Tomcat編

改版履歴			
版数	日付	内容	担当
V.1.1	2014/12/22	初版	NII
V.1.2	2015/5/25	中間CA証明書のファイル名を修正	NII
V.1.3	2015/12/11	誤記修正	NII
V.2.0	2018/2/26	SHA1の記載内容の削除	NII
V.2.1	2018/3/26	CT対応版の中間CA証明書について説明を追加	NII
V.2.2	2018/7/9	ECDSA対応版について説明を追加 事前準備の記述の修正 DNのルールの修正	NII
V.2.4	2019/4/22	ECC認証局 中間CA証明書の名称を変更	NII
V.2.5	2019/6/10	DNのルール(Locality Name)の修正	NII
V.2.6	2020/4/13	中間CA証明書のファイル名を修正 DNのルール(State or Province Name、Locality Name)の修正	NII
V.2.7	2020/7/15	DNのルール、TSVファイル形式のSTおよびLの値の説明、リンクの変更	NII
V.2.8	2020/8/25	中間CA証明書の記載内容を修正	NII
V.2.9	2020/12/22	中間CA証明書を修正 サーバー証明書L、STを必須に修正 サーバー証明書OUの利用条件を修正	NII
V2.10	2022/9/22	サーバー証明書OU廃止について修正	NII

目次

1. Tomcat(JavaKeytool)によるサーバ証明書の利用

1-1. 前提条件

Tomcat(JavaKeytool)でサーバ証明書を使用する場合の前提条件について記載します。適宜、サーバ証明書をインストールする利用管理者様の環境により、読み替えをお願いします。（本マニュアルではRedhat Enterprise Linux 6.3、 OpenJDK Runtime Environmen Java 1.7、Apache Tomcat7.0.57での実行例を記載しております。　）

前提条件

1. Tomcat(JavaKeytool)がインストールされていること（対応：7系～8.5系）
2. 使用中のTomcat(JavaKeytool)に最適なJaveがインストールされていること(Jave1.6以降を前提とする)
3. Tomcatの設定ファイルserver.xmlファイルまでの絶対パス：$CATALINA_HOME/conf/server.xml
   ※Tomcat7系でサポートしているサーブレット3.0に対応するためには、Java1.6以上が必要です

CSR作成時は既存の鍵ペアは使わずに、必ず新たにCSR作成用に生成した鍵ペアを利用してください。更新時も同様に、鍵ペアおよびCSRを新たに作成してください。鍵ペアの鍵長は
RSA鍵の場合、2048bit
ECDSA鍵の場合、384bit
にしてください。

1-2. 事前準備

鍵ペア・CSRを生成する前に、事前に以下の項目の準備をしてください。

事前準備

1. キーストアファイル名：***<server_yyyymmdd.keystore>***（**「1-3-1、1-3-2、1-5-2、1-5-3、1-5-4」**で使用）
   例)server_20141226.keystore
2. サーバ DN（※サーバDNについては、本サービス証明書ポリシまたは、下記DNのルールをご確認ください）：***<サーバDN>***（**「1-3-1」**で使用）
   例)CN=www.nii.ac.jp, O=National Institute of Informatics, L=Chiyoda-ku,ST=Tokyo, C=JP
3. 鍵ペアalias名：***<tomcat>***（**「1-3-1、1-3-2、1-5-4」**で使用）
   例)tomcat
4. 鍵ストア・パスワード：***<keystore_pass>***（**「1-3-1、1-3-2、1-5-2、1-5-3、1-5-4」**で使用）
   ※Tomcat7系では、鍵ストア・パスワードと鍵のパスワードを同一にする仕様となっているため、changeit以外を設定ください。
5. 鍵のパスワード：***<key_pass>***（**「1-3-1 キーストアの生成」**で使用）
   ※[4．鍵ストア・パスワード]にも記載のとおり、Tomcat7系では、鍵ストア・パスワードと鍵のパスワードを同一にする仕様となっているため、changeit以外で鍵ストア・パスワードと同じパスワードを設定ください。
6. CSRファイル名：***<servername.csr>***（**「1-3-2 CSRの生成」**で使用）

