Eclipse Adoption Study

November 2006

(with comparisons to November 2005 and September 2004 Study)

BZ Research

BZ Research
7 High Street, Suite 407
Huntington, NY 11743
631-421-4158
www.bzresearch.com

Report revised January 5, 2007

Table of Contents

Research Methodology

Methodology – November 2006 Study

An e-mail was sent out to 19,500 randomly selected SD Times subscribers inviting them to participate in a study. Excluding bounces, the net number of surveys delivered was 19,246. The purpose was to ascertain their plans to use Eclipse-based technology and Eclipse plug-ins. The study was e-mailed on November 9, 2006, and cut off and tabulated by BZ Research on November 20, 2006. 769 subscribers, or 4.0% of the respondents, completed the survey, although not all of them answered every question.

BZ Research designed, conducted and tabulated the study. BZ Research is a division of BZ Media, publisher of SD Times.

Methodology – November 2005 Study

An e-mail was sent out to 17,500 randomly selected SD Times subscribers inviting them to participate in a study. Excluding bounces, the net number of surveys delivered was 16,358. The purpose was to ascertain their plans to use Eclipse-based technology and Eclipse plug-ins. The study was e-mailed on November 4, 2005, and cut off and tabulated by BZ Research on November 7, 2005. 772 subscribers, or 4.7% of the respondents, completed the survey, although not all of them answered every question.

BZ Research designed, conducted and tabulated the study. BZ Research is a division of BZ Media, publisher of SD Times.

Methodology – September 2004 Study

An e-mail was sent out to 14,200 randomly selected SD Times subscribers inviting them to participate in a study. Excluding bounces, the net number of surveys delivered was 14,158. The purpose was to ascertain their plans to use Eclipse-based technology and Eclipse plug-ins. The study was e-mailed on August 23, 2004, and cut off and tabulated by BZ Research on August 31, 2004. 719 subscribers, or 5.1% of the respondents, completed the survey, although not all of them answered every question.

BZ Research designed, conducted and tabulated the study. BZ Research is a division of BZ Media, publisher of SD Times.

QUESTION 1: Do the developers within your organization use the Eclipse framework?

(please check one)

	2004	2005	2006
Yes, all developers use Eclipse	13.2%	12.8%	13.5%
Yes, most (more than half) use Eclipse	15.9%	20.2%	19.5%
Yes, some (half or fewer) use Eclipse	24.8%	29.5%	33.3%
Subtotal: At least some use Eclipse framework	53.9%	62.5%	66.3%
No, it is not in use	43.5%	35.1%	31.3%
Don't know	2.6%	2.3%	2.3%
Base	718	772	769

QUESTION 2: Which version(s) of Eclipse are you using?

(please check all that apply)

	2004	2005	2006
Eclipse 3.2.x	n/a	n/a	48.7%
Eclipse 3.1.x	n/a	48.4%	31.8%
Eclipse 3.0.x	41.5%	24.7%	10.2%
Eclipse 2.x	23.9%	4.2%	3.2%
Eclipse 1.x	0.0%	0.3%	0.4%
Not using Eclipse	39.3%	33.0%	28.2%
Don't know	5.0%	4.7%	5.9%
Other	1.9%	2.8%	1.3%
Total Respondents	698	746	745

QUESTION 3: Which commercial Eclipse-based IDEs are you using?

(please check all that apply)

	2006
Other (107 responses, see Appendix I)	33.6%
IBM WebSphere Application Developer	30.5%
Genuitec MyEclipse	19.8%
IBM Software Development Platform	17.3%
Borland JBuilder/Core SDP	9.7%
BEA Weblogic Workshop	9.1%
Exadel Studio	3.5%
SAP NetWeaver Studio	3.5%
Wind River Workbench	3.1%
Sybase Workspace	1.3%
Total Respondents	318

QUESTION 4: If your organization is not currently using Eclipse or Eclipse-based tools, do you expect to do so?

(please check one)

	2004	2005	2006
Within the next 6 months	16.6%	13.6%	11.2%
Within the next 6-12 months	8.9%	5.0%	5.3%
Within the next 12-18 months	3.7%	3.7%	5.1%
Further than the next 18 months	3.3%	6.3%	5.6%
Never	20.5%	19.9%	18.8%
Not sure	47.1%	51.4%	53.9%
Total Respondents	459	381	356

QUESTION 5: Do you have any comments about your reasons for using Eclipse or for not using Eclipse?

Open-ended question. For 384 verbatim responses, see Appendix II

QUESTION 6: If you use Eclipse, or Eclipse-based tools, why do you use it?

(please check all that apply)

	2004	2005	2006
Because….			
it is a low-cost solution	59.8%	62.2%	64.7%
it is an open-source platform	65.5%	62.4%	61.5%
there is a wide range of plug-ins available	n/a	58.2%	60.4%
it's extensible	n/a	49.6%	47.6%
it's easy to learn/use	42.6%	35.4%	40.3%
it is cross-platform	53.2%	39.8%	37.4%
it targets multiple deployment platforms	n/a	33.4%	34.6%
won't get locked in to a particular vendor	n/a	39.0%	34.2%
my development team prefers it	n/a	30.8%	32.4%
we like the license model	n/a	34.0%	31.5%
it's technologically superior	n/a	30.6%	31.1%
it is multi-language	25.3%	21.2%	26.7%
we can write our own plug-ins	n/a	27.8%	25.8%
our preferred tools are based on it	14.0%	16.6%	19.8%
it's an alternative to Microsoft	12.3%	18.6%	19.4%
we can modify the platform	n/a	20.4%	16.1%
it is more stable	n/a	17.6%	16.1%
my IT/development management prefers it	n/a	10.2%	10.1%
we're following IBM's lead	15.3%	10.6%	9.3%
I think it's a great career move	n/a	8.0%	8.6%
it generates better code	n/a	7.8%	8.4%
Other (35 responses, see Appendix III)	14.0%	5.8%	6.4%
there are no other viable alternatives	n/a	5.2%	3.8%
I don't know why we use Eclipse	4.7%	1.6%	2.0%
an executive mandate	n/a	3.6%	1.8%
Base	470	500	546

QUESTION 7: What types of applications are you building with Eclipse or Eclipse-based tools?

(please check all that apply)

	2004	2005	2006
Java	82.9%	84.8%	81.5%
Web	n/a	57.0%	57.4%
Web services	40.9%	43.3%	42.4%
Server	37.2%	34.7%	34.4%
AJAX	n/a	n/a	27.4%
Desktop	n/a	25.8%	26.5%
Rich Client	n/a	n/a	23.9%
C/C++	10.0%	15.5%	19.9%
Other (73 responses, see Appendix IV)	12.7%	13.2%	12.7%
Embedded	n/a	n/a	9.1%
Native Linux	9.0%	9.1%	9.1%
Native Windows	12.5%	7.9%	7.5%
Mobile	n/a	8.2%	5.9%
Native Unix	4.0%	3.1%	4.9%
Total Respondents	479	547	573

QUESTION 8: Do you have any comments about the types of applications that you're building with Eclipse or with Eclipse-based tools?

Open-ended question. For 89 verbatim responses, see Appendix V

QUESTION 9: Are you and your team currently using Eclipse plug-ins?

(please check one)

	2004	2005	2006
Yes	53.5%	58.6%	63.1%
No	38.1%	36.1%	28.7%
Don't know	8.4%	5.3%	8.2%
Total Respondents	667	718	721

QUESTION 10: If you or your team are using Eclipse plug-ins how many are you using?

(please check one)

	2004	2005	2006
1 only	n/a	8.5%	9.5%
2-4	n/a	51.4%	46.2%
5-9	n/a	26.3%	27.7%
10 or more	n/a	13.8%	16.6%
Total Respondents	n/a	434	476

QUESTION 11: If you or your team are using Eclipse plug-ins, are they…

(please check one)

	2004	2005	2006
Exclusively open-source	33.7%	37.8%	42.9%
Exclusively proprietary/closed-source	3.1%	3.5%	1.5%
A mixture	32.6%	47.7%	42.9%
Don't know	30.6%	11.0%	12.6%
Total Respondents	549	463	517

QUESTION 12: What types of Eclipse plug-ins are you currently using or considering using in the future?

(please check all that apply)

	2004	2005	2006
Code/text editor	51.0%	42.0%	58.3%
HTML/Web development	41.9%	51.8%	49.8%
XML editor/parser	43.1%	55.5%	49.4%
Unit testing	45.3%	45.9%	45.5%
Debugger	36.0%	35.5%	44.7%
Compiler	36.4%	26.1%	41.4%
Make/build utility	35.3%	34.2%	39.3%
Source configuration management	33.0%	35.9%	38.7%
UML or other modeling	42.8%	44.1%	30.8%
Database development	35.3%	38.0%	30.0%
GUI builder/designer	31.2%	36.3%	28.3%
Static code analyzer	18.2%	23.6%	24.9%
Team collaboration	21.6%	27.6%	24.5%
Web services framework	24.1%	34.0%	21.3%
Change management	15.0%	25.9%	19.0%
Runtime profiler	n/a	20.3%	17.8%
AJAX	n/a	n/a	16.2%
Rich client development	n/a	23.0%	15.2%
Functional tester	n/a	17.1%	11.3%
Stress/load/performance testing	13.0%	20.3%	10.9%
Help/documentation builder	13.2%	14.6%	9.3%
Project management	18.7%	18.2%	8.7%
SOA framework	11.8%	19.2%	7.9%
Installer builder	10.3%	12.5%	6.5%
Business rules/process tools	9.6%	14.6%	5.7%
Security analysis	n/a	6.1%	3.4%
Multimedia development	n/a	4.2%	2.8%
Total Respondents	439	479	494

QUESTION 13: Have you written or are you writing/considering writing any Eclipse plug-ins or extensions?

(please check one)

	2004	2005	2006
Yes, for use within our organization	n/a	18.9%	19.4%
Yes, for our partners	n/a	3.9%	3.1%
Yes, to give away free	n/a	7.0%	7.1%
Yes, to sell	n/a	6.8%	5.1%
Yes, to support our platforms/products	n/a	9.6%	7.5%
No, we don't write plug-ins	n/a	56.4%	56.7%
Don't know	n/a	14.0%	14.6%
Other (10 responses, see Appendix VI)	n/a	2.0%	1.5%
Total Respondents	n/a	615	651

QUESTION 14: Do you have any comments about your experiences with Eclipse plug-ins -- good bad or otherwise?

Open-ended question. For 128 verbatim responses, see Appendix VII

QUESTION 15: Do you (or does your organization) require that new development tools be Eclipse-based or Eclipse-compatible?

