Readme
RSA SecurID
Software Token Converter 2.5

November 15, 2010

Introduction

RSA SecurID Software Token Converter is a command line utility that converts a software token file (SDTID file) from XML format to a Compressed Token Format consisting of an 81-digit numeric string. The token data can then be provisioned to an RSA SecurID application running on a mobile device.

The mechanisms available for delivering the converted token data to the device depend on the device platform. For instructions, see the documentation for your RSA SecurID software token application.

You can use RSA SecurID Software Token Converter 2.5 with the following applications:

- RSA SecurID Software Token for Android
- RSA SecurID Software Token for iPhone Devices
- Verizon Software Token for RSA SecurID
- RSA SecurID Software Token for Windows Mobile
- RSA SecurID Token for the Java ME Platform

The Token Converter supports 128-bit (AES) software tokens. It does not support 64-bit (SID) tokens.

The Token Converter runs on Windows operating systems and on Red Hat Linux. The Linux version runs on Red Hat Enterprise Server 4 (RH ES 4) or later.

Before You Begin

Before you use the Token Converter, keep the following in mind:

- You can convert only one token at a time and only one token per token file.
- If you bound the token in RSA Authentication Manager, the binding information is transmitted as part of the token data.

Command Line Syntax

The Token Converter command syntax is as follows. The SDTID filename (*filename*.sdtid) is required.

On Windows:

```
C:\path_name\TokenConverter filename.sdtid options
```

On Linux:

```
/path_name/TokenConverter filename.sdtid options
```

Options

The command line options are described in the following table.

Option/ Parameter	Description
-android	Outputs a numeric string in the format required by the SecurID for Android application, version 1.0, as follows: `http://rsa.com/android/token/ctf?ctfData=nnnnnnnn` where n represents a digit in the 81-character numeric string. When -android is specified, the -f option and -v option are ignored, and the converted token file is always formatted as -v 2.
-f	Recommended for the Windows Mobile and Java ME applications. Outputs a numeric string consisting of blocks of five digits separated by delimiters (hyphens), for example, nnnnn-nnnnn-nnnnn, and so on. The hyphens are intended to assist users who have a device that requires manually entering the numeric string. If you do not use the -f option, the output is a numeric string without delimiters. **Note:** Do not use the -f option with the Android, iPhone, or Verizon software token applications.
-iphone	Outputs a numeric string in the format required by the SecurID for iPhone application, version 1.1, as follows: `com.rsa.securid.iphone://ctf?ctfData=nnnnnnnn` where n represents a digit in the in the 81-character numeric string. When -iphone is specified, the -f option and -v option are ignored, and the converted token file is always formatted as -v 2. **Note:** The -iphone option is deprecated and should only be used to support legacy scripts that were written against previous versions of the Token Converter.
-mobile	Currently used with the SecurID for iPhone application, version 1.2 or later. Outputs a numeric string in the format required by the application, as follows: `com.rsa.securid://ctf?ctfData=nnnnnnnn` where n represents a digit in the in the 81-character numeric string. When -mobile is specified, the -f option and -v option are ignored, and the converted token file is always formatted as -v 2. **Note:** Future RSA SecurID software token applications may support -mobile to allow you to use a single setting when provisioning tokens for multiple applications.
-o *filename*	Outputs a text file (ASCII) containing an 81-character numeric string. If you enter a filename that already exists, the Token Converter overwrites the original file. If you do not use this option, the numeric string is written to the screen.
-p *password*	Specifies the password used to decrypt the XML token record and re-encrypt the numeric string. Required to convert password-protected tokens. The converted token retains the password, and the user is prompted to enter it to complete the token import.

Option/ Parameter	Description
-v *value*	Required for converting tokens used with the Windows Mobile and Java ME applications. Specifies the format of the converted token file. Valid values are 1 and 2 (default). Choose one of the following: • Use the -v 1 option when converting token files for version 2.2 of the Windows Mobile or Java ME applications. The -v 1 option does not support device binding. • Use the -v 2 option when converting token files for version 2.3 of the Windows Mobile or Java ME applications. The -v 2 option is required if you bind the token to a specific device in Authentication Manager.
-version	Displays the Token Converter version on the screen. Used to assist RSA Customer Support in troubleshooting. For example, the token file may not be converted properly if you use an earlier version of the Token Converter that does not support your software token application.

