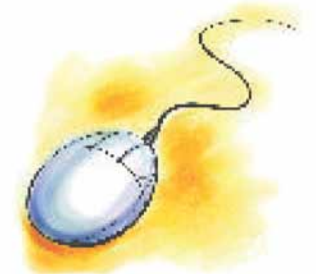


ANNEXURE-I

Binary Codes

It is important to know how data is processed inside a computer. For example, if we press an alphabet key then how that alphabet goes inside the computer? Does the computer understand the language which we speak or use? Certainly not. After all, a computer is made up of various electronic components. The human language or instructions have to be transformed into a form which can be processed by these electronic components.

It is noted that any of the electronic components can possess two physical states. To represent a two-state system, the binary system is the most suited one since it works on only two digits 0 and 1 called bits (short notation of binary digits). When some data or instruction is fed into a computer, each of its electronic components would have one of the two states that it can have. As such the instruction gets converted into a combination of bits (0's and 1's). Thus each alphabet, character or number gets converted into codes containing bits.

We require lots and lots of unique combinations of bits in order to store each and every possible character and number. For example, if we allow only 2 bits, then there are four possible combinations, namely 00, 01, 10, 11. Thus using 2 bits, only four numbers or alphabets can be accommodated. Similarly, using 3 bits, there are 8 ($=2^3$) different combinations. These combinations are not sufficient enough even to store 10 numbers (0-9) for which we require at least 10 different combinations. To meet this requirement, we need at least 4-bit code which can accommodate 16($=2^4$) numbers or characters. If we write these 16 combinations in increasing order and use the first 10 for the numbers 0-9, then the corresponding code is the well known BCD Code.

BCD	CODE
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	-
1011	-
1100	-
1101	-
1110	-
1111	-

In a computer, we need to process not only numbers but also alphabets and many other characters. To accommodate all these, a higher bit code is required. Among such codes, the most popular is the ASCII code (American Standard Code for Information Interchange). It is a 7 bit code that can store $2^7 = 128$ characters. For the ASCII table giving the representation of these 128 combinations, one may refer to Annexure II.

In order to work with Hindi language also, another code ISCII (Indian Standard Code for Information Interchange) was introduced. It is a 8-bit code and therefore gives possibilities to store more characters than ASCII code. While introducing this code, care was taken so that the common characters are represented by the same combination in ASCII as well as ISCII. An additional 0 bit has been added on the left most position to make it 8-bit combination without changing the converted decimal value. The ISCII table has been placed in Annexure III.

This idea of having a 8-bit code has been further extended to generate a 16-bit code which has the possibilities to store characters of all major languages of the world. This makes the computer multilingual. Such a code is termed as UNICODE. The first 128 combinations in UNICODE have been set same as those given by ASCII. One may refer to Annexur IV where Unicode allocation has been given and also in this code, the characters of Devanagari, Bengali, Gujarati and Tamil languages have been given.

Now you will learn how to convert a decimal number (integer or fraction) into binary and vice versa.

Conversion of Decimal Integer to Binary

In order to convert a decimal integer number into binary, the integer number is divided by 2 and the remainder is recorded. The quotient is again divided by 2 and the remainder is again recorded. This process goes on till the quotient becomes 0. At this stage the remainders obtained at each step are written starting from the last one. The resulting number is the desired binary number. Let us convert the decimal number 29 into binary:

- Divide 29 by 2
- The quotient is 14 and the remainder is 1
- Divide the quotient 14 again by 2
- The quotient is 7 and remainder is 0

The entire process is shown below

2	29	1	↑
2	14	0	↑
2	7	1	↑ Remainders
2	3	1	↑
2	1	1	↑
	0		

The binary number 11101 is the binary equivalent of the decimal number 29.

Conversion of Decimal Fractions to Binary

Here the fraction is multiplied by 2 and the carry which is either 0 or 1 is recorded. In the resulting number, the fraction is again multiplied by 2 and the carry is again recorded. The process goes on till the fraction becomes 0. At this stage write all the carries obtained from each step in the forward order. The number so obtained is the desired binary fraction. Let us convert the decimal fraction .59375 into binary fraction. We do as follows:

- .59375 × 2 = .1875 with a carry 1
- .1875 × 2 = .375 with a carry 0
- . 375 × 2 = . 75 with a carry 0
- .75 × 2 = .5 with a carry 1
- .5 × 2 = .00 with a carry 1

Thus the converted binary fraction is .10011. If a decimal number contains both, integral part as well as fraction part then both these parts are converted into binary separately by the procedures as described above. Then the two binary parts are written together yielding the desired binary number. For example, the binary conversion of 29.59375 is 11101.10011.

Conversion from Binary to Decimal

We shall consider the combined case when a binary number has both integral as well as fractional part. The conversion of such a number to decimal goes as follows. First consider the integral part. Going from right to left, each binary digit is multiplied, respectively, by 2^0, 2^1, 2^2, 2^3, and so on. Then all the resulting numbers are added. As for the fractional part, we go from left to right and multiply each binary digit, respectively, by 2^{-1}, 2^{-2}, 2^{-3}, and so on. The resulting numbers are added. Putting together both the parts gives the desired converted decimal number. Let us convert 11101.10011 into decimal. The conversion is described below:

1	1	1	0	1	1	0	0	1	1
2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}

$2^4 \times 1 + 2^3 \times 1 + 2^2 \times 1 + 2^1 \times 0 + 2^0 \times 1 = 16 + 8 + 4 + 1 = 29$

$2^{-1} \times 1 + 2^{-2} \times 0 + 2^{-3} \times 0 + 2^{-4} \times 1 + 2^{-5} \times 1 = ½ + 1/16 + 1/32 = .59375$

Hence the converted decimal number is 29.59375.

Floating Point Representation

In Computer, representation of numbers plays an important role. Normally, a general user is not aware of it. However what goes at the back end is quite technical. Let us try to understand it. Consider number 35. It can be represented as 3.5×10 or 0.35×102 or even 350×10^{-1}. Similarly the binary equivalent 100011 of 35 can be represented as 100.011×2^{3} or 10.0011×2^{4} etc. Since the decimal or binary point can change places, these representations are called floating point representations. In general the floating point representation of a number has the form

$$\pm m \times 2^{E}$$

In this form m is called mantissa and E exponent. In order to maintain uniformity across computers, certain standards have been fixed.

In modern computers, the length of data words can be 32 bits or 64 bits. In a 32 bits word(0 - 31), the format of bits is as follows

0	1 - 7	8 - 31
Sign	Exponent	Mantissa

Also, in order to have unique representation for each number, the mantissa is to be set between 1 and 2, i.e. $1<m<2$. This form of the number is known as the normalized form. Thus the normalized form of the 100011 is 1.00011×25 or

0	00000101	1.0011000000000…………..0
Sign	E (8 bits)	M(23 bits)

Further, the exponents could be negative as well. Now to remove negative sign from exponent, number 127 is added in the actual exponent so as to make all exponents positive. For example, consider the decimal number 2 which has the binary equivalent 10. The normalized form for 10 is 1.0×2^{1}, i.e. the exponent has the value 1. But according to the standard, the exponent actually stored would be 1+127 i.e. 128. Thus it will be represented as

0	10000000	1.00000000000……………...0
Sign	(8 bits)	(23 bits)

We note that in the 8 bit exponent system, the range of exponents is from 00000001 to 11111110, i.e. the actual exponents from -126 to 127. Consequently, the smallest normalized number is

0 00000001 1.000000………………0

i.e. 1×2^{-126} or 1.2×2^{-38} approx. on the other hand, the largest number that can be stored is

0 11111110 1.1111111……………….1

i.e. (1.1111……………1) $\times 2^{127}$, i.e. $(2 - 2^{-23})\times 2^{127}$ or 3.4×10^{38} approx.

