VincSCalc
User Manual

Author:	Vincenzo Sambito VincSCalc@gmail.com	
Revision:	20150903 draft	
Related Software Release	0.2.x.x.x beta	
File	VincSCalc User Manual 20151115 Sw Rel. 0.2.x.x.x - ENG	
Revision history	**Revision date**	**Description**
	xxth Yyy, 2015	Draft

Index

1. User License for VincSCalc software

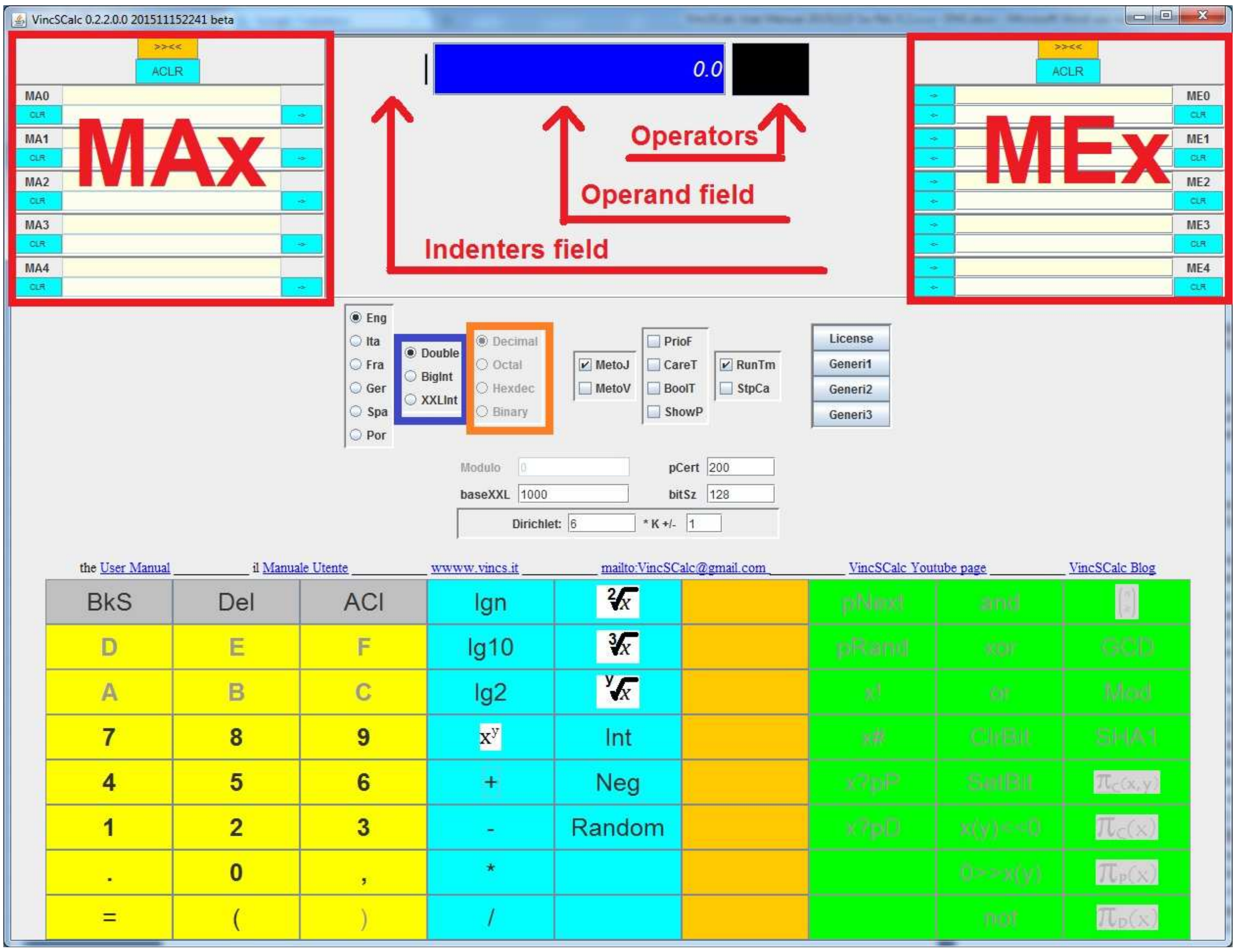
We here just remind that the use of the software is underling to the acceptance of some conditions. The only valid text that is valid as contract with the user is what reported when the user launch the software for the first time and that might be recalled back whenever the user would like having so the chance to confirm, refuse or keep as it is. Once accepted the license, terms of use and distribution, the choice is stored in the computer. Next time that it will be launched, the software will be executed straight to the application window.

