# Introduced in version 2.4.0

## Major code re-factoring

* SftpSubSystemFactory,ScpCommandFactory and their respective Builder(s) as well as the AbstractGitCommandFactory use a Supplier<CloseableExecutorService> instead of an executor instance in order to allow users to provide a "fresh" instance every time a new command instance is initiated and protect their instance from shutdown when session is destroyed:

CloseableExecutorService mySpecialExecutor = ...;  
 SftpSubsystemFactory factory = new SftpSubsystemFactory.Builder()  
 .withExecutorServiceProvider(() -> ThreadUtils.noClose(mySpecialExecutor))  
 .build();  
 server.setSubsystemFactories(Collections.singletonList(factory));

* SubsystemFactory is a proper interface and it has been refactored to contain a createSubsystem method that accepts the ChannelSession through which the request has been made
* AbstractSftpSubsystemHelper#resolvePathResolutionFollowLinks is consulted wherever the standard does not specifically specify the behavior regarding symbolic links handling.
* UserAuthFactory is a proper interface and it has been refactored to contain a createUserAuth method that accepts the session instance through which the request is made.
* ChannelFactory is a proper interface and it has been refactored to contain a createChannel method that accepts the session instance through which the request is made.
* KeyExchangeFactory is a proper interface and it has been refactored to contain a createKeyExchange method that accepts the session instance through which the request is made.
* Signature methods accept a SessionContext argument representing the session context of their invocation (if any).
* Default MAC(s) list is set according to the [ssh\_config(5)](https://www.freebsd.org/cgi/man.cgi?query=ssh_config&sektion=5) order as **first** ones, where the supported MAC(s) that do no appear in it come last.
* PasswordAuthenticator has a handleClientPasswordChangeRequest method that is invoked if a password change has been indicated by the user during authentication via the "password" method - by default throws UnsupportedOperationException.

## Minor code helpers

* SessionListener supports sessionPeerIdentificationReceived method that is invoked once successful peer version data is received.
* SessionListener supports sessionEstablished method that is invoked when initial constructor is executed.
* ChannelIdTrackingUnknownChannelReferenceHandler extends the functionality of the DefaultUnknownChannelReferenceHandler by tracking the initialized channels identifiers and being lenient only if command is received for a channel that was initialized in the past.
* The internal moduli used in Diffie-Hellman group exchange are **cached** - lazy-loaded the 1st time such an exchange occurs. The cache can be invalidated (and thus force a re-load) by invoking Moduli#clearInternalModuliCache.
* DHGEXClient implementation allows overriding the min./max. key sizes for a specific session Diffi-Helman group exchange via properties - see DHGEXClient#PROP\_DHGEX\_CLIENT\_MIN/MAX/PRF\_KEY. Similar applies for DHGEXServer but only for the message type=30 (old request).
* AbstractSignature#doInitSignature is now provided also with the Key instance for which it is invoked.
* The MacInformation interface has an extra isEncryptThenMac method (default=*false*) to enable distinction of this mode.
* Provide configurable control over the client-side ChannelSession *stdin* pump chunk size.
* Client side UserAuthKeyboardInteractive allows configurable detection of plain-text password prompt.
* Moved a few informative getters from Session to SessionContext.

## Behavioral changes and enhancements

* [SSHD-926](https://issues.apache.org/jira/browse/SSHD-930) - Add support for OpenSSH 'lsetstat@openssh.com' SFTP protocol extension.
* [SSHD-930](https://issues.apache.org/jira/browse/SSHD-930) - Added configuration allowing the user to specify whether client should wait for the server's identification before sending its own.
* [SSHD-931](https://issues.apache.org/jira/browse/SSHD-931) - Using an executor supplier instead of a specific instance in SftpSubsystemFactory and ScpCommandFactory.
* [SSHD-934](https://issues.apache.org/jira/browse/SSHD-934) - Fixed ECDSA public key encoding into OpenSSH format.
* [SSHD-937](https://issues.apache.org/jira/browse/SSHD-937) - Provide session instance when creating a subsystem, user authentication, channel.
* [SSHD-941](https://issues.apache.org/jira/browse/SSHD-941) - Allow user to override min./max. key sizes for a specific session Diffi-Helman group exchange via properties.
* [SSHD-943](https://issues.apache.org/jira/browse/SSHD-943) - Provide session instance when KEX factory is invoked in order to create a KeyExchange instance.
* [SSHD-945](https://issues.apache.org/jira/browse/SSHD-945) - Added sshd-contrib code that uses SHA1 with DSA regardless of its key length.
* [SSHD-946](https://issues.apache.org/jira/browse/SSHD-946) - Supporting 'encrypt-then-MAC' mode.
* [SSHD-947](https://issues.apache.org/jira/browse/SSHD-947) - Added configuration allowing the user to specify whether client should wait for the server's identification before sending KEX-INIT message.
* [SSHD-948](https://issues.apache.org/jira/browse/SSHD-948) - Do not accept password authentication if the session is not encrypted.
* [SSHD-949](https://issues.apache.org/jira/browse/SSHD-949) - Session should use cipher block size and not IV size to calculate padding.
* [SSHD-953](https://issues.apache.org/jira/browse/SSHD-953) - Parse and strip quoted command arguments when executing a server-side command via local shell.
* [SSHD-955](https://issues.apache.org/jira/browse/SSHD-955) - Provide configurable control over auto-detected password prompt in client-side UserAuthKeyboardInteractive implementation.
* [SSHD-956](https://issues.apache.org/jira/browse/SSHD-956) - Using System#nanoTime to measure session idle/authentication timeouts