(please check one)

	2004	2005	2006
Yes it's required	7.6%	6.7%	6.6%
No but it's preferred	29.4%	32.9%	34.2%
No it's neither required nor preferred	53.2%	53.8%	49.7%
Don't know	9.8%	6.5%	9.3%
Total Respondents	643	689	708

QUESTION 16: Is your organization formally involved with the Eclipse Foundation?

(please check all that apply)

	2004	2005	2006
Yes as a Strategic Member	n/a	2.4%	2.5%
Yes as an Add-In Provider	n/a	1.0%	0.8%
Yes as an Associate Member	n/a	0.4%	0.1%
Yes as a Committer Member	n/a	0.4%	0.7%
Yes on the Board of Directors	n/a	0.4%	0.7%
No we are not formally involved	n/a	89.4%	86.1%
Don't know	n/a	7.1%	10.8%
Total Respondents	n/a	716	710

QUESTION 17: Do you or your team members participate in any Eclipse projects?

(please check one)

	2004	2005	2006
Yes we are committers on a project	n/a	1.9%	1.5%
Yes we contribute (bug submissions, newsgroup postings) as part of our jobs	n/a	2.2%	2.5%
Yes we contribute (bug submissions, newsgroup postings) on our own time	n/a	7.4%	7.2%
No we don't participate	n/a	79.7%	77.3%
Don't know	n/a	8.8%	11.4%
Total Respondents	n/a	718	720

QUESTION 18: Do you have any comments about your involvement with the Eclipse Foundation or your experiences with contributing to its projects?

Open-ended question. For 48 verbatim responses, see Appendix VIII

QUESTION 19a: Which Eclipse "bits" are currently used by your organization?

(please check all that apply)

	2005	2006
Java Development Tools (JDT)	79.9%	70.6%
J2EE Standard Tools (JST)	46.9%	48.8%
Web Standard Tools (WST)	36.4%	32.5%
Eclipse Modeling Framework	26.6%	31.7%
Visual Editor (VE)	33.1%	28.6%
Graphical Editor Framework (GEF)	27.0%	27.0%
SQL Development Tools	n/a	25.3%
C/C++ IDE (CDE)	18.8%	23.9%
Graphical Modeling Framework (GMF)	n/a	20.0%
Plug-In Development Environment (PDE)	25.3%	19.8%
JavaServer Faces Tools	13.2%	18.5%
Rich Client Platform (RCP)	15.3%	18.3%
UML2 Project (UML2)	14.9%	18.1%
Test & Performance Tools (TPTP)	24.5%	17.7%
Data Tools Platform	12.6%	13.8%
PHP IDE	n/a	9.7%
Business Intelligence & Reporting Tools (BIRT)	5.0%	8.8%
AspectJ Development Tools (AJDT)	7.9%	7.6%
Generative Model Transformer (GMT)	n/a	4.7%
Mobile Tools for Java	n/a	4.3%
Equinox Framework	1.0%	4.3%
Native Application Builder	n/a	3.9%
Connectivity (ODA Component)	n/a	3.7%
Embedded Rich Client Platform (eRCP)	n/a	3.5%
Device Software Development Platform	2.1%	3.3%
Model Driven Development Infrastructure	n/a	2.9%
COBOL IDE	1.3%	1.0%
Dali JPA Tools	n/a	0.8%
Total Respondents	478	486

QUESTION 19b: Which Eclipse "bits" do you expect to be used within the next 12 months?

(please check all that apply)

	2005	2006
SQL Development Tools	n/a	19.5%
UML2 Project (UML2)	28.2%	19.5%
Test & Performance Tools (TPTP)	27.8%	19.3%
Business Intelligence & Reporting Tools (BIRT)	n/a	18.3%
Eclipse Modeling Framework (EMF)	20.1%	16.7%
JavaServer Faces Tools	25.3%	15.8%
Data Tools Platform	19.5%	13.0%
Visual Editor (VE)	19.7%	12.6%
Rich Client Platform (RCP)	14.9%	12.6%
Graphical Modeling Framework (GMF)	n/a	11.9%
J2EE Standard Tools (JST)	16.5%	11.3%
Graphical Editor Framework (GEF)	20.3%	11.1%
AspectJ Development Tools (AJDT)	11.3%	10.5%
Web Standard Tools (WST)	16.9%	10.3%
Mobile Tools for Java	n/a	9.9%
C/C++ IDE (CDE)	12.3%	8.8%
PHP IDE	n/a	8.0%
Model Driven Development Infrastructure (MDDI)	n/a	7.8%
Java Development Tools (JDT)	13.4%	6.6%
Plug-In Development Environment (PDE)	12.6%	6.6%
Embedded Rich Client Platform (eRCP)	n/a	6.6%
Device Software Development Platform	8.6%	6.0%
Native Application Builder	n/a	4.7%
Generative Model Transformer (GMT)	n/a	4.3%
Equinox Framework	2.5%	3.7%
Connectivity (ODA Component)	n/a	3.5%
Dali JPA Tools	n/a	3.5%
COBOL IDE	2.9%	2.1%
Total Respondents	478	486

QUESTION 19c: Which Eclipse "bits" are currently used OR do you expect will be used by your organization within the next year? *(Note: This is a summation of any who selected an item in either 19a or 19b.)*

(please check all that apply)

	2005	2006
Java Development Tools (JDT)	84.9%	76.1%
J2EE Standard Tools (JST)	60.3%	59.5%
Eclipse Modeling Framework (EMF)	43.7%	47.3%
SQL Development Tools	n/a	44.4%
Web Standard Tools (WST)	50.6%	42.6%
Visual Editor (VE)	49.8%	40.9%
Graphical Editor Framework (GEF)	44.4%	37.9%
UML2 Project (UML2)	41.2%	37.4%
Test & Performance Tools (TPTP)	50.6%	36.8%
JavaServer Faces Tools	37.4%	34.2%
C/C++ IDE (CDE)	29.3%	32.3%
Graphical Modeling Framework (GMF)	n/a	31.5%
Rich Client Platform (RCP)	28.5%	30.7%
Business Intelligence & Reporting Tools (BIRT)	24.7%	27.2%
Data Tools Platform	30.1%	26.7%
Plug-In Development Environment (PDE)	34.7%	25.7%
AspectJ Development Tools (AJDT)	18.8%	17.9%
PHP IDE	n/a	17.7%
Mobile Tools for Java	n/a	13.8%
Model Driven Development Infrastructure (MDDI)	n/a	10.7%
Embedded Rich Client Platform (eRCP)	n/a	9.9%
Generative Model Transformer (GMT)	n/a	9.1%
Device Software Development Platform	10.3%	8.8%
Native Application Builder	n/a	8.4%
Equinox Framework	3.1%	8.0%
Connectivity (ODA Component)	n/a	6.8%
Dali JPA Tools	n/a	4.3%
COBOL IDE	4.2%	3.1%
Total Respondents	478	486

QUESTION 20: What is your company's interest/involvement in the Rich Client Platform?

(please check one)

	2005	2006
We have built and deployed rich clients using the RCP	4.8%	6.0%
We are currently building rich clients using the RCP	5.3%	4.2%
We plan to start using the RCP in the next six months	4.7%	5.7%
We plan to start using the RCP in the next 6-12 months	3.5%	6.1%
We plan to start using the RCP in the next 12-24 months	6.7%	6.7%
We do not plan to use the RCP, but will use other technologies to build rich-client applications	12.3%	11.2%
We do not plan to use the RCP, and do not plan to build rich client applications	20.4%	22.4%
Don't know	42.4%	37.8%
Total Respondents	661	688

QUESTION 21: Do you have any comments about your experiences with the Eclipse Rich Client Platform?

Open-ended question. For 48 verbatim responses, see Appendix IX

QUESTION 22: Which tools frameworks are currently in use within your organization?

(please check all that apply)

	2004**	2005	2006
Eclipse- or Eclipse-based	n/a	59.0%	68.7%
Visual Studio/VS.NET/Team System	69.4%	57.0%	60.1%
JBuilder (2006 added “Borland Core SDP”)	29.3%	19.3%	13.6%
NetBeans	18.0%	19.5%	23.5%
IntelliJ	n/a	n/a	8.6%
Other (73 responses, see Appendix X)	19.0%	18.2%	11.0%
Total Respondents	601	688	664

** 2004 Study asked the question as: “What other tools frameworks are currently in use within your organization?”

QUESTION 23: How many developers are in your organization?

(please check one)

	2004	2005	2006
101 or more	22.1%	22.1%	23.9%
51 to 100	7.5%	9.1%	6.9%
26 to 50	11.2%	9.0%	10.7%
11 to 25	19.0%	17.8%	18.9%
Fewer than 10	40.1%	42.0%	39.5%
Total Respondents	705	736	737

QUESTION 24: Where do you live?

(please check one)

	2006
North America	61.7%
Europe/Middle East	27.3%
Asia/Pacific Rim	6.0%
South/Central America	4.1%
Africa	0.9%
Total Respondents	737

QUESTION 25: Have you heard of any of the following companies?

(please check all that apply)

Ranked in Order of Awareness

		2005	2006
1	Sybase	n/a	89.0%
2	SlickEdit	50.8%	53.4%
3	Business Objects	n/a	49.9%
4	PalmSource	57.7%	47.6%
5	QNX	42.3%	41.6%
6	CollabNet	37.4%	34.6%
7	Zend	n/a	31.9%
8	Progress Software	31.9%	31.4%
9	Actuate	29.2%	29.9%
10	ARM	n/a	23.7%
11	Instantiations	16.0%	20.3%
12	LynuxWorks	n/a	19.0%
13	Catalyst Systems	19.8%	19.0%
14	ActiveGrid	n/a	17.9%
15	Agitar	13.4%	17.0%
16	Fortify Software	n/a	16.1%
17	ENEA	14.5%	15.6%
18	Exadel	14.0%	15.4%
19	Laszlo	11.0%	14.9%
20	MontaVista	15.4%	14.4%
21	Enerjy	10.8%	14.0%
22	Genuitec	12.8%	13.9%
23	Aonix	12.5%	13.4%
24	Klocwork	9.0%	12.6%
25	Accelerated Technology	16.3%	12.0%

continued….

continued….