2. Features

2.1 Introducing the VincSCalc

As easily you can guess, the VincSCalc is primarily a *calculator* which aim is to give the same features in the average of such kind of software. The user can take advantage in daily use of the calculator.

Differently from many other calculators, the VincSCalc is a multi language, multi data type, multi base (the numeric base as the number is shown and input) calculator. More than this the VincSCalc has many other not common features that will be illustrated later on.



[✓ Eng] Ita Fra Ger Spa Por

[✓ Double] BigInt XXLInt

[✓ Decimal] Octal Hexdec Binary

[✓ MetoJ] MetoV

PrioF CareT BoolT ShowP

[✓ RunTm] StpCa

Modulo 0
baseXXL 1000
pCert 200
bitSz 128
Dirichlet: 6 * K +/- 1

2.2 Key Features

- One of the few calculators with resizable window (up to full screen) in order to allow a wide input/output field for huge numbers (horizontal) and deeply nested calculation (vertical).
- Nesting of calculation kept shown in calculation area that may become vertically scrollable for deeply nested calculation.
- Input/Output operand fields horizontally scrollable to allow arbitrary hugeness of integers (as huge as your hardware will allow)
- Digit Keyboard disabled automatically when data is consistent to avoid involuntary overwrite (only operators or "clears" input are accepted)
- 5 "Memories Absolute" (MAx) on the left side of calculation area - these are used by the calculator in order to give extra output (see as example HMPx functions) - The panel can be horizontally shrinked to permit the widest calculation area as possible.
- 5 "Memories Editable" (MEx) on the right side of calculation area - the user can store results of operation by "input" arrow key in the dedicated panel as well as input directly a value inside the field giving also a "name" to any stored value in order to remind the meaning of such data (just close below the stored value) - Recall the value in the calculation area input it is so a simple as a click on the counter "output" arrow button in the panel. - The panel can be horizontally shrinked to permit the widest calculation area as possible.
- 3 Data Types (Double, BigInt[eger], XXLInt[eger])
- 4 numeric bases (only for integer types - decimal, octal, hexdecimal, binary)
- Calculations can be performed "run time", for a quick response, or also postponed to the "=" operator request to keep track and double check (RunTm checkbox is "run time" by default). - In case of postponed operations, the user could also eventually go back (🡐BkS button) in order to correct his input.
- Integer calculations can be performed in Modulo (settable directly in its field) allowing experiments with modular arithmetic.
- Integers primality test in probabilistic and deterministic methods.
- Integers HowManyPrimes functions to test the Gauss formulas (logarithmic, Integral Logarithmic) or the Riemann's derived formulas

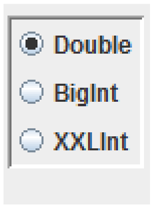
2.3 Software's security

The VincSCalc is written in Java source code. Java offers the advantage of a great emphasis in security matters, embedded in its constantly upgraded technology, in order to avoid deployment of viruses and malicious software. In facts the VincSCalc is a signed application that means that the identity of the software developer has been verified by a Certifications Authority (Symantec.Thawte) and that the software cannot be modified in order to introduce viruses or similar hacking attacks (because it has been encrypted with the same signature).

One of the main advantages of using Java is the availability of the *BigInteger* library (a class derived from Numbers class) particularly suitable for Numbers Theory experiments (such as prime numbers research) and application such as *communication security* (cryptography and similar technology). So the VincSCalc, among classic Double precision data type, has built in the BigInt data type that allows to perform such kind of calculation.

2.4 Number Data Types

The user can select the preferred data type by proper radio buttons provided in the GUI as shown in the pictures here below.



If a section of the calculator keyboard make sense only in a specific Data Type (as example operation among integers for BigInt) this will be enabled/shown when that specific Data Type is selected.

2.4.1 Double precision Data Type

The most common data type for commercial calculators. It offers a limited precision and it uses exponent of 10 (Exx) to extend wideness of numbers. Using the Double precision Data Type, the choice of numeric base (for viewing) is limited to Decimal (see also Chapter about these radio buttons).

