# Introduced in version 2.2.0

## Major code re-factoring

* AttributeStore "read" methods moved to (new class) AttributeRepository.
  + AttributeKey moved to AttributeRepository.
  + getAttribute and resolveAttribute moved to AttributeRepository.
  + Added attributeKeys enumeration method to AttributeRepository.
  + In this context, please note that all associated session/channel/client/server attributes are cleared when entity is closed. The clearing usually occurs as after all relevant registered event listeners have been invoked.
* DEFAULT\_PORT moved from SshConfigFileReader to SshConstants.
* Moved some session "summary" related definitions from Session to SessionContext (which Session extends).
* Added new sessionDisconnect method to SessionListener.
* ReservedSessionMessagesHandler#handleUnimplementedMessage has an extra cmd argument and is called both for SSH\_MSG\_UNIMPLEMENTED as well as for any other unexpected/unrecognized command encountered during the session message processing loop.
* AttributeRepository optional context propagated during initial connection establishment
  + ClientSessionCreator has extra connect methods with an AttributeRepository connection context argument
  + The context is also propagated to HostConfigEntryResolver#resolveEffectiveHost method
  + connectionEstablished and abortEstablishedConnection methods of IoServiceEventListener accept also an AttributeRepository connection context argument (propagated from the ClientSessionCreator#connect invocation).
* FilePasswordProvider
  + Added an extra method (handleDecodeAttemptResult) that enables users to try and repeat an encrypted private key decoding using a different password.
  + The interface methods are also provided with a retry index that indicates the number of times they have been re-invoked for the same resource (including on success).
  + The available session context (if any) is also provided as an argument to the interface methods.
  + The interface methods use a NamedResource as the resource key instead of a plain string.
* SshAgent#getIdentities returns an Iterable rather than a List
* SftpFileSystemProvider and its associated helper classes have been moved to org.apache.sshd.client.subsystem.sftp.fs package.
* KeyPairProvider accepts a SessionContext argument in its getKeyTypes/loadKey methods.
* KeyIdentityProvider accepts a SessionContext argument in its loadKeys method.
* ClientIdentityProvider accepts a SessionContext argument in its getClientIdentity method.
* ClientIdentityLoader
  + Accepts a SessionContext argument in its loadClientIdentity method.
  + Uses a NamedResource as the identity location indicator instead of a plain old string.
* ApacheSshdSftpSessionFactory#get/setPrivateKey has been renamed to get/setPrivateKeyLocation.
* SshClient and ClientSession use a KeyIdentityProvider instead of a full blown KeyPairProvider. KeyPairProvider is used only in the context of an SshServer and/or ServerSession.
* SshClient#loadClientIdentities has been renamed to preloadClientIdentities + it returns a KeyIdentityProvider instead of a collection of strings representing paths.
* The various ClientIdentitiesWatcher(s) use a type-safe ClientIdentityLoaderHolder and FilePasswordProviderHolder instead of the generic Supplier definition.
* Removed API(s) that used string file paths to create FileInputStream-s - using only java.nio.file.Path-s
* Converted most of the key-pair identity loaders (e.g., ClientIdentityLoader, ClientIdentityProvider, etc.) to return an Iterable<KeyPair> instead of single KeyPair instance.
* Code that converts authorized keys entries into PublicKey-s has been renamed to resolvePublicKeyEntries and moved to PublicKeyEntry class.
  + Note that the parameters **order** has also been modified
* PublicKeyEntryResolver (and its derived classes) accept an extra SessionContext parameter.
* All methods ScpTransferEventListener accept an extra Session parameter indicating the SSH client/server session context for the listener's invocation.
  + Same applies for ScpFileOpener and ScpReceiveLineHandler
* Re-provide expected SCP byte count transfer and permissions when invoking ScpFileOpener#openRead/openWrite
* CipherInformation#getBlockSize has been renamed to getKdfSize in order to emphasize that its value represents the number of bytes used to derive the cipher's secret key value and not the cipher's underlying block size.
  + See [SSHD-873](https://issues.apache.org/jira/browse/SSHD-873) enhancement remark below for the cipher block size information.
* IdentityResourceLoader exposes getSupportedKeyTypes() instead of getSupportedTypeNames()
  + This method is now also implemented by the Identity interface as well
* HostConfigEntryResolver#resolveEffectiveHost accepts also an (optional) initial connection context and/or local peer binding address - propagated from the ClientSessionCreator#connect invocation.
* connectionAccepted/abortAcceptedConnection methods of IoServiceEventListener accept an extra argument - the service listen endpoint through which the connection was accepted.