CSRに記述するDNのルールは以下のとおりです。

DNのルール			
項目	指定内容の説明と注意	必須	文字数および注意点
Country(C)	本認証局では必ず「JP」と設定してください。 例）C=JP	○	JP固定
State or Province Name(ST)	「都道府県」（ST）は利用管理者及び利用者が所属する組織の所在地の都道府県名としサービス窓口に事前に届出したとおりの所在地の都道府県名をローマ字表記で指定してください。この情報は各所属機関の登録担当者にお問い合わせください。 例）ST=Tokyo	○	STとして指定できる値は下記リンクを参照してください。機関ごとに固定となります。 UPKI証明書 主体者DNにおける ST および L の値一覧 ※STおよびLが必須。（2020年12月22日以降）
Locality Name(L)	「場所」（L）は利用管理者及び利用者が所属する組織の所在地の市区町村名とし、サービス窓口に事前に届出したとおりの所在地の市区町村名をローマ字表記で指定してください。この情報は各所属機関の登録担当者にお問い合わせください。 例)L=Chiyoda-ku	○	Lとして指定できる値は下記リンクを参照してください。機関ごとに固定となります。 UPKI証明書 主体者DNにおける ST および L の値一覧 ※STおよびLが必須。（2020年12月22日以降）
Organization Name(O)	サービス参加申請時の機関名英語表記を設定してください。この情報は各所属機関の登録担当者にお問い合わせください。 例)O=National Institute of Informatics	○	半角の英数字64文字以内 (記号は「'(),-./:=」と半角スペースのみ使用可能)
Organizational Unit Name(OU)	証明書を使用する部局等の名前を設定してください。 （この値は省略可能です） （この値は複数設定することが可能です。複数指定する方法につきましては、CSR作成時ご使用のアプリケーションのマニュアルをご確認ください。） 例)OU=Cyber Science Infrastructure Development Department **2022年7月26日以降に新規発行・更新する証明書については、OUが廃止となります。詳細はこちら**	△	・半角の英数字64文字以内 (記号は「'(),-./:=」と半角スペースのみ使用可能) ・複数OUを指定する場合は、全体で64文字以内 UPKI証明書 主体者DNにおける OU の値一覧

Common Name(CN)	サーバ証明書URLに表示されるウェブ・サーバの名前をFQDNで設定してください。例えばSSL/TLSを行うサイトが https://www.nii.ac.jp の場合には、「www.nii.ac.jp」となります。FQDNにはサービス参加申請時に登録いただいた対象ドメイン名を含むFQDNのみ、証明書発行が可能となります。 例)www.nii.ac.jp	○	証明書をインストールする対象サーバのFQDNで64文字以内 半角英数字、"."、"-"のみ使用可能。 また、先頭と末尾に"."と"-"は使用不可
Email	本認証局では使用しないでください。	×	
鍵長			
RSA 2048bit ECDSA 384bit			

○・・・必須　×・・・入力不可　△・・・省略可

注意：証明書の更新を行う場合は、先に1-7をご確認ください。

1-3. 鍵ペアの生成とCSRの作成

1-3-1. キーストアの生成

以下に鍵ペアの生成方法を記述します。

鍵ペアの作成

1. キーストアを作成するため、以下のコマンドを実施してください。-dnameの引数に関しましては、「1-2.事前準備」の「DNルール」に従いDN情報を入力してください。

   RSA鍵の場合

   ```
   $keytool -genkey -alias <tomcat> -keyalg RSA -keysize 2048 -keystore <server_yyyymmdd.keystore> -dname "<サーバDN> "
   鍵ストア・パスワードを入力してください：<keystore_pass> ←鍵のパスワードを入力。
   <tomcat>の鍵パスワードを入力してください
         （鍵ストア・パスワードと同じ場合にはEnterを押してください）：←鍵ストア・パスワードと同じパスワードにする必要があるため[Enter]を入力。
   ```

