>Rule Data</r1>
    ...
  </rules>
  <process> (Optional)
    <streams>Number of Process Streams</streams> (Optional)
    <speed file-delay='X' block-delay='Y'>Mode</speed> (Optional)
    <copy>Copy Options</copy> (Optional)
    <verify>Verification Mode</verify> (Optional)
  </process>
  <report type='Report Type'> (Optional)
    <title>Custom Report Title</title> (Optional)
    <save-to>Report File Name</save-to> (Optional)
    <export-levels> Levels To Export</export-levels> (Optional)
    <items-per-level>Items Per Level</items-per-level> (Optional)
  </report>
  <exclude> (Optional)
    <dir>Exclude Directory 1</dir>
    ...
  </exclude>
</classify>
```

Command Inputs

The inputs list should specify all the directories and files that should be processed. Each directory or file in the input list should include a full path name. Depending on the file scanning mode, all the specified directories will be processed sequentially (one after one) or in parallel.

Special Input Modes

In addition to the ability to process explicitly specified inputs, the FlexTk GUI application provides two special input modes allowing one to process external parameters or the current directory instead of the inputs list. User-defined commands with the inputs mode set to the external-parameters mode are accepting drag-and-drop operations into the command item in the commands tool pane. In order to set a user-defined command to the external-parameters mode, specify `<inputs>USER-PARAMS</inputs>` instead of the regular input list.

User-defined commands with the input mode set to the current-directory mode are processing the file navigator's current directory. In order to set a user-defined command to the current-directory inputs mode, specify `<inputs>CURRENT-DIR</inputs>` instead of the input list.

File Matching Rules (Optional)

FlexTk Ultimate provides the user with the ability to limit the file classification process to files matching user-defined criteria. An optional list of file matching rules may specify one or more rules that should be used during the file classification process. All the specified file matching rules will be evaluated in the same order as they are specified in the user-defined command. FlexTk supports a large number of various types of rules capable of matching files by the file extension, name, path, size, time, content, etc. For detailed information about all the supported types of file matching rules refer to the section 1 in this document.

Miscellaneous Parameters (Optional)

Parameter	Description
action	The 'action' element allows one to specify an automatic file classification action that should be performed instead of displaying file classification results. The 'type' attribute may be set to one of the following values: 'LINK' , 'COPY' , 'MOVE' , 'MOVE-AND-LINK' or 'EXECUTE' . For detailed information about all action types and values refer to the FlexTk product manual.
filter	The 'filter' element allows one to specify a file name filter instead of specifying file matching rules. For example, set the file name filter to <code>'*.pdf'</code> to process all PDF documents.
scanning	FlexTk is capable of scanning multiple disks or directories in parallel thus improving the speed of the file classification process on multi-CPU/multi-core systems. The 'scanning' element allows one to control the file scanning mode. Set the scanning mode to 'PARALLEL' if you wish to scan all input disks and directories simultaneously.

Process Options (Optional)

Parameter	Description
streams	FlexTk is optimized for multi-core/multi-CPU computers and advanced RAID storage systems and capable of scanning multiple file systems in parallel. In order to speed up the file classification process, use multiple processing streams when scanning input directories located on multiple physical hard disks or a RAID disk array.
speed	In order to minimize potential performance impact on running production systems, FlexTk allows one to intentionally slow down the file classification process. According to your specific needs, set the process speed mode to 'FULL' , 'MEDIUM' , 'LOW' or 'MANUAL' . The 'MANUAL' performance mode required the 'file-delay' and 'block-delay' parameters to be specified (ms.).

Automatic Report Generation (Optional)

The **'report'** element instructs FlexTk to automatically save results to a report file. The **'type'** attribute may be set to **'HTML'**, **'TEXT'** or **'CSV'**. An optional **'title'** element may be used to set a custom report title. If the **'save-to'** element is not specified, FlexTk will automatically generate a file name and save the report into the user's home directory. If the specified **'save-to'** value represents an existing directory, FlexTk will automatically generate a file name and save report into the specified directory. In all other cases, FlexTk will save the report using the **'save-to'** value as an absolute file name.

4.3 Storage Utilization Analysis

Overtime, disks on home computers and enterprise storage systems collect outdated files wasting storage space and making it difficult to find the required information. The FlexTk file management toolkit provides advanced storage utilization analysis capabilities allowing one to analyze one or more local disks or network shares, identify abnormal storage utilization patterns and automatically execute (FlexTk Ultimate) user-defined actions.

In addition to the ability to define storage analysis commands using the FlexTk GUI application, FlexTk allows one to specify storage analysis commands using an XML-Based format. The XML format provides support for all the storage analysis features and capabilities available in the FlexTk main GUI application. Storage analysis commands defined in the XML format may be imported into the FlexTk product configuration as user-defined commands or executed on-the-fly using the FlexTk command line tool (FlexTk Ultimate).

Format:

```

<analyze name='Command Name'>
  <inputs>
    <dir>Directory Name 1</dir>
    ...
    <file>File Name 1</file>
    ...
  </inputs>

  <actions mode='Actions Mode'> (Optional)
    <action>
      <for mode='M' op='O' value='V' units='U'>Condition</for>
      <do mode='M'>Action</do>
    </action>
    ...
  </actions>

  <filter>File Scanning Filter</filter> (Optional)
  <scanning>File Scanning Mode</scanning> (Optional)
  <shortcut>Keyboard Shortcut</shortcut> (Optional)

  <rules> (Optional)
    <r1>Rule Data</r1>
    ...
  </rules>

  <process> (Optional)
    <streams>Number of Process Streams</streams> (Optional)
    <speed file-delay='X' block-delay='Y'>Mode</speed> (Optional)
  </process>

