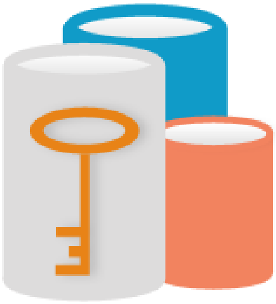
edu.Library

CERTivity® Release Notes



June 30, 2025

Document Version 2.0.16
CERTivity® 2.0

www.edulib.com

edu.Library

Legal Notice

No part of this publication may be reproduced stored in a retrieval system, or transmitted, in any form or by any means, without the prior written permission of EduLib S.R.L..

EDULIB S.R.L. MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OR MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE.

CERTivity IS A REGISTERED TRADEMARK OF EDULIB S.R.L. OTHER PRODUCT NAMES AND SERVICE NAMES ARE THE TRADEMARKS OR REGISTERED TRADEMARKS OF THEIR RESPECTIVE OWNERS AND ARE USED FOR IDENTIFICATION ONLY.

Copyright

2025, EduLib S.R.L.

www.edulib.com

Calea Bucuresti, Bl. 27B, Sc. 1, Ap. 2
Craiova, DJ, 200675, Romania
Phone: +40 351 420970
Fax: +40 351 420971
E-mail: office@edulib.ro

Table of Contents

1. Changes in CERTivity 2.0, build 16

Release Date: 2025-06-30

1.1 Bug Fixes

- Keep up with the deprecation of MD5 and SHA1 related algorithms.

2. Changes in CERTivity 2.0, build 15

Release Date: 2017-07-10

2.1 New Features

- Added support for Timestamp Information when signing Jar files.

2.2 Bug Fixes

- The wizard for installing Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files works for Java 1.8 as well.
- For Java 1.8 the type is now correctly identified for PKCS12 KeyStores.

3. Changes in CERTivity 2.0

Release Date: 2014-01-23

3.1 New Features

- Added support for Elliptic Curve (EC) Algorithms in Key Pair generation. Supported EC algorithms, their parameter specifications and signature algorithms are:
  - EC (ECDSA)
    - Parameter specification:

      c2pnb272w1, c2tnb191v3, c2pnb208w1, c2tnb191v2, c2tnb191v1, c2tnb359v1, prime192v1, prime192v2, prime192v3, c2tnb239v3, c2pnb163v3, c2tnb239v2, c2pnb163v2, c2tnb239v1, c2pnb163v1, c2pnb176w1, prime256v1, c2pnb304w1, c2pnb368w1, c2tnb431r1, prime239v3, prime239v2, prime239v1, sect233r1, secp112r2, secp112r1, secp256k1, sect113r2, secp521r1, sect113r1, sect409r1, secp192r1, sect193r2, sect131r2, sect193r1, sect131r1, secp160k1, sect571r1, sect283k1, secp384r1, sect163k1, secp256r1, secp128r2, secp128r1, secp224k1, sect233k1, secp160r2, secp160r1, sect409k1, sect283r1, sect163r2, sect163r1, secp192k1, secp224r1, sect239k1, sect571k1, B-163, P-521, P-256, B-233, P-224, B-409, P-384, B-283, B-571, P-192, brainpoolp512r1, brainpoolp384t1, brainpoolp256r1, brainpoolp192r1, brainpoolp512t1, brainpoolp256t1, brainpoolp224r1, brainpoolp320r1, brainpoolp192t1, brainpoolp160r1, brainpoolp224t1, brainpoolp384r1, brainpoolp320t1, brainpoolp160t1.
    - Signature algorithm:

      SHA1withECDSA, SHA224withECDSA, SHA256withECDSA, SHA384withECDSA, SHA512withECDSA.
  - ECGOST3410
    - Parameter specification:

      GostR3410-2001-CryptoPro-A, GostR3410-2001-CryptoPro-XchB, GostR3410-2001-CryptoPro-XchA, GostR3410-2001-CryptoPro-C, GostR3410-2001-CryptoPro-B.
    - Signature algorithm:

      GOST3411withECGOST3410.
- Added support for managing and using Elliptic Curve (ECDSA and ECGOST) Algorithms Key Pairs and Certificates. These can be used for signature, import, export and any other operations where RSA and DSA keys can be used, as long as the underlying standards support them. Management support for EC keys include: Select CA Issuer, Sign by CA Issuer, Generate CSR, Import CA Reply, Generate Key Pair Signed by CA Issuer, Regenerate Key Pair, Extend Validity, Convert KeyStore type, Append to certificate chain, export/import EC Key Pairs and parts of the Key Pair, Copy/Paste and other operations.

- Added an innovative and easy to use wizard for installing Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files via an integrated JavaFX browser.

  The JCE Unlimited Strength Jurisdiction Policy Files are required in order to remove default restrictions regarding the cryptographic algorithms and maximum cryptographic strengths available by default in the JCE framework. For example, the JCE Unlimited Strength Jurisdiction Policy Files are required in order to support PKCS#12 / Uber KeyStore files that use passwords larger than 7 characters. Note that the usage of these files is under the import restrictions of some countries, so it is the user's responsibility to decide if (s)he is entitled to use them.