   ECDSA鍵の場合

   ```
   $keytool -genkey -alias <tomcat> -keyalg EC -keysize 384 -keystore <server_yyyymmdd.keystore> -dname "<サーバDN> "
   鍵ストア・パスワードを入力してください：<keystore_pass> ←鍵のパスワードを入力。
   <tomcat>の鍵パスワードを入力してください
         （鍵ストア・パスワードと同じ場合にはEnterを押してください）：←鍵ストア・パスワードと同じパスワードにする必要があるため[Enter]を入力。
   ```

   重要：更新時、キーストアファイルを上書きすることの無いように、キーストアファイルに日付等のファイル名をつけることを推奨します。

2. 作成したキーストアの情報は以下のコマンドで確認することができます。

   ```
   $keytool -list -v -keystore < server_yyyymmdd.keystore >
   キーストアのパスワードを入力してください: <keystore_pass> ←鍵のパスワードを入力。

   鍵ストア・タイプ: JKS
   ・・・
   別名: tomcat
   ・・・
   所有者: CN=www.nii.ac.jp, O=National Institute of Informatics, L=Chiyoda-ku, ST=Tokyo, C=JP
   発行者: CN=www.nii.ac.jp, O=National Institute of Informatics, L=Chiyoda-ku, ST=Tokyo, C=JP
   ```

3. 作成したキーストアファイルを保存します。バックアップは外部媒体等に保存し、安全な場所に保管してください。
   キーストアファイルの中の私有鍵を利用すれば、お使いのウェブ・サーバがSSL/TLS で保護して送受信したデータを、解読することができてしまいます。
   従って保存するキーストアファイルへのアクセス権は利用管理者自身とSSL/TLS サーバのプロセス等必要最小限になるよう設定してください。
   またバックアップを保存した外部媒体等も利用管理者のみまたは同じ権限のある方のみ利用できる場所へ保管してください。
   また、キーストアファイル作成時のパスワードの管理も、確実に行ってください。キーストアファイルの紛失、パスワード忘れ等が発生した場合、証明書のインストールが行えなくなります。
   この場合、新たに証明書を申請しなおしていただくことになりますので、ご注意ください。

1-3-2. CSRの生成

キーストアが作成されたことを確認後、CSRを生成します。

CSRの作成

1. 次のコマンドを入力し、CSRの作成を開始してください。パスフレーズの入力が求められますので、「1-3-1 キーストアの生成」の手順1で作成したキーストアのパスワードを入力してください。

   RSA鍵の場合
   コマンドでは、署名アルゴリズムSHA2でCSRを作成し、「servername.csr」（ファイル名は任意）というファイル名で保存することを示しています。

   ```
   $keytool -certreq -sigalg SHA256withRSA -alias <tomcat> -file <servername.csr> -
   keystore
   <server_yyyymmdd.keystore>
   鍵ストア・パスワードを入力してください：<keystore_pass>
   ```

   「-sigalg SHA256withRSA」:署名アルゴリズムを示すオプション。
   署名アルゴリズムSHA1でCSRを作成する場合は、「-sigalg SHA1withRSA」に置き換えてください。

   ECDSA鍵の場合
   コマンドでは、署名アルゴリズムecdsa-with-SHA256でCSRを作成し、「servername.csr」（ファイル名は任意）というファイル名で保存することを示しています。

   ```
   $keytool -certreq -sigalg SHA256withECDSA -alias <tomcat> -file <servername.csr> -
   keystore
   <server_yyyymmdd.keystore>
   鍵ストア・パスワードを入力してください：<keystore_pass>
   ```

   「-sigalg SHA256withECDSA」:署名アルゴリズムを示すオプション。
   署名アルゴリズムecdsa-with-SHA384でCSRを作成する場合は、「-sigalg SHA384withECDSA」に置き換えてください。

2. パスワードの入力が成功するとCSR が生成され、要求された情報の入力が完了すると CSR が生成され、servername.csrに保存されます。なお、このファイルもバックアップを保存して証明書を受領するまでは別途保管することをお勧めします。