2.4.2 BigInt Data Type

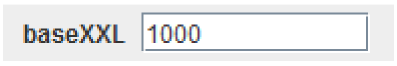
Documentation about this data type is widely available at Oracle's as its class of object is the BigInteger embedded in JDK since v1.1.

Sumarizing:

- operations just among integers
- arbitrary precision (it relies only on hardware capability)
- Java optimized bytecode available for operations
- etc. etc.

2.4.3 XXLInt Data Type

In case of doubts, up to now, this is a proprietary Data Type derived by BigInteger since every single digit of this data type is a BigInteger. The aim of this class of object is to extend the capability of BigInteger data type. Digits are separed by a dot (or comma in the future in order to support other cultures) The maximum value of every digit is represented and can be chosen by …

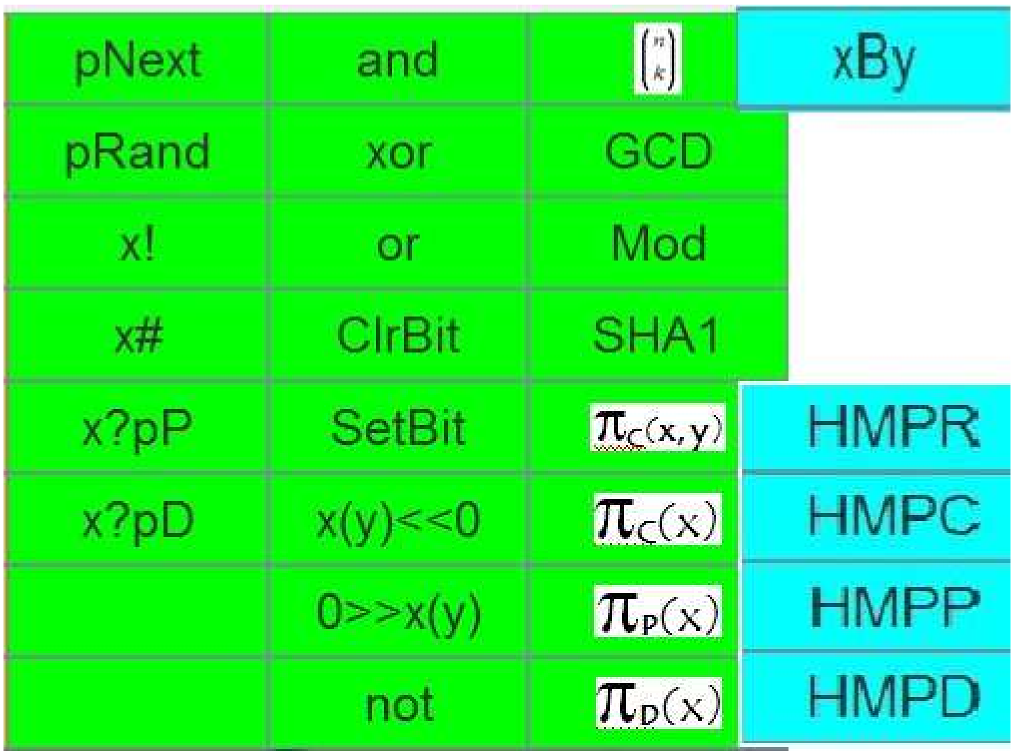

baseXXL 1000

Up to now, for XXLInt data type, only basic operations (+,-,*,/) are supported. Other operations are currently under development.

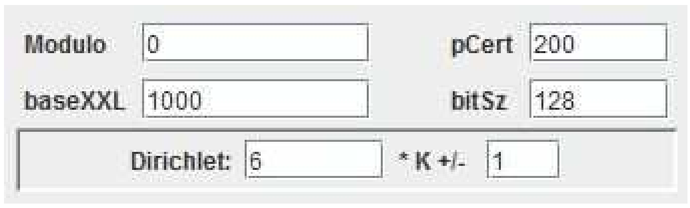
3. Operations with Integers

3.1 Dedicated keyboard and parameters

When the **BigInt** data type is selected (as well as in the coming implementation of XXInt data type), a dedicated portion of keyboard will be shown/enabled with suitable operations and function for such kind of data.



Also some settable parameters are shown/enabled. This parameters can be left as they are in default value or may be changed, by the user, to fit his/her needs.



