BigFix Enterprise Suite (BES™)

Upload and Archive Manager

BigFix, Inc.

Emeryville, CA

Last Modified: June 2, 2009

Version 7.2

Contents

Introduction

Starting with version 4.1, BES introduced the ability to collect multiple files from BES Clients into an archive and move them through the relay system to the server. This allows the BES Administrator to automatically log data from specific managed computers.

To do this, a new component called the **Archive Manager** has been added to the BES Client which can collect files periodically or on command. It passes the resultant compressed tar-ball to the **Upload Manager** on the BES Client. The Upload Manager has an input directory which queues the files for uploading.

The Upload Manager performs one upload operation at a time, moving the data in manageable chunks to reduce network traffic. It sends these chunks to the nearest BES Relay or Server, where the **PostFile** program reassembles the chunks back into the original file.

PostFile then passes the file up the chain, to the next BES Relay or to its ultimate destination at the BES Server. It again uses the Upload Manager to slice the file into chunks and send them on to the next PostFile program in the hierarchy. Once the file finally arrives at the BES Server, it is saved in a special directory location based on the ID of the client computer.

Along the way, both the Upload Manager and the PostFile program can alter the chunk size or throttle the upload speed to smooth out network traffic.

NOTE: When it encounters an unregistered BES Client, the Upload Manager pauses. This can happen for a variety of reasons, including a downed network, a busy server or a disconnected client. As soon as the BES Client can register with the BES system again, it will restart the Upload Manager and continue where it left off.

Editing the Settings

A typical archive is a collection of logs and configuration files that are compiled regularly and posted to the server. There are many settings available to help you customize your logging needs.

To initialize the various archive settings, follow these steps:

1. Start up the BES Console.
2. Select the **Computers** tab.
3. From the filter/list, select the set of computers you wish to start archiving.
4. Select **Edit Computer Settings** from the **Edit** menu. Typically, you will have selected multiple computers, so you will see a tabbed dialog box.
5. Select the **Settings** tab.
6. Check the **Custom Setting** box.
7. Enter the **Names** and corresponding **Values** of the desired settings, as discussed in the rest of this document.

Creating a Custom Action

You can create custom actions that can post attributes about the BES Client to an archive file.

To create a custom action:

1. Start up the BES Console.
2. Select the **Computers** tab.
3. From the filter/list, select the set of computers you wish to target for the action.
4. Select **Take Custom Action** from the **Tools** menu.
5. Select the **Action Script** tab.
6. Enter the desired **Action Script** in the text box provided.

Archive Manager

Archive Manager Settings

_BESClient_ArchiveManager_OperatingMode

The OperatingMode dictates the style of archiving, allowing periodic or triggered archiving. The following modes are available:

0: Disable all archival operations (default value).

1: Automatic, with a period = BESClient_ArchiveManager_IntervalSeconds.

2: Enables the **archive now** action command.

To allow a custom action to post client attributes to an archive file, make sure the OperatingMode is set to 2.

The default value of 0 disables archiving.

_BESClient_ArchiveManager_FileSet-<tag>

This setting (actually a group of settings with optional tags) specifies the files to be archived. This technique lets you specify multiple named batches of files. Each setting starts with "_BESClient_ArchiveManager_FileSet-" and ends with a batch name (the <tag> part).

The value of each setting is a path on the client file system. It may be a single file, in which case that file is part of the archive; a single directory, in which case all files in the directory will be part of the archive; or a directory path ending with wild cards, in which case all files in the directory matching the wild cards will be part of the archive.

For instance, the setting _BESClient_ArchiveManager_FileSet-(log), representing all the log files in a temporary log folder, could have a value like c:\temp\log.

Everything after the dash (-) is used as the default prefix of the files as they are unpacked on the root server. So a file named x.log in the c:\temp\log folder would be unpacked as (Log)x.log.

_BESClient_ArchiveManager_SendAll

This setting allows you to send just the archives that have changed, avoiding redundant uploads. There are two possible values for this setting:

0: Only send files that have changed since the last archive operation (default).

1: Send all files, even if they have not changed.

The default value of 0 is recommended for most applications.

_BESClient_ArchiveManager_MaxArchiveSize

This setting limits the size (in bytes) of the uploaded archive. Since a typical archive may be composed of several files, the archive size corresponds to the sum of the file sizes.

If the limit is exceeded, an archive that contains only the index file will be created and uploaded by the Archive Manager. The index will contain the following header line:

```
MaxArchiveSize: Exceeded
```

The default value is 1000000 (one million bytes).

_BESClient_ArchiveManager_IntervalSeconds

When the OperatingMode is set to 1, this setting determines the interval at which the client will trigger an archive.

The default value is 86400 seconds (24 hours).