   ```
   ----BEGIN CERTIFICATE REQUEST----
   MIIBhDCB7gIBADBFMQswCQYDVQQGEwJKUDEQMA4GA1UEBxMHQWNhZGVtZTEMMAo
   G
   例
   Um0E3vq8Ajg=
   ----END CERTIFICATE REQUEST----
   ```

3. 以下のコマンドを入力することにより、CSRの内容を確認することができます。

RSA鍵で作成したCSRの場合

```
$ openssl req -noout -text -in servername.csr
Certificate Request:
    Data:
        Version: 0 (0x0)
        Subject: C=JP, ST=Tokyo, L=Chiyoda-ku, O=National Institute of Informatics, CN=www.nii.ac.jp ←CSR生成時に入力したDNと一致していることを確認してください。
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            Public Key: (2048 bit) ←鍵長が2048bitであることを確認してください。
                Modulus:
                    00:c9:0e:99:5c:8a:4a:e3:b2:e2:0d:3d:60:4d:30:
                            :
                           例
                            :
                    ca:2e:56:f7:66:bd:01:44:ea:f3:ca:d2:f6:e0:5e:
                    6c:57:4b:65:e4:e7:f7:ca:dd
                Exponent: 65537 (0x10001)
        Attributes:
            a0:00
    Signature Algorithm: sha256WithRSAEncryption←　CSR生成時に指定した署名アルゴリズムで あることを確認してください。
                                                  署名アルゴリズムにSHA1を指定した場合は「sha1WithRSAEncryption」と表示されます。

         88:44:e5:27:06:02:ec:85:6c:29:6a:0f:a3:92:87:4e:e2:f1:
                                :
                               例
```

ECDSA鍵で作成したCSRの場合

```
$ openssl req -noout -text -in servername.csr
Certificate Request:
    Data:
        Version: 0 (0x0)
        Subject: C=JP, ST=Tokyo, L=Chiyoda-ku, O=National Institute of Informatics, CN=www.nii.ac.jp ←CSR生成時に入力したDNと一致していることを確認してください。
        Subject Public Key Info:
            Public Key Algorithm: id-ecPublicKey
            Public Key: (384 bit) ←鍵長が384bitであることを確認してください。
                Modulus:
                    00:c9:0e:99:5c:8a:4a:e3:b2:e2:0d:3d:60:4d:30:
                            :
                           例
                            :
                    ca:2e:56:f7:66:bd:01:44:ea:f3:ca:d2:f6:e0:5e:
                    6c:57:4b:65:e4:e7:f7:ca:dd
                ASN1 OID: secp384r1
                NIST CURVE: P-384
        Attributes:
        Requested Extensions:
            X509v3 Subject Key Identifier:
                98:5A:D9:7B:3C:5D:4E:C1:62:8C:5F:2D:89:1A:B3:DC:F7:6C:1C:E2
    Signature Algorithm: ecdsa-with-SHA256←　CSR生成時に指定した署名アルゴリズムで あることを確認してください。
                                            署名アルゴリズムにSHA384withECDSAを指定した場合は「ecdsa-with-SHA384」と表示されます。

         88:44:e5:27:06:02:ec:85:6c:29:6a:0f:a3:92:87:4e:e2:f1:
                                :
                               例
```

1-4. 証明書の申請から取得まで

CSRを作成しましたら登録担当者へ送付するための証明書発行申請TSVファイルを作成し申請します。証明書発行申請TSVファイルの作成方法、申請方法等につきましては、「**証明書自動発行支援システム操作手順書(利用管理者用)**」をご確認ください。
TSVファイル作成用Webアプリケーション（TSVツール）を提供しておりますので、ご利用ください
証明書の発行が完了すると、本システムより以下のメールが送信されます。メール本文に記載された証明書取得URLにアクセスし、証明書の取得を実施してください。

証明書取得URLの通知

【件名】
Webサーバ証明書発行受付通知

・・・・・

#以下に証明書の取得先が記述されています。
貴機関の登録担当者経由で発行申請をいただきましたサーバ証明書を配付いたします。
本日から1ヶ月以内に以下の証明書取得URLへアクセスし、サーバ証明書の取得を行ってください。
証明書取得URL：https://scia.secomtrust.net/～　←左記URLにアクセスし証明書の取得を行ってください。