Modulo: 0
pCert: 200
baseXXL: 1000
bitSz: 128
Dirichlet: 6 * K +/- 1

3.2 Parameters

Shortly explained these parameters one by one:

- **Modulo**: after every operation, the final result will be automatically re-calculated to be equal to **result MOD Modulo**. Very useful working with modular arithmetic.

- **pCert**: is the "certainty" given when using the Java BigIntegers library functions related to determine if a number is a prime number or not. This because, of course, these functions use probabilistic methods to determine it. Taken from Oracle documentation: "`certainty` - a measure of the uncertainty that the caller is willing to tolerate. The probability that the new BigInteger represents a prime number will exceed ($1 - 1/2^{\texttt{certainty}}$). The execution time of this constructor is proportional to the value of this parameter.". Some functions affected by it are so **x?pP** (is x prime probabilistically?), **pNext** (next prime), **HMPP** (How Many Primes probabilistically?). etcetera.

- **bitSz**: is the number of bit size used to create, for example, a random prime number with the function **pRandom** (random prime number). Very useful with cryptography applications.

- **baseXXL**: the boundary of a "digit" in XXLInt data type.

- **Dirichlet**: a couple of parameters (dedicated to the mathematician but not used to prove his theorem) used by **HMPC** (HowManyPrimesCalculated) and **HMPR** (HowManyPrimes[in]Range) functions to determine not only how many primes there are in a given range but also how many primes of those are in the format **z***K+/-**w** (where **z** and **w** are the input parameters). As we know, the famous Dirichlet theorem says that **z** and **w** must not have common factors but, as already told, we don't care (it's not a method to prove his theorem). Note: default value are **z=6** and **w=1** just because it's easy and funny to verify (and also to prove) that all prime numbers are in the format **6***K+/-**1** but also how many of them are in the format *6*K+1* versus how many are in the format *6*K-1*.

3.3 Keyboard

Let's go now to calculator keys explained one by one:

- **pNext**: next prime greater than the number in the operand field.
  Examples: 0 --> 2 , 20 --> 23.
- **pRand**: a random prime number with a bit size given by bitSz parameter.
  Examples (bitSz=8 and so 127<x<256): 191, 167, 157, 163.
- **x!**: the factorial of the given operand.
- **x#**: the primorial of the given operand.
- **x?pP**: is x a prime number? The test is done using a probabilistic method using *pCert* parameter (see above). If yes the operand becomes 1 (true), if not the operand becomes 0 (false).
- **x?pD**: is x a prime number? The test is done using a deterministic method (a sieve). If yes the operand becomes 1 (true), if not the operand becomes 0 (false).
- **and**, **xor**, **or**; bitwise operations among BigInteger operands.
- **CrlBit**: clears the bit indicated by the given second parameter. To obtain the result press =.
  Example: 5 **ClrBit** 2 = 1.
- **SetBit**: sets the bit indicated by the given second parameter. To obtain the result press =.
  Example: 5 **SetBit** 1 = 7.
- **x(y)<<0**; the operand **x** (currently in the operand field) **left**-shifted by **y** *digits* and filled by 0s. Note: the shift is made digit-wise and not bit-wise (that's if you choose Binary data type).
- **x(y)<<0**; the operand **x** (currently in the operand field) **right**-shifted by **y** *digits* and filled by 0s. Note: the shift is made digit-wise and not bit-wise (that's if you choose Binary data type).
- **not**: the operand is sign reverted and subtracted by 1 (differently by **neg**ation).
- **xBy**: the Newton's Binomial coefficient (Pascal's triangle etcetera).
  Example: for the 2nd coefficient in the 3rd row (1,3,3,1) insert 3 **xBy** 1 and you will get 3.
  Example: for the 3rd coefficient in the 4th row (1,4,6,4,1) insert 4 **xBy** 2 and you will get 6.
- **GCD**: the Greater Common Divisor among the two operands. Example: 15 **GCD** 6 = 3.
- **Mod**: the Modulo operator. Example: 11 **Mod** 5 = 1.
- **SHA1**: the Secure Hash Algorithm revision 1.