The above discussion was regarding 32-bit word arrangement which is generally termed as single precision. The double precision case for the 64 bit word length can be discussed similarly.

ANNEXURE-II

AMERICAN STANDARD CODE FOR INFORMATION INTERCHANGE (ASCII)

Decimal	Octal	Hex	Binary	Value	
000	000	000	00000000	NUL	(Null char.)
001	001	001	00000001	SOH	(Start of Header)
002	002	002	00000010	STX	(Start of Text)
003	003	003	00000011	ETX	(End of Text)
004	004	004	00000100	EOT	(End of Transmission)
005	005	005	00000101	ENQ	(Enquiry)
006	006	006	00000110	ACK	(Acknowledgment)
007	007	007	00000111	BEL	(Bell)
008	010	008	00001000	BS	(Backspace)
009	011	009	00001001	HT	(Horizontal Tab)
010	012	00A	00001010	LF	(Line Feed)
011	013	00B	00001011	VT	(Vertical Tab)
012	014	00C	00001100	FF	(Form Feed)
013	015	00D	00001101	CR	(Carriage Return)
014	016	00E	00001110	SO	(Shift Out)
015	017	00F	00001111	SI	(Shift In)
016	020	010	00010000	DLE	(Data Link Escape)
017	021	011	00010001	DC1 (XON)	(Device Control 1)
018	022	012	00010010	DC2	(Device Control 2)
019	023	013	00010011	DC3 (XOFF)	(Device Control 3)
020	024	014	00010100	DC4	(Device Control 4)

021	025	015	00010101	NAK	(Negativ Acknowledgemnt)
022	026	016	00010110	SYN	(Synchronous Idle)
023	027	017	00010111	ETB	(End of Trans. Block)
024	030	018	00011000	CAN	(Cancel)
025	031	019	00011001	EM	(End of Medium)
026	032	01A	00011010	SUB	(Substitute)
027	033	01B	00011011	ESC	(Escape)
028	034	01C	00011100	FS	(File Separator)
029	035	01D	00011101	GS	(Group Separator)
030	036	01E	00011110	RS	(Reqst to Send) (Rec. Sep.)
031	037	01F	00011111	US	(Unit Separator)
032	040	020	00100000	SP	(Space)
033	041	021	00100001	!	(exclamation mark)
034	042	022	00100010	"	(double quote)
035	043	023	00100011	#	(number sign)
036	044	024	00100100	$	(dollar sign)
037	045	025	00100101	%	(percent)
038	046	026	00100110	&	(ampersand)
039	047	027	00100111	'	(single quote)
040	050	028	00101000	(	(left/open parenthesis)
041	051	029	00101001	)	(right/closing parenth.)
042	052	02A	00101010	*	(asterisk)
043	053	02B	00101011	+	(plus)
044	054	02C	00101100	,	(comma)
045	055	02D	00101101	-	(minus or dash)
046	056	02E	00101110	.	(dot)
047	057	02F	00101111	/	(forward slash)
048	060	030	00110000	0	

049	061	031	00110001	1	
050	062	032	00110010	2	
051	063	033	00110011	3	
052	064	034	00110100	4	
053	065	035	00110101	5	
054	066	036	00110110	6	
055	067	037	00110111	7	
056	070	038	00111000	8	
057	071	039	00111001	9	
058	072	03A	00111010	:	(colon)
059	073	03B	00111011	;	(semi-colon)
060	074	03C	00111100	<	(less than)
061	075	03D	00111101	=	(equal sign)
062	076	03E	00111110	>	(greater than)
063	077	03F	00111111	?	(question mark)
064	100	040	01000000	@	(AT symbol)
065	101	041	01000001	A	
066	102	042	01000010	B	
067	103	043	01000011	C	
068	104	044	01000100	D	
069	105	045	01000101	E	
070	106	046	01000110	F	
071	107	047	01000111	G	
072	110	048	01001000	H	
073	111	049	01001001	I	
074	112	04A	01001010	J	
075	113	04B	01001011	K	
076	114	04C	01001100	L	
077	115	04D	01001101	M	
078	116	04E	01001110	N	

079	117	04F	01001111	O	
080	120	050	01010000	P	
081	121	051	01010001	Q	
082	122	052	01010010	R	
083	123	053	01010011	S	
084	124	054	01010100	T	
085	125	055	01010101	U	
086	126	056	01010110	V	
087	127	057	01010111	W	
088	130	058	01011000	X	
089	131	059	01011001	Y	
090	132	05A	01011010	Z	
091	133	05B	01011011	[	(left/opening bracket)
092	134	05C	01011100	\	(back slash)
093	135	05D	01011101	]	(right/closing bracket)
094	136	05E	01011110	^	(caret/circumflex)
095	137	05F	01011111	_	(underscore)
096	140	060	01100000	`	
097	141	061	01100001	a	
098	142	062	01100010	b	
099	143	063	01100011	c	
100	144	064	01100100	d	
101	145	065	01100101	e	
102	146	066	01100110	f	
103	147	067	01100111	g	
104	150	068	01101000	h	
105	151	069	01101001	i	
106	152	06A	01101010	j	

107	153	06B	01101011	k	
108	154	06C	01101100	l	
109	155	06D	01101101	m	
110	156	06E	01101110	n	
111	157	06F	01101111	o	
112	160	070	01110000	p	
113	161	071	01110001	q	
114	162	072	01110010	r	
115	163	073	01110011	s	
116	164	074	01110100	t	
117	165	075	01110101	u	
118	166	076	01110110	v	
119	167	077	01110111	w	
120	170	078	01111000	x	
121	171	079	01111001	y	
122	172	07A	01111010	z	
123	173	07B	01111011	{	(left/opening brace)
124	174	07C	01111100	|	(vertical bar)
125	175	07D	01111101	}	(right/closing brace)
126	176	07E	01111110	~	(tilde)
127	177	07F	01111111	DEL	(delete)