Archive Manager internal variables

__BESClient_ArchiveManager_LastArchive

The Archive Manager updates this setting whenever it posts an archive. The value of the setting is the secure hash algorithm (sha1) of the file that was posted.

__BESClient_ArchiveManager_LastIntervalNumber

The BES Client updates this setting whenever it posts an archive. It represents the interval number since 1970 that the archive was last collected. If the interval is a day long (the default), then the setting indicates the number of days since 1970 that it created the last archive. It is calculated such that when the interval number changes, it is time to create a new archive.

The value is also offset by a time corresponding to the computer id, to stagger the collecting of archives.

Archive Manager Index File Format

During the building of the archive, the Archive Manager creates an index containing metadata about the archive. This is a sample index from an archive with a single file:

```
MIME-Version: 1.0
Content-Type: multipart/x-directory2; boundary="==="
Unique-ID: 1077307147
Archive-Size: 105
SendAll: 0
Date: Wed, 17 Mar 2004 02:23:01 +0000
FileSet-(LOG): c:\temp\log\newfile.log

--===

URL: file:///c:/temp/log/newfile.log
NAME: (LOG)newfile.log
SIZE: 105
TYPE: FILE
HASH: 3a2952e0db8b1e31683f801c6384943aae7fb273
MODIFIED: Sun, 14 Mar 2004 18:32:58 +0000

--=====--
```

Upload Manager

The Upload Manager coordinates the sending of files in chunks to the Post File program. You can throttle the upload dataflow to conserve bandwidth.

Upload Manager Settings

__BESClient_UploadManager_Progress

This internal setting provides a value that tracks the progress of the upload. When an upload is underway, this setting takes on values of the form:

```
<sha1>: <position> of <size> bytes in <duration> seconds
```

For example:

```
51ee4cf2196c4cb73abc6c6698944cd321593007: 672 of 672 bytes
in 5 seconds
```

The default value is "No Progress".

_BESClient_UploadManager_BufferDirectory

The Archive Manager always sets the Upload Manager's input Buffer Directory to "__BESData/__Global/Upload". This directory is on the client computer, in the BES Client folder.

_BESRelay_UploadManager_BufferDirectory

Like the BES Client, the BES Relay also has an Upload Manager, and it also has a buffer directory, whose path is specified by this setting. The Upload Manager will upload the files in the sha1 subdirectory of the specified directory. It sorts the files by modification time and then, just like the BES Client, it uploads them in chunks to smooth out the bandwidth requirements.

_BESRelay_UploadManager_BufferDirectoryMaxSize

This setting denotes the maximum amount of space on disk that the server will be allowed to take from the client using the Upload Manager.

_BESRelay_Uploadmanager_BufferDirectoryMaxCount

This setting denotes the number of files that the buffer directory will be allowed to hold. Its default is 10,000.

_BESRelay_UploadManager_CompressedFileMaxSize

This setting denotes the amount of space of the largest compressed file the Upload Manager will be allowed to handle.

_BESRelay_UploadManager_ChunkSize

Uploads are done one chunk at a time. In case of a conflict between this computer and the upstream computer, the size of the chunk is set to the smaller of the two.

A value of 0 for the chunk size indicates that the upload should be done in a single chunk, and effectively disables restarts and throttling (this is NOT recommended). The local chunk size setting is specified in bytes.

The default value is 128K.

_BESRelay_UploadManager_ThrottleKBPS

After each chunk has been uploaded, the Upload Manager calculates an amount of time to sleep to maintain the throttle speed in kilobytes per second (ThrottleKBPS). This allows you to compensate for network bottlenecks. For instance, a BES Relay connected over a slow VPN to the server might have a low upload throttle rate to minimize the bandwidth on that network segment.

In case of a conflict between this computer and the upstream server (or relay), the throttle KBPS is set to the smaller of the two.

The default value is 0, which disables throttling.

_BESRelay_UploadManager_ParentURL

This setting specifies the location of the parent computer, which is either a BES Relay or a Server. The Upload Manager posts the file chunks to the PostFile program using a url such as:

http://host.domain.com:52311/cgi-bin/bfenterprise/PostFile.exe

This specifies an absolute address to the desired server. More typically, however, Upload Manager will point to a folder on the current parent relay:

/cgi-bin/bfenterprise/PostFile.exe

This is the default setting.

__BESRelay_UploadManager_Progress

The Upload Manager writes a status or error string into this setting. Like the BES Client progress setting, it takes on values of the form:

```
<sha1>: <position> of <size> bytes in <duration> seconds
```

The default value is "No Progress".

NOTE: There are two leading underscores on this setting, which prevents the BES Console Operator from affecting this internal value.

_BESRelay_UploadManager_CleanupHours

Sometimes archived files accumulate but don't get uploaded. This could happen with a network outage, a downed server or other communication problem. To avoid overloading the system, these old files are deleted or cleaned up. This setting determines how old a file may get before it will be deleted.