  This wizard can be started from the "Advanced Details" dialog that opens from the "Help / Advanced Details" menu item. In the "Advanced Details" dialog, under the "Security Properties" tab, the "Policy Status" section indicates whether the JCE Unlimited Strength Jurisdiction Policy Files are installed. If the JCE unlimited strength jurisdiction policy files are not installed, an "Install" button is present and enabled for starting the wizard.

  ### Note

  If the JCE Unlimited Strength Jurisdiction Policy Files are already installed on the system, the "Install" button will be disabled.

  A web browser window will be displayed and you will be driven to the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files download page. Here, if you are entitled to use the policy files, you click to accept the "Oracle Binary Code License Agreement for the Java SE Platform Products", and then click on the "zip" file to download the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files.

  The installation process is automatic, so, except the License Agreement, no other action is required. At the end, a message confirming the installation will be displayed, or, in case of failure, instructions of how to install the JCE Unlimited Strength Jurisdiction Policy Files semi-manually.

  After that, you will need to restart CERTivity in order for the changes to take effect. You should now see, in the "Advanced Details" dialog, in the "Policy Status" section, that the JCE Unlimited Strength Jurisdiction Policy Files are installed.

  ### Note

  This feature is available only when using Oracle's Java Virtual Machine version 1.7.0_13 or later;

- Added an "Advanced Details" option under the "Help" menu, that displays detailed information about system and security properties.

  - `Security Properties` - here are displayed detailed information about the Java Virtual Machine's security settings:

    - `Policy Status` - what kind of security policy is installed on the system, if the unlimited strength jurisdiction policy files are installed;

      It also provides a wizard for automatic installation of the unlimited strength jurisdiction policy files. In order to use this feature, you must click on the "`Install`" button.

    - `Policy Files` - the system's currently installed policy files, with the ability to view files, view each file content and browse java security directory where the policy files are installed.
  - `System Properties` - Here are displayed detailed information about the system: Operating System properties and Java Virtual Machine's properties:
    - `Java Home` - path to the currently used Java Virtual Machine;
    - `Java Vendor` - name of the vendor of the currently used Java Virtual Machine;
    - `Java Vendor URL` - URL to the vendor's site;
    - `Java Version` - the version of the currently used Java Virtual Machine;
    - `Java Class Path` - class path of the currently used Java Virtual Machine;
    - `OS Arch` - architecture of the installed Operating System;
    - `OS Name` - the name of the installed Operating System;
    - `User Dir` - the current user directory;
    - `User Home` - home directory of the currently logged user;
    - `User Name` - name of the currently logged user.
- Added full support for eleven new certificate extensions, thus covering all standard certificate extensions according to Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile - RFC 5280. The new certificate extensions available to add are:
  - Certificate policies;
  - Policy mappings;
  - Subject alternative name;
  - Issuer alternative name;
  - Authority information access;
  - Subject information access;
  - Inhibit any policy;
  - Name constraints;
  - Policy constraints;
  - Freshest CRL (a.k.a. Delta CRL Distribution Point);
  - Subject Directory Attributes.
- Interface improvements for certificate extensions management such as:
  - When adding another extension the tree is not entirely re-expanded, keeping the folding options;
  - The extension root cannot be folded anymore, thus the extension tree is always visible;
  - Made the Generate Key Pair Dialog larger in order to better view certificate extensions. Resizing the Generate Key Pair dialog resizes the Extension tree area proportionally;
  - Click anywhere in the error table row to open the extensions which needs attention.

- Added Quick Search feature for application, consisting of a text field in the top right corner of application. As a search string is typed in the field, a drop-down list appears, showing matching items. The items come from the names of actions, options registered in the application and help topics in the application's JavaHelp. When an action item is selected, the action is invoked; when an option item is selected, the options dialog is displayed; when a help item is selected, the topic opens in JavaHelp.
- The Quick Search feature for an opened KeyStore, triggered when starting to type a search string, having the focus on an opened KeyStore tree, searches through values of all the entry fields (alias, subject, issuer, etc.) not just the entry alias and the visible columns.
- Added filter to the Options Panel, consisting of a text field in the top right corner of the Options Panel. As a filter string is typed in the field, the matched tab from the available options is activated.
- KeyStore related actions are now displayed in the context menu, when right-clicking on an empty area inside an opened KeyStore.
- Updated to version 1.49 of the Bouncy Castle library.
- With the update of the Bouncy Castle library the default BKS KeyStore type is not compatible with older versions of Bouncy Castle. In order to keep compatibility a new KeyStore type, BKS-V1, has been added for people needing to create key stores compatible with earlier versions of Bouncy Castle. Depending on your needs when you create a new KeyStore you can either select `bks` (the new version, 2) or `bks-v1` for legacy version. When opening an existent `bks-v1` KeyStore its type is maintained over the usage and save. You can also convert back and forth between `bks` and `bks-1` types.
- Updated to version 7.4 of NetBeans RPC.
- Updated to version 5.1.7 of install4j.
- Added Windows installer with bundled 64bit Java Runtime Environment.
- Added MacOS installer with bundled 64bit Java 7 Runtime Environment.
- Added new "Advanced Options" category in the options dialog under the "Tools / Options" menu. This category offers some advanced settings that can be set to enhance some CERTivity features:
  - `Download URL` - specifies the link to the Oracle site from where the JCE Unlimited Strength Jurisdiction Policy Files need to be downloaded;
  - `File download pattern` - pattern identifying the direct authenticated link to the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files. When clicking on download, the browser is redirected to other page, that is the final, authenticated direct download link (e.g.: for JRE 7, the identifying pattern can look like this: *".zip?AuthParam="*,because the link is identifying a *"zip"* file and it has an *"AuthParam"* parameter);
  - `Use Secure Validation for XML Signatures` - when set to `true`, this option instructs the implementation to process XML signatures more securely. This will set limits on various XML signature constructs to avoid conditions such as denial of service attacks. When set to `false`, the property instructs the implementation to process XML signatures according to the XML Signature specification without any special limits.
- Updated the display CSR file panel, in order to also display the CSR file challenge.