Reference : http://www.neurophys.wisc.edu/comp/docs/ascii/

ANNEXURE-III

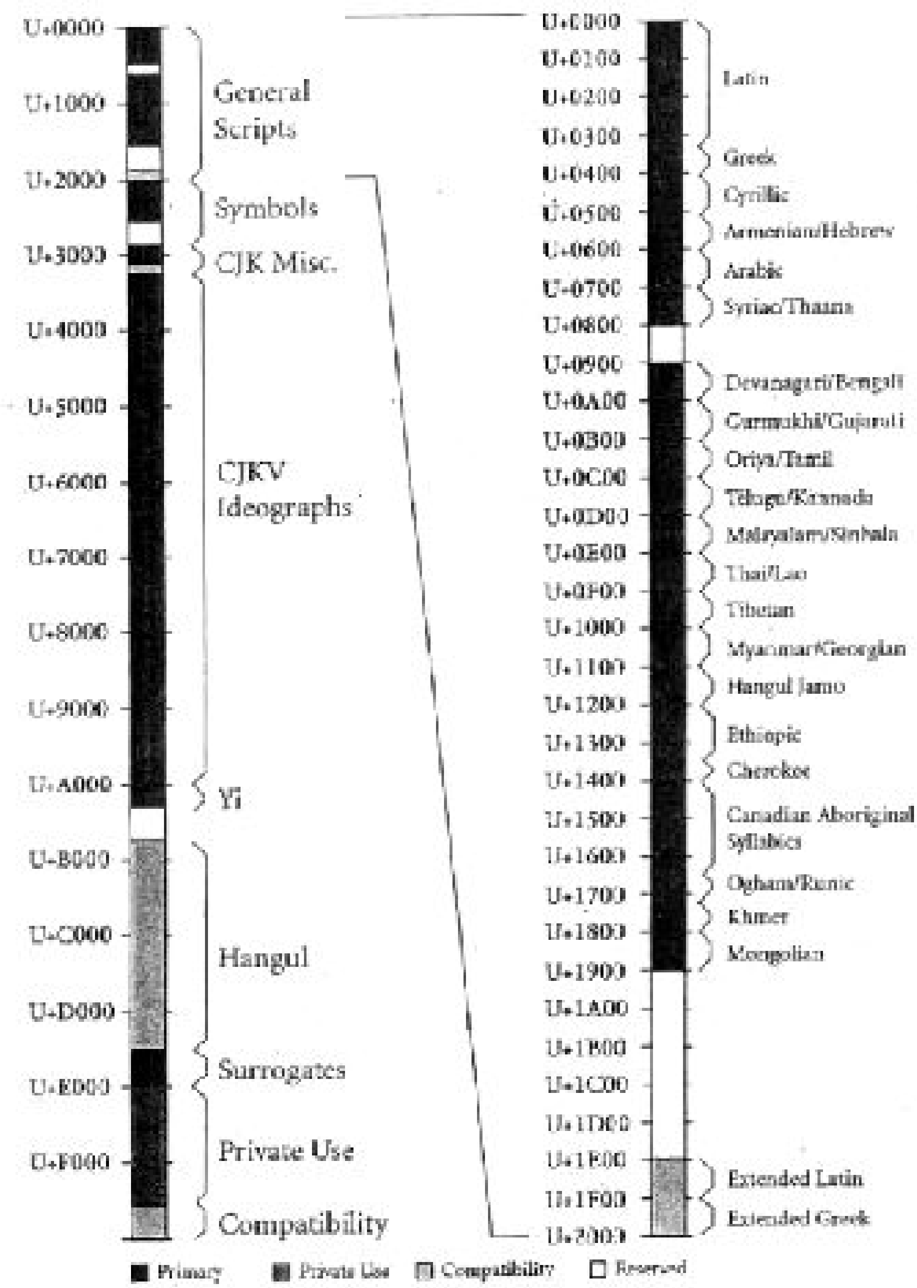
INDIAN STANDARD CODE FOR INFORMATION INTERCHANGE (ISCII)

	Hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Hex	Dec	0	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240
0	0	NUL	DLE	SP	0	@	P	`	p				ओ	ढ	ऱ	◌ॆ	EXT
1	1	SOH	DC1	!	1	A	Q	a	q			◌ँ	औ	ण	ल	◌े	०
2	2	STX	DC2	"	2	B	R	b	r			◌ं	ऑ	त	ळ	◌ै	१
3	3	ETX	DC3	#	3	C	S	c	s			◌ः	क	थ	ऴ	◌ॅ	२
4	4	EOT	DC4	$	4	D	T	d	t			अ	ख	द	व	◌ॊ	३
5	5	ENQ	NAK	%	5	E	U	e	u			आ	ग	ध	श	◌ो	४
6	6	ACK	SYN	&	6	F	V	f	v			इ	घ	न	ष	◌ौ	५
7	7	BEL	ETB	'	7	G	W	g	w			ई	ङ	ऩ	स	◌ॉ	६
8	8	BS	CAN	(	8	H	X	h	x			उ	च	प	ह	◌्	७
9	9	HT	EM	)	9	I	Y	i	y			ऊ	छ	फ	INV	◌़	८
A	10	LF	SUB	*	:	J	Z	j	z			ऋ	ज	ब	◌ा	।	९
B	11	VT	ESC	+	;	K	[	k	{			ऎ	झ	भ	◌ि		
C	12	FF	FS	,	<	L	\	l					ए	ञ	म	◌ी		
D	13	CR	GS	-	=	M	]	m	}			ऐ	ट	य	◌ु		
E	14	SO	RS	.	>	N	^	n	~			ऍ	ठ	य़	◌ू		
F	15	SI	US	/	?	O	_	o	DEL			ऒ	ड	र	◌ृ	ATR	

Reference : http://www.tdil.mit.gov.in/standards.htm

ANNEXURE-IV

UNICODE



Unicode Allocation

	090	091	092	093	094	095	096	097
0		ऐ 0910	ठ 0920	र 0930	◌ी 0940	ॐ 0950	ॠ 0960	॰ 0970
1	◌ँ 0901	ऑ 0911	ड 0921	ऱ 0931	◌ु 0941	◌॑ 0951	ॡ 0961	
2	◌ं 0902	ऒ 0912	ढ 0922	ल 0932	◌ू 0942	◌॒ 0952	◌ॢ 0962	
3	◌ः 0903	ओ 0913	ण 0923	ळ 0933	◌ृ 0943	◌॓ 0953	◌ॣ 0963	
4		औ 0914	त 0924	ऴ 0934	◌ॄ 0944	◌॔ 0954	। 0964	
5	अ 0905	क 0915	थ 0925	व 0935	◌ॅ 0945		॥ 0965	
6	आ 0906	ख 0916	द 0926	श 0936	◌ॆ 0946		० 0966	
7	इ 0907	ग 0917	ध 0927	ष 0937	◌े 0947		१ 0967	
8	ई 0908	घ 0918	न 0928	स 0938	◌ै 0948	क़ 0958	२ 0968	
9	उ 0909	ङ 0919	ऩ 0929	ह 0939	◌ॉ 0949	ख़ 0959	३ 0969	
A	ऊ 090A	च 091A	प 092A		◌ॊ 094A	ग़ 095A	४ 096A	
B	ऋ 090B	छ 091B	फ 092B		◌ो 094B	ज़ 095B	५ 096B	
C	ऌ 090C	ज 091C	ब 092C	◌़ 093C	◌ौ 094C	ड़ 095C	६ 096C	
D	ऍ 090D	झ 091D	भ 092D	ऽ 093D	◌् 094D	ढ़ 095D	७ 096D	
E	ऎ 090E	ञ 091E	म 092E	◌ा 093E		फ़ 095E	८ 096E	
F	ए 090F	ट 091F	य 092F	ि◌ 093F		य़ 095F	९ 096F	