Ranked in Order of Awareness

		2005	2006
26	Secure Software	7.5%	9.7%
27	TimeSys	8.9%	8.4%
28	Acucorp	7.5%	8.3%
29	Aldon	3.5%	8.0%
30	INNOOPRACT	5.2%	7.4%
31	Nexaweb	n/a	6.9%
32	Logic Library	6.1%	6.7%
33	Genitech Group	4.1%	6.7%
34	PegaSystems	9.3%	6.6%
35	AvantSoft	5.3%	6.4%
36	eCube Sustems	n/a	6.1%
37	Open Logic	n/a	6.0%
38	DDC-I	0.9%	5.6%
39	Lombardi Software	3.7%	5.0%
40	IVIS Technologies	3.4%	5.0%
41	Scapa Technologies	5.3%	4.9%
42	M1 Global	4.0%	4.6%
43	Simula Labs	n/a	4.4%
44	Omondo	3.1%	3.9%
45	Codign Software	n/a	2.9%
46	OC Systems	3.7%	2.7%
47	Metallect	n/a	2.0%
48	mValent	2.6%	1.9%
49	Optena	0.6%	1.1%
50	Kinzan	2.1%	1.0%
Total Respondents		655	700

QUESTION 26: Any final comments about Eclipse?

Open-ended question. For 97 verbatim responses, see Appendix XI

Index to Appendices

Verbatim Answers to Open-Ended Questions and Questions With an Option to Answer "Other (Please Specify)"

Question Number	Resp.	Question
3	107	Which commercial Eclipse-based IDEs are you using? (Other, please specify)
5	384	Do you have any comments about your reasons for using Eclipse or for not using Eclipse? (Open-ended question)
6	35	If you use Eclipse, or Eclipse-based tools, why? (Other, please specify)
7	72	What types of applications are you building with Eclipse or Eclipse-based tools? (Other, please specify)
8	89	Do you have any comments about the types of applications that you're building with Eclipse or with Eclipse-based tools? (Open-ended question)
13	10	Have you written or are you writing/considering writing any Eclipse plug-ins or extensions? (Other, please specify)
14	128	Do you have any comments about your experiences with Eclipse plug-ins -- good bad or otherwise? (Open-ended question)
18	48	Do you have any comments about your involvement with the Eclipse Foundation or your experiences with contributing to its projects? (Open-ended question)
21	48	Do you have any comments about your experiences with the Eclipse Rich Client Platform? (Open-ended question)
22	73	Which tools frameworks are currently in use within your organization? (Other, please specify)
26	97	Any final comments about Eclipse? (Open-ended question)

Appendix I (QUESTION 3): Which commercial Eclipse-based IDEs are you using?

(Answers to “Other, please specify”)

- IBM Rational Application Developer
- none.
- IBM/Rational tools
- jCompany
- Eclipse Java IDE (custom plug-ins)
- Not sure
- Eclipse only
- None of the above
- I tried the IBM but couldn't add in the Perl module due an error that was unexplained.
- Adobe Flex Builder 2
- Borland Together Architect 2006
- Not using
- none at this time
- Don't use eclipse
- Various commercial Eclipse Plugins
- Yoxos
- myEclipse
- no delelopment inhouse
- Flex Builder 2
- Don't know
- Fujitsu Interstage Studio Enterprise
- IBM RSA
- 1 team is using MyEclipse; (since June 2006) Parasoft JTest; near future (1~1.5 year's time) SAP NetWeaver
- Nexaweb Studio
- Our own ....
- borland together
- IBM Websphere Portlet Factory
- QNX Momentics
- TTCN Workbench
- JBoss Eclipse IDE
- Don't know
- IBM Rational Application Developer
- yoxos
- perforce
- Borland JBuilder
- unknown
- Lombardi TeamWorks
- Sun JDK jre
- qnx Momentix

- Rational Application Developer
- not using
- Don't know
- myeclipse, epic
- Palm
- JBoss
- Not Applicable
- MyEclipse
- easyeclipse-maven-subclipse
- Just the default
- Dont use
- Borland Together Architect
- RadRails
- Open Edge 10.1
- eclipse.org
- Nucleus EDGE
- Rational Application Developer and Rational Software Arch.
- Mostly none
- progress sonic plugin
- clearcase
- one
- not using eclispe
- Not using
- EasyEclipse (package)
- IBM Rational Software Architect
- MyEclipse
- Designer for Identity Manager (http://www.novell.com/coolsolutions/dirxml/designer/)
- myeclipse ide
- NVU and EditPlus and Rational Platform
- MyEclipse
- JBOSS
- only free, no commersial
- Fujitsu Apworks
- NA
- PerCobol
- DO NOT KNOW
- Borland(TogetherSoft) TEC
- NetBeans
- IBM Rational Application Developer, IBM DB2 Developers Workbench
- SlickEdit

Appendix II (QUESTION 5): Do you have any comments about your reasons for using Eclipse or for not using Eclipse?

(Open-ended question)

- Not using Eclipse due to management not willing to have technical staff spend time on 'new' products.
- Microsoft Visual Studio is good enough.
- Open source, very complete and well integrated feature set, low cost, multi-platform, multi-language and many useful plug-ins.
- We're a large organization and it may be that one of our groups my be using it.
- We also use NetBeans 5 for java development, because of the ease of web services integration.
- I like IntelliJ better.
- We are a VisualStudio Shop. If Eclipse supported .Net easily, we would consider using it.
- We're using it for select Rational tools only because they moved to that platform. Since we're a .NET shop, we're used to faster and better performing IDE's.
- Not had opportunity to fully explore its functionality
- Have not tried it.
- Very good tool. It is used by the IDE our company chose.
- Great, open source IDE with lots of plugins and support for our niche area, WebObjects.
- The capacity of my position does not allow me to make those decisions
- Free, easy-to-use, becoming a defacto standard
- Eclipse is an awesome platform...its portable, customizable and truly a staple in my day-to-day existence as a Technology professional.
- We're a commercial software company, don't do open source.
- Happy using our existing IDEs... plus tend to work a lot from the command line instead of in a GUI.
- Good plug-in support. Free/open source.
- Excellent Migrating more and more projects to utilize this build framework and related utilities
- Eclipse is one of the best IDE I have ever used. The ease of plugins installation and development is pretty awesome. I am hoping all my development can be on Eclipse eventually.
- Eclipse provides a lot of tools that used to cost a ton of money for free. This free model is pushing vendors to create innovative features not available before.
- To further elaborate on the point. The error persisted. It seemed to be a module that was required to update any modules. No explanation was given on how to solve the problem either in the message, the documentation, or when I searched using Google.
- There is not enough common information about it. I had heard you had to do something special for it to work with different languages

- It needs a full vi plugin
- Palm Development, Philips ARM Embedded Applications
- Because of the wide use of plug-ins, it is difficult to just get up and get started. I also had trouble locating good introductory materials on how to get started with a language other than Java.
- We also use the RCP as part of our product.
- I was using Eclipse for OpenLaszlo development, as it was freely available, but now I have moved to Adobe FlexBuilder 2 to develop in Flex because the environment suited our needs best.
- The integration with CVS is a major benefit
- Our small team is developing custom GIS (ESRI)applications. The last three versions of the host application, ArcMap, has built-in VBA IDE. It is also compatible with VB6, VB.NET, VBScript and Python. So, traditionally, all my programming tools are dependent on Microsoft (except Python). Being in a deadline-oriented production environment, we rely on paid vendor customer support. I don't want to underestimate the efforts of open-source developers, but with an open-source language it is never sure when, if ever, a bug will be resolved.
- We evaluated NetBeans and Eclipse. Eclipse was a little more mature and had more functionality at the time. We now have an investment in plugins that would make a switch to NetBeans difficult.
- I like Eclipse a lot. It took me a while to get hang of it.
- After years of hype I still don't know what it's for.
- It's not user friendly, too hard to get started with basic stuff.
- I had a lot of trouble getting it to uncompress. It still is not fully setup because of errors durring uncompression.
- The best feature of this framework is how different development plugins have the same 'feel'. Once you know how Eclipse works with one plugin, that is knowledge is immediately transferable to a new plugin.
- It's part of our IBM toolset and extends to our non-IBM tools like JBoss
- Just wish I could use Eclipse 3.2 instead of WSAD 5.02!
- It's price is right. One large project, which used to use JBuilder, cost, and the fact that future JBuilder will be eclipse-based prompted the change. It's been simpler to use than NetBeans, so it was the preferred low-cost platform.
- Open source, flexible
- Actually I prefer to use Netbeans because it's easier to use, it comes out of the box with all the functionality I need, but Eclipse needs a bunch of add-ins to make it operational.
- core parts good and free but most plugins do not blend into overall product design both in terms of UI interactions and design (layout, icons,....).
- We are moving toward a .net solution.
- Beyond my area of expertise.
- The organization is primarily a Microsoft Development house so we use .NET framwork based development tools. Not currently looking at cross platform development but if we do, we will be looking at Sun Solaris and best of breed tools for that platform.

- I've started working with it but got stuck on a perl extension and didn't feel like taking the time to figure it out.
- Provides GUI support out of box which will make our app dev simple.
- Our expertise is half focused on Netbeans
- We have entrenched in-house development tools.
- For one, it is at the core of RAD & WSAD. Other than that, we are seeing other vendors (and we are pushing them) migrating their home grown IDEs over to Eclipse as the base.
- We are not ready at this point to use this type of software.
- We have placed Java in a Software Restriction policy not allowing it to run, and we don't do Java development. Therefore we don't need Eclipse, besides Visual Studio has everything we need.
- runs on linux and windows talks to subversion
- Started with PHPEclipse and slowly moved everything else over including now Borland Builder C++ projects. On IDE for eveything.
- It's THE Java standard platform for software development
- Developers are currently using NetBeans. I'm not sure why they chose NetBeans over Eclipse.
- I just works for many languages and support enough platforms where we as users of computers are not tied to on operating system.
- Support good open source projects.
- My organization is starting to use Eclipse because of the ongoing, largely vendor-independent innovation within the Eclipse environment.
- Eclipse is the best ide for java development at the moment.
- We're an IBM partner, for both WebSphere and Rational... how could we not use Eclipse? <grin>
- Just not looking at this technology product at this time.
- Development has almost ceased: it has been outsourced for the main applications
- I use it at home because a friend who works on a multinational company recommended it to me.
- Eclipse is simply the best! And with great technology projects like Buckminster that helps to download, install and configure my projects I saves a tremendous amount of time and get help to increase the quality of my code.
- We're using netbeans which we see as a superior IDE
- very stable, lots of great plugins including cfeclipse and subversion
- None of the for pay IDE's offer anything significantly better
- Organization does not trust open source.
- These days it's possibly one of the few IDEs that let you write applets in what ever way *you* want, without too much constraint.
- Not Applicable
- Speeds up various development tasks such as code refactoring, executing unit tests and generating boilerplate code (getters, setters, abstract method stubs, etc).
- We are a traditional 3 tier, 2 vendor enterprise (IBM mainframe and Microsoft midrange and desktop). We only use Eclipse to support Java-based COTS solutions. We're not aware of any Eclipse based offering for the mainframe world

of zOS/CICS/DB2, nor are there strong Eclipse-based competitors to VisualStudio 2005 - but Oh that there were!!