Using the Token Converter on Microsoft Windows

To use the Token Converter on Windows:

1. Save the Token Converter kit to a directory on your computer. Extract the Token Converter executable (**TokenConverter.exe**) and the DLL file (**sdti2tsf.dll**) to the same directory.
2. Place the token files (SDTID files) that you want to convert into the same directory where you saved the executable and DLL.
3. Open a command line, and change to the directory where you saved the executable and DLL.
4. Enter a command using syntax similar to the following:

   `C:\`*`path_name`*`\TokenConverter` *`filename`*`.sdtid` *`options`*

Windows Command Examples

Android 1.0

The following command uses the -android option to convert a password-protected token file. The -android option is currently required with the SecurID for Android application, 1.0.

```
C:\path_name\TokenConverter user2-passwordtoken.sdtid -p t0kenpw1 -android -o
tokenfile.txt
```

The converted token data looks similar to the following:

```
http://rsa.com/android/token/ctf?ctfData=200002068164720663136011170432774461076477
1646324562010261721150440460627167126 50
```

iPhone 1.2 or Later

The following command uses the -mobile option to convert a password-protected token file. The -mobile option is currently used with the SecurID for iPhone application, 1.2 or later.

```
C:\path_name\TokenConverter user2-passwordtoken.sdtid -p t0kenpw1 -mobile -o
tokenfile.txt
```

The converted token data looks similar to the following:

```
com.rsa.securid://ctf?ctfData=200002068164720663136011170432774461076477164632456201
026172115044046062716712650
```

iPhone 1.1

The following command uses the -iphone option to convert a password-protected token file. Use this option only to support legacy scripts for the SecurID for iPhone application that were written against previous versions of the Token Converter.

```
C:\path_name\TokenConverter user2-passwordtoken.sdtid -p t0kenpw1 -iphone -o
tokenfile.txt
```

The converted token data looks similar to the following:

```
com.rsa.securid.iphone://ctf?ctfData=200002068164720663136011170432774461076477164632
456201026172115044046062716712650
```

Windows Mobile and Java ME

The following command converts the token file to a text file containing the numeric string with delimiters. Use the -f option with the Windows Mobile or Java ME applications if the application requires the user to enter the string, but the device does not support copy and paste. Instruct users not to enter the hyphens.

```
C:\path_name\TokenConverter user2-50.sdtid -f -o tokenfile.txt
```

The following command converts a password-protected token file to a text file containing the numeric string with delimiters. Use the -f option with the Windows Mobile or Java ME applications if the application requires the user to enter the string, but the device does not support copy and paste. Instruct users not to enter the hyphens. The numeric string format is version 1, which does not support device binding and is required with version 2.2 of the Windows Mobile or Java ME applications.

```
C:\path_name\TokenConverter user2-passwordtoken.sdtid -v 1 -p t0kenpw1 -f -o
tokenfile.txt
```

Additional Windows Examples

The following command converts the token file to a text file containing the numeric string without delimiters. Use this command if the application requires the user to import the token manually, and the device has a copy/paste function. The user can copy the number from an e-mail and paste it into the application.

```
C:\path_name\TokenConverter user2-50.sdtid -o tokenfile.txt
```

The following command converts a password-protected token file to a text file containing the numeric string with no delimiters.

```
C:\path_name\TokenConverter user2-passwordtoken.sdtid -p t0kenpw1 -o tokenfile.txt
```

The following command displays the version of the Token Converter.

```
C:\path_name\TokenConverter -version
```

Using the Token Converter on Red Hat Linux

To use the Token Converter on Linux:

1. Save the Token Converter executable (**TokenConverter**) to a directory.
2. Open a command line, and change to the directory where you saved the executable.
3. Enter a command using syntax similar to the following:

   /*path_name*/TokenConverter *filename*.sdtid *options*

Linux Command Examples

Android

The following command uses the -android option to convert a password-protected token file. The -android option is currently required with the SecurID for Android application, 1.0.

```
/path_name/TokenConverter user2-passwordtoken.sdtid -p t0kenpw1 -android -o
tokenfile.txt
```