Devnagri

	098	099	09A	09B	09C	09D	09E	09F
0		ঐ	ঠ	র	◌ী		ৠ	ৰ
1	◌ঁ		ড		◌ু		ৡ	ৱ
2	◌ং		ঢ	ল	◌ূ		◌ৢ	৲
3	◌ঃ	ও	ণ		◌ৃ		◌ৣ	৳
4		ঔ	ত		◌ৄ			৴
5	অ	ক	থ					৵
6	আ	খ	দ	শ			০	৶
7	ই	গ	ধ	ষ	◌ে	◌ৗ	১	৷
8	ঈ	ঘ	ন	স	◌ৈ		২	৸
9	উ	ঙ		হ			৩	৹
A	ঊ	চ	প				৪	৺
B	ঋ	ছ	ফ		◌ো		৫	
C	ঌ	জ	ব	◌়	◌ৌ	ড়	৬	
D		ঝ	ভ		◌্	ঢ়	৭	
E		ঞ	ম	◌া			৮	
F	এ	ট	য	◌ি		য়	৯	

Bengali

	0A8	0A9	0AA	0AB	0AC	0AD	0AE	0AF
0		ઐ 0A90	ઠ 0AA0	ર 0AB0	ી 0AC0	ૐ 0AD0	ૠ 0AE0	
1	ઁ 0A81	ઑ 0A91	ડ 0AA1		ુ 0AC1			
2	ં 0A82		ઢ 0AA2	લ 0AB2	ૂ 0AC2			
3	ઃ 0A83	ઓ 0A93	ણ 0AA3	ળ 0AB3	ૃ 0AC3			
4		ઔ 0A94	ત 0AA4		ૄ 0AC4			
5	અ 0A85	ક 0A95	થ 0AA5	વ 0AB5	ૅ 0AC5			
6	આ 0A86	ખ 0A96	દ 0AA6	શ 0AB6			૦ 0AE6	
7	ઇ 0A87	ગ 0A97	ધ 0AA7	ષ 0AB7	ે 0AC7		૧ 0AE7	
8	ઈ 0A88	ઘ 0A98	ન 0AA8	સ 0AB8	ૈ 0AC8		૨ 0AE8	
9	ઉ 0A89	ઙ 0A99		હ 0AB9	ૉ 0AC9		૩ 0AE9	
A	ઊ 0A8A	ચ 0A9A	પ 0AAA				૪ 0AEA	
B	ઋ 0A8B	છ 0A9B	ફ 0AAB		ો 0ACB		૫ 0AEB	
C		જ 0A9C	બ 0AAC	઼ 0ABC	ૌ 0ACC		૬ 0AEC	
D	ઍ 0A8D	ઝ 0A9D	ભ 0AAD	ઽ 0ABD	્ 0ACD		૭ 0AED	
E		ઞ 0A9E	મ 0AAE	ા 0ABE			૮ 0AEE	
F	એ 0A8F	ટ 0A9F	ય 0AAF	િ 0ABF			૯ 0AEF	

Gujarati

	0B8	0B9	0BA	0BB	0BC	0BD	0BE	0BF
0		ஐ 0B90		ர 0BB0	◌ீ 0BC0			௰ 0BF0
1				ற 0BB1	◌ு 0BC1			௱ 0BF1
2	◌ஂ 0B82	ஒ 0B92		ல 0BB2	◌ூ 0BC2			௲ 0BF2
3	◌ஃ 0B83	ஓ 0B93	ண 0BA3	ள 0BB3				
4		ஔ 0B94	த 0BA4	ழ 0BB4				
5	அ 0B85	க 0B95		வ 0BB5				
6	ஆ 0B86				ெ◌ 0BC6			
7	இ 0B87			ஷ 0BB7	ே◌ 0BC7	◌ௗ 0BD7	௧ 0BE7	
8	ஈ 0B88		ந 0BA8	ஸ 0BB8	ை◌ 0BC8		௨ 0BE8	
9	உ 0B89	ங 0B99	ன 0BA9	ஹ 0BB9			௩ 0BE9	
A	ஊ 0B8A	ச 0B9A	ப 0BAA		ொ 0BCA		௪ 0BEA	
B					ோ 0BCB		௫ 0BEB	
C		ஜ 0B9C			ௌ 0BCC		௬ 0BEC	
D					◌் 0BCD		௭ 0BED	
E	எ 0B8E	ஞ 0B9E	ம 0BAE	◌ா 0BBE			௮ 0BEE	
F	ஏ 0B8F	ட 0B9F	ய 0BAF	◌ி 0BBF			௯ 0BEF	

Tamil

Reference : The Unicode Standard, Version 3.0, The Unicode Consortium, Addison-Wesley, An Imprint of Addison Wesley Longman, Inc., 2000.

ANNEXURE-U

Installation of Netbeans IDE

To install the NetBeans Integrated Development Environment (IDE), first it must be downloaded from the NetBeans web page. It can also be downloaded in a bundle with the Java 2 Standard Edition (J2SE) or Java 2 Enterprise Edition (J2EE).

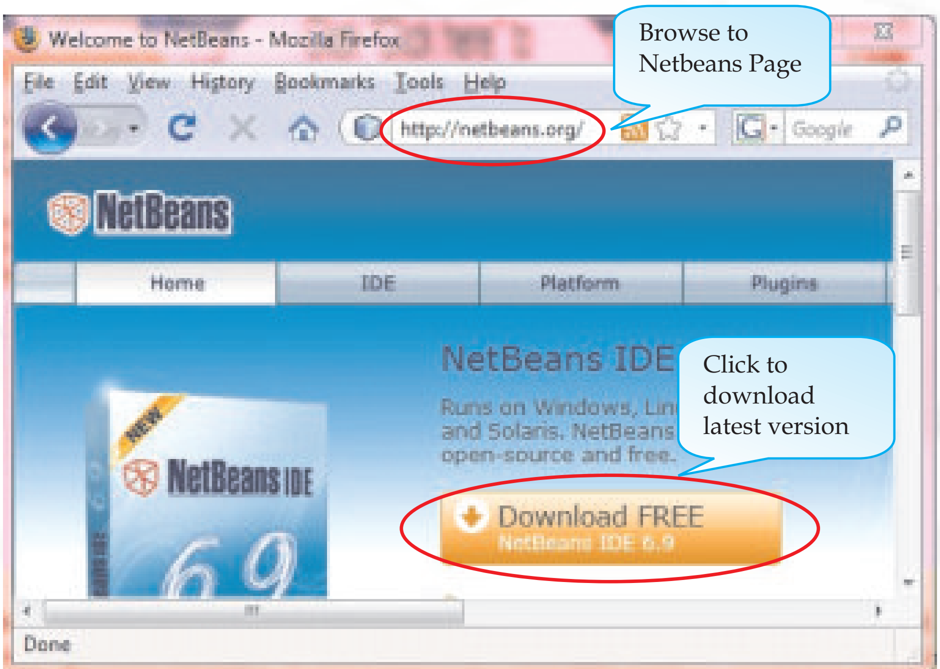
> **!** You should have already installed the Java Development Kit (JDK) before installing Netbeans. If you have not done so, then install JDK first before starting the Netbeans installation . The JDK consists of the Java compiler and related tools which enable users to create applications in Java.