3.2 Bug Fixes

- Implemented a separate Security Provider in order to correctly manage the cut/copy/paste actions for PKCS#12 KeyStore type.
- Changed the way the Generate Key Pair Dialog responds to the Enter key: the default button which responds to Enter Key is the "OK" button; When the user adds a certificate extension that shows a panel with an input text and a "Set Value" button, when the user focuses the input text to enter a value, the default button (which responds to Enter key) becomes the "Set Value" button. This prevents the accidental close of the dialog, when the user actually wanted to insert a value for the extension.
- Fixed always showing the warning "Could Not establish trust Path" when signing another Key Pair with a Key Pair selected as CA Issuer, although the current KeyStore is selected in the Trust Path options, or the KeyStore path is explicitly specified.
- Fixed showing the message "Successfully signed by <X> and Imported its certificate in chain" even when selecting not trusting the root, hence the signature not taking place.
- Fixed requesting Key Pair password for PKCS 12 which do not have passwords.
- Fixed Help window displaying behind the main window on Mac OS X.
- Fixed error when trying to open a certificate chain with 6 or more certificates exported in PKCS#7 format.
- Fixed incorrect display of certificate chain when generating a new Key Pair in a PKCS#12 KeyStore (only the user certificate was displayed).
- Made the Secret Key icon transparent.
- Fixed a focus lost bug when using the persistence option "Persist only KeyStore file name (without password)" and the password was requested for a KeyStore located in a different tab than the one the application started with.
- Fixed a focus lost bug when using the application Quick Search function which opened the Options dialog. The focus was in the KeyStore panel, not in the Options dialog, and this is now fixed.

4. Changes in CERTivity 1.2

Release Date: 2013-05-09

4.1 New Features

- Added support for viewing CRL files - Implemented a CRL viewer that acts as a top component when opening a CRL file. The user can open a CRL file from two sources:

  - From a file;
  - From a URL.

  The information contained in a CRL file is displayed as follows: the fields of the CRL file are displayed in the left part of the application window using a tree structure and the content of the CRL fields or tree node selected is displayed in the right part of the application window. The revoked certificates can be viewed individually by clicking on each revoked certificate node, or they can be displayed as a list, by selecting the Revoked Certificates List.

  Also implemented CRL extensions and CRL entry extensions as defined in the RFC 3280, and the extensions Issuer Alternative Name and Subject Alternative Name which can be used also as CRL or Certificate extensions.

  Updated the "`Open Recent File`" option from the "`File`" menu to also remember URL locations, not just file locations.

  Added a new feature that allows the user to start viewing a CRL from the certificate details, if the certificate has an associated CRL. For this, a new button was added in the toolbar found at the top of the window section that displays the certificate details. This button is called "View Associated CRL" and it is active only when the certificate contains the corresponding information in the "CRL Distribution Points" extension.

- Added selective drag and drop - We are now able to open using the drag and drop action the following files: certificates, CRLs, CSRs and KeyStores. This is working on Microsoft Windows and Linux platforms.

- Added new signature algorithms for Key Pair/CSR generation and for CSR signing. Also updated the Digest algorithms for the Sign Jar action. The algorithms supported now for generation of Key Pair/CSR files and for signing CSR files are:

  - For DSA:
    - SHA1withDSA;
    - SHA224withDSA;
    - SHA256withDSA;
  - For RSA:
    - MD2withRSA;
    - MD5withRSA;
    - RIPEMD128withRSA;
    - RIPEMD160withRSA;
    - RIPEMD256withRSAE;
    - SHA1withRSAandMGF1;

    - SHA1withRSA;
    - SHA224withRSAandMGF1;
    - SHA224withRSA;
    - SHA256withRSAandMGF1;
    - SHA256withRSA;
    - SHA384withRSAandMGF1;
    - SHA384withRSA;
    - SHA512withRSAandMGF1;
    - SHA512withRSA.

  The Digest algorithms that we support now for the Sign Jar action are:

  - MD2;
  - MD5;
  - SHA-1;
  - SHA-224;
  - SHA-256;
  - SHA-384;
  - SHA-512.

- Added new Key Algorithms for Secret Key generation. The algorithms are defined for 2 providers: for the Bouncy Castle Provider and for the Sun JCE Provider (if it exists on the system where CERTivity is running), allowing the user to select only the supported key sizes for each algorithm depending on the algorithm type and provider. In case the Sun JCE Provider is not available, the Default provider will be used which means that all the Secret Key algorithms (that CERTivity supports) will be displayed with the key sizes starting from 1 for each algorithm. For this case, if the algorithm or the key size is not supported by the Default provider, an error will be displayed.