・・・・・

1-5. 証明書のインストール

本章ではTomcat(JavaKeytool)への証明書のインストール方法について記述します。

1-5-1. 事前準備

事前準備として、サーバ証明書、中間CA証明書、ルートCA証明書を取得してください。

前提条件

1. サーバ証明書を準備します。「1-4.証明書の申請から取得まで」で受領したサーバ証明書をserver.cerという名前で保存してください。

2. 中間CA証明書を準備します。
   次のURLにアクセスすることでリポジトリにアクセスすることが可能です。

   ●リポジトリ（証明書の発行日時が2020年12月25日0時以降の場合）：https://repo1.secomtrust.net/sppca/nii/odca4/index.html
   サーバー証明書 RSA認証局　中間CA証明書
   「NII Open Domain CA - G7 RSA(SC Organization Validation CA) CA証明書(nii-odca4g7rsa.cer)」
   サーバー証明書 ECC認証局　中間CA証明書
   「NII Open Domain CA - G7 ECC(SC Organization Validation CA) CA証明書(nii-odca4g7ecc.cer)」

   ●リポジトリ（証明書の発行日時が2020年12月25日0時以前の場合）：https://repo1.secomtrust.net/sppca/nii/odca3/index.html
   SHA-2認証局CT対応版サーバ証明書
   「国立情報学研究所 オープンドメイン SHA-2認証局 CT対応版 CA証明書(nii-odca3sha2ct.cer)」
   ECC認証局サーバ証明書
   「国立情報学研究所 オープンドメイン ECC認証局 CA証明書(nii-odca3ecdsa201903.cer)」

   【サーバー証明書(ecdsa-with-SHA384)をインストールする場合】
   ECC認証局 中間CA証明書 をnii-odca3ecdsa.cerという名前で保存したと仮定して以降記載します。

3. ルートCA証明書を準備します。
   【サーバー証明書(sha256WithRSAEncryption)インストールする場合】
   以下URLよりSecurity Communication RootCA2 証明書 - Security Communication RootCA2 Certificateを取得して、SCRoot2ca.cerという名前で場所に保存してください。本ファイルはデフォルトではSCRoot2ca.cerという名前でダウンロードされます。
   リポジトリ：https://repository.secomtrust.net/SC-Root2/index.html

   【サーバー証明書(ecdsa-with-SHA384)インストールする場合】
   以下URLよりSecurity Communication ECC RootCA1 証明書 - Security Communication ECC RootCA1 Certificateを取得して、SCECCRoot1ca.cerという名前で場所に保存してください。本ファイルはデフォルトではSCECCRoot1ca.cerという名前でダウンロードされます。
   リポジトリ：https://repository.secomtrust.net/SC-ECC-Root1/index.html

1-5-2. ルートCA証明書のインストール

以下の手順に従って、ルートCA証明書のインストールを行ってください。

ルートCA証明書のインストール

「1-5-1.事前準備」で取得したルートCA証明書をキーストアにインストールしてください。

SHA2のルートCA証明書の場合

```
$ keytool -import -alias root -keystore < server_yyyymmdd.keystore > -file SCRoot2ca.cer
←証明書のフィンガープリントが表示されるので、リポジトリに公開されたフィンガープリントと等しいことを確認してください
Enter keystore password: <keystore_pass> ←キーストアのパスワードを入力してください
Trust this certificate? [no]: Y ←Yもしくはyesと入力してください
Certificate was added to keystore
```

証明書のフィンガープリントが「SHA 1 :5f 3b 8c f2 f8 10 b3 7d 78 b4 ce ec 19 19 c3 73 34 b9 c7 74]であることを確認してください。

ECDSAのルートCA証明書の場合

```
$ keytool -import -alias root -keystore < server_yyyymmdd.keystore > -file SCECCRoot1ca.cer
←証明書のフィンガープリントが表示されるので、リポジトリに公開されたフィンガープリントと等しいことを確認してください
Enter keystore password: <keystore_pass> ←キーストアのパスワードを入力してください
Trust this certificate? [no]: Y ←Yもしくはyesと入力してください
Certificate was added to keystore
```