- Which it handled MS/C++ better.
- Its the best IDE so far.
- Has a lot of OO features Help J2EE development Plenty of plug-ins available Open source
- Free open source tools, widely accepted and available. Good support for MDA and other J2EE technologies
- Good plugin support, CFEclipse, RDS support using Adobe plugins
- Not sure if it works on our older (SCO Openserver 5) systems. Our developers will investigate it and consider its use as we migrate to other systems.
- free and widely used
- Currently using ASP.Net with plans to move toward open-source tools in 2007.
- Cause it like is totally awesome.
- I see Eclipse as a new standard gaining momentum. Something that will be around for awhile and worth learning to use.
- Very very cool, great developer productivity.
- It's not only one of the richest Java IDE around, but its plug-in model and its perspectives also make a Subversion client, an XML editor, etc... so except for the mainframe folks we've got a single all-encompassing development environment -- our developers like it, in large part because they don't have to familiarize themselves with a new development environment each time they tackle a new technology.
- Best Java IDE
- Freely available. Has most of the features of IDE. Also, there is a large following, with plug-ins. Now, commercial vendors are adding to it (though must pay license).
- Initial migration was cost based. CVS integration, ease of use, and reliability were a key requirements.
- Cross-platform, open source, extesibility, RCP
- Tried it, found significant performance issues in earlier versions, did not like the way that it forced project-based usage.
- Fast, just like it!
- First-class editor functionalities. * First-class Java development environment. * Lot of plugins.
- Price and the ability to create plugins needed tools for development.
- Eclipse IDE and related tools are just great and really productive. We build even custom tools within Eclipse that our developers and our customers are using to work with our application.
- it's the best development tool fo J2EE application developing
- we are oriented to .NET
- I am using Eclipse as Java IDE. It helps me to better analyze the existing code base which has cumulated for more than 8 years.
- By far the best IDE available.
- I don't know enough about Eclipse.

- We use Eclipse because we provide toolsets based on Eclipse technology which is very well appreciated by our customers.
- a great tool, and improving all the time.
- Best free IDE available for java development
- Excellent support for all the java projects lifecycle; many open source high quality plugins
- Actually, we have both Eclipse and Netbeans in use. Netbeans seem to have a better means for GUI development. With server software it's almost the same. Developers can have their personal solution in case of JAVA development tools.
- It is the de facto standard. We are basing our whole Java development line on eclipse, because we can use it in the same constellation in a small project as in pretty large projects. The open source plugins that are available are sufficient, and if we need something, there is a big market for commercial plugins.
- I prefer to use IDEA because this tool is more useful for my 8-)
- We use Netbeans and KDevelop
- Price, Ease of use, covers a large amount of technologies
- For coding C and Java I am using CodeWarrior (on Win and Mac); now I am transitioning to Xcode and VisualStudio for C and to Eclipse for Java. However I have a lot of old code. I am investigating the use of Eclipse for C.
- Excellent IDE, lots of plugins, widely adopted. free.
- developers don't use eclipse but we use rational software management/test tools
- Developing an RCP application.
- It's free and has extensive features.
- We will only use Eclipse when Oracle Corp. migrates their Forms & Reports development toolset onto Eclipse.
- Free, lots of plugins, multi-language
- Powerfull free open source IDE
- Same framework, look and feel.
- It's a good IDE with so good flexibility.
- Our current development projects are re-using existing tools.
- Because it is the best, and free, and has a huge community that integrates it with many things. The commercial Eclipse plugins have not interested me so far.
- It is free, updated constantly, and getting better and speedier with each new release.
- It is a very good IDE and it is free.
- Don't use it
- It is a nice ide but also it is faster than many of the ids's we had prior to moving to Eclipse.
- It is great, extensable, open, has many plugins in the community. And does what I want
- don't know why we should bother
- Encourages best practice: code layout, refactoring etc.
- we seem to be doing less and less java developement
- Best IDE since sliced bread I wish it supported C# as well as Java
- Not using because ours is a Cobol based product
- I love the chnages it is driving, and the refactoring tools.

- Used it since v2 .. originally because it was/is free ! Replaced Jbuilder pretty quickly as IDE of choice due to 'snappier' performance ..
- Quality, Price, Features
- It is easy to use
- Eclipse has lots of plugin options and a lot of industry support.
- We don't make software here. We manage programs and hire contractors to do the work. I don't know 'how' they create their work.
- We have not made the move to use open sourced software yet.
- We have gone down the .net route, and are still pretty glad that we have. We might not like the fact that Microsoft are a big monopoly, but their stuff does deliver.
- Not using Eclipse - too slow
- Free and very good
- We are using Oracle's JDeveloper. It provides a rich development environment with Oracle's Application Developer's Framework, a J2EE framework.
- We do not use Java. Even if we did, we find that Visual Studio is more interesting because of its support for a greater number of languages.
- the best IDE for those of us writing java code (RPG programmers not using Eclipse).
- Eclipse v.3.2 has good functionality, usibility, reliable, performance and supportability and it's free!
- Steep learning curve; I only have time to use what I know already.
- Simple the developers are more productive with Eclipse.
- Use NetBeans on OpenVMS and Visual Studio on Windows.
- Have used NetBeans instead.
- Several people in my group want to develop with Eclipse. For the past 8 months we have planned to design an entire application out of Eclipse plug-ins. However, it was recently brough to our attention that Eclipse is not approved to be loaded onto US Army computer systems.
- We decided to use Eclipse as it is free and we currently only need it for one small Java project - not large enough to warrant purchasing a professional tool. The bulk of our development is C++ and for this we use MSVC++. We would consider using Eclipse more if it were easier to use and better documented - but VC++ seems easier to get into, and is better integrated with the system/tools we develop with (Qt, Windows).
- Other developers in my company were already using it ... to minimize the learning curve.
- Best Java development IDE. Will want to integrate our test tool product with Eclipse.
- I'm not a developera but a professor who works with developers in an academic environment. The developers I know are rather Spartan. They describe Eclipse as an environment that has a lot of overhead and gets in their way rather than helps them
- Eclipse had more features than NetBeans when we started (~2003). Since then, we have continued to use Eclipse.
- Runs on almost anything. Lots of useful plugins. Works well with WebSphere.

- I'm tired of using various tools to do different development effort. It is time consuming to learn and be efficient in all. It is useful to have a single versatile tool which is well understood so the development time can be used more effectively. However, lack of Open Source and Commercial plug-in and compatible plug-ins are forcing us not to upgrade or make us use other tools.
- Eclipse (or Momentix) is a lifesaver. qnx corporation cannot afford to make a proprietary development system from scratch.
- Eclipse provices an excellent environment for server side Java development - and it's free.
- Very limited open source involvement - looking to do much more in the future.
- We're .Net based, and so using the MS IDEs
- I've tried past versions of eclipse to do java programmin on mac os x. Just getting started seemed much more difficult than using C#/VB.NET in Microsoft's Visual Studio.NET. My company has deployed a few x-servers from apple and we were thinking of doing some of our websites using Java and Eclipse to run on the x-serves, but development took so much longer that it just wasn't viable at the time. I still want to learn Java and do some web pages with it and Eclipse, but I realize that the learning curve is much steeper than with Visual Studio and even then from what little I've seen it appears that development will take longer.
- I don't work in development
- we use .Net dev platform and open source (e.g., dotnetnuke)around that
- C/C++ support should be stronger.
- It's often slow and not self-consistent (e.g. feature update available but depends on something that is NOT available). Update and maintenance stinks (I've complained about it, listing 15 legitimate problems, and was rebuked). Startup is too slow. After many upgrades at a particular version it has failed and had to be reinstalled from scratch. What's good about it is the rich variety of tools and add-ons available for a wide variety of tasks.
- The interface is very nice. The open architecture make it very easy fro me to use with multiple languages.
- Java IDE and remote debugging are very good.
- Used for Flash/Flex development for some of the content we deliver. Required by Flex, preferred for older ActionScript work using open source Flash compilers.
- Our corporate office determines which applications and hardware we will use.
- It has become the quickest IDE for Java Development.
- There is a mind set at this organization against open source tools. I, however, intend to start using Eclipse at home and after getting some experience with it there begin to make use of it in a limited way at work.
- We used IntelliJ's IDEA for a couple years, at the time it was a far more featureful platform for java developers. However, Eclipse has caught up and is now a great platform for us to use without the expense of IDEA.
- Working with management to decide overall tools direction.
- We do most of our development in Visual Studio.net (VB.net, C#).
- Open source Java tools
- We use eclipse because it is an enterprise quality development environment. Also, the fact that it is free helps keep our budget in line.

- Excellent Java IDE, almost up to par with Microsoft VS 2005 in terms of features and integration. We tried NetBeans, found it more difficult to use and less feature-rich.
- Eclipse has lots of capability and with plugins can be adapted to most needs. But it is heavy weight and there are some annoyances (such as openning a file that's not part of a workspace/project should have full editor/tool support but it doesn't - seems like this should be simple but the internal model makes implementing this non trivial).
- Next generation product uses Eclipse as a run-time framework, current generation uses native C++ and .net compiles.
- He he... It was the first and only one IDE we tried. And we liked it. We are still using it because it is highly versatile and has fulfilled our expectations since then.
- Whatever increases efficency is what is used.
- I like eclipse because of it's ease of use. I also like the plug-in development because it helps to distribute tools for use in our development environment. I also like the plug-ins that are available to speed up certain development tasks.
- Unemployed
- We have been forced to use IBM WSAD and RAD due to the failure of ANT deployment tasks for websphere to run outside these environments and recurring problems with the eclipse open source version's attempts to deploy to websphere servers. It works great wth other servers.
- Eclipse is an excellent free IDE. I personally think it is better than many I have paid to use in the past.
- We have tested Eclipse. We found the core functionality did not meet our needs. Plug-ins are available, but at a significan cost when you find quality, suitable ones. Sun Java Studio Enterprise 8 supplied better, cleaner, out-of-the-box functionality and was more cost effective once Sun reduced the cost to 0.
- We use Visual Studio 2005 instead.
- No knowledge of the product. We are a .NET shop.
- I need C#, C++, and managed C++ in the same project. I will use it when it can do the same kinds of things I do in Visual Studio. It would be nice if I could also use Eclipse, for writing FPGA code in VHDL, C++, and schematic entry.
- Using MS Visual Studio
- We use eclipse because is a good option for Java development.
- I like its ability to serve as a common IDE across many programming languages and platforms/operating systems.
- We are using Eclipse because of is free, it has a lot of helpful plugins and a great community and foundation supporting it.
- I personally don't like Eclipse. I just don't think like it is structured. I think File | Open, not Window | Open Perspective. Too often, when I open an existing project, Eclipse just spends too much time to open it up. I find KDevelop a much faster and more intuitive developemnt environment.
- The main reason is because it is free. H this were not true! am sure most of my organization would not be using it. Surely the existence of plugins, such as TP-IP. make it an easier sell.