  The available algorithms for the Bouncy Castle Provider are the followings:

  - AES;
  - AESWrap;
  - Blowfish;
  - Camellia;
  - Cast5;
  - Cast6;
  - DES;
  - DESede;
  - DESedeWrap;
  - GOST28147;
  - Grainv1;

- Grain128;
- HC128;
- HC256;
- Noekeon;
- RC2;
- RC4;
- RC5;
- RC5-64;
- RC6;
- Rijndael;
- Salsa20;
- SEED;
- Serpent;
- Skipjack;
- TEA;
- Twofish;
- VMPC;
- VMPC-KSA3;
- XTEA;
- HmacMD2;
- HmacMD4;
- HmacMD5;
- HmacRIPEMD128;
- HmacRIPEMD160;
- HmacSHA1;
- HmacSHA224;
- HmacSHA256;
- HmacSHA384;
- HmacSHA512;
- HmacTIGER.

The available algorithms for the Sun JCE Provider are the followings:

- AES;
- ARCFOUR;
- Blowfish;

  - DES;
  - DESede;
  - RC2;
  - HmacMD5;
  - HmacSHA1;
  - HmacSHA256;
  - HmacSHA384;
  - HmacSHA512.
- Added support for file type inspection - Implemented a new feature that inspects the type of a chosen file based on a greedy heuristic algorithm. This feature was added in the "`File`" menu and it is called "`Inspect Type`". The cryptographic file types detected are:
  - JKS KeyStore;
  - JCEKS KeyStore;
  - BKS KeyStore;
  - UBER KeyStore;
  - PKCS#12 KeyStore or Key Pair;
  - Certificate;
  - Certificate Signing Request (CSR) of PKCS10 type;
  - Certificate Signing Request (CSR) of SPKAC type;
  - Certificate Revocation List (CRL);
  - Encrypted Microsoft PVK Private Key;
  - Unencrypted Microsoft PVK Private Key;
  - Unencrypted OpenSSL Private Key;
  - Unencrypted PKCS#8 Private Key;
  - Encrypted OpenSSL Private Key;
  - Encrypted PKCS#8 Private Key;
  - OpenSSL Public Key.

  If the type of the file inspected is not among the ones listed above, the user will be prompted with the following message: "`The type of the <InspectedFileName> file was not detected.`".
- Added support for viewing CSR files - Implemented a CSR viewer that acts as a top component when opening a CSR file and that allows the user to view all the possible fields in a CSR file. This action is called "`Open CSR`" and can be accessed from the "`File/Open`" menu.
- Improved the appearance of the "Certificate Extensions" tab to be more user friendly for adding extensions - Updated the appearance of the "Certificate Extensions" tab found in the "Generate Key Pair" and "Sign CSR File" windows. This was done by adding a

self-explanatory text in the right panel from the "Certificate Extensions" tab, to make the interface more user friendly when it comes to adding extensions using the contextual menu that appears when right-clicking on the tree nodes.

The text added is the following: "`To manage extensions, right-click on the tree nodes (initially the "extensions" root) and follow the contextual menu that appears.`"

- As many CAs only return the issued certificate, with no supporting chain, we improved the certificates management in certificate chain through the following features:
  - Insert/Delete Certificate to/from chain: these two operations can be done using two methods:
    - Using the two new actions ("Append Signer Certificate" and "Remove Signer Certificate"), which can be found in the "`Certificate Chain Actions`" menu that is available in the contextual menu of the Certificate Chain node and Key Pair node.
    - Using the Copy/Paste/Delete actions for inserting and deleting the top certificate from the chain. Undo/Redo functionality was also implemented for pasting and deleting.

    The action for inserting certificates appends the top certificate or another chain of certificates on top of the existing one. If there should be inserted a chain of certificates, the chain is ordered first. When inserting a new certificate or chain, we display an error message if the user is trying to insert one or more certificates "on top" of a self-signed certificate or if this operation will invalidate the existing chain of certificates.

    The action for removing certificates removes the certificate from the top of the chain, only if the chain does not have only one certificate, in which case a notice will be displayed for the user.
  - The "Import Key Pair" action was improved to allow the user to specify a set of certificate files instead of a single certificate file - Support was added for the "Import Key Pair" action so that the user can create more Certificate input fields by pressing the "Add More Certificates..." button. In the "Certificate(s) File" input fields, the user can enter either certificate files or Certificate Chain files in which order (s)he desires, because the certificates are collected from the files and ordered, the resulting chain being validated. If the resulting chain is not valid, an error message is prompted and the previous dialog is displayed allowing modifications.
  - Extended the Certificate viewer to display multiple certificates including certificate chains - Because the certificate files can contain more certificates, including chains, support was added for an extended Certificate viewer. If an opened certificate file contains more than one certificate, then in the left part of the new tab these certificates will be displayed in a tree view reflecting their hierarchy. When a certificate is selected in the tree view, the information associated with it will be displayed in the right part of the window.
- Extended the Fingerprints in all Certificates Details sections - Support was added so that in the Certificate Details sections, the certificate fingerprints to be also available in the following hashes: MD2, MD4, RIPEMD-128, RIPEMD-160, RIPEMD-256, SHA-224, SHA-256, SHA-384 and SHA-512.
- Added the "`Windows Menu Key`" as a shortcut for the System/Popup action. Also added the `Shift-F10` keys combination as Shortcut for the contextual menu.