Download Netbeans Installer File from the Internet

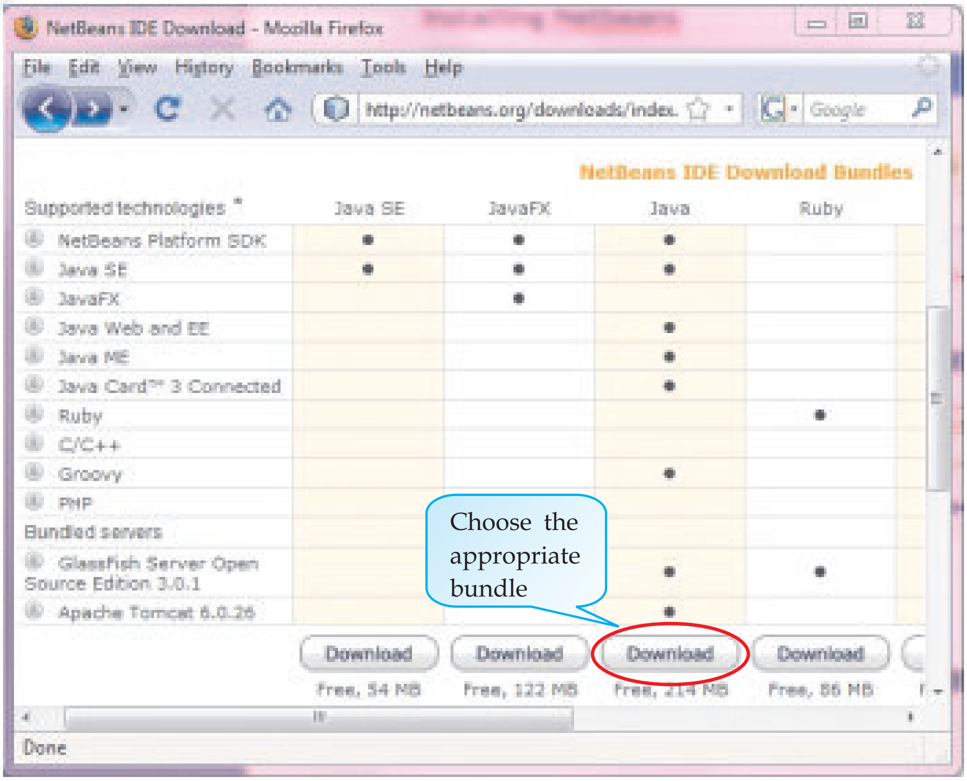
To download Netbeans Installer files, follow the steps enumerated below:

STEP 1: Browse to the Netbeans web page - http://netbeans.org/

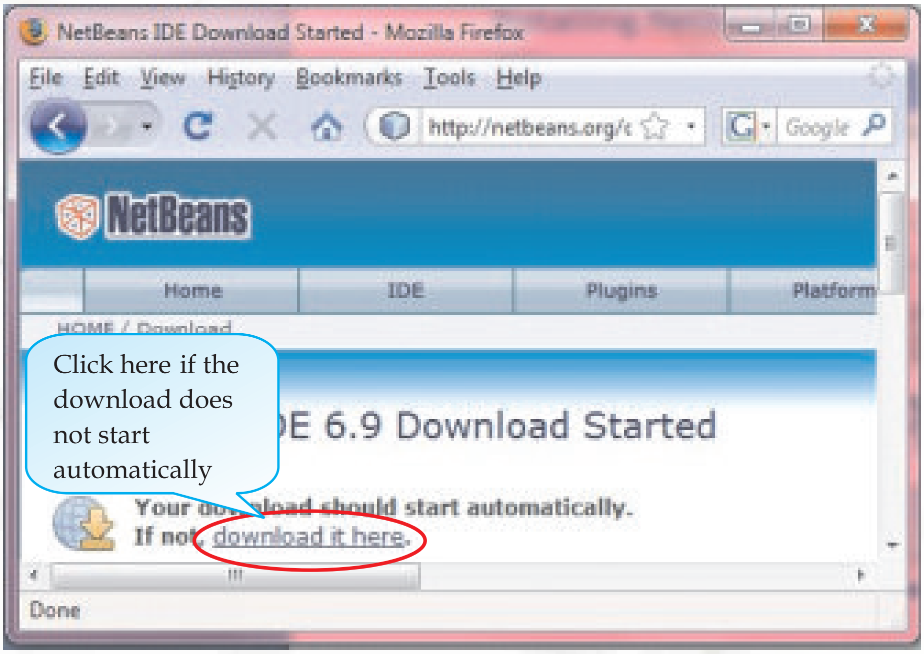
STEP 2: Choose to download the latest version of the IDE



STEP 3: Choose the appropriate bundle to be downloaded



Once you click on the Download button for the appropriate bundle, the following screen appears and then the download starts automatically.