- Eclipse is pretty nice platform, really popular and with good support. Our organization mostly uses IBM Rational ClearCase Remote Client as well as several Open Source plugins like, for example, Pydev. Deploying and configuring the plugins is really easy and this is what we like.
- Very slow when McAfee antivirus is running at the same time. Can sometimes cause eclipse to crash.
- Unknown what it is
- It is a powerful IDE with availability of significant extensibilty options. It provides huge productivity improvement over jdk and editors.
- Preference to intelliJ because of supposed simplicity by some developers.
- Most of our development is done in C++ and C#, using Microsoft tools, but our back-end and embedded systems are moving to a Java environment and Eclipse is the dominant IDE. Currently, our use of commercial Eclipse-based tools is strictly for working in the SAP and WebSphere environment; not enough of our developers are doing other Java development for the company to invest in commercial tools (e.g., MyEclipse) for 'everyday' Java development.
- Excellent IDE, integrated debugger, great productivity features
- strong support for jUnit and other tools makes it a good teaching tool
- I've used JBuilder, and it's not bad, but I find that for most things Eclipse makes 'more sense' as a development tool. Also, the fact that there is a large development community around Eclipse really helps.
- Quite happy with JB!
- we maintain a bunch of old MS ASP VB script based apps but even those we are looking at using an eclipse IDE for the JS aspects of those apps
- It's a great IDE. I prefer it over visual studio.
- We are not a Java shop (currently C++ and moving towards C#). I have heard good things about Eclipse, particularly in the open source geospatial market (it you can call it that) for the uDIG application. If I ever get into working in Java then I will definitely give Eclipse a close look, particularly in the open source tools.
- I use eclipse as my Java development environment because of it's stability, extensive feature set, rapid evolution, and of course the price is right.
- Not involved in that part of the process.
- It easily double our productivity
- I simply want to have another IDE other than NetBeans.
- Curiosity. As an old-fashioned programmer who has always worked in vi, I decided to try something new. I didn't like earlier IDEs for Java, but Eclipse got rave reviews from my friends and I am slowly acclimitizing. Also, since I work in an academic setting, it is nice to have non-commercial IDEs to work with.
- Awesome Cool Stuff (ACS) !! :D
- I like Eclipse as a development platform because it is vendor and technology neutral. It allows us to use a common framework for all of our projects regardless of deployment environment or application types. Being an open source framework, Eclipse is very well suited for using best of breed development tools, which positions it well against other technology or OS specific tools like Visual Studio, which are also good tools, but more restrictive than Eclipse

- What is Eclipse
- We had a large debate in our organisation about IDEs. We decided to go with Eclipse for a number of reasons: 1) Opensource 2) Fits with Rational Products that we use 3) Amount of resources/plugins available etc
- We actually design in Rhapsody 6.1 (a UML tool) and set it to use the NucleusOS and EDGE compiler from its environment, so we do not plan to use the compiler IDE (which is Eclipse) for anything else but the Firmware for the ARM board.
- We attempted it before - but it proved too painful. Expecting to use NetBeans soon.
- Too complex to get started for novice Java developers. Not required for strong Java developers working on small projects. Only used here by experienced Java folks on large projects.
- We used to use IntelliJ idea, but Eclipse has become so powerful, and stable that it's hard to justify paying for idea anymore, so we've switched to Eclipse. The fact that some of us do C++ development and others do Java means that we can all use the same environment. However, we still use Visual Studio for .NET development.
- It is the best java development tool out there.
- It is bundled with an IBM tool
- It works well and the price is right.
- We use MS Access and can get by with VBA and SQL. Can't afford web-hosting services at this time.
- Eclipse has the best feature set compared to other IDEs.
- Not using java for development and non-java aspects of Eclipse are not understood or known. 2) Eclipse changes too fast for us. We are always waiting for the next significant update to try it.
- It provides an open platform for tool integration, this allows us to avoid lock-in from proprietary platforms.
- No Comments
- Visual Studio is pretty much enough. Not sure if we have human resources to handle Eclipse learning curve.
- IntelliJ IDEA is far superior. It is quick, clean, and does not distract from the task at hand. Eclipse UI is hopelessly cluttered thanks largely to its self-congratulatory and irritating rant about views and perspectives. It tries to do do much and apparently no one there knows anything about information hiding and complexity management as they apply to user interface design. Eclipse is bloated and unstable. I tried some plugins for Ruby and Javascript and both crashed. I tried a simple JSP example and I got a spew of wierd error messages out the console and then it hung.
- Good feature set, stable & it's free (in that order)
- Use NetBeans instead
- Specific plugin only available for eclipse
- As a Borland house, we use product specific IDEs.
- Small company, doing C# almost exclusively (and certainly not Java - yuck), and quite satisfied with VS2005.
- Its productive.

- Netbeans is better.
- We are just starting to use it. NetBeans is still more popular. Developers who also use PHP, found a PHP plugin and switched
- Itr is on the docket for evaluation Q@ 07, we just haven't been forced into it until recently.
- Free and there exists many plugins for different uses.
- Eclipse allows us to achieve increased productivity for our developers. It is a multi-purpose developing tool that a company can standardize on.
- I downloaded Eclipse, but so far have not had the time to try it out other than to quickly fire it up and look around. Our company currently does not use a standard IDE, but I'm interested in at least taking a look at them. The main reason I even looked at Eclipse was that it was mentioned in the documentation for tools such as Daikon and Fjalar. Apparently those tools have ways of integrating with Eclipse, but I don't know the details. Our product is developed primarily in Linux, but is shipped to customers in both Linux and Windows. We currently use vi, xemacs, gdb, and various other Linux-based tools for development. The languages we use include C/C++, Tcl/Tk, and Python. For scripting, we primarily use Bash, but I've seen scripts here written in Awk, Perl, Sh, Lisp (for Xemacs functions), and Expect. I don't konw much about our web site development, but I'm guessing they'd be interested in editing support for HTML, JavaScript, and Flash. I suspect a few of the developers here would be eager to use an IDE, while a few others would rather have the option to stick with simpler tools such as vi and shell commands. If I can manage to find the time to get Eclipse to work in our build environment, and if I can get it tied in to some nice debugging tools such as Fjalar, Daikon, or some graphical gdb equivalent, that would probably go a long way towards convincing more programmers to start using Eclipse. The trick is to find the time to get it set up and configured, when I'm already swamped with a bunch of development tasks in progress. I hope to get around to it sometime in early 2007.
- we do so llittle direct development that tools are barely an afterthought. most development is done as minor mods to vendor software, so we use vendor specific tools.
- Free and full-featured
- Bloated, slow, resource greedy and too confusing.
- It is the only IDE that does everything what we want to do today.
- Eclipse is counter-intuitive and doesn'thost the necessary platforms.
- Support, integration with external tools, C# support
- Doesn't support .NET
- IDEs have commoditized and we do Java development in a number of areas of the business. Therefore we use Eclipse.
- It's free and it's good and getter better all the time. The Eclipse based projects are growing and expanding, offering new and exciting ways to utilize it. It also fun to feel part of a growing community with expanding opportunities that you can choose to contribute to.
- It has cutting edge features, it makes me productive, it has a lot of community support and it's free!

- We seldom get asked to develop for Java anymore, and have better IDE's for other languages
- It does everything I need an IDE to do and it is within my price range (free).
- Free cost and availability of plugins. Particularly the ability to easily edit multiple languages in a single IDE.
- Open Plugin framework, powerful available plugins
- Very heavy, hogs memory and is complex to understand.
- touchy, as is most open source. Although the platform seems to offer much more than we are using. As more materials and training sources emerge, teams/product continues to mature I am sure this will be quite a standard. For the time, our core apps will continue on the tried and true platforms.
- We are in the stage project
- WOLips WOLips is a set of Eclipse tools for WebObjects development. http://objectstyle.org/confluence/display/WOL/WOLips
- Open Source, excellent tool.
- Free to purchase. Does a good job. Rich Client Platform architecture. BIRT
- Microsoft Standards such as .Net and their compatability with other Microsoft products are too prevaliant
- Import features for using Eclipse: - Its modular (plug-in) architect. - Good community support available (forums, etc) - Lots of good and cheap plug-ins - Good performance - Good update strategy for plataform and plug-ins
- developing on client-server application with API call that may not support on Java (up to now)
- Its a great technical platform for our tools and an excellent community to work with.
- I don't like the editor. I really like TextPad for editing.
- I use Eclipse because it's free and it's powerful.
- Initially (about 4 years ago) because it was free and I wanted an IDE rather thatn programmer's editor plus command line tools. Now, the quality of the environment, and availablility of high-quality plug-ins and increasing buy-in from 'major players' keeps me from wandering.
- Free, excellent, plugin support, versioning tool support.
- Not sure what Eclipse would be used for.
- It's probably the best IDE for development I have ever used. We actually dumped JBuilder in favor of it.
- Have not had a need to upgrade skill levels and web complexity to include services requiring Eclipse platform functionability
- Good: multi-platform environment (e.g., Windows + Unix + Linux) Good: multi-language IDE (e.g., C++, Java, Ruby) Bad: bloated, ugly design. Way to complicated for what it does.
- We're comfortable with our existing tools
- private: 1) I don't have time to test it :(2) I think, that I have to small memory on my PC (only 512MB) profesional: Currently I support INFORMIX 4GL system for my clients.
- I always liked it, since the inception phase. Now, 3.2 version is better than ever. And most of all, it is FREE