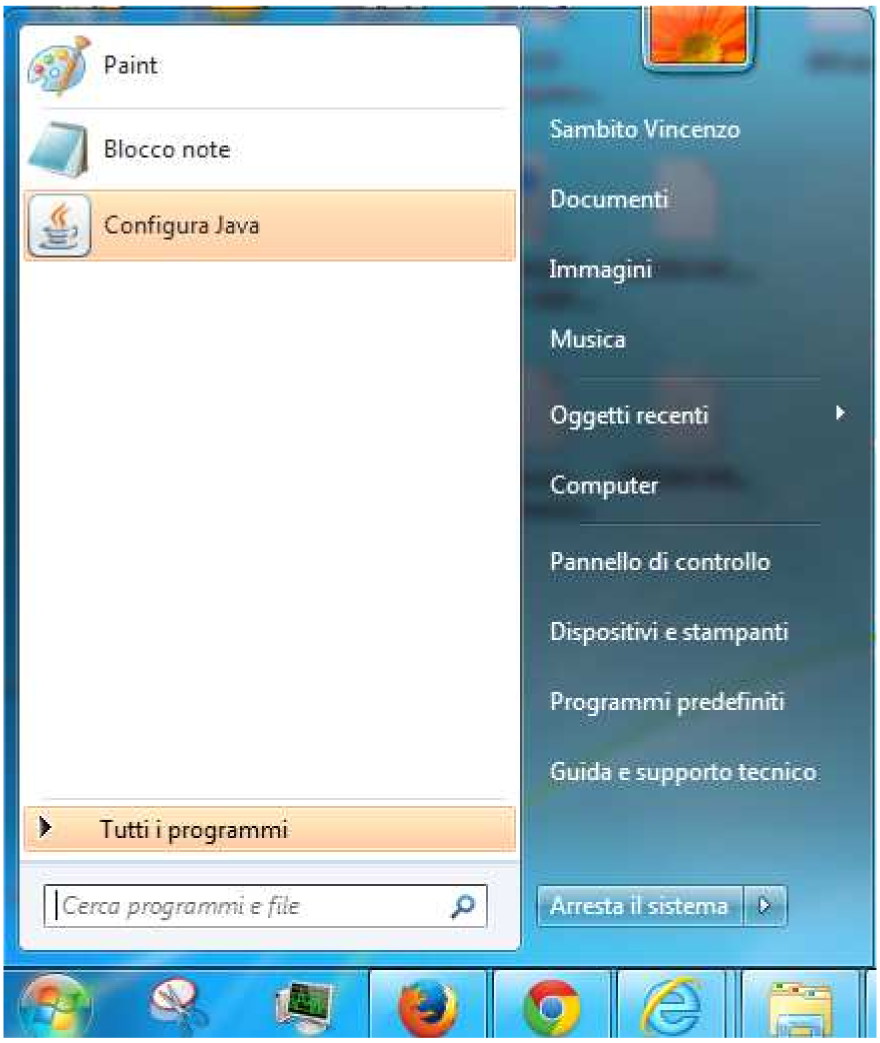
(HowManyPrimes? functions - where π stands for Gauss's function symbol)