The default value is 72 hours (3 days).

PostFile

The PostFile program receives the chunks of files posted by the Upload Manager and appends them to its own copy of the file. The Upload Manager specifies the range of bytes being posted and the sha1 of the file, which is used as the filename. These parameters are appended to the URL as in the following example:

```
postfile.exe?sha1=51ee4cf2196c4cb73abc6c6698944cd321593007&ran
ge=1000,1999,20000
```

Here the sha1 value identifies the file, and the range in this case specifies the second 1,000 byte chunk of a 20,000 byte file.

When PostFile receives a chunk of the file it first checks to make sure it is the correct segment. If so, it appends the posted data to its local copy of the file. It returns the size of this file, as well as the current chunk size and throttle BPS settings.

PostFile has to deal with several BES Clients feeding into it at once. To balance that load, it adjusts the throttle rate. The effective throttling rate is determined by dividing the limiting PostFile rate by the number of concurrently uploading files.

For instance, if PostFile has a throttle setting of 100 KBPS and 50 clients are simultaneously uploading files, the throttle value returned to each client would be adjusted to 2 KBPS. By setting custom throttle values to specific BES Relays, you can efficiently deal with any bottlenecks in your network.

PostFile stores the partially uploaded files in the Upload Manager's buffer directory with an underscore in front of them (the Upload Manager does not upload files that begin with underscore). When PostFile receives the last chunk of the file, it calculates the sha1 of the file and checks that it matches the sha1 parameter in the URL. If so, it removes the leading underscore.

The Upload Manager can then upload the file to the next relay up the hierarchy (or any other server, if so specified).

PostFile determines whether or not the Upload Manager is running. If not, PostFile assumes that it has reached its root server destination. It renames the uploaded file, extracts the files from the archive and deposits them in a subfolder of the Upload Manager's buffer directory.

The program calculates the subfolder path using a modulus of the computer id. This has the effect of spreading out file directory accesses and preventing an overpopulation of any single directory.

For example, the path to file "log" from computer id 1076028615 would be converted to the path "BufferDir/sha1/**15**/1076028615/log" where 15 is the remainder modulo 100 (the lower two digits) of the id.

If the uploaded file is a valid BES archive and is successfully extracted, then the original uploaded file is deleted.

PostFile Settings

_BESRelay_PostFile_ChunkSize

_BESRelay_PostFile_ThrottleKBPS

PostFile uses these settings for the chunk size and throttle values for incoming data. These values can be adjusted for varying connection speeds or other network anomalies.

When PostFile communicates with the upload manager, it passes along these values. As mentioned before, if there is a conflict between any two computers over these settings, it favors the smaller value.

The default values are 128K for ChunkSize and 0 for ThrottleKBPS (disable throttling).

Examples

Example 1

In this example, we want to collect all the files in the c:\log folder and all the .ini files in the c:\myapp folder once an hour. Send up only the differences and don't send the archive if it exceeds 1,000,000 bytes in size. To set this up, create the following settings in the BES Console:

```
_BESClient_ArchiveManager_FileSet-(Log) = c:\log
_BESClient_ArchiveManager_FileSet-(Ini) = c:\myapp\*.ini
_BESClient_ArchiveManager_OperatingMode = 1
_BESClient_ArchiveManager_Interval_Seconds = 3600
_BESClient_ArchiveManager_SendAll = 0
_BESClient_ArchiveManager_MaxArchiveSize = 1000000
```

Example 2

In this example, we want the same set of files as above, but we also want to collect some useful attributes (retrieved properties) from the client computer. A custom action can generate these attributes and trigger an archive when it completes. It uses the same settings as above, but sets the operating mode to 2 to enable the **archive now** action command:

```
_BESClient_ArchiveManager_OperatingMode = 2
```

You may then create a custom action, specifying the attributes you want to collect. For instance, to append the operating system, computer name and DNS name to the log file, create a custom action like this:

```
appendfile {"System:" & name of operating system}
appendfile {"Computer:" & computer name}
appendfile {"DNS name:" & dns name}
delete "c:\log\properties.log"
copy __appendfile "c:\log\properties.log"
archive now
```

The **appendfile** command creates a temporary text file named **__appendfile** . Each time you invoke the command, it appends the text you specify to the end of this temporary file.

The **delete** and **copy** commands clear out the old log file (if any) and copy the __appendfile to the log. This has the effect of creating a new properties.log file.

The **archive now** command immediately creates an archive, as long as the OperatingMode is set to 2.

You can then target this action at any subset of BES Clients, using whatever scheduling you choose. Using variations on this scheme, you could perform a full archive once a week, in addition to nightly differences.