- multiplatform tool good for linux and windows
- Cannot afford not to use that same platform that various customers/prospects are already using themselves.
- For: zero-cost, fully-featured IDE. Fast (much faster than NetBeans) when running. Against: Slow startup, somewhat clumsy workspace mechanism.
- Visual Studio is the only tool which a software developer would ever need.
- I don't know much about eclipse framework.
- We have been moving to NetBeans for Java EE 5 support and generally more all-in-one capabilities
- I don't jump on every bandwagon that pops up. Most of these 'high level languages' lead to bloated, incomprehensible code that disregards efficiency and memory usage concerns.
- rdt,radrails,subclipse
- Happy with NetBeans
- Open source, Lot of plugin development that is useful to the community.
- an open development platform comprised of extensible frameworks, tools and runtimes for building, deploying and managing software across the lifecycle.,i.e., Enterprise Development, Embedded + Device Development,Rich Client Platform, Language IDE and so forth
- We are currently a BEA shop.
- Not there yet.
- We are primarily testing consultants, we have to stay familiar with what our clients use. Eclipse is *not* our preference when we have a choice, but we feel it is important to stay current with common platforms across the industry.
- No employers in NE Ohio need us to offer / teach this product.
- it is vere flexible
- We are using .NET
- Most developers were already using Eclipse when I arrived.
- What is Eclipse?
- We are not actively involved in Software development right now. But we are watching the development of Eclipse for when we do get a software development project.
- PLUGINS, PLUGINS, PLUGINS and more PLUGINS.
- This is the first I have heard of Eclipse. I may look into using it if freeware tools are available.
- I use eclipse for a simple question, Eclipse is a platform, not a IDE like NetBeans and others. I can make several applications and distribut them like a RCP Eclipse.
- Best tool for Java development, easy to install and use, cost-free, availability of plugins for almost every task needed by developers, evolution guaranteed by the Eclipse Community.
- I think Eclipse is exceptional for Java, and adequate for C, C++. I am very, very pleased that i can use the tool for C, C++, Java, Perl, and web development all at once - on both Linux and Windows. I really like the fact that it runs exceptionally well on Linux and also adequately well on Windows, while maintaining the same superior look and feel. The concept of open plug-ins is tremendous and gives it an automatic superiority to closed (proprietary) systems. The software updates tool

in the later editions is very good and helps productivity by minimizing administration. I really like eclipse, and since it is open source (i.e low cost) I find it a tremendous benefit to my small development company.

- no comments for now
- Stay in touch with direction of Java development
- no present requirement for commercial product of this kind
- I really love eclipse, the rest of the company is going to use it in the near future
- Good standard.
- The Java guys use it as a standard in our company. they used to use Borland JBuilder but were unhappy with price and support given. They find Eclipse more powerfull and easier to use. They also use some open source plugins like the JBoss IDE. I love it for the PHP plugins ;)
- It is good for Java development. Also it comes with many useful plug-ins like Guidepost.
- Haven't really tried it out.
- We use MS VS. There is no room for two tool sets in small software shop.
- #NAME?
- Wide support from the open source community.
- Well, it is much better and faster compared to the other IDEs. It also has great refactoring features. Unfortunately it's download size is too big and it is a little bit bloated - as example Netbeans is smalled and with more features. However, Eclipse is still the best.
- Very effective and seems equally as good as other tools if not better.
- Standardization, Integrated environment, Plug-ins, App server independence
- We let the developers free. IntelliJ is the alternative.
- free refactoring tools
- current projects do not need eclipse at the moment, within a year this is bound to change
- #NAME?
- It's simple enough to enable developers to adopt it fairly quickly, and at the same time it has very nice productivity features.
- eclipse is cool...though IBM should make more effort to keep their WSAD/RAD versions close to the latest eclipse releases.
- We are currently using Borland JBuilder as our main IDE, but started using Eclipse because we are moving from a Swing-based GUI to a web based user interface using the Wicket framework. For this we prefer eclipse because of its support for web frameworks and its user community.
- For using Eclipse: -Open Source -Very good -Availablity of plugins, tools,... for the kernel -Flexibility in its usage
- We use a combination of Java and developer tools from Software AG. Software AG is in the process of making many of their tools available as Eclipse plug-ins. We are switching our Java development to Eclipse which will allow us to standardize on a single tools platform.
- I have found Eclipse to be a fairly robust environment. It is not flawless, but it is the closest to that that I have worked with.
- The price is right and it is a good tool.

- very good open source tool
- Eclipse editor does some primitive code slicing. Eclipse is a full-featured IDE that is robust, stable, and free.
- I needed a PHP development tool.
- don't even know what it is
- Lots of great plugins
- Unaware of what it is or why i would switch to it
- free great extensible industry support
- works well; it's open source; on-going development allows Eclipse to adapt to new directions in tool use; price is right
- It is a great product.
- The MyEclipse package and Eclipse is the best tool, paid or otherwise, on the market. It's outstanding.
- Pure .net based shop -- only use Visual Studio
- Our organization is based on Microsoft technologies, but we do develop ocasionally in Java.
- its open source and its free
- We have standardized on MS's Visual Studio. We are using .NET. We looked at but eventually decided against J2EE.
- Flexibility. Plugins added tremendous value.
- It's free, flexible, and gives us great, powerful tools to do things like refactoring, monitor for flaws in code, and provide point-of use API documentation and source browsing for our own libraries as well as the third-party libraries we use.
- Great plugin effort development
- Open source + lots of open source plug-ins
- I am evaluating CDT to find out how useful it maybe as a replacement for VC++ 6.0
- We are primarily a Microsoft .NET/Visual Studio shop
- We feel that the .Net Framework and Microsoft Visual Studio present a better environment for our development needs.
- Excellent IDE. Incomparable flexibility.
- I use it to build java components to interact with our .NET based web services.
- Eclipse has finally matured into a very usable IDE. Once I started using MyEclipse I couldn't see myself going back a manual development environment.
- As a hobbyist programmer, I use eclipse just to try to keep up with the state of the art. Using features/plugins for test and development of small projects of my own. In my company for now all the tools are stand-alone, the first attempt to use eclipse a couple of years was unsuccessful.
- I'm a C++ clientside developer - I only use it for Web servlets. Rest of team use it for Java serverside development.
- Due to the wide range of target platforms, often specified by the customer, Eclipse is not normally useable/useful.
- I'm not familiar with eclipse
- We use eclipse because it is well designed and widely supported. It is extremely cost efficient (obviously) and globally understood. The ability to add functionality

via plug-ins in a matter of minutes (from download to use) is unquestionably increasing our efficiency.

- Great tool!
- I didn't know about it. I need to read about it and see if it something we may use in the future.
- Unaware of Eclipse software applications, and have never heard anything about any of them.
- why bother
- The ability to look under the hood and effectively self-troubleshoot is very attractive and has come in handy a couple of times, but at this point what drives Eclipse adoption is its relative stability, freeness of licensing fees, empowerment of developers to self-select Eclipse distributions and plugins. If developers wanted to use something more expensive we'd almost certainly provision them with whatever tools they want. No sense in being penny wise and pound foolish. In practice, what they want to use happens to be free of charge.
- Eclipse tends to be quite slow on older computers; Netbeans works better on these systems.
- The best Java IDE out there
- Eclipse need too many computing resources. There are other efficient tools than Eclipse.
- Great ide, light weight
- Slow to change.
- IDEs are of no use in our working environment.
- Some do not use IDE's at all, others think of Eclipse as strictly for Java guys, others have drank the MS cool-aid and only use Visual Studio.
- THe kernel developers do not use Eclipse. Many of the developers that are working with our Java-based apps are using Eclipse.
- We program in C#.NET
- not yet
- Our current development environment is not yet supported on Eclipse
- Not using Eclipse currently because of tie-in to vendor specific IDE. Would like to move to Eclipse to take advantage of richer IDE, available add-ons/plugins (e.g. source control), and commercial code quality products and other development tools.
- Theoretically, it's perfect for us, since we use QNX. But even though Eclipse is available on QNX, it won't run without certain prerequisites that can only be met by purchasing a license from QNX.
- My particular usage heavily leverages the Eclipse Modeling Framework (EMF) along with the Java development Toolkit (JDT). The limited but extremely versatile model driven development environment has proved extremely useful for developing in house administrative related tools.

Appendix III (QUESTION 6): If you use Eclipse, or Eclipse-based tools, why?

(Answers to “Other, please specify”)

- CVS Integration
- supports darcs and scons
- ONLY because Rational requires it.
- not using it
- Don't use eclipse
- We don't
- personal development
- Don't know if it's used
- See prior comment. It makes various development tasks faster.
- We use IBM Websphere app server and need the test environments that come with RSA
- Our customer wanted it
- because support will be available for a long time
- It does RCP
- Because it it provided and setup for development with QNX RTOS
- Because that's what our vendor chose
- RAD
- don't use it
- Build RCP / Plugin features
- Note that this is PERSONAL use, not at my company.
- wind river workbench is based on it
- Don't know
- Some tools in use require it.
- The best IDE for JSF
- Because our consulting firm used it to create the apps in the first place
- because it supports test-first programming and refactoring very well
- It is bundled with an IBM tool
- Not using eclipse
- To stay familiar with what others are doing.
- We are not using Eclipse.
- don't use it for now
- it boosts productivity

Appendix IV (QUESTION 7): What types of applications are you building with Eclipse or Eclipse-based tools?

(Answers to "Other, please specify")

- DB/C
- None. Only some design, and requirements analysis, use Eclipse based tools.
- not sure
- Spring, EJB 3
- We ARN'T using Eclipse.
- PHP
- PALM
- php and BI
- Perl CGI
- EPF Process models, web sites
- Don't know
- Suppprting COTS solutions
- ColdFusion
- Sonic ESB services and iWay Adapter Manager agents
- we are evaluating Eclipse
- RPG
- Automated Test suites with IEUnit
- Native OSX
- signal processing
- Grid services
- OS/400
- Portlets
- Jython
- TTCN
- don't know
- WidgetServer
- plug-ins
- BPM
- XSL
- Python
- real-time data acquisition and control
- Don't know
- perl
- cold fusion, ruby on rails, ruby
- PHP
- JSF (MyFaces)
- RMI, EJB
- Python
- unknown
- ruby
- Progress

- Test automation
- SOA
- PHP
- Perl
- PHP
- Flex 2
- PHP
- Not using eclipse
- EJB, JSF, Facelets, Apache Tomhawk
- PHP
- OSGi
- Eclipse plugin
- ColdFusion
- digital library preservation systems

Appendix V (QUESTION 8): Do you have any comments about the types of applications that you're building with Eclipse or with Eclipse-based tools?