- Support was added to make certificate signing easier - To decrease the number of steps taken when signing a certificate and acting as a test or in-house CA, the following items were implemented:

  - Added in the contextual menu available for Key Pairs a new option called "`Select CA Issuer`", which acts as a check box. When this option is selected for a Key Pair node, the Private Key contained in the selected Key Pair is selected as CA Issuer. Note that before the Private Key is selected as CA Issuer, the password (if any) is requested for the Private Key.

  - Added another option in the contextual menu available for Key Pairs, option called "Sign Certificate by <...>". When a Key Pair node is selected as the CA Issuer (the status bar and contextual menu reflect if a node is selected as the CA Issuer), then the "`Sign Certificate by <...>`" menu option dynamically changes text to contain the alias of the Key Pair selected as the CA Issuer.

    If the Key Pair node chosen to act as CA Issuer is selected, the "`Sign Certificate by <aliasForIssuer>`" menu item will be inactive for that Key Pair. This is done in order to avoid signing a certificate by the same Key Pair that was assigned as CA Issuer.

    If a Key Pair node is selected as CA Issuer in a KeyStore and another KeyStore is opened, the selection will also work on the Key Pairs of the second KeyStore. To know which Key Pair is selected as CA Issuer the status line was updated to contain the following information: "`CA Signer: keystore / alias`".

  When the "`Sign Certificate by <aliasForIssuer>`" contextual menu item is accessed the following steps happen transparently:

  - a CSR is generated for the selected Key Pair. The CSR generated will not be saved, will be kept in memory. The type of the generated CSR will be PKCS10 and the algorithm used will be SHA1withDSA for keys of DSA type and SHA1withRSA for keys of RSA type.

  - the generated CSR file will be signed using the CA Issuer selected. The signing process will differ from the standard "Sign CSR File" action in that the file choosers will not appear anymore, since we are using the CSR and CA Replies stored in memory. Only the second window will appear ("Sign CSR File").

  - after the CSR generated is signed by the current CA Issuer and the "OK" button is pressed, the CA Reply produced will be automatically imported in the current Key Pair.

  This new feature was also made available for the "Generate Key Pair" action, by doing the following:

  - Added in the "Certificate Info" tab from the "Generate Key Pair" dialog a check-box with the label "`Sign By <...>`". This check-box is placed above the "Signature Algorithm" combo-box. When no CA Issuer is selected the "`Sign by <...>`" is inactive. When a CA Issuer is selected the "`Sign by <...>`" check-box is active and the "`...`" string is replaced with the name of the selected CA Issuer.

    When this check-box is checked and the "OK" button is pressed, a CSR will be generated for the newly created Key Pair. This CSR will be signed by the selected CA Issuer and the CA Reply obtained will be imported in the newly created Key Pair - all these being done automatically.

- Support was added to detect CSR files and resulting CA Replies signed by/imported in the same Key Pair - In order to prevent CSR files to be signed by the same Key Pair that generated them, we verify each certificate from the ordered chain not to have the same public key as the next certificate in the chain. If we find such a case, the user will be presented with the warning message: "`The public key of the CSR file is contained in the Key Pair you are trying to sign it with.`" and (s)he will be allowed to continue the signing process if (s)he chooses so.
- Support added to standardize the country code contained in a DN - To make sure that the Country code contained in a DN is a two-letter ISO code, we added a combo-box populated with all the ISO countries, combo-box that has enabled automatic completion. The automatic completion is strict, only items from the combo-box can be selected. Also, if the user does not want to set a country code, the "None" option is available in the combo-box. The combo-box will be active in the "Generate Key Pair" and "Regenerate Key Pair" dialogs.
- Warning message added when "Esc" key is pressed in the "Generate Key Pair" or "Sign CSR File" dialogs - Pressing "Esc" key when in the "Generate Key Pair" or "Sign CSR File" dialogs closed the dialogs without any notice, thus making possible to accidentally loose the information already filled in many fields contained in the dialog, including the Certificate Extensions. We now added a warning message when "Esc" key is pressed, allowing the user to choose whether to close or not the current dialog.
- Extended the Options panel to support more settings through the following features:
  - Renamed the "Certificate Options" option found in the "Tools/Options" dialog, to "Main Options".
  - Added in the "Tools/Options/Main Options" dialog a new preference field, called "Certificates Retriever connection type" that is populated with all connection types available for the java version used by the user. This new preference field should be adjusted when the "SSL Certificates Retriever" action fails with the message: "`The connection type (SSL) may not be supported or client authentication may be required. To choose another connection type, access the "Tools/Options/Main Options" menu option.`"
  - Added a new option in the "Tools/Options" dialog, option called "Trust Path Options", to allow dynamic TrustStore management at runtime. This new option contains two tabs called "TrustStores Selection" and "Trust Validation Options", these tabs containing in turn new options for setting the TrustStores and validation options. For setting the TrustStores, there are more options: selecting from one of the existing JVM CA KeyStores, selecting from the Windows KeyStores, setting custom KeyStores, or using the currently opened KeyStore. Each of these options can be enabled using a check-box, and for selecting the KeyStores (in case there are more available), each KeyStore can be selected from a check-box list. The options available in the "Trust Validation Options" tab are the following:
    - Inhibit any policy - this option sets the value of the any policy inhibited flag. The default value for the flag, if not specified, is false.
    - Explicit policy required - this option sets the value of the explicit policy required flag. The default value for the flag, if not specified, is false.
    - Inhibit policy mapping - this option sets the value of the policy mapping inhibited flag. The default value for the flag, if not specified, is false.
    - Use revocation checking - this option allows the user to disable revocation checking. Revocation checking is disabled by default.