  <report type='Report Type'> (Optional)
    <title>Custom Report Title</title> (Optional)
    <save-to>Report File Name</save-to> (Optional)
    <export-levels> Levels To Export</export-levels> (Optional)
    <items-per-level>Items Per Level</items-per-level> (Optional)
  </report>

  <exclude> (Optional)
    <dir>Exclude Directory 1</dir>
    ...
  </exclude>
</analyze>

```

Command Inputs

The inputs list should specify all the directories and files that should be processed. Each directory or file in the input list should include a full path name. Depending on the file scanning mode, all the specified directories will be processed sequentially (one after one) or in parallel.

Special Input Modes

In addition to the ability to process explicitly specified inputs, the FlexTk GUI application provides two special input modes allowing one to process external parameters or the current directory instead of the inputs list.

User-defined commands with the inputs mode set to the external-parameters mode are accepting drag-and-drop operations into the command item in the commands tool pane. In order to set a user-defined command to the external-parameters mode, specify `<inputs>USER-PARAMS</inputs>` instead of the regular input list.

User-defined commands with the input mode set to the current-directory mode are processing the file navigator's current directory. In order to set a user-defined command to the current-directory inputs mode, specify `<inputs>CURRENT-DIR</inputs>` instead of the input list.

File Matching Rules (Optional)

FlexTk Ultimate provides the user with the ability to limit the storage analysis process to files matching user-defined criteria. An optional list of file matching rules may specify one or more rules that should be used during the storage analysis process. All the specified file matching rules will be evaluated in the same order as they are specified in the user-defined command. FlexTk supports a large number of various types of rules capable of matching files by the file extension, name, path, size, time, content, etc. For detailed information about all the supported types of file matching rules refer to the section 1 in this document.

Miscellaneous Parameters (Optional)

filter	The ' filter ' element allows one to specify a file name filter instead of specifying file matching rules. For example, set the file name filter to '*.pdf' to process all PDF documents.
scanning	FlexTk is capable of scanning multiple disks or directories in parallel thus improving the speed of the storage analysis process on multi-CPU/multi-core systems. The ' scanning ' element allows one to control the file scanning mode. Set the scanning mode to ' PARALLEL ' if you wish to scan all input disks and directories simultaneously.
shortcut	The FlexTk GUI application provides the user with the ability to execute user-defined commands using keyboard shortcuts. Use the ' shortcut ' element to specify a keyboard shortcut for the command.

Process Options (Optional)

Parameter	Description
streams	FlexTk is optimized for multi-core/multi-CPU computers and advanced RAID storage systems and capable of scanning multiple file systems in parallel. In order to speed up the storage analysis process, use multiple processing streams when scanning input directories located on multiple physical hard disks or a RAID disk array.
speed	In order to minimize potential performance impact on running production systems, FlexTk allows one to intentionally slow down the storage analysis process. According to your specific needs, set the process speed mode to ' FULL ', ' MEDIUM ', ' LOW ' or ' MANUAL '. The ' MANUAL ' performance mode required the ' file-delay ' and ' block-delay ' parameters to be specified (ms.).

Automatic Report Generation (Optional)

The '**report**' element instructs FlexTk to automatically save results to a report file. The '**type**' attribute may be set to '**HTML**', '**TEXT**' or '**CSV**'. An optional '**title**' element may be used to set a custom report title. If the '**save-to**' element is not specified, FlexTk will automatically generate a file name and save the report into the user's home directory. If the specified '**save-to**' value represents an existing directory, FlexTk will automatically generate a file name and save report into the specified directory. In all other cases, FlexTk will save the report using the '**save-to**' value as an absolute file name.

Storage Analysis Actions (Optional)

FlexTk Ultimate allows one to define one or more conditional actions capable of detecting user-specified storage utilization patterns, executing automatic actions and sending e-mail notifications. The **'mode'** attribute may be set to **'PREVIEW'** or **'EXECUTE'**. In the **'PREVIEW'** mode, the command will display the storage analysis results and a list of actions that were triggered. In the FlexTk GUI application, the user will be provided with the ability to manually select actions that should be executed. In the **'EXECUTE'** mode, FlexTk will automatically execute all the triggered actions without showing storage analysis results.

The actions list may include one or more conditional actions specifying conditions to detect and actions to execute for each directory meeting the specified condition. The **'for'** element allows one to specify a condition that should be detected.

Use the **'mode'** attribute to set the condition mode to one of the following values:

Mode	Description
EID	Instructs to evaluate the condition for each input directory.
EDU	Instructs to evaluate the condition for each directory under the directory specified as the value in the 'for' element.
DIR	Instructs to evaluate the condition for the directory specified as the value in the 'for' element.

Use the **'op'** attribute to set the condition operator to one of the following values:

Operator	Description
EQ	Triggers action when the evaluated value equal to the condition value.
GT	Triggers action when the evaluated value greater than the condition value.
LT	Triggers action when the evaluated value less than the condition value.

Use the **'value'** attribute to set the condition value. The condition value, which should be specified as a decimal integer, will be evaluated according to the specified mode and operator.

Use the **'units'** attribute to set one of the following condition units.

Value	Description
FILES	The condition value specifies a number of files.
MBYTES	The condition value specifies an amount of used space in MBytes.
GBYTES	The condition value specifies an amount of used space in GBytes.
TBYTES	The condition value specifies an amount of used space in TBytes.
MSPACE	The condition value specifies an amount of free space in MBytes.
GSPACE	The condition value specifies an amount of free space in GBytes.
TSPACE	The condition value specifies an amount of free space in TBytes.
PSPACE	The condition value specifies an amount of free space in percents.

Use the **'do'** element to specify one of the following actions that should be executed:

Mode	Description
EMAIL-HTML	Send an HTML e-mail notification to the specified e-mail address.
EMAIL-TEXT	Send a text e-mail notification to the specified e-mail address.
USER-CMD	Execute the specified user-defined command.
CUSTOM-CMD	Execute the specified custom command.
REPORT-HTML	Save an HTML storage analysis report to the specified file.
REPORT-TEXT	Save a text storage analysis report to the specified file.
REPORT-CSV	Save a CSV storage analysis report to the specified file.

Depending on the specified action type, the **'do'** element's value will be used as an e-mail address, a command name or a report file name. For detailed information about all features related to storage analysis conditional actions refer to the FlexTk product manual.

4.4 Duplicate Files Search

FlexTk's built-in duplicate files finder provides a large number of advanced features and capabilities allowing one to identify and cleanup duplicate files on personal computers and enterprise storage systems. In addition, power users and IT professionals are provided with policy-based duplicate files detection and removal capabilities (FlexTk Ultimate only) allowing one to define custom duplicate files detection and removal commands and execute them in a fully unattended, automatic mode.

In addition to the ability to define duplicate files search commands using the FlexTk GUI application, FlexTk allows one to specify duplicate files search commands using an XML-Based format. The XML format provides support for all the duplicate files search features and capabilities available in the FlexTk main GUI application. Duplicate files search commands defined in the XML format may be imported into the FlexTk product configuration as user-defined commands or executed on-the-fly using the FlexTk command line tool (FlexTk Ultimate).

Format:

```

<duplicates name='Command Name'>
  <inputs>
    <dir>Directory Name 1</dir>
    ...
    <file>File Name 1</file>
    ...
  </inputs>
  <actions mode='Actions Mode'> (Optional)
    <action>Action 1</action>
    ...
    <action>Action X</action>
  </actions>
  <signature>Signature Type</signature> (Optional)
  <sets>Max Duplicate File Sets</sets> (Optional)
  <filter>File Scanning Filter</filter> (Optional)
  <scanning>File Scanning Mode</scanning> (Optional)
  <shortcut>Keyboard Shortcut</shortcut> (Optional)
  <rules> (Optional)
    <r1>Rule Data</r1>
    ...
  </rules>
  <process> (Optional)
    <streams>Number of Process Streams</streams> (Optional)
    <speed file-delay='X' block-delay='Y'>Mode</speed> (Optional)
  </process>
  <exclude> (Optional)
    <dir>Exclude Directory 1</dir>
    ...
  </exclude>
</duplicates>

```

Command Inputs

The inputs list should specify all the directories and files that should be processed. Each directory or file in the input list should include a full path name. Depending on the file scanning mode, all the specified directories will be processed sequentially (one after one) or in parallel.

Special Input Modes

In addition to the ability to process explicitly specified inputs, the FlexTk GUI application provides two special input modes allowing one to process external parameters or the current directory instead of the inputs list.

User-defined commands with the inputs mode set to the external-parameters mode are accepting drag-and-drop operations into the command item in the commands tool pane. In order to set a user-defined command to the external-parameters mode, specify `<inputs>USER-PARAMS</inputs>` instead of the regular input list.

User-defined commands with the input mode set to the current-directory mode are processing the file navigator's current directory. In order to set a user-defined command to the current-directory inputs mode, specify `<inputs>CURRENT-DIR</inputs>` instead of the input list.

File Matching Rules (Optional)

FlexTk Ultimate provides the user with the ability to limit the duplicates search process to files matching user-defined criteria. An optional list of file matching rules may specify one or more rules that should be used during the duplicates search process. All the specified file matching rules will be evaluated in the same order as they are specified in the user-defined command. FlexTk supports a large number of various types of rules capable of matching files by the file extension, name, path, size, time, content, etc. For detailed information about all the supported types of file matching rules refer to the section 1 in this document.

Miscellaneous Parameters (Optional)

signature	The ' signature ' element allows one to set one of the following duplicate files search algorithms: MD5, SHA1 or SHA256 (default). The SHA256 algorithm is the most reliable one and it is used by default. The MD5 and SHA1 algorithms are significantly faster, but less reliable.
sets	The ' sets ' element provides the user with the ability to control the maximum number of duplicate file sets (default is 1000) reported by FlexTk. After finishing the search process, FlexTk will sort all the detected duplicate file sets by the amount of the wasted storage space and display the top X duplicate file sets as specified by this parameter
filter	The ' filter ' element allows one to specify a file name filter instead of specifying file matching rules. For example, set the file name filter to '*.pdf' to process all PDF documents.
scanning	FlexTk is capable of scanning multiple disks or directories in parallel thus improving the speed of the duplicates search process on multi-CPU/multi-core systems. The ' scanning ' element allows one to control the file scanning mode. Set the scanning mode to ' PARALLEL ' if you wish to scan all input disks and directories simultaneously.
shortcut	The FlexTk GUI application provides the user with the ability to execute user-defined commands using keyboard shortcuts. Use the ' shortcut ' element to specify a keyboard shortcut for the command.

Process Options (Optional)

Parameter	Description
streams	FlexTk is optimized for multi-core/multi-CPU computers and advanced RAID storage systems and capable of scanning multiple file systems in parallel. In order to speed up the duplicates search process, use multiple processing streams when scanning input directories located on multiple physical hard disks or a RAID disk array.
speed	In order to minimize potential performance impact on running production systems, FlexTk allows one to intentionally slow down the duplicates search process. According to your specific needs, set the process speed mode to ' FULL ', ' MEDIUM ', ' LOW ' or ' MANUAL '. The ' MANUAL ' performance mode required the ' file-delay ' and ' block-delay ' parameters to be specified (ms.).

Duplicate Files Removal Actions (Optional)

FlexTk Ultimate provides the user with the ability to predefine one or more automatic duplicates detection and removal actions. Each action may contain a number of rules specifying how to detect the original file in each duplicate file set and what action to perform if the original was successfully detected.

Format:

```

<actions mode='Actions Mode'>                                     (Optional)
  <action>
    <original mode='original mode'>original value</original>
    <rules>                                                         (Optional)
      <r1>Rule Data</r1>
      ...
      <rX>Rule Data</rX>
    </rules>
    <do>Action</do>
  </action>
  ...
  <action>Action X</action>
</actions>

```

Use the **'mode'** attribute to set the actions mode to one of the following values: **'SELECT'** or **'EXECUTE'**. In the **'SELECT'** actions mode, FlexTk will evaluate duplicate files and try to detect the original file in each set of duplicate files according to the specified original detection rules and policies. Actions containing the original file detection rules will be evaluated one after one in the order as they specified in the actions list. If a duplicate file will match rules defined in an action, the duplicate file will be set as the original and the matching action will be set as the active action for the duplicate set.

The **'EXECUTE'** mode is the most advanced duplicates detection and removal move, which is intended for automatic execution of actions matching one or more duplicate file sets. During the search process, FlexTk will evaluate all duplicate files and set matching actions. After finishing the search process, FlexTk will automatically execute all actions that were set and remove all duplicate files using policies defined in each specific action.

The **'original'** element controls the original file detection mode. If the **'mode'** attribute is set to **'MTIME'**, the original file value should be set to **'OLDEST'** or **'NEWEST'** and in this case the original file will be set to the oldest or the newest file respectively. If the **'mode'** attribute is set to **'DIR'**, the original value should specify a directory where original files should be located in.

The **'rules'** element is an optional element, which may be used to limit the duplicates removal action to specific file types according to the specified file matching rules. For detailed information about file matching rules refer to the section 1 in this document.

Use the **'do'** element to specify the action that should be performed (or selected) for duplicate file sets with successfully detected original files. The **'do'** element may be specified using one of the following formats:

Action	Description
<do>LINK</do>	Replaces all duplicate files with links to originals.
<do>DELETE</do>	Deletes all duplicate files.
<do mode='MOVE-TO'>Directory</do>	Moves all duplicate files to the specified directory.

4.5 File Synchronization

The FlexTk file management toolkit provides advanced file synchronization capabilities allowing one to synchronize files between directories, local disks and network shares. The FlexTk's built-in file synchronization engine offers numerous one-way and two-way file synchronization modes, rule-based file matching capabilities and advanced performance tuning options. Frequently used file synchronization operations may be pre-configured as user-defined commands and executed in a single mouse click.

In addition to the ability to define file synchronization commands using the FlexTk GUI application, FlexTk allows one to specify file synchronization commands using an XML-Based format. The XML format provides support for all the file synchronization features and capabilities available in the FlexTk main GUI application. File synchronization commands defined in the XML format may be imported into the FlexTk product configuration as user-defined commands or executed on-the-fly using the FlexTk command line tool (FlexTk Ultimate).

Format:

```
<sync name='Command Name'>
  <source>Source Directory</source>
  <dest>Destination Directory</dest>

  <mode>Synchronization Mode</mode> (Optional)
  <recursive>Recursive Mode</recursive> (Optional)
  <preview>Preview Mode</preview> (Optional)
  <shortcut>Keyboard Shortcut</shortcut> (Optional)

  <rules> (Optional)
    <r1>Rule Data</r1>
    ...
  </rules>

  <process> (Optional)
    <streams>Number of Process Streams</streams> (Optional)
    <speed file-delay='X' block-delay='Y'>Mode</speed> (Optional)
    <copy>Copy Options</copy> (Optional)
  </process>