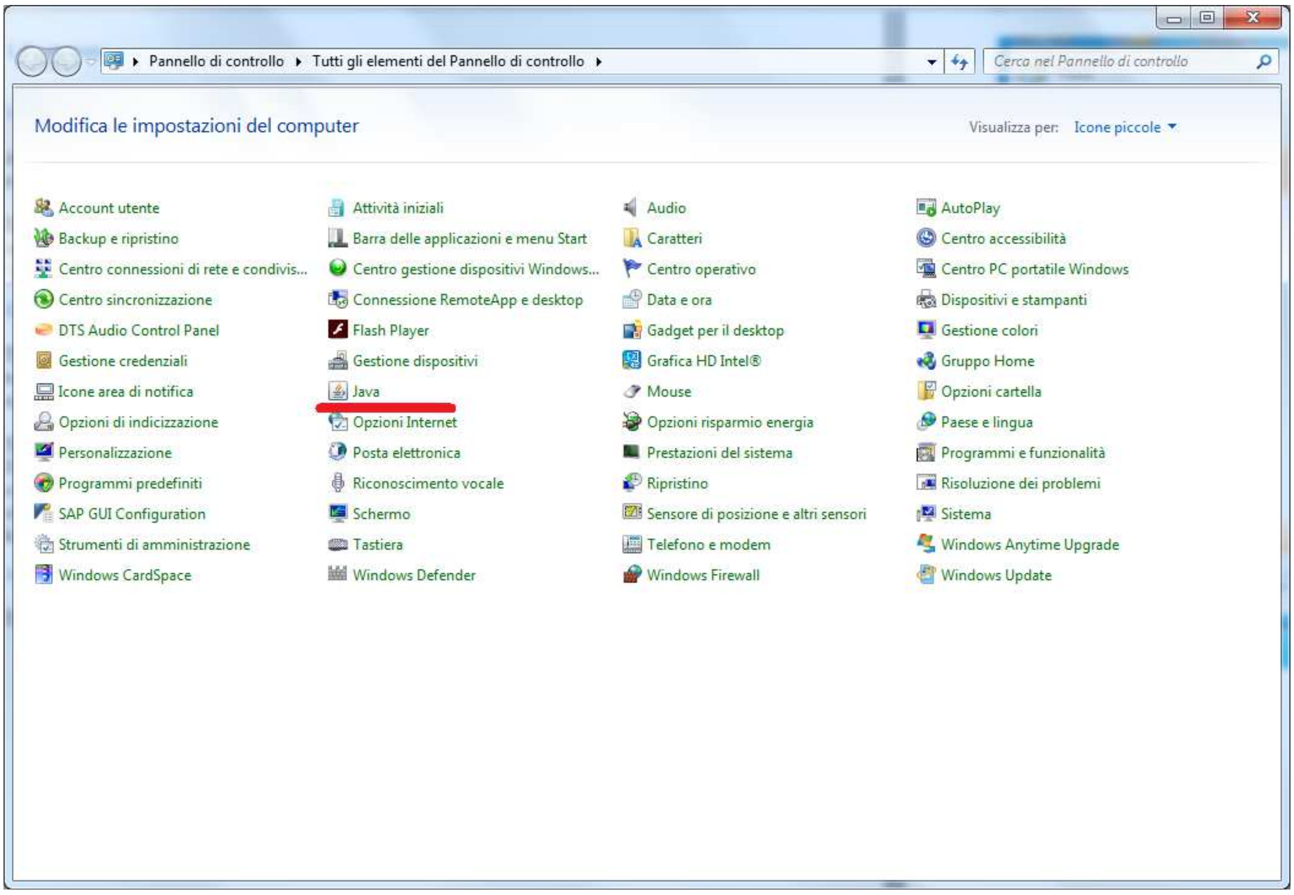
- **HMPR** - $\pi_C(x,y)$: how many primes (calculated) in a given range.
  Note: don't exceed a range of 10000000 numbers (calculation lasts about 30 secs).
  Example: 100 **HMPR** 200 = 21 : there are 21 primes among 100 and 200.
- **HMPC** - $\pi_C(x)$: how many primes (calculated) in a given range (starting from 0).
  Example: 100 **HMPC** ➔ 25 : there are 25 primes among 0 and 100.
- **HMPP** - $\pi_P(x)$: how many primes (counted with probabilistic method) in a given range (starting from 0). Note: don't exceed 1000000 (calculation lasts about 10 secs).
- **HMPD** - $\pi_D(x)$: how many primes (counted with deterministic method - a sieve) in a given range (starting from 0). Note: don't exceed 100000 (calculation lasts about 10 secs).

Note: **HMPR** and **HMPC** functions output is enhanced (in MAx memories block) also with the calculation of "how many primes" are in a specific format (that the user can trim with dedicated "Dirichlet" parameters - see above).

To be continued ….

Appendix


Paint
Blocco note
Configura Java
Tutti i programmi
Cerca programmi e file
Sambito Vincenzo
Documenti
Immagini
Musica
Oggetti recenti
Computer
Pannello di controllo
Dispositivi e stampanti
Programmi predefiniti
Guida e supporto tecnico
Arresta il sistema



