# Introduced in version 2.3.0

## Major code re-factoring

* The ChannelSession provides a mechanism for supporting non-standard extended data (a.k.a. STDERR data) in a similar manner as the "regular" data. Please read the relevant section in the main documentation page.
* The user can use a registered SessionDisconnectHandler in order be informed and also intervene in cases where the code decides to disconnect the session due to various protocol or configuration parameters violations.
* ScpFileOpener#getMatchingFilesToSend has been modified to accept a Path as the base directory and also return an Iterable<Path>.
* The SFTP command line client provides a kex command that displays the KEX parameters of the current sesssion - client/server proposals and what has been negotiated.
* The Session object provides a KexExtensionHandler for usage with [KEX extension negotiation](https://tools.wordtothewise.com/rfc/rfc8308)
* The SignalListener accepts a Channel argument indicating the channel instance through which the signal was received.
* When creating a client shell or command channel one can provide optional PTY and/or environment values in order to override the internal default ones.
  + In this context, the PtyCapableChannelSession#setEnv method has been modified to accept ANY object. When the environment values are sent to the server, the object's toString() will be used. Furthermore, if one provides a null value, the previous registered value (if any) is **removed**.
* The SftpFileSystemAccessor callbacks are now provided with the relevant Handle they are servicing (*Note:* in special cases a null value is provided to indicate invocation outside the scope of such a handle).
  + Closing of file channel/directory streams created by the accessor are also closed via callbacks to the same accessor
  + When closing a file channel that may have been potentially modified, the default implementation forces a synchronization of the data with the file-system. This behavior can be modified by setting the sftp-auto-fsync-on-close property to *false*.
* The ScpFileOpener methods are also invoked in order to close input/output streams created through it when they are no longer needed once data has been successfully copied.
* The CommandFactory and ShellFactory have been modified to accept the server's ChannelSession instance through which they are being invoked.
* The various implementations of public/private keys/pairs decoders/loaders are provided with a Map of any headers that may be available in the relevant data file.
* org.apache.sshd.agent.unix.AgentClient constructor expects a non-*null* FactoryManager instance which it then exposes via its getFactoryManager.
* SftpEventListener#removing/removed callbacks accept an isDirectory flag indicating the type of Path being removed - file or directory.

## Minor code helpers

* The Session object provides a isServerSession method that can be used to distinguish between client/server instances without having to resort to instanceof.
* When creating a CLI SSH client one can specify -o KexExtensionHandler=XXX option to initialize a client-side KexExtensionHandler using an FQCN. If default is specified as the option value, then the internal DefaultClientKexExtensionHandler is used.

## Behavioral changes and enhancements

* [SSHD-782](https://issues.apache.org/jira/browse/SSHD-882) - Added session level heartbeat mechanism via SSH\_MSG\_IGNORE or customized user provided code.

In order to support customized user code for this feature, the ReservedSessionMessagesHandler can be used to implement any kind of user-defined heartbeat. *Note:* if the user configured such a mechanism, then the sendReservedHeartbeat method **must** be implemented since the default throws UnsupportedOperationException which will cause the session to be terminated the 1st time the method is invoked.

* [SSHD-882](https://issues.apache.org/jira/browse/SSHD-882) - Provide hooks to allow users to register a consumer for STDERR data sent via the ChannelSession - especially for the SFTP subsystem.
* [SSHD-892](https://issues.apache.org/jira/browse/SSHD-882) - Inform user about possible session disconnect prior to disconnecting and allow intervention via SessionDisconnectHandler.
* [SSHD-893](https://issues.apache.org/jira/browse/SSHD-893) - Using Path(s) instead of String(s) as DirectoryScanner results
* [SSHD-895](https://issues.apache.org/jira/browse/SSHD-895) - Add support for RSA + SHA-256/512 signatures. **Note:** according to [RFC - 8332 - section 3.3](https://tools.ietf.org/html/rfc8332#section-3.3):

Implementation experience has shown that there are servers that apply authentication penalties to clients attempting public key algorithms that the SSH server does not support.