The converted token data looks similar to the following:

```
http://rsa.com/android/token/ctf?ctfData=200002068164720663136011170432774461076477
1646324562010261721150440460627167126 50
```

iPhone 1.2 or Later

The following command uses the -mobile option to convert a password-protected token file. The -mobile option is currently used with the SecurID for iPhone application, 1.2 or later.

```
/path_name/TokenConverter user2-passwordtoken.sdtid -p t0kenpw1 -mobile -o tokenfile
```

The converted token data looks similar to the following:

```
com.rsa.securid://ctf?ctfData=200002068164720663136011170432774461076477164632456201
026172115044046062716712650
```

iPhone 1.1

The following command uses the -iphone option to convert a password-protected token file. Use this option only to support legacy scripts for the SecurID for iPhone application that were written against previous versions of the Token Converter.

```
/path_name/TokenConverter user2-passwordtoken.sdtid -p t0kenpw1 -iphone -o tokenfile
```

The text file contains token data similar to the following:

```
com.rsa.securid.iphone://ctf?ctfData=200002068164720663136011170432774461076477164632
45620102617211504404606271671265 0
```

Windows Mobile and Java ME

The following command converts the token file to a text file containing the numeric string with delimiters. Use the -f option with the Windows Mobile or Java ME applications if the application requires the user to enter the string, but the device does not support copy and paste. Instruct users not to enter the hyphens.

```
/path_name/TokenConverter user2_50.sdtid -f -o tokenfile
```

The following command converts a password-protected token file to a text file containing the numeric string with delimiters. Use the -f option with the Windows Mobile or Java ME applications if the application requires the user to enter the string, but the device does not support copy and paste. Instruct users not to enter the hyphens. The numeric string format is version 1, which does not support device binding and is required with version 2.2 of the Windows Mobile or Java ME applications.

```
/path_name/TokenConverter user2_passwordtoken.sdtid -v 1 -p t0kenpw1 -f -o tokenfile
```

Additional Linux Examples

The following command converts the token file to an 81-digit numeric string with no delimiters, and creates a text file containing the numeric string. Use this command if the application requires the user to import the token manually, and the device has a copy/paste function. The user can copy the number from an e-mail and paste it into the application.

```
/path_name/TokenConverter user2_50.sdtid -o tokenfile
```

The following command converts a password-protected token to an 81-digit numeric string with no delimiters, and creates a text file containing the numeric string.

```
/path_name/TokenConverter user2_passwordtoken.sdtid -p t0kenpw1 -o tokenfile
```

The following command displays the version of the Token Converter.

```
/path_name/TokenConverter -version
```

Exit Codes

You can use RSA SecurID Software Token Converter 2.5 within your own scripts to do additional processing, such as bulk conversion of tokens or, for the Android or iPhone applications, inserting the generated URL into an e-mail template. The following table lists the exit codes that the Token Converter returns to support scripting.

Exit Code	Meaning
0	Token converted properly
1	Invalid command line parameter - output file
2	Invalid command line parameter - password
3	Invalid command line parameter - version
4	Invalid command line parameter - insufficient arguments
5	Invalid command line parameter - Invalid arguments
6	Record is not a software token
7	Correct password needed
8	Parse failed
9	General ctf parse failure
10	Error opening output file

Known Issues

Token cannot be converted if SDTID file contains double-byte characters in UserFirstName, UserLastName, or UserLogin fields

Tracking Number: 119824
Problem: The Token Converter cannot convert SDTID files that contain double-byte characters in the token record's **UserFirstName**, **UserLastName**, and **UserLogin** fields. A double-byte character set contains characters encoded in two bytes. This typically occurs with Asian languages. If any of these fields contain double-byte characters, the token is not converted, and the user receives the following message: "Error parsing XML for the following reason: Failed to parse token record." However, SDTID files that contain double-byte characters in the **Nickname** field are properly converted.
Solution: This is a limitation in RSA Authentication Manager 7.1. To work around the problem, edit the user's first name, last name, and User ID in Authentication Manager to remove double-byte characters, and then issue the token as an SDTID file. If you cannot modify the user in Authentication Manager (read-only identity source), create a dummy internal user, assign the token to the user, issue the software token, and keep the SDTID file. Unassign the token from the dummy user, and assign it to the user with the double-byte characters. Do not re-issue the token.

Getting Support and Service

RSA SecurCare Online	**https://knowledge.rsasecurity.com**
Customer Support Information	**www.rsa.com/support**
RSA Secured Partner Solutions Directory	**www.rsasecured.com**

Trademarks