(Open-ended question)

- Again, none developed with Eclipse, but some modeling (up to but not including class modeling), noe targeted for Eclipse.
- Many existing apps are written in C++ either using Microsoft or Borland C++ builder. Migrating some of these to use QT and GNU tools and now using Eclipse. This is going to take a long time and it is a long term objective over the next few years to elininate our dependence on technologies which may have no future.
- Tools are maturing. Getting better.
- I intended to use it for writing Perl scripts used in our testing activities.
- I use Zend PHP's IDE for my web editing/scripting /unix script maintenance
- Inhouse and Commercial
- Mostly Web based applications, or using it to port/rewrite applications.
- j2EE
- Client side should be better integrated - perhaps 3.2 w/ WTP will reach this milestone?
- Not using Eclipse
- would like some way to deal with GTK apps
- As well as applications it also handles all the documentation!
- My group is using eclipse to build a web framework that will be used across my entire company. It is based in Java, JSF and Ajax.
- I'd like that the GUI design with VE would be as easy as it is with Netbeans.
- Mainly I am experimenting as a matter of personal education.
- Not Applicable
- Also using Adobe Flex tools (built on Eclipse)
- Documentation seems to be lacking. I like the new framework for web development, but key assumptions seem to be undocumented. Getting things up and running with an existing project is trial and error.
- We are trying to learn about Eclipse, and see if it has any value to us.
- we have a multi lang app, w/ Java, c, and fortran. The c and fortran parts of eclipse are a bit immature, so we have to use another ide for those. The most important missing part is that cdt does not support ms cl compiler / debug using it.
- J2ee Web based applications
- all relevant quality tools (unit testing, metrics, source versioning, refactoring, blueprint support) are immediately available in an integrated environment. it is simpler to introduce organizational concepts this way.
- C Applications and JRE plugings (written in C) for Windows and Mac OSX
- not building with eclipse
- We are building compilers and language translators, so we virtually do not user any GIUs.

- The next version of Lotus Notes (v. 8.0) is also going to be built around Eclipse technology which is very exciting!
- We are building new tools with RCP.
- Desktop apps are Swing based.
- No Comments - I am NEW to this tool set / development environment
- regardless of types of application, they have to work together and not mess up the general menu and tools look when the perspective is not active. The IDE has to be smart enough to only show the toolbar and menu items which are relevant. This is most distracting. As for type of applications: Java, static and dynamic web, J2EE, portal/portlet, other java and web technology, web services, HTML, XHTML, C/C++, Perl, Python, Linux shell (bash),
- We've used eclipse (and MyEclipse) for J2EE apps and are currently heavily using RadRails for a lot of our in-house development.
- critical healthcare applications
- We expect them to be easily developed and maintained, and Eclipse seems to be the right choice.
- Embedded manufacturing control system products, manufacturing execution systems (MES) and enterprise connectors and enterprise portal content.
- Remote debugging with Eclipse is excellent, and is an essential tool. What would absolutely float my boat is if it was possible to develop and debug Flash ActionScript with Eclipse.
- Mostly web servers. Some down-and-dirty data processing applications.
- I would like Ruby on Rails.
- We plan to work on Xgrid applications in the future.
- We are now starting to look at RCP and development has started on an RCP application.
- Only the firmware (which is assembly and a little C code) is done with Eclipse.
- JSF, JSP support is too primitive.
- We found Eclipse to be a perfect platform for out cross-platform network management solution.
- System Monitoring and Management Tools Java based CRM and Sales Automation Others
- -
- ColdFusion based web applications
- We mostly build products and do custom engineering for clients
- lots of xml (jibx, ajax, web services, soa) implemented on J2EE
- J2EE
- need more and faster c++ ide for server development
- Java / J2EE applications
- assets management, claims management, hr, finance, medical, records managements, and so forth.
- We are not using Eclipse.
- J2EE
- We have really only started looking at this
- To support corporate productivity and performance.
- RCP Desktops and Web.

- Not going open source with our software development, which seem to be the requirements for using Eclipse.
- A big system, more than 800.000 lines, in plain C. In the near future we are going to develop for our clients.
- We don't use AJAX yet, but will soon...
- Ruby, REBOL, D, Common Lisp are current development languages.
- Not many J2EE plug-ins available
- for web based applications netbeans has an edge
- not only Java but also Ruby (mostly on Rails)
- see question 5
- CFEclipse rocks
- Mostly desktop based applications.
- Just small tests and personal utilities.
- never used
- The bulk of our work on Eclipse is in Java, but some are starting to work with C/C++.
- Need a road map for the various plug-ins that are being offered.
- not yet
- EJB/Web Services
- The current applications being built with Eclipse are in house tools that aggregate information from source code, organizational structure, requirements traceability, point of contact information, etc. that enable a unique view of a project's status that is difficult to achieve with existing COTS tooling. EMF enables building extremely versatile and powerful applications that are uniquely tailored to a project's specific needs. These specialized applications can be built in a short period of time with minimal manpower (thought the learning curve is non-trivial).

Appendix VI (QUESTION 13): Have you written or are you writing/considering writing any Eclipse plug-ins or extensions?

(Answers to “Other, please specify”)

- We are migrating Servoy to Eclipse
- Never
- I work for IBM. We do. I do not.
- we may do this, if we see a need
- I am personally trying to write plugins - more as a hobby
- Generally no, but it's not out of the question.
- not Yet
- I plan on writing a plugin
- If the need arises we will

Appendix VII (QUESTION 14): Do you have any comments about your experiences with Eclipse plug-ins -- good bad or otherwise?

(Open-ended question)

- GOOD: JUnit integration. Very customizable. BAD: Visual editor doesn't provide a way to position components using the keyboard.
- would like to see an open source ADA plugin (currently we're getting ready to install one from a leading vendor.
- Adequate, but slow. Completely unimpressed with Eclipse implementation. The MODEL of Eclipse seems very good, but java on the desktop is worthless to us.
- The large, extensive API is hard to wrap your head around.
- Plug-ins and RCP are some of the best features.
- No comments.
- What does a clipse plugin do - is this like a netscape plugin ?
- Great stuff.
- Not all plugs in work in all versions of Eclipse. You have to use different versions to use tools you want.
- The Eclipse API is very extensive. It is often difficult to discover the correct API to use to accomplish a task. The newsgroup is a good place to ask questions about the Eclipse platform API when the docs are not sufficient.
- The open source tools are wonderfully rich development environments to work within. Excluding the proprietary eclipse plugins, no proprietary development suite is in the same league for development.
- They work out of the box.
- Some are better than others. ClearCase and MKS plug-ins aren't as nice as they could be, both have short-comings. WTP prior to 1.0 release didn't refactor scriptlet code with Java code.
- Frequently difficult to install/manage. Management of different (but compatible) versions of the same plugins is not well managed. (E.g. MyEclipse installs part of the Callisto project, but so does Callisto, and then the updater throws a warning on every update.)
- Tried to install the perl plugin and never quite figured out how to make it work.
- like the subclipse plugin a lot
- Most are pretty good. They offer the basic functionality of the third party application they interface with, i.e. source control environments. The benefits of using plug-ins is that you don't need to have multiple applications open to accomplish the same tasks.
- good. We like most of the plug-ins. They are well written and always work.
- It has a rich (read 'dauntingly large') feature set.
- It seems to be easier to install and use them. A time ago, it was rather complicated.
- Not Applicable

- I've tested a few plugins such as xml editors, a source code formatter, some jsp editors, a vss plugin and maybe a few other misc plugins. Out of all the plugins I've tested, every single one significantly reduced the stability of the Ecilpse platform. So I don't use plugins at all.
- mostly good
- Rational XDE plugin is pretty good
- We are in the process of rewriting our Development tool Servoy, a powerful Java IDE to build and deploy smart and web client applications, to Eclipse. Servoy is used by over 7000 developers worldwide and over 1200 applications have been developed in Servoy .
- Plugins tend to be lacking documentation.
- Exemplary tool support and example plugins made quick adoption easy. Expert use required detailed understanding of Eclipse architecture which was painful to come by.
- Good in some, bad in some, depends on originator of the plugins
- While the Eclipse plugin community and availability is very big, the plugin-management mechanism in Eclipse is not very good. The Update Site can be troublesome from times to times, Uninstalling plugins is very hard if not impossible, the usage is not intuitive, etc. In the end-user perspective, the experience with NetBeans plugin mechanism is a lot more appreciated.
- Eclipse plug-in architecture provides a great way to extend the base IDE with our tools and provide our customers an integrated environment to work with our application products.
- Good
- Good BUT some compatibility issues (NOT DESCRIBED) between Eclipse versions!!!
- sometimes the plugins do not play very well together. Most notedly, I have been unable to stably run Borland Together and AJDT at the same time - Eclipse keeps crashing very frequently. But in general the feeling is positive - there are great plugins out there.
- good experience
- most are very good. sometime we found some unsupported or not very mature plug-in, but until now they are few.
- Mostly good, sometimes I don't understand the differences of old (pre 3.x) plugins, 3.1 and 3.2. But there is a lot stuff to look it up.
- As they are scattered through the web they can be hard to find
- Mostly good. We occasionaly have a little difficulty with the SVN plugin. But not being experianced at SVN before this I don't know if it's SVN or the plugin.
- There have are stability issues with Swing GUI tools
- the installation mechanism could do with being standardised. the plugins tend to be all packed in different ways. some plugins are only available if you have a web connection (eg. callisto) and it would be good to always have an alternate install method as not every corporate employee has direct web access.
- Tutorial information is quite hard to come by at times. The eclipse books are ok but often don't get down to the real gritty details.

- The product/feature management is fairly complex - you get a product that will run fine in the IDE and spend days getting the dependencies correct so that it can be exported as an exe.
- Very good. The main issue I've detected is that in some cases Plug-ins are heavily clouped with the runtime version of Eclipse. So, if you are interested in several plugins... you have to have available in your platform severals Eclipse runtimes
- Would love a Oracle pl/sql development tool
- Acceptable :)
- The resources and files are a bit trickier than just Java, and it would be nice to find a conversion from standard file to Eclipse IFile
- hard to learn to do much more than the basic stuff.
- have not done enough yet to comment
- Many not ready or prime-time
- All plugins should be made to work the the URL update. Unzipping files for plugins should be banned.
- Most of the eclipse documentation seems geared to developers who write plug-ins for eclipse. We just want to develop code with eclipse - not for it.
- No Comment
- Each release seems to break critical aspects of plugin development, and that causes a lot of problems.
- bad. but I have high hopes that the community and commercial offering work together to make it work. Learning new tools is extremely time consuming and unproductive. If Eclipse delivers on its promise of abundance and compatible of plug-ins then it is fantastics. I have not seen it yet. Open source plug-ins don't behave and the commercial ones don't let any other plug-in to work with them. disappointing.
- Clarification: we use C++ plug-ins
- Mostly good. A few are a little immature, but nothing unexpected.
- Painful. Internal interfacses are inconsistent.
- Exadel is excellent.
- The ease of creating, distributing, and using plug-ins (both internal and third party) are excellent! Eclipse is doing very well with this functionality!
- Too expensive and quite annoying to have to 'track down' and install functionality that should be in the IDE out-of-the-box.
- Still trying to find time to figure out how to write them.
- Your mileage will vary. Some are good. Some are horrible.
- Most are good. the updates are nice. When they work, they work. When they don't it is not cool.
- Eclipse is nothing without it's plug-ins. It's a huge pain to have to assemble a set of plug-ins for a specific type of development. There are may holy wars about which plug-ins to use and how well they integrate. We will use it to build a tool chain for our product but to as an IDE it stinks. I would rather by a best of breed IDE like VisualStudio or Netbeans and just do the work. Too my developers spend too much time building an Eclipse IDE and not developing.
- Great way to extend features Drawback - new Eclipse version sometimes incompatible with old plug-in