  - Use policy qualifier processing - this option allows the user to enable or disable policy qualifier processing. This setting is set to true by default and reflects the most common (and simplest) strategy for processing policy qualifiers.
  - Use a path length constraint of: <default value 5> certificates - this option specifies the maximum number of non-self-issued intermediate certificates that may exist in a certification path. If the value is 0, the path can only contain a single certificate. If the value is -1, the path length is unconstrained. The default maximum path length, if not specified, is 5. This option is useful to prevent from spending resources and time constructing long paths that may or may not meet the requirements.
  - Use this date for validation: <default value - current date> - this option sets the time for which the validity of the path should be determined. If this option is not set, the current date is used by default.
- Added Trust Path validation - Implemented Trust Path validation for the "Import Certificate" and "Import Ca Reply" actions. Also added an additional column to the KeyStores view, to display if the certificates are trusted or not. This column applies only to Certificate Entries, not to Key Pair entries.
- Added support for entering a new serial number when extending validity of a self-signed certificate. On the "Extend Validity" dialog, the user can now also see the current serial number of the certificate for which the validity period should be extended, and can enter a new serial number for the new resulting certificate (either by generating one using the "Generate" button, or by entering a custom serial number in the corresponding text box).
- A Quick Search triggered by the keyboard input is now available in the KeyStore panel allowing selective column search.
- The main icon of CERTivity KeyStores Manager was facelifted and CERTivity Help has a suitable 32x32 icon as well.
- Some GUI labels uniformization - We made sure all labels are Title Case when they need to be, also updated some of the labels to be more readable, in order to provide a more uniform view for the application.

4.2 Bug Fixes

- The "SSL Certificates Retriever" option availability - The "SSL Certificates Retriever" option is now always available in the "Toolbars/KeyStore" and in the "Menu/KeyStore", even when no KeyStore is opened. When the active component is not a KeyStore, the "Import to KeyStore" button from the "SSL Certificates Retriever" dialog is not enabled.
- Comparison of issuer DN and subject DN of the certificates in the certificate chain - In some situations, when validating the certificate chain, the comparison of issuer DN and subject DN of the certificates in the chain failed because of different ways of representing the DN names. We modified the issuer - subject name comparison mechanism, so that the comparison to be done between X500Name objects using the "equals" method which is implemented by Bouncy Castle. To obtain X500Name we wrapped the certificates in X509CertificateHolder objects.
- Importing PEM certificate chains files issue - When importing a certificate chain from a PEM file where the PEM certificates are separated by white spaces, for example, or the file contains invalid content not within the `BEGIN` and `END` sections, the loading of the certificates from the file failed. A workaround was implemented for filtering the content of PEM certificate files, so that the empty lines (and the invalid content not within the `BEGIN` and `END` sections of the PEM file) to be ignored.

- Certificates wrongly labeled as Self Signed - In the "Key Pair details" section (that appearss when the user selects a Key Pair) the certificate was sometimes labeled as Self Signed when the issuer was missing from the chain. This is now fixed.
- Date validation for "Extend Certificate Validity" action - Implemented date validation for the new expiration date introduced for the selected certificate when using the "Extend Certificate Validity" action.
- Focus lost on File Chooser for unlocked Key Pairs/Private Keys on Linux - On Linux focus was lost when trying to export an unlocked Key Pair/Private Key and one cannot write anything in the file name field of the File Chooser. This is now fixed.
- The contextual menu opened with the first time right click in a freshly opened KeyStore showed only the standard options - Support was added so that the contextual menu shows from the beginning the specific options when opened with right click.
- Full Screen after registering the license - Using View/Full Screen after registering the license resulted in a pop-up error message "`License Installation canceled. Application will now exit.`". This happened only in the first run when a License Key File was installed. This was fixed and the message shows now only when the License Key installation is indeed canceled.
- Sporadic `NullPointerExceptions` are fixed by switching to a newer version of NetBeans RPC, RELEASE721 - the `NullPointerException` occurred from the `DelegateAction` class from `GeneralAction` because in the "`removePropertyChangeListener`" method, the `PropertyChangeSupport` object is used without checking if it is null, and apparently somehow it gets null until that step. Other items such as the Quick search for TreeTableView are now available and the bug with Redo label which changed after first Undo is now transparently fixed.
- Fixed revocation status check when the URL in the certificate extension pointed to a page which redirects to another URL.
- Fixed opening the same certificate file multiple times in different tabs - Made sure that when opening a certificate file we first check if the file is already opened, and if it is, we switch to it.
- Updated the Public Key information label in the Certificate Details panel to show the Public Key size in bits and not in bytes.
- Fixed the opening of the Public Key (from the Certificate Panel) to not throw an error and to display the content of the Elliptic Curve Public Keys also. The key size displayed in the "Key Type" column of the KeyStore view now also displays the size for Elliptic Curve (EC) keys.