When authenticating with an RSA key against a server that does not implement the "server-sig-algs" extension, clients MAY default to an "ssh-rsa" signature to avoid authentication penalties. When the new rsa-sha2-\* algorithms have been sufficiently widely adopted to warrant disabling "ssh-rsa", clients MAY default to one of the new algorithms.

Therefore we do not include by default the "rsa-sha-\*" signature factories in the SshClient. They can be easily added by using the relevant BuiltinSignatures:

SshClient client = SshClient.setupDefaultClient();  
client.setSignatureFactories(  
 Arrays.asList(  
 /\* This is the full list in the recommended preference order,  
 \* but the initialization code can choose and/or re-order  
 \*/  
 BuiltinSignatures.nistp256,  
 BuiltinSignatures.nistp384,  
 BuiltinSignatures.nistp521,  
 BuiltinSignatures.ed25519,  
 BuiltinSignatures.rsaSHA512,  
 BuiltinSignatures.rsaSHA256, // should check if isSupported since not required by default for Java 8  
 BuiltinSignatures.rsa,  
 BuiltinSignatures.dsa));

* [SSHD-896](https://issues.apache.org/jira/browse/SSHD-896) - Added support for [KEX extension negotiation](https://tools.ietf.org/html/rfc8308)
* [SSHD-870](https://issues.apache.org/jira/browse/SSHD-896) - Added support for GPGv2 public keyring (Note: requires upgraded [Bouncycastle](https://mvnrepository.com/artifact/org.bouncycastle/bcpg-jdk15on/1.61) and [jpgpj](https://mvnrepository.com/artifact/org.c02e.jpgpj/jpgpj/0.6.1) versions).
* [SSHD-897](https://issues.apache.org/jira/browse/SSHD-897) - The default CLI code automatically tries to detect the PTY settings to use if opening a shell or command channel.
* [SSHD-901](https://issues.apache.org/jira/browse/SSHD-901) - Added capability to request a reply for the keepalive@... heartbeat request in order to avoid client-side session timeout due to no traffic from server.
* [SSHD-902](https://issues.apache.org/jira/browse/SSHD-902) - Shutdown output when receiving SSH\_MSG\_CHANNEL\_EOF message via port forwarding channel.
* [SSHD-903](https://issues.apache.org/jira/browse/SSHD-903) - Fixed the SFTP version negotiation behavior in case client proposed version is higher than server supported one.
* [SSHD-904](https://issues.apache.org/jira/browse/SSHD-904) - Add option to enable/disable 'fsync' on modified file contents via SFTP (default=enabled).
* [SSHD-905](https://issues.apache.org/jira/browse/SSHD-905) - Add option to enable/disable 'fsync' on modified file contents via SCP (default=enabled).
* [SSHD-907](https://issues.apache.org/jira/browse/SSHD-907) - StpEventListener invokes (new) exiting method to inform about SFTP subsystem exiting and therefore closing all currently tracked file/directory handles.
* [SSHD-909](https://issues.apache.org/jira/browse/SSHD-909) - SFTP versions extension handler ignores non-numerical versions when resolving the available ones.
* [SSHD-913](https://issues.apache.org/jira/browse/SSHD-913) - Provide channel session instance to command and/or shell factories creators
* [SSHD-912](https://issues.apache.org/jira/browse/SSHD-912) - Use separate locks for Future(s) and Session/Channel instances.
* [SSHD-916](https://issues.apache.org/jira/browse/SSHD-916) - Avoid locking the session lock when signalling client session authentication failure.
* [SSHD-917](https://issues.apache.org/jira/browse/SSHD-917) - Add support for SSH2 public key file format.
* [SSHD-921](https://issues.apache.org/jira/browse/SSHD-921) - Do not send session disconnect message due to timeout expiration if already done so.
* [SSHD-923](https://issues.apache.org/jira/browse/SSHD-923) - Added agent close detection mechanisms to avoid infinite waits on incoming messages.
* [SSHD-929](https://issues.apache.org/jira/browse/SSHD-929) - Provide file/directory flag indicator to SFTP event listener callback for removal.