  <exclude> (Optional)
    <dir>Exclude Directory 1</dir>
    ...
  </exclude>
</sync>
```

Mandatory Parameters

Parameter	Description
source	Use the ' source ' element to specify the source directory for the file synchronization command.
dest	Use the ' dest ' element to specify the destination directory for the file synchronization command.

File Matching Rules (Optional)

FlexTk Ultimate provides the user with the ability to limit the file synchronization process to files matching user-defined criteria. An optional list of file matching rules may specify one or more rules that should be used during the file synchronization process. All the specified file matching rules will be evaluated in the same order as they are specified in the user-defined command. FlexTk supports a large number of various types of rules capable of matching files by the file extension, name, path, size, time, content, etc. For detailed information about all the supported types of file matching rules refer to the section 1 in this document.

Miscellaneous Parameters (Optional)

Parameter	Description
mode	<p>The 'mode' element allows one to set one of the following sync modes:</p> <ul style="list-style-type: none"> • SD (Synchronize Destination Directory, one-way, default mode) - All changes made in the source directory will be propagated to the destination directory. Files deleted from the source directory will be deleted from the destination directory. All files changed or deleted in the destination directory will be restored using files from the source directory. No changes will be made in the source directory. After the file synchronization process is complete both locations will be identical. • SUD (Update Destination Directory, one-way) - Newly created and modified source files will be copied to the destination directory. Files deleted from the source directory will be deleted from the destination directory. Files deleted from the destination directory will be restored using files from the source directory. Newly created and modified destination files will be kept in place. No changes will be made in the source directory. • SAD (Accumulate and Update Files in Destination, one-way) - Newly created and modified source files will be copied to the destination directory. Files deleted from the destination directory will be restored using files from the source directory. Newly created and modified destination files will be kept in place. No files will be deleted from the destination directory. No changes will be made in the source directory. • STW (Two-Way File Synchronization) - Newly created and modified files will be synchronized in both directions. Files deleted from the source directory will be deleted from the destination directory. Files deleted from the destination directory will be restored from the source directory. Changes may be made in both locations. After the file synchronization process is complete both locations will be identical. • STA (Two-Way Accumulation and Update) - Newly created and modified files will be updated in both directions. Files deleted from one location will be restored from the second location. Changes may be made in both locations. After the file synchronization process is complete both locations will be identical.
recursive	The 'recursive' element may be set to 'YES' or 'NO' . If the 'recursive' element is set to 'YES' (default), FlexTk will synchronize directories recursively including all subdirectories.
preview	The 'preview' element may be set to 'YES' or 'NO' . If the 'preview' element is set to 'YES' (default), FlexTk will show the file synchronization preview dialog allowing one to manually select or unselect files that should be synchronized.
shortcut	The FlexTk GUI application provides the user with the ability to execute user-defined commands using keyboard shortcuts. Use the 'shortcut' element to specify a keyboard shortcut for the command.

Process Options (Optional)

Parameter	Description
streams	FlexTk is optimized for multi-core/multi-CPU computers and advanced RAID storage systems and capable of scanning multiple file systems in parallel. In order to speed up the file synchronization process, use multiple processing streams when synchronizing directories located on multiple physical hard disks or a RAID disk array.
speed	In order to minimize potential performance impact on running production systems, FlexTk allows one to intentionally slow down the file synchronization process. According to your specific needs, set the process speed mode to 'FULL' , 'MEDIUM' , 'LOW' or 'MANUAL' . The 'MANUAL' performance mode required the 'file-delay' and 'block-delay' parameters to be specified (ms.).
copy	<p>The 'copy' element may specify one or more of the following copy flags:</p> <ul style="list-style-type: none"> • ATTR - Instructs to copy file attributes • DACL - Instructs to copy discrete access control lists (DACL) • SACL - Instructs to copy security control lists (SACL) • OWNER - Instructs to copy user/group ownership information

4.6 File Organizing

FlexTk's file organizing capabilities allow one to perform various file management operations on files matching user-defined criteria. The user is provided with the ability to specify one or more file matching rules and define what action should be performed on all the matching files. Multiple file organizing operations may be performed during a single pass through the entire file system. For example, the user may backup recently modified documents, archive unused music files and cleanup temporary files during a single pass through one or more file systems containing millions of files.

In addition to the ability to define file organizing commands using the FlexTk GUI application, FlexTk allows one to specify file organizing commands using an XML-Based format. The XML format provides support for all the file organizing features and capabilities available in the FlexTk main GUI application. File organizing commands defined in the XML format may be imported into the FlexTk product configuration as user-defined commands or executed on-the-fly using the FlexTk command line tool (FlexTk Ultimate).

Format:

```

<organize name='Command Name'>
  <inputs>
    <dir>Directory Name 1</dir>
    ...
    <file>File Name 1</file>
    ...
  </inputs>
  <action name='Action Name 1'>
    <rules>
      <r1>Rule Data</r1>
      ...
      <rX>Rule Data</rX>
    </rules>
    <do>Action</do>
    <process> (Optional)
      <copy>Copy Options</copy>
      <verify>Verification Mode</verify>
    </process>
  </action>
  ...
  <action name='Action Name X'>Action Data</action> (Optional)
  <preview>Preview Mode</preview> (Optional)
  <shortcut>Keyboard Shortcut</shortcut> (Optional)
  <process> (Optional)
    <streams>Number of Process Streams</streams> (Optional)
    <speed file-delay='X' block-delay='Y'>Mode</speed> (Optional)
    <progress>Progress Mode</progress> (Optional)
    <retry-count>X</retry-count> (Optional)
    <retry-delay>X</retry-delay> (Optional)
  </process>
</organize>

```

Command Inputs

The inputs list should specify all the directories and files that should be processed. Each directory or file in the input list should include a full path name. Depending on the file scanning mode, all the specified directories will be processed sequentially (one after one) or in parallel.

Special Input Modes

In addition to the ability to process explicitly specified inputs, the FlexTk GUI application provides two special input modes allowing one to process external parameters or the current directory instead of the inputs list.

User-defined commands with the inputs mode set to the external-parameters mode are accepting drag-and-drop operations into the command item in the commands tool pane. In order to set a user-defined command to the external-parameters mode, specify `<inputs>USER-PARAMS</inputs>` instead of the regular input list.

User-defined commands with the input mode set to the current-directory mode are processing the file navigator's current directory. In order to set a user-defined command to the current-directory inputs mode, specify `<inputs>CURRENT-DIR</inputs>` instead of the input list.

File Matching Rules

Each file organizing action should include a list of file matching rules specifying files that should be processed. All the specified file matching rules will be evaluated in the same order as they are specified in the user-defined command. FlexTk supports a large number of various types of rules capable of matching files by the file extension, name, path, size, time, content, etc. For detailed information about all the supported types of file matching rules refer to the section 1 in this document.

Actions

Use the `'do'` element to specify the action that should be performed for all files matching the specified file matching rules. The `'do'` element may be specified using one of the following formats:

Action	Description
<code><do mode='COPY-TO'>Directory</do></code>	Copies files matching the specified file matching rules to the specified directory.
<code><do mode='MOVE-TO'>Directory</do></code>	Moves files matching the specified file matching rules to the specified directory.
<code><do mode='MOVE-AND-LINK'>Dir</do></code>	Moves files matching the specified file matching rules to the specified directory and creates links pointing to the moved files in the destination directory.
<code><do>DELETE</do></code>	Deletes all files matching the specified file matching rules.

Global Process Options (Optional)

Parameter	Description
streams	FlexTk is optimized for multi-core/multi-CPU computers and advanced RAID storage systems and capable of scanning multiple file systems in parallel. In order to speed up the file organizing process, use multiple processing streams when processing input directories located on multiple physical hard disks or a RAID disk array.
speed	In order to minimize potential performance impact on running production systems, FlexTk allows one to intentionally slow down the file organizing process. According to your specific needs, set the process speed mode to ' FULL ', ' MEDIUM ', ' LOW ' or ' MANUAL '. The ' MANUAL ' performance mode required the ' file-delay ' and ' block-delay ' parameters to be specified (ms.).
progress	The ' progress ' element may be set to ' YES ' or ' NO '. If the ' progress ' element is set to ' YES ', FlexTk will display a progress bar on the operation process dialog. In order to optimize the file organizing performance when processing large amounts of files, it may be significantly more efficient to disable the progress bar.
retry-count	The ' retry-count ' element allows one to specify the number of retries should be made on locked source files for copy/move operations.
retry-delay	The ' retry-delay ' element allows one to specify a retry delay that should be made between retries to copy/move locked source files.

Action Process Options (Optional)

Parameter	Description
copy	The ' copy ' element may specify one or more of the following copy flags: <ul style="list-style-type: none"> • ATTR – Instructs to copy file attributes • DACL – Instructs to copy discrete access control lists (DACL) • SACL – Instructs to copy security control lists (SACL) • OWNER – Instructs to copy user/group ownership information • TIME – Instructs to copy file timestamps
verify	The ' verify ' element may be set to ' YES ' or ' NO ' (default). If the ' verify ' element is set to ' YES ', FlexTk will verify each copied or moved file.

Miscellaneous Parameters (Optional)

Parameter	Description
preview	The ' preview ' element may be set to ' YES ' or ' NO '. If the ' preview ' element is set to ' YES ' (default), FlexTk will show the file organizing preview dialog allowing one to manually select or unselect actions that should be performed.
shortcut	The FlexTk GUI application provides the user with the ability to execute user-defined commands using keyboard shortcuts. Use the ' shortcut ' element to specify a keyboard shortcut for the command.

4.7 File Copy

The FlexTk file management toolkit provides a large number of advanced multi-stream file copy and rule-based data migration capabilities allowing one to copy vast amounts of files fast, efficiently and reliably. FlexTk allows one to copy files including access control lists (ACLs), security attributes, ownership information, timestamps and file attributes. The user is provided with a large number of flexible copy performance tuning and fault tolerance options allowing one to copy data at various speeds, recover failed copy operations and minimize the potential impact on running production systems.

In addition to the ability to define file copy commands using the FlexTk GUI application, FlexTk allows one to specify file copy commands using an XML-Based format. The XML format provides support for all the file copy features and capabilities available in the FlexTk main GUI application. File copy commands defined in the XML format may be imported into the FlexTk product configuration as user-defined commands or executed on-the-fly using the FlexTk command line tool (FlexTk Ultimate).

Format:

```
<copy name='Command Name'>
  <inputs>
    <dir>Directory Name 1</dir>
    ...
    <file>File Name 1</file>
    ...
  </inputs>

  <dest>Destination Directory</dest>
  <shortcut>Keyboard Shortcut</shortcut> (Optional)

  <rules> (Optional)
    <r1>Rule Data</r1>
    ...
  </rules>

  <process> (Optional)
    <progress>Progress Mode</progress> (Optional)
    <overwrite>Overwrite Mode</overwrite> (Optional)
    <streams>Number of Process Streams</streams> (Optional)
    <speed file-delay='X' block-delay='Y'>Mode</speed> (Optional)
    <copy>Copy Options</copy> (Optional)
    <retry-count>1</retry-count> (Optional)
    <retry-timeout>62</retry-timeout> (Optional)
    <verify>Verification Mode</verify> (Optional)
  </process>
</copy>
```

A file move command may be defined by using the **'move'** command type instead of the **'copy'** command type. All other parameters and options are completely identical.