証明書のフィンガープリントが「SHA 1 :b8 0e 26 a9 bf d2 b2 3b c0 ef 46 c9 ba c7 bb f6 1d 0d 41 41]であることを確認してください。

1-5-3. 中間CA証明書のインストール

以下の手順に従って、中間CA証明書のインストールを行ってください。

中間CA証明書のインストール

「1-5-1.事前準備」で取得した中間CA証明書をキーストアにインストールしてください。

SHA-2認証局 中間CA証明書の場合

```
$ keytool -import -alias niica3 -keystore < server_yyyymmdd.keystore > -file nii-odca3sha2ct.cer

Enter keystore password: <keystore_pass> ←キーストアのパスワードを入力してください
Trust this certificate? [no]: Y ←Yもしくはyesと入力してください
Certificate was added to keystore
```

ECC認証局 中間CA証明書の場合

```
$ keytool -import -alias niica3 -keystore < server_yyyymmdd.keystore > -file nii-odca3ecdsa.cer

Enter keystore password: <keystore_pass> ←キーストアのパスワードを入力してください
Trust this certificate? [no]: Y ←Yもしくはyesと入力してください
Certificate was added to keystore
```

1-5-4. サーバ証明書のインストール

サーバ証明書をインストールする場合は以下の手続きを実施してください。

サーバ証明書のインストール

「1-5-1.事前準備」で取得したサーバ証明書をキーストアにインストールしてください。

```
$keytool -import -alias <tomcat> -keystore <server_yyyymmdd.keystore> -file server.cer
Enter keystore password: <keystore_pass> ←キーストアのパスワードを入力してください
Trust this certificate? [no]: Y ←Yもしくはyesと入力してください
Certificate was added to keystore
```

1-6. Tomcatの設定変更

本章ではTomcatに証明書を適用するための設定方法について記述します。

Tomcatの設定変更

証明書のインストール終了後、「1-1. 前提条件」で記述したserver.xmlファイルの編集を行ってください。証明書の更新を行った場合は新たに作成したキーストアファイルのファイルパスを**KeystoreFile**に、新たに作成したキーストアファイルのパスワードを**KeystorePass**に設定してください。

【7（7.0.52以前）系の場合】

```
<Connector port="443" ←SSL通信を行うポート番号を指定
protocol="HTTP/1.1" SSLEnabled="true"
maxThreads="150" scheme="https" secure="true"
keystoreFile="[< server_yyyymmdd.keystore >までのパス]" keystorePass="<keystore_pass>"

↑キーストアファイルまでのパス                    ↑キーストアのパスワード
clientAuth="false" sslProtocol="TLS"/>
```

【Tomcat7（7.0.53以降）、8、8.5系の場合】

```
<Connector port="443" ←SSL通信を行うポート番号を指定
protocol="org.apache.coyote.http11.Http11Protocol"
maxThreads="150" SSLEnabled="true" scheme="https" secure="true"
keystoreFile="[< server_yyyymmdd.keystore >までのパス]" keystorePass="<keystore_pass>"

↑キーストアファイルまでのパス                    ↑キーストアのパスワード
clientAuth="false" sslProtocol="TLS"/>
```

1-7. 証明書の更新

証明書の更新時はキーストアを新たに作成して頂く必要があります。本マニュアルに従い、キーストアを作成後、「1-6.Tomcatの設定変更」の**keystore File**、**KeystorePass**の値を新たに作成したキーストアに合わせて変更してください。

1-8. 起動確認

本章ではインストールした証明書によるSSL通信に問題がないか確認する方法を記述します。

証明書の反映・確認

1. Tomcatを再起動し、変更した設定を反映させます。

   ```
   $ /sbin/service tomcat7 stop
   $ /sbin/service tomcat7 start
   ```

   ※Tomcatの起動と停止は、ご使用の環境によって大きく異なりますので、適宜読み替えてください。

2. ブラウザ経由で、当該のサーバへアクセスし、SSL通信に問題がないことを確認してください。