5. Changes in CERTivity 1.1

Release Date: 2012-10-18

5.1 New Features

- Opening the machine's JRE CA TrustStore(s) - Added support for opening the CA TrustStore(s) of the JRE(s) discovered on the current system. We search for the TrustStores in the following locations:

  - The Java property `${java.home}` of the JRE CERTivity started with;
  - The system environment variables `JAVA_HOME` and `JRE_HOME`;
  - For Windows platforms searching the installed Java JDKs and JREs in the Windows registry;
  - For Unix and Mac we are looking for traditional Java installation directories such as `/usr/java` for Unix, `/usr/lib/jvm` for Linux (Debian, RedHat) and for Mac `/Library/Java/Home/`, `/System/Library/Java/JavaVirtualMachines/`. Various patterns are then used.

  If the KeyStore persistence settings in `Tools > Options` is set to "`Fully persist (file name & encrypted password)`" the passwords of the TrustStores are saved in the preferences after closing the application.

- Support for transition to secure RSA keys - As Microsoft announces that the use of certificates that have RSA keys that are less than 1024 bits long will be blocked, and in the future other systems and even Java Virtual Machine may do this as well, we have introduced features to easily spot such certificates, and to warn when generating RSA Key Pairs with less than 1024 bits. Also, we have made the minimum size allowed for generating RSA Key Pairs configurable and it can be set in the `Tools > Options`. The minimum size can not be less than 1024.

  The Certificates and Key Pairs which have RSA Keys that are less than the minimum size set are now marked in the KeyStores by having the name and the Key Type / Size colored in red. The Key Type / Size is also colored in red in the certificate panel.

  Another measure is that the default out of the box RSA Key Pair size is now doubled to 2048, too. This value can, at any moment, be increased by users (from `Tools > Options`) and it will remain persistent for future uses.

- Signing APK (Android Application Package) files and verifying signatures on signed APK files - Added support for singing and verifying signatures on APK files in the Sign JAR and Verify JAR actions. By selecting the APK file filter (or by selecting the "All files" file filter), the user is now able to select APK files and sign them in a similar manner as it is done for signing JAR files or for verifying signatures on signed JARs.

- SSL Certificates Retriever details using HTTPS URLs - Added an additional field to the SSL Certificates Retriever for entering an URL (HTTPS protocol) from which the host and port will be parsed. If the URL does not start with a protocol designator the `https://` one will be added automatically. Also, if the URL has no port specified, the default port for HTTPS protocol (443) will be used. The parsed host and port will also be set into the initial fields "Host name" and "Port", which can still be used.

- Importing certificates from the signature verification results of the verify JAR, PDF and XML actions into the active KeyStore (the active KeyStore tab) - Implemented support to allow importing a selected certificate found in the signature of a JAR, PDF or XML file into the active KeyStore (the active top component), directly from the verification results panels of the verify JAR, PDF and XML actions.

- Basic display of certificate extensions - Added support for identifying and displaying the detailed content of the following certificate extensions:
  - Authority Key Identifier;
  - Basic Constraints;
  - CRL Distribution Points;
  - Extended Key Usage;
  - Key Usage;
  - Netscape Cert Type;
  - Private Key Usage Period;
  - Subject Key Identifier.

  For the extensions which are not identified, the OID will be displayed and a rough representation of the extension content.
- ASN.1 display of each certificate extension - added support for displaying the ASN.1 representation of the content of each extension of the certificate (even for the ones for which we display only the rough representation of its content).
- Extensions for certificates when generating Key Pairs - Implemented support for adding certificate extensions to Certificates when generating Key Pairs using an easy to use tree-like structure for adding and representing the extensions. The following extensions can be added:
  - Authority Key Identifier;
  - Basic Constraints;
  - CRL Distribution Points;
  - Extended Key Usage;
  - Key Usage;
  - Netscape Cert Type;
  - Private Key Usage Period;
  - Subject Key Identifier.

  The extension structure is validated in real time, as the extensions and their sub-items are being added, showing information about the validity status, and details about the validation errors (if any). Also, some values from some extension fields (such as the Directory Name components from the General Name of Authority Certificate Issuer component of the Authority Key Identifier extension, and others) are filled automatically using the values provided in the certificate fields (if they are not empty). More than this, when adding extension subitems which require some information from the certificate which was not provided yet, for example if the Authority Certificate Serial Number is required and a Serial Number was not provided yet, the information can be generated at that time, and it will also be set for the corresponding certificate field. Also, the mandatory subitems of the extensions are added automatically in the tree structure, to ease the job of the user, and for the optional ones, the user can select which one to use from a context menu.