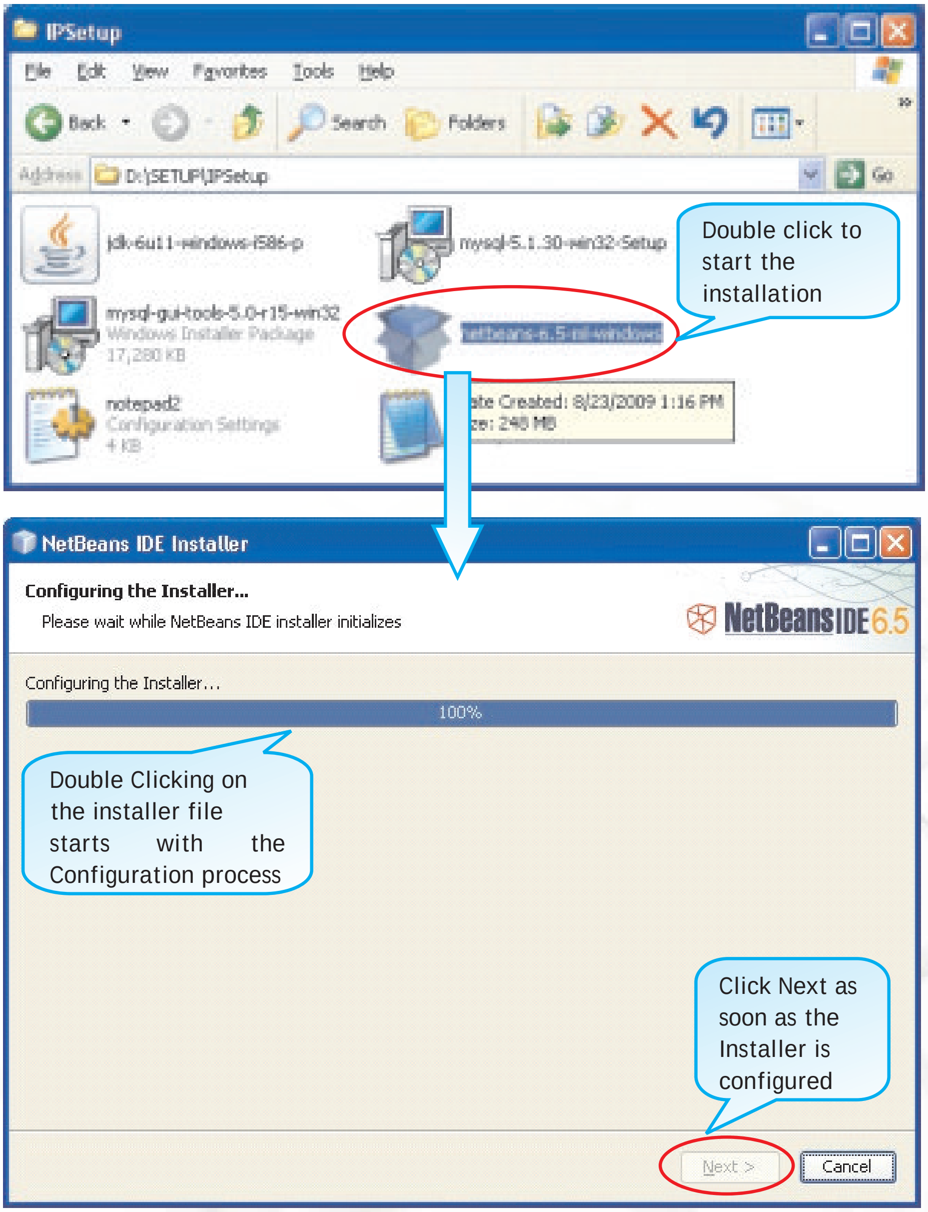


Installing Netbeans Using The Installer File

After downloading Netbeans Installer files, follow the steps enumerated below to install the IDE on your system:

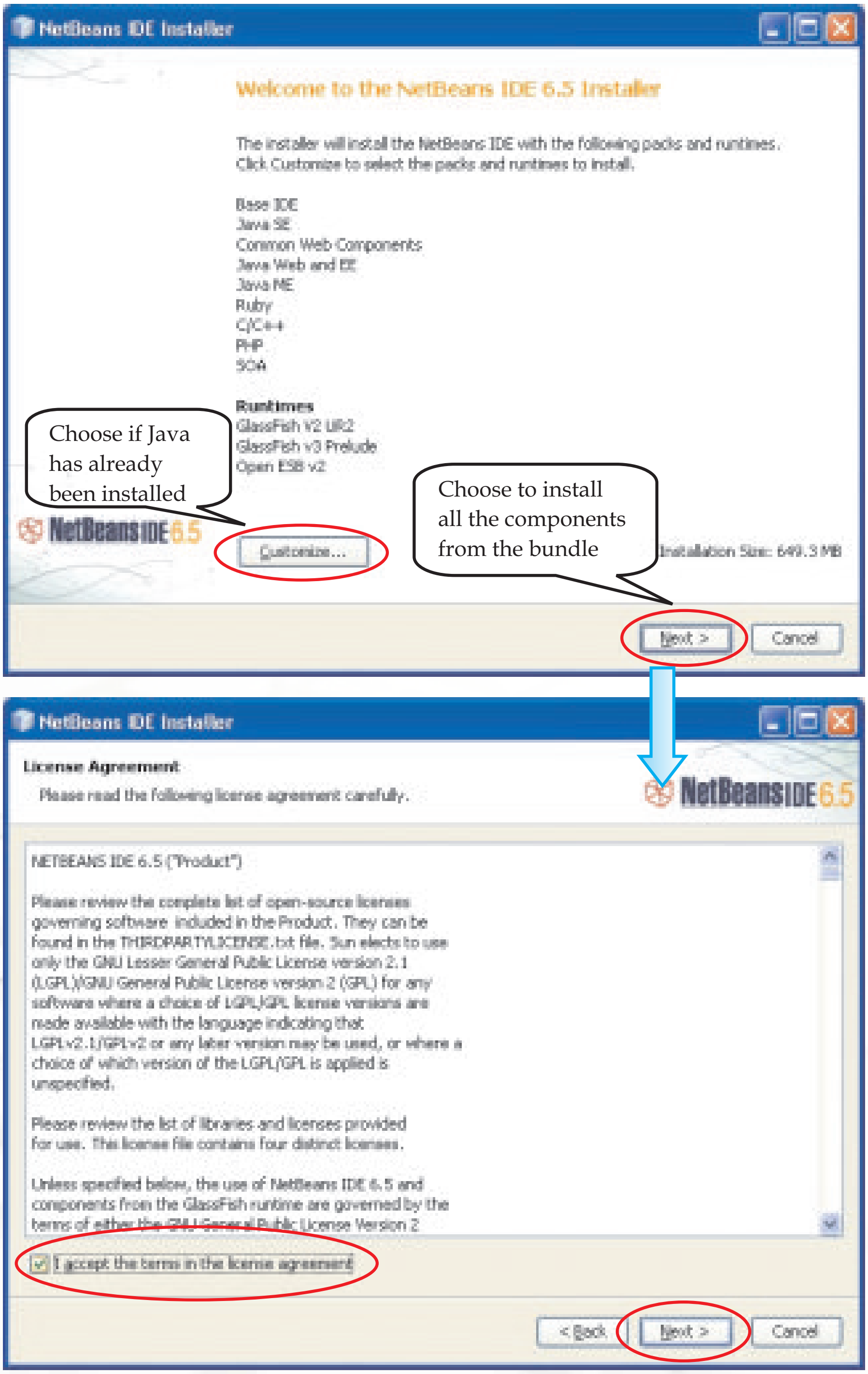
STEP 1: Start the Installer Program

Use a file explorer to navigate to the file that you just downloaded. Double click on the file name to start the installation process and then follow the instructions as given in the dialog boxes (shown in the following figures).