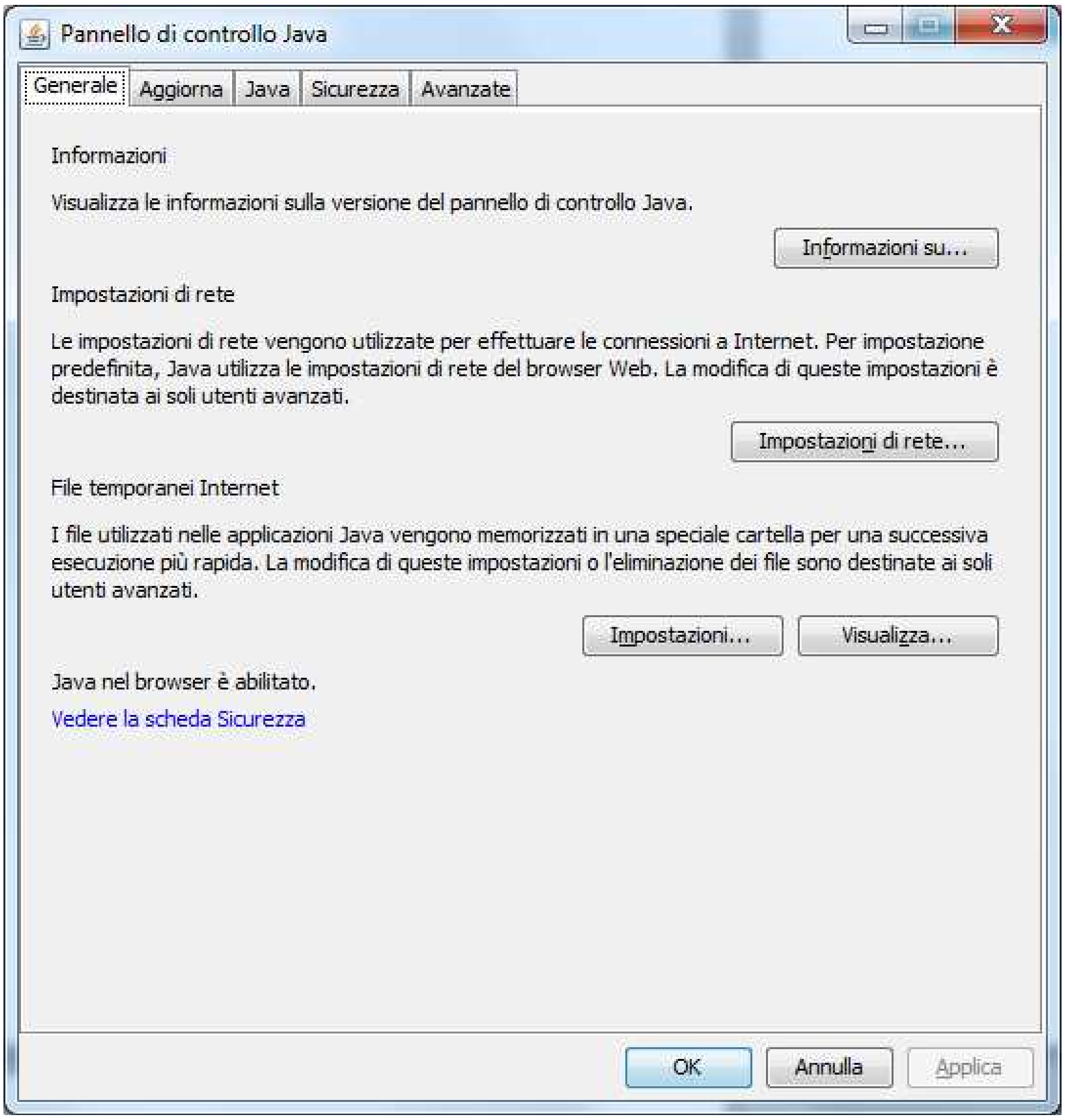
Pannello di controllo ▸ Tutti gli elementi del Pannello di controllo ▸

Cerca nel Pannello di controllo

Modifica le impostazioni del computer

Visualizza per: Icone piccole

Account utente	Attività iniziali	Audio	AutoPlay
Backup e ripristino	Barra delle applicazioni e menu Start	Caratteri	Centro accessibilità
Centro connessioni di rete e condivis...	Centro gestione dispositivi Windows...	Centro operativo	Centro PC portatile Windows
Centro sincronizzazione	Connessione RemoteApp e desktop	Data e ora	Dispositivi e stampanti
DTS Audio Control Panel	Flash Player	Gadget per il desktop	Gestione colori
Gestione credenziali	Gestione dispositivi	Grafica HD Intel®	Gruppo Home
Icone area di notifica	Java	Mouse	Opzioni cartella
Opzioni di indicizzazione	Opzioni Internet	Opzioni risparmio energia	Paese e lingua
Personalizzazione	Posta elettronica	Prestazioni del sistema	Programmi e funzionalità
Programmi predefiniti	Riconoscimento vocale	Ripristino	Risoluzione dei problemi
SAP GUI Configuration	Schermo	Sensore di posizione e altri sensori	Sistema
Strumenti di amministrazione	Tastiera	Telefono e modem	Windows Anytime Upgrade
Windows CardSpace	Windows Defender	Windows Firewall	Windows Update