- Handling extensions for CA Reply certificates when signing a CSR file - Implemented support for adding certificate extensions to the CA Reply user certificate as well, which will be obtained after signing a CSR file. The mechanism for creating and adding extensions is the same one described for adding extensions when generating a new Key Pair, with the small difference that the information regarding the issuer is now taken from the signer certificate.
- Viewing extensions structure at creation time as XML - Implemented support for viewing the tree like structure of the extensions in XML format, for easy visualization or for easy copying into other documents. Each extension is a node, and each subitem of an extension is a child node of the extension node.
- Saving extensions structure at creation time as XML templates for later usage and loading them back into the tree structure - Added support for saving the extensions structure as an XML document into a file as an extensions template for later usage. These templates can be loaded later, when generating a new Key Pair, or when creating a CA Reply, which needs for its Certificate a similar or maybe identical extensions structure.
- Reopening the last used files (Key Stores, Certificates) - Added "`Open Recent File`" menu (in the `File` menu), to allow reopening the last used files like KeyStores and Certificates (and the "Readme" file). For the KeyStore files, if the KeyStore persistence option in the `Tools > Options` is set to "`Fully persist (file name & encrypted password)`", the passwords will also be stored, so that the user won't have to re-enter them again when reopening the KeyStores. Otherwise, the user will be prompted to enter the KeyStore password again.
- Renaming an entry by the `F2` key - The `F2` key is the default standard for the Windows Users for renaming, so we also use it in CERTivity for renaming entries, to improve usability, besides the existent shortcut `Ctrl + R`.
- Generating a new Key Pair using the information from an existing Key Pair - Implemented functionality to allow the user to generate a new Key Pair using some information from an existing Key Pair such as:
  - Key Type (RSA or DSA);
  - Key Size;
  - Certificate Version;
  - Signature Algorithm;
  - Certificate Issuer / Subject Distinguished Name (Common Name (CN), Organization Unit (OU), Organization Name (O), Locality Name (L), State Name (ST), Country (C), Email (E)).
- Opening Software Publisher Certificate (SPC) Certificate Files.
- The CA Reply file chooser contains the "All files" filter - This way, the user can open and import a CA Reply even if the file name does not contain the required file extension, if the file really contains a CA Reply.
- Opening the "Readme.txt" file on the first run of the application - Now the "Readme.txt" file is opened and displayed as a new tab in CERTivity on the first run of the application. This file can also be later opened using the `File > Open Recent File` menu (if the file is still in the recent files list). This file contains details, such as passwords, related to the samples provided with CERTivity.

- The "Validity" column from the KeyStore view was renamed to "Validity Status".

5.2 Bug Fixes

- JAR Signature Block compatibility - The Signature Block file (e.g. *.RSA) was bigger than the one produced by JDK's `jarsigner` for the same jar, the same key because there were two equivalences one at logical level, and one at binary (encoding) level. Although the `jarsigner` from Sun (Oracle) JDK was OK with these equivalences we are now using really the same, unequivocal productions by involving the Distinguished Encoding Rules (DER) and by using a PKCS7 with a direct signature, without the default signed attributes. This makes CERTivity compatible with other tools than JDK's, such as the Android SDK tools.
- Focus lost and entry selection issues - Fixed the focus lost and entry selection issues after operations involving KeyStores, including Undo/Redo. Now the focus remains in the Tree Table (allowing navigation through the KeyStore and performing actions using the keyboard) after performing operations such as deleting or adding new entries, converting KeyStore type, Extend Validity period for self signed Key Pairs. Also now the entries remain selected after performing Undo/Redo operations (for example after performing Undo on a delete operation, the entry will be selected). A detailed list of fixes regarding the focus and selection issue can be seen below:
  - After adding entries (by generating, importing or pasting new entries) and using Undo and then Redo, the entries remain now selected (either if there is one entry or more entries), and the focus is in the Tree Table;
  - After deleting or cutting an entry, when performing Undo, the resurrected entry is now selected in the tree, and the tree has the focus allowing navigation;
  - After performing Undo/Redo on a rename operation, the entry which is being renamed remains selected and the focus is in the tree;
  - After changing the password of an entry, the entry remains selected, as well after performing Undo or Redo on the password change action;
  - After extending the validity period of a self signed Key Pair, the entry remains selected and the focus in the tree, as well after performing Undo or Redo on the Extend Validity action;
  - After performing Undo/Redo for Importing the CA Reply, the entry for which the action takes place remains selected and the focus in the tree;
  - After closing modal dialogs invoked through contextual menus, the focus and selection remain in the tree.
- Contextual Help for the `Tools > Options` panel describing the Options from JavaHelp - Now the Help from the Options panel points to the description of the Options in the JavaHelp, and not to the default help anymore.
- Line terminator in "`${certivity.home}/etc/certivity.conf`" configuration file is now platform specific - The line terminator from the file "`${certivity.home}/etc/certivity.conf`" consisted of only one LF (Line Feed), and this could have been problematic on Windows platforms when trying to edit the configuration file using a simple editor such as Notepad. A fix was made to change the line feeds from the mentioned file to the platform specific type at installation time.
- Corrections and uniformization for the File Choosers - For some actions such as opening KeyStores or Certificates the default file filter for the file chooser was "All files". This was

updated so that the default selected filter to be the one in cause (e. g. for Verifying PDF it must be PDF files, for key store files it must be KeyStore Files).

- Clipboard shortcuts on Mac OS X - The clipboard shortcuts (`Meta-C`, `Meta-V`, `Meta-X`) on MAC OS X are now correctly functioning in the Tree Table for KeyStore entries.