```
<move name='Command Name'>
  <!-- Command Data -->
</move>
```

Mandatory Parameters

Parameter	Description
dest	Use the 'dest' element to specify a destination directory for the copy command.

Command Inputs

The inputs list should specify all the directories and files that should be processed. Each directory or file in the input list should include a full path name. Depending on the file scanning mode, all the specified directories will be processed sequentially (one after one) or in parallel.

Special Input Modes

In addition to the ability to process explicitly specified inputs, the FlexTk GUI application provides two special input modes allowing one to process external parameters or the current directory instead of the inputs list.

User-defined commands with the inputs mode set to the external-parameters mode are accepting drag-and-drop operations into the command item in the commands tool pane. In order to set a user-defined command to the external-parameters mode, specify `<inputs>USER-PARAMS</inputs>` instead of the regular input list.

User-defined commands with the input mode set to the current-directory mode are processing the file navigator's current directory. In order to set a user-defined command to the current-directory inputs mode, specify `<inputs>CURRENT-DIR</inputs>` instead of the input list.

File Matching Rules (Optional)

FlexTk Ultimate provides the user with the ability to limit the file copy process to files matching user-defined criteria. An optional list of file matching rules may specify one or more rules that should be used during the file copy process. All the specified file matching rules will be evaluated in the same order as they are specified in the user-defined command. FlexTk supports a large number of various types of rules capable of matching files by the file extension, name, path, size, time, content, etc. For detailed information about all the supported types of file matching rules refer to the section 1 in this document.

Overwrite Mode (Optional)

Mode	Description
ALLWAYS	Always overwrites files already existing in the destination directory.
SKIP	Skips files already existing in the destination directory.
GEN-NAME	Appends a unique file suffix and copies to the destination directory.
IF-NEWER	Overwrites if the source file is newer than the destination file.

Process Options (Optional)

Parameter	Description
streams	FlexTk is optimized for multi-core/multi-CPU computers and advanced RAID storage systems and capable of scanning multiple file systems in parallel. In order to speed up the copy process, use multiple processing streams when processing input directories located on multiple physical hard disks or a RAID disk array.
speed	In order to minimize potential performance impact on running production systems, FlexTk allows one to intentionally slow down the copy process. According to your specific needs, set the process speed mode to 'FULL' , 'MEDIUM' , 'LOW' or 'MANUAL' . The 'MANUAL' performance mode required the 'file-delay' and 'block-delay' parameters to be specified (ms.).
progress	The 'progress' element may be set to 'YES' or 'NO' . If the 'progress' element is set to 'YES' , FlexTk will display a progress bar on the operation process dialog. In order to optimize the file copy performance when processing large amounts of files, it may be significantly more efficient to disable the progress bar.
retry-count	The 'retry-count' element allows one to specify the number of retries should be made on locked source files for copy/move operations.
retry-delay	The 'retry-delay' element allows one to specify a retry delay that should be made between retries to copy/move locked source files.
copy	The 'copy' element may specify one or more of the following copy flags: <ul style="list-style-type: none"> • ATTR – Instructs to copy file attributes • DACL – Instructs to copy discrete access control lists (DACL) • SACL – Instructs to copy security control lists (SACL) • OWNER – Instructs to copy user/group ownership information • TIME – Instructs to copy file timestamps
verify	The 'verify' element may be set to 'YES' or 'NO' (default). If the 'verify' element is set to 'YES' , FlexTk will verify each copied or moved file.

4.8 File Delete

The FlexTk file management toolkit provides the user with a large number of flexible file delete capabilities allowing one to delete huge amounts of files very fast and effectively. In addition, power computer users and IT professionals are provided with the ability to optimize performance of delete operations for user-specific storage configurations and performance requirements. Finally, users required to physically destroy confidential or classified information are provided with numerous multi-pass data wiping capabilities ranging from a simple single-pass wipe to a powerful, 7-Pass DOD 5220.22-M compliant (FlexTk Ultimate) data wiping algorithm.

In addition to the ability to define file delete commands using the FlexTk GUI application, FlexTk allows one to specify file delete commands using an XML-Based format. The XML format provides support for all the file delete features and capabilities available in the FlexTk main GUI application. File delete commands defined in the XML format may be imported into the FlexTk product configuration as user-defined commands or executed on-the-fly using the FlexTk command line tool (FlexTk Ultimate).

Format:

```
<delete name='Command Name'>
  <inputs>
    <dir>Directory Name 1</dir>
    ...
    <dir>Directory Name X</dir>
    <file>File Name 1</file>
    ...
    <file>File Name X</file>
  </inputs>
  <rules> (Optional)
    <r1>Rule Data</r1>
    ...
    <rX>Rule Data</rX>
  </rules>
  <process> (Optional)
    <streams>Number of Process Streams</streams> (Optional)
    <speed file-delay='X' block-delay='Y'>Mode</speed> (Optional)
    <progress>Progress Mode </progress> (Optional)
    <wipe>Wipe Mode</wipe>
  </process>
</delete>
```

Command Inputs

The inputs list should specify all the directories and files that should be processed. Each directory or file in the input list should include a full path name. Depending on the file scanning mode, all the specified directories will be processed sequentially (one after one) or in parallel.

Special Input Modes

In addition to the ability to process explicitly specified inputs, the FlexTk GUI application provides two special input modes allowing one to process external parameters or the current directory instead of the inputs list.

User-defined commands with the inputs mode set to the external-parameters mode are accepting drag-and-drop operations into the command item in the commands tool pane. In order to set a user-defined command to the external-parameters mode, specify `<inputs>USER-PARAMS</inputs>` instead of the regular input list.

User-defined commands with the input mode set to the current-directory mode are processing the file navigator's current directory. In order to set a user-defined command to the current-directory inputs mode, specify `<inputs>CURRENT-DIR</inputs>` instead of the input list.

File Matching Rules (Optional)

FlexTk Ultimate provides the user with the ability to limit the file delete process to files matching user-defined criteria. An optional list of file matching rules may specify one or more rules that should be used during the file delete process. All the specified file matching rules will be evaluated in the same order as they are specified in the user-defined command. FlexTk supports a large number of various types of rules capable of matching files by the file extension, name, path, size, time, content, etc. For detailed information about all the supported types of file matching rules refer to the section 1 in this document.

Process Options (Optional)

Parameter	Description
streams	FlexTk is optimized for multi-core/multi-CPU computers and advanced RAID storage systems and capable of scanning multiple file systems in parallel. In order to speed up the delete process, use multiple processing streams when processing input directories located on multiple physical hard disks or a RAID disk array.
speed	In order to minimize potential performance impact on running production systems, FlexTk allows one to intentionally slow down the delete process. According to your specific needs, set the process speed mode to 'FULL' , 'MEDIUM' , 'LOW' or 'MANUAL' . The 'MANUAL' performance mode required the 'file-delay' and 'block-delay' parameters to be specified (ms.).
progress	The 'progress' element may be set to 'YES' or 'NO' . If the 'progress' element is set to 'YES' , FlexTk will display a progress bar on the operation process dialog. In order to optimize the file delete performance when processing large amounts of files, it may be significantly more efficient to disable the progress bar.
wipe	The 'wipe' element may be set to one of the following values: ZERO-1 – wipes data using a single-pass, zero pattern algorithm ALT-2 – wipes data using a two-pass, alternating-pattern algorithm ALT-3 – wipes data using a three-pass, alternating-pattern algorithm ALT-4 – wipes data using a four-pass, alternating-pattern algorithm ALT-5 – wipes data using a five-pass, alternating algorithm DOD-7 – wipes data using a seven-pass, DOD 5220.22-M compliant algorithm BS-7 – wipes data using a seven-pass, Bruce Schneier's algorithm GMS-35 – wipes data using a 35-pass, Gutmann's Maximum Security algorithm

4.9 Disk Monitoring

FlexTk Ultimate includes a built-in disk monitor capable of detecting file creations, modifications, attribute changes and deletions in real-time. The real-time disk monitor provides the user with the ability to monitor one or more disks, directories or network shares and detect all changes made in the monitored file systems. In addition, the user is provided with the ability to specify conditional real-time actions capable of executing FlexTk's user-defined commands, custom commands, scripts or batch files.

In addition to the ability to define disk monitoring commands using the FlexTk GUI application, FlexTk allows one to specify disk monitoring commands using an XML-Based format. The XML format provides support for all the disk monitoring features and capabilities available in the FlexTk main GUI application. Disk monitoring commands defined in the XML format may be imported into the FlexTk product configuration as user-defined commands or executed on-the-fly using the FlexTk command line tool.

Format:

```
<monitor name='Command Name'>
  <inputs>
    <dir>Directory Name 1</dir>
    ...
  </inputs>
  <detect>Change Types</detect>
  <recursive>yes</recursive> (Optional)
  <shortcut>Keyboard Shortcut</shortcut> (Optional)
  <rules> (Optional)
    <r1>Rule Data</r1>
    ...
  </rules>
  <actions changes='Change Count'> (Optional)
    <report type='Report Type'> (Optional)
      <title>Custom Report Title</title> (Optional)
      <save-to>Report File Name</save-to> (Optional)
    </report>
    <user-command>Command </user-command> (Optional)
    <custom-command>Command </custom-command> (Optional)
    <email>E-Mail Address</email> (Optional)
  </actions>
</monitor>
```

Mandatory Parameters:

Parameter	Description
detect	Use the ' detect ' element to specify changes that should be detected: <ul style="list-style-type: none"> FNAME – detect file name changes DNAME – detect directory name changes ATTR – detect file and directory attribute changes SIZE – detect file size changes ATIME – detect file access time changes CTIME – detect file creation time changes MTIME – detect file modification time changes SECURITY – detect file security attributes changes

Command Inputs

The inputs list should specify all the directories that should be monitored. Each directory in the input list should include a full path name.

Special Input Modes

In addition to the ability to process explicitly specified inputs, the FlexTk GUI application provides two special input modes allowing one to process external parameters or the current directory instead of the inputs list.

User-defined commands with the inputs mode set to the external-parameters mode are accepting drag-and-drop operations into the command item in the commands tool pane. In order to set a user-defined command to the external-parameters mode, specify `<inputs>USER-PARAMS</inputs>` instead of the regular input list.

User-defined commands with the input mode set to the current-directory mode are processing the file navigator's current directory. In order to set a user-defined command to the current-directory inputs mode, specify `<inputs>CURRENT-DIR</inputs>` instead of the input list.

Miscellaneous Parameters (Optional)

Parameter	Description
recursive	The <code>'recursive'</code> element may be set to <code>'YES'</code> or <code>'NO'</code> . If the <code>'recursive'</code> element is set to <code>'YES'</code> (default), FlexTk will monitor directories recursively including all subdirectories.
shortcut	The FlexTk GUI application provides the user with the ability to execute user-defined commands using keyboard shortcuts. Use the <code>'shortcut'</code> element to specify a keyboard shortcut for the command.

File Matching Rules (Optional)

FlexTk provides the user with the ability to limit the disk monitoring process to files matching user-defined criteria. An optional list of file matching rules may specify one or more rules that should be used during the disk monitoring process. All the specified file matching rules will be evaluated in the same order as they are specified in the user-defined command. FlexTk supports a large number of various types of rules capable of matching files by the file extension, name, path, size, time, content, etc. For detailed information about all the supported types of file matching rules refer to the section 1 in this document.

Actions (Optional)

One of the most powerful capabilities of the FlexTk's real-time disk monitor is the ability to perform user-defined actions each time the disk monitor reaches a user-specified number of changes. Specify one or more actions and set the `'changes'` attribute to the required number of changes. Each specified action may save a change report, execute a user-defined command and/or execute a custom command.

The `'report'` element instructs FlexTk to automatically save changes to a report file. The `'type'` attribute may be set to `'HTML'`, `'TEXT'` or `'CSV'`. An optional `'title'` element may be used to set a custom report title. If the `'save-to'` element is not specified, FlexTk will automatically generate a file name and save the report into the user's home directory. If the specified `'save-to'` value represents an existing directory, FlexTk will automatically generate a file name and save report into the specified directory. In all other cases, FlexTk will save the report using the `'save-to'` value as an absolute file name.

Use the `'user-command'` element to specify the name of a user-defined command that should be executed each time the monitoring command reaches the specified number of changes. Use the `'custom-command'` element to specify the name of an executable or a batch file that should be executed. The `'email'` element allows one to specify an E-Mail address to send notifications to when the monitoring operation will reach the specified number of changes.