Pannello di controllo Java

Informazioni

Visualizza le informazioni sulla versione del pannello di controllo Java.

Impostazioni di rete

Le impostazioni di rete vengono utilizzate per effettuare le connessioni a Internet. Per impostazione predefinita, Java utilizza le impostazioni di rete del browser Web. La modifica di queste impostazioni è destinata ai soli utenti avanzati.

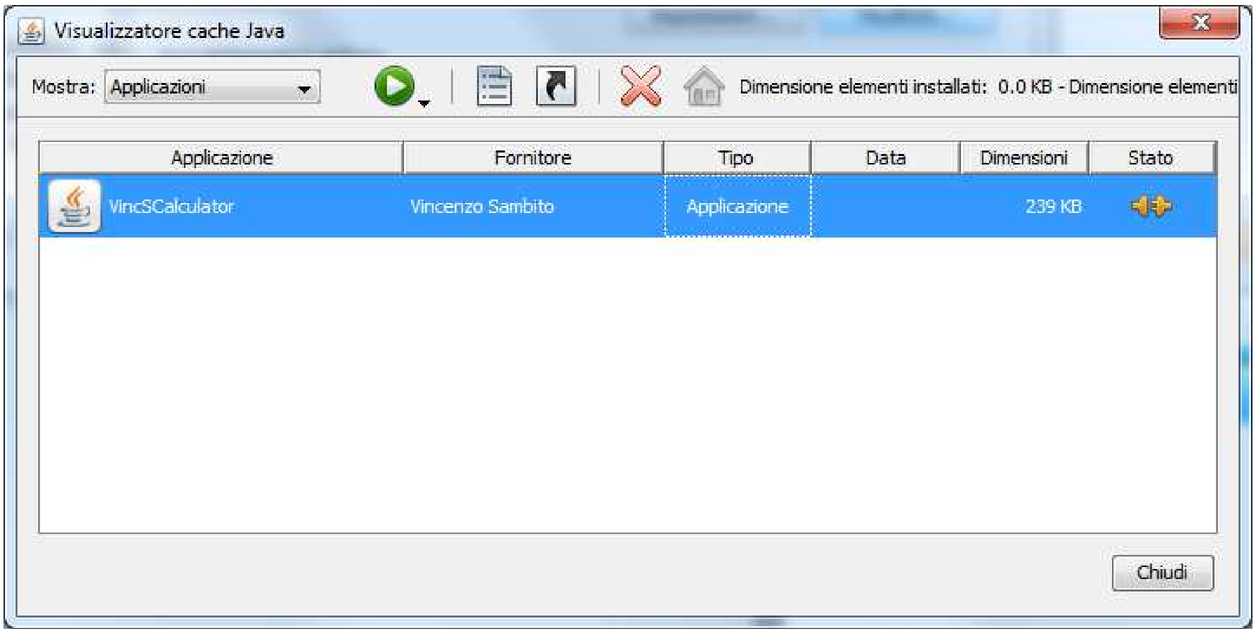
File temporanei Internet

I file utilizzati nelle applicazioni Java vengono memorizzati in una speciale cartella per una successiva esecuzione più rapida. La modifica di queste impostazioni o l'eliminazione dei file sono destinate ai soli utenti avanzati.

Java nel browser è abilitato.

Vedere la scheda Sicurezza

Note: in case View (Visualizza) is not enlighted (cannot be pressed), click before on Apply button.


Visualizzatore cache Java
Mostra: Applicazioni
Dimensione elementi installati: 0.0 KB - Dimensione elementi
Chiudi

Applicazione	Fornitore	Tipo	Data	Dimensioni	Stato
VincSCalculator	Vincenzo Sambito	Applicazione		239 KB	