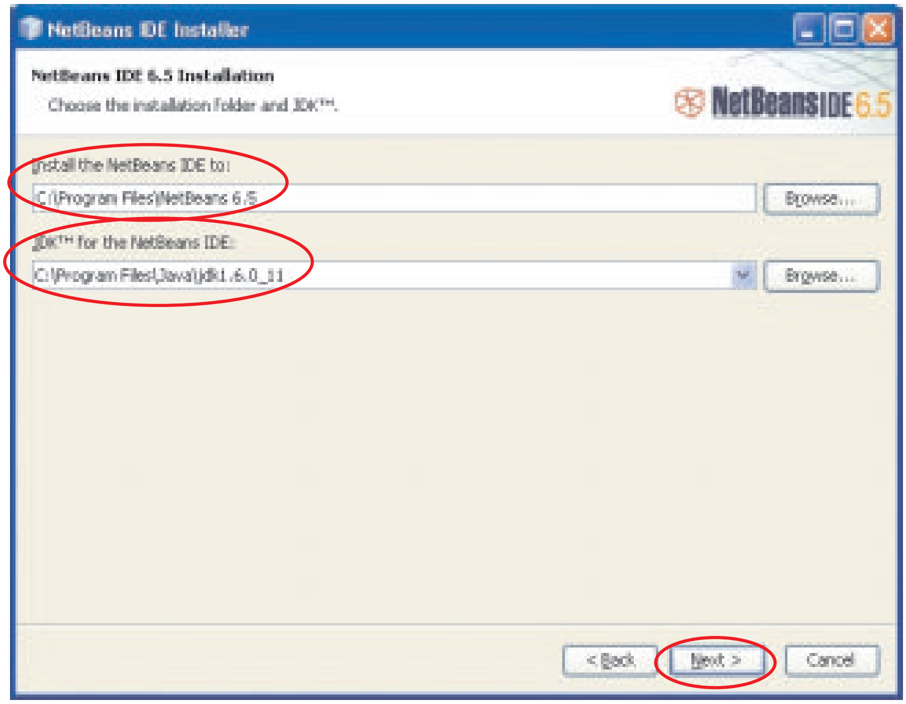


STEP 2: Customize the Netbeans Installer

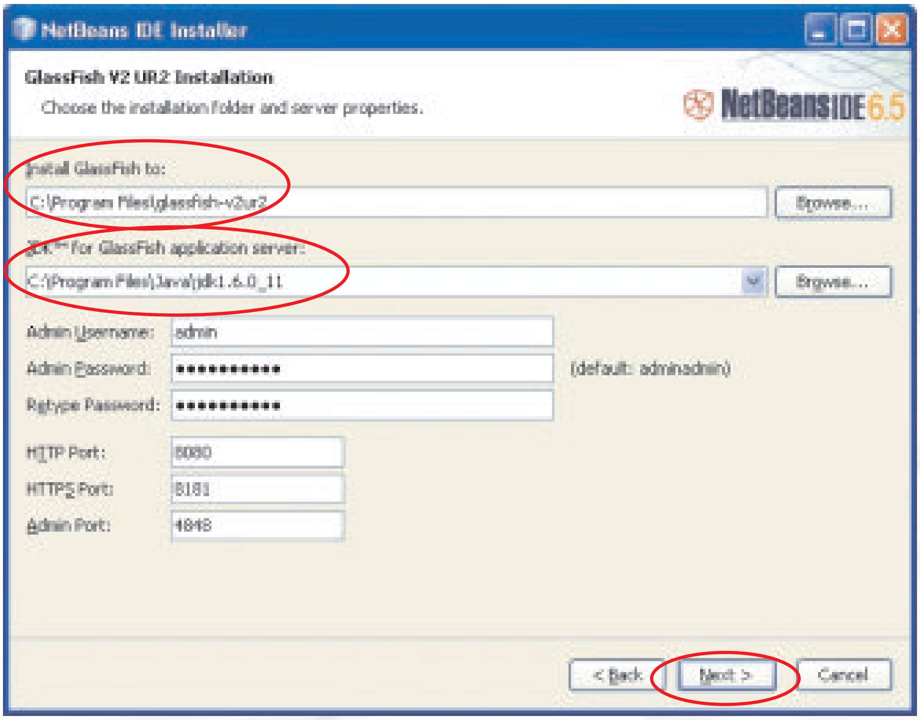
If you have already installed Java, then choose the customize button to select which Java version to use else simply click the Next button. In the next dialog box select the accept terms and conditions check box and click on Next button.

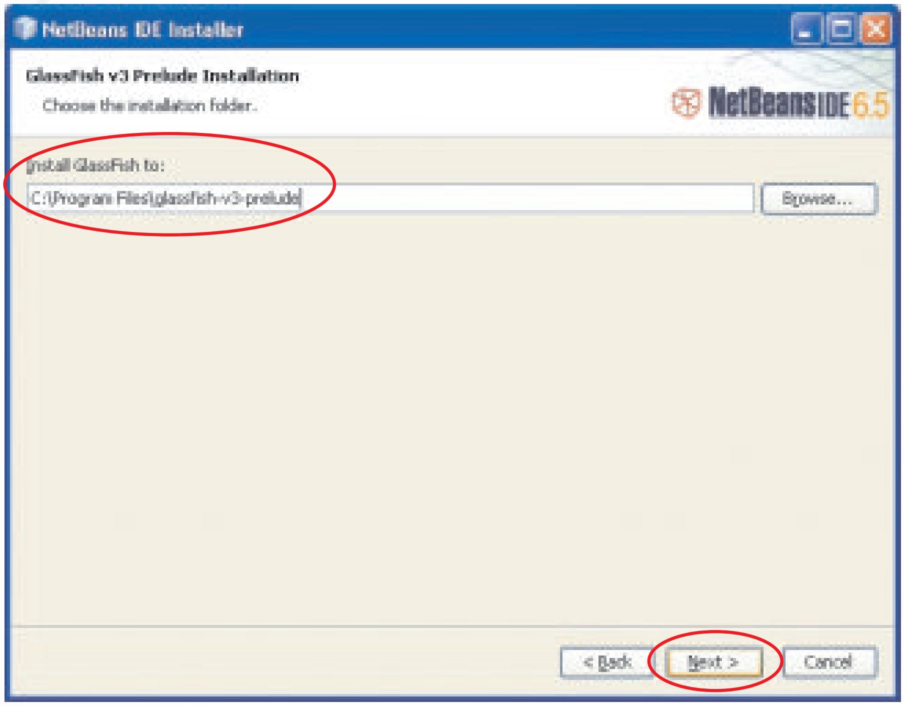


STEP 3: Choose the Installation Folder for all components to be installed (one by one) and also the appropriate JDK version.



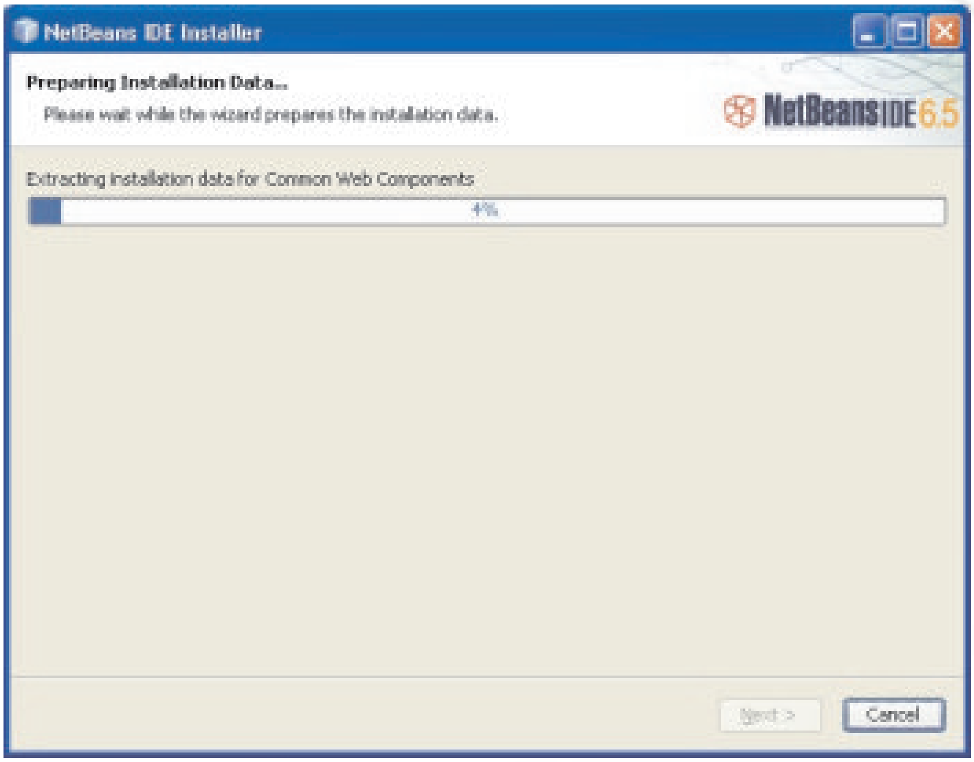
In the subsequent dialog boxes choose the appropriate destination folder for the other components of the bundle.