## Behavioral changes and enhancements

* [SSHD-708](https://issues.apache.org/jira/browse/SSHD-708) - Add support for password encrypted OpenSSH private key files.
* [SSHD-757](https://issues.apache.org/jira/browse/SSHD-757) - Added hooks and some initial code to allow (limited) usage of [OpenPGP](https://www.openpgp.org/) key files - e.g. in authorized\_keys files or as client identities.
* [SSHD-849](https://issues.apache.org/jira/browse/SSHD-849) - Data forwarding code makes sure all pending packets have been sent to the peer channel when closing the tunnel gracefully.
* [SSHD-850](https://issues.apache.org/jira/browse/SSHD-850) - Add capability to retry a failed private key decryption.
* [SSHD-857](https://issues.apache.org/jira/browse/SSHD-857) - Add session disconnect event signalling to SessionListener.
  + Also calling ReservedSessionMessagesHandler#handleUnimplementedMessage not only for SSH\_MSG\_UNIMPLEMENTED but also for any unexpected/unrecognized command encountered during the session message processing loop.
* [SSHD-859](https://issues.apache.org/jira/browse/SSHD-859) - Provide client session connection context that is propagated to the SSH session.
  + Also added connection context argument (propagated from the ClientSessionCreator#connect invocation) toconnectionEstablished and abortEstablishedConnection methods of IoServiceEventListener.
* [SSHD-860](https://issues.apache.org/jira/browse/SSHD-860) - Use lazy loading of public key identities.
* [SSHD-861](https://issues.apache.org/jira/browse/SSHD-861) - Fixed username/password encoding for SftpFileSystem URI(s).
  + Also added SftpFileSystemClientSessionInitializer support in SftpFileSystemProvider
* [SSHD-862](https://issues.apache.org/jira/browse/SSHD-862) - Provide session context argument (if available) when key loading methods are invoked.
* [SSHD-864](https://issues.apache.org/jira/browse/SSHD-864) - Using a NamedResource instead of plain old string in order to provide key file(s) location information
* [SSHD-865](https://issues.apache.org/jira/browse/SSHD-865) - Key identities overrides specified in the [ssh\_config](http://www.gsp.com/cgi-bin/man.cgi?topic=ssh_config) configuration file are also lazy loaded
* [SSHD-866](https://issues.apache.org/jira/browse/SSHD-866) - Counting empty challenges separately when enforcing max. attempts during keyboard-interactive authentication
* [SSHD-870](https://issues.apache.org/jira/browse/SSHD-870) - Added hooks and some initial code to allow (limited) usage of [OpenPGP](https://www.openpgp.org/) key rings in authorized\_keys files
* [SSHD-873](https://issues.apache.org/jira/browse/SSHD-873) - CipherInformation#getCipherBlockSize method has been added for exposing the cipher's block size. **Note:** for the time being we declare a virtual block size for stream ciphers as well (e.g., RC4) in order to facilitate the automatic re-keying mechanism described in [RFC 4253 - section 9](https://tools.ietf.org/html/rfc4253#section-9) and [RFC 4344 - section 3.2](https://tools.ietf.org/html/rfc4344#section-3.2).
* [SSHD-876](https://issues.apache.org/jira/browse/SSHD-876) - Looking through the resolvable class-loaders "hierarchy" (thread-context => anchor => system) for sshd-version.properties file instead of just in the thread context class loader.
  + In this context, the default reported client/server SSH version string has been set to APACHE-SSHD-...version.... Reminder: the user can override this default via configuration properties set on the client/server instance (see AbstractSession#resolveIdentificationString, ClientFactoryManager#CLIENT\_IDENTIFICATION, and ServerFactoryManager#SERVER\_IDENTIFICATION).
* [SSHD-878](https://issues.apache.org/jira/browse/SSHD-876) - The File/DirectoryHandle(s) used by the SFTP subsystem implement AttributeStore interface - which means that SftpEventListener(s) can now attach user-defined attributes to the generated handle(s).
* [SSHD-886](https://issues.apache.org/jira/browse/SSHD-886) - Do not send SSH\_MSG\_UNIMPLEMENTED reply if registered ReservedSessionMessagesHandler signals that it has handled the unknown packet type.
* SftpCommandMain shows by default get/put command progress using the hash sign (#) marker. The marker can be enabled/disabled via the progress command:

> progress  
  
 ... reponse is whether it is 'on' or 'off'  
  
 > progress on/off  
  
 ... set the progress marker indicator ...