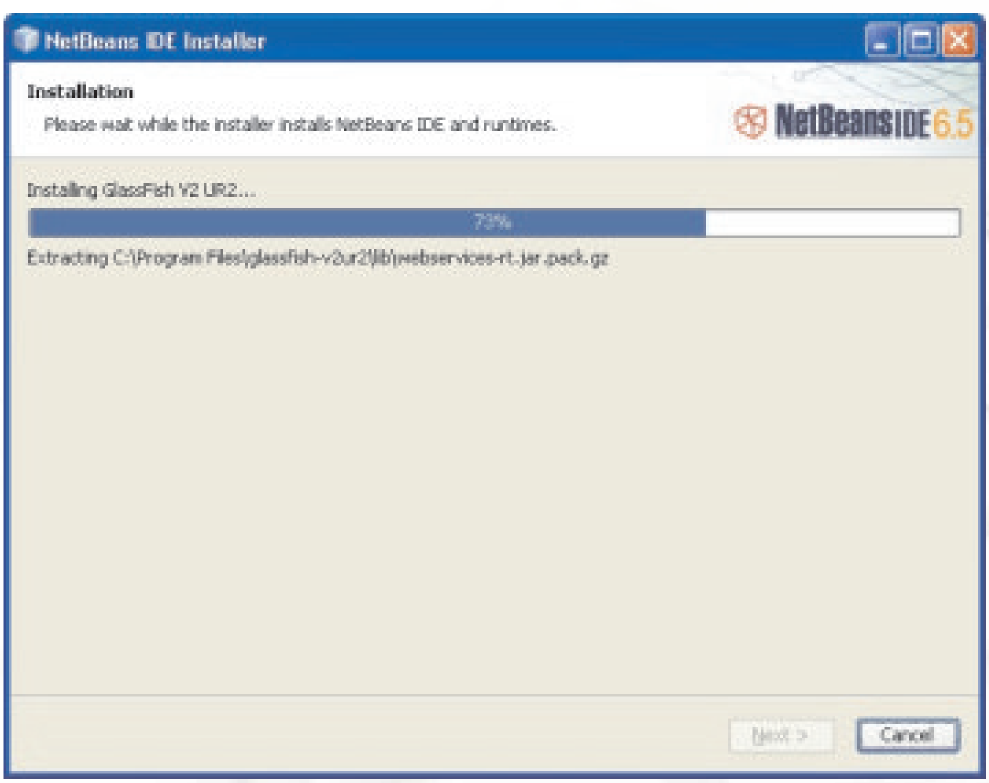




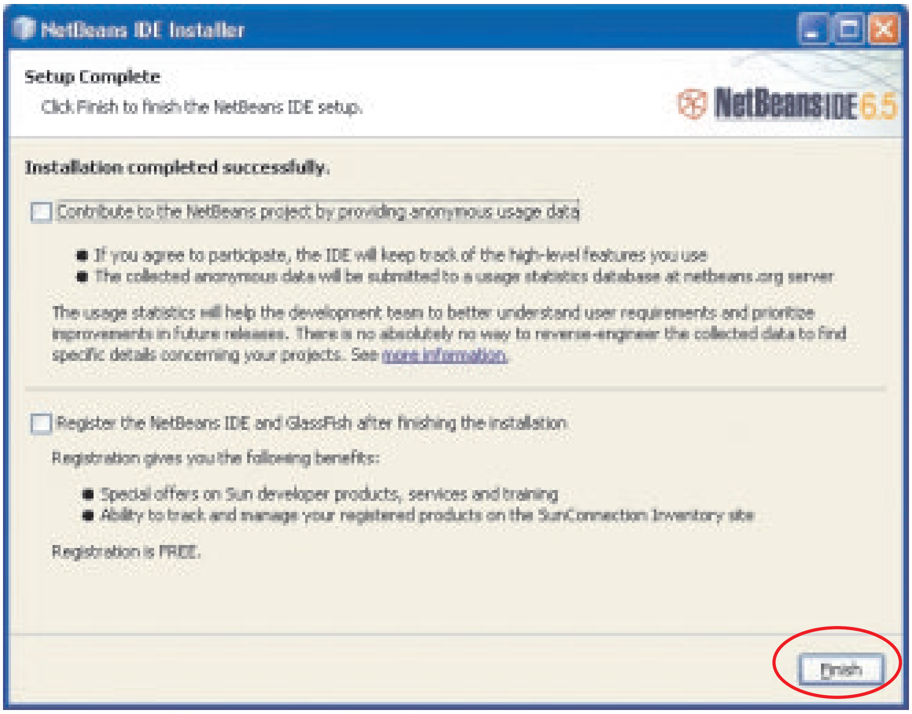
STEP 4: Start the Installation of Netbeans and the related components by clicking on the Install button. Continue with the dialog boxes until you have completed installing NetBeans on your PC.

The progress window keeps you informed about the installation status.



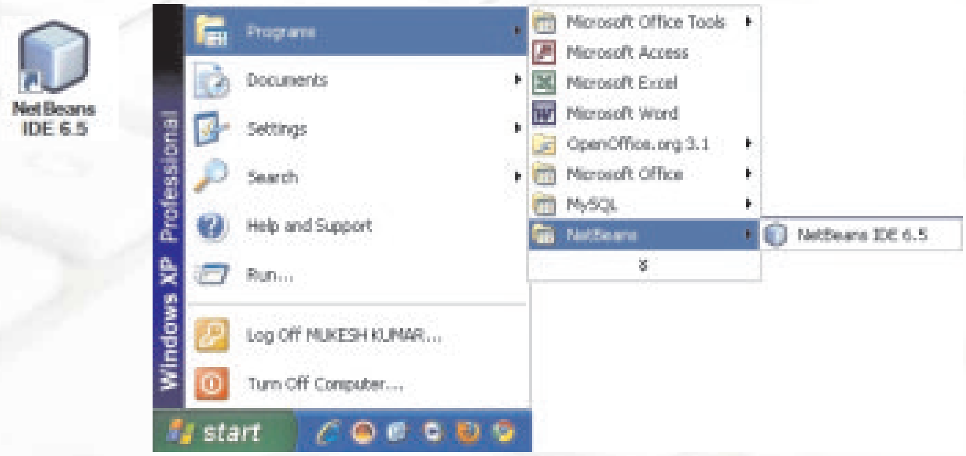


Click on the Finish button as soon as the Installation process is completed and the following dialog box is shown.



Starting Netbeans

If you use the default location for the install on Windows, NetBeans will be installed in the directory C:\Program Files\NetBeans6.5. An icon to start NetBeans will be installed on your desktop. On Windows start NetBeans with a double mouse click on this icon or use the Start button to navigate to the program name.



The opening screen of the Netbeans IDE is as shown in the following Figure.