- I've been pleasantly surprised at the quality of Eclipse. For an open source project, the quality of the code is excellent and I've found very few bugs (and no more than I've found in other commercial products).
- I wish I had more time in my day to learn more about plug-in development.
- Good.
- So far, our experience with Eclipse plug ins has been largely positive.
- Not required for current projects
- They generally are of high quality, however, I find that plugins tend to be so different from each other (in terms of look and feel, and interface), that there is a learning curve to each one.
- Fairly steep learning curve, but you get a lot of functionality out of the box.
- All the open source plugins I tried were rubbish that either crashed Eclipse or 'lost' their ability to operate normally after an hour or so. By 'crash' i mean that the operating system process in which Eclipse was running vanished.
- Not bad so far, but our experience is limited
- Overall, our experience developing Eclipse plugins is excellent. There is certain learning curve that took impact on our initial development, but once we reached certain level development becomes 'blitzing-fast.'
- Good
- Do they work?
- Experiences range from very good to limited use, depends on type of plug-in. Would be great, if it would be ensured that a wrongly coded plug-in could not impact start of Eclipse itself.
- e.g. svn plug-in is very poorly thought out - branching UI in particular sucks big time.
- Can be problematic to find what you want, and to make sure they will all work together
- Good.
- All good.
- Generally good.
- I am having trouble with Eclipse, CDT, cygwin -- debugger doesn't work, can't find/link libraries... I am using the VI plugin -- it has saved my sanity (if it were a full implementation I would be happier)
- some have compatibility issues (running plugin version X on Eclipse 3.2.1 may require 3.2.0 install for older plugin versions.
- Dependency management can become a nightmare.
- Gnerally good. SOme lack of consistency accross the editors (e.g., use of soft tabs must be set in each editor)
- good could be better
- The main problem is trying to coordinate versions of Eclipse and the plugins. A plugin will work on 3.2, but not on 3.0, and the vendor based their product on 3.0. Here's a concrete example I've recently had - IBM RAD 6.x (Eclipse 3.0) and IvyDE (Eclipse 3.1). I had some problems with putting some plugins into the Designer for Identity Manager. They were caused by different dependecies between some plugins. I cannot remember the actual example, but it went something like this (simplified version): - Web Services designer needs version A

of GEF. GEF version A needs version B of EMF. - UML designer needs version C of GEF, that works only with version D of EMF. - I end up with 2 Cclipse installs, one to work with UML in, and the other for web services. DLL hell, just in Java world. Another problem is that commercial vendors are a little behind in the version of Eclipse platform, and the plugin writers are on the edge. If they could meet halfway, it would be great. For a while, I was hoping that JSR 198 might help a little with this, but I don't think Eclipse will implement this JSR (and knowing some other JSRs, this one may be useless anyways). Maybe the new JSR 277 will also help with internal plugin versioning, but then again, maybe not.

- all good
- Our experience is a mixed bag, but the fact that there are so many options we view as positive.
- We are not using Eclipse.
- Mostly good. Tried some perl plugin a while back (EPIC) and it was pretty bad.
- Make Eclipse plug-ins is not simple, because you need understand Eclipse architecture. I think we need visual tools to help us. I write this, because in Brazil(where I live) there is a big moviment to adopt Netbeans. I have been a moviment to adopt Eclipse, but it is not easy.
- Very pleased, I enthusiastically support the whole concept of open plug-ins.
- they do not interact well, and one loses them all with an eclipse upgrade.
- Love the PHP plugin, although I wish it had shorter release cycles. One comment I have is the dependencies of the plugins. I like to download them as a package instead of using the built-in installer. I find that if I download them there are lots of other dependencies I need to download too and sometimes it is not that straight forward which ones. I know there are pre-built Eclipse deployments (with the plugins) but none that have multiple plugins which I might be using.
- Some are good and some are not upto the mark.
- Seem quite good.
- Very memory hungry. Tends to perform slow when RAM is at premium
- installation has recently improved a lot
- just started: not much experience so far
- The plug in architecture is one of the features that makes Eclipse so attractive.
- no comment.
- as always, some are good, some not. Have to look and analyse before installing/using.
- Some are better than others
- If a vendor supplies both a rich client and a plug-in, it is very difficult to tell which one is being used... Is there a way to make this more apparent? E.g., IBM/Rational has both an Eclipse plug-in and a rich client for their ClearQuest CRM product. Both are installed during a ClearQuest installation. Once you are in Eclipse and have started either the plug-in or the rich client, it is easy to forget which one you are using and there is no easy way to tell without closing it and (re)opening the one you need.
- Good.
- Not very stable with older versions
- hard to code but doable

- We haven't had any negative experiences with Eclipse plug-ins.
- I always feel I need more desktop using eclipse, I like to have all 'at hand' :-)
- Just started using them, so just coming up to speed on their full usefulness. So far, however, I am very impressed with the ease of integration, having had less than optimal results from other tool suites in the past.
- It is extremely hard to remove/identify all plug-ins installed. Once installed, plug-ins don't always show up in the management console and/or do not allow easy removal. It is extremely hard to back out some plug-ins.
- Good
- On the whole, we have had excellent experiences with the plugins we use. They are the major strength of Eclipse.
- Still looking for a really good, cheap or free, WYSIG Java GUI Builder.
- not yet
- My experience with the: - JDT has been extremely positive. - Plugin development has a non-trivial learning curve to use effectively (thought the concepts are simple enough). But worth the learning curve. - EMF development has required a lot of time and mistakes. It requires a strong background in OO to even think of starting out on this track. The book is very good but EMF is so encompassing that it is easy to get side tracked. But once some level of mastery is obtained the quality of the apps that can be built in a short period of time will alter the way your company looks at building/buying tooling.

Appendix VIII (QUESTION 18): Do you have any comments about your involvement with the Eclipse Foundation or your experiences with contributing to its projects?

(Open-ended question)

- Unfortunately Not much free time at present
- We are planning to participate more.
- We recently joined so we don't yet have an opinion.
- We actively submit bug reports, and contribute patches when we discover problems that block our progres. We are looking into the BPEL related activities and may contribute formally if this proves to be a strategic move for us.
- Dislike the way PHP-IDE a commercially biased PHP framework is being promoted over the much more established PHPEclipse. 'Money' should not be a reason for trying to kill off established open source projects!
- Not Applicable
- We plan to participate/contribute to the DSDP project.
- Would like to contribute more if time permits
- I work for Oracle, so I'd be surprised if they aren't involved somehow, but I don't know anything about that involvement.
- I may get involved next year.
- Our contributions have been relatively minor, but our experiences have been positive. Hopefully we can be more involved in the future
- Would like to become more involved in the future.
- Most of the online documentation related to Eclipse strikes me as recruitment for more Eclipse developers. Most of the 'help' documentation for newbies is written from the point of view that the new user should aspire to become an Eclipse contributor some day. The problem report I sent in resulted in request for information that is really only the business of an Eclipse developer. This disease seems to inflict most open source projects. I'll happily send $500 on IDEA because it saves me 10 times that in lost productivity every year.
- I would like to use eclipse but it's just too slow compared to Netbeans
- I use Eclipse as a hobby enabler. I core dumped SWT once and got a reproducible bug submission! I tend to be good at breaking things.
- If I came across a bug or decided to add a feature, I would participate in the eclipse project. As neither of those situations have occurred, I have not gotten involved.
- Sometimes works slowly, but it works!
- The Foundation is the glue that holds the open source and commercial elements of the ecosystem together.
- I have a severe bug in Eclipse Ant support that keeps me from moving to 3.2 final but since not many others are experiencing it I may have to fix it myself (on my own time, when I get around to it).

- Eclipse is quite stable and the features are numerous, so there is no pressing need to contribute at the moment.
- i should done more to contributing
- Would like to at some point. But we do not have the resources allocated to it currently
- Very minimal participation because what we do is mostly evaluation... mostly we try to read and keep current. We are primarily a testing organization, we only develop to stay current or build in house utilities.
- We are not using Eclipse.
- I have been involve in ECF(Eclipse Communication Framework). I have been development a plug-in basead in ECF, to help distributed teams. http://wiki.eclipse.org/index.php/Shared_Code_Plugin
- We are based in South Africa and don't know of any happening here.
- sw is not our core
- no comment.
- I'm brand new to the tool so I don't contribute today. I could see contributing at a later point in time.
- not yet, sorry, we are evaluating...
- Do to the extremely open nature of Eclipse certain companies (like mine) strongly discourage any direct collaboration with the open source community, because the participants are unknown. So the approach to using open source is more one of an anonymous user or, if some form of collaboration is required, then a US based company is contracted to limit the direct interaction with the open source community.

Appendix IX (QUESTION 21): Do you have any comments about your experiences with the Eclipse Rich Client Platform?

(Open-ended question)

- Doesn't come close to Microsoft's .NET rich client development platform.
- Very feature rich platform, but also a very steep learning curve for new RCP application developers.
- It gets better each release. Please keep it up.
- I use the Eclipse RCP whenever I can.
- Not Applicable
- Early adoption was difficult. Refactoring for RCP was outstanding. Tool support is improving and love every bit.
- Very interesting area and we hope to see it develop further in the near future
- Quite a steep learning curve.
- It rocks, so much built in functionality, so much that can be imported from existing eclipse plugins.
- mainly done plugins, not specifically targeted RCP
- I have written a couple of prototype applications to explore whether or not it would be interesting to develope commercial products. We have to see how such a tool will fit into our long term product line goals.
- have not done enough yet to comment
- No Comments
- too slow and fat. should use OnDemand loading
- It's something I would really like to explore if/when the opportunity presents itself.
- We build clients and pluging at the level just below RCP because most of what it provides is not flexible enough for us. e.g. SWT needs to be replaced in our apps.
- We are using Flash for Rich Client Platforms. This is largely because of the products we build. Eclipse is being used primarily for server-side communications with Flash clients, and standalone Java development.
- I wish I had more time in my day to learn more about rcp development.
- Just starting to use it now. Looks to be very powerful and should reduce our code production costs as well as bring a uniformity of look to our Front ends applications.
- I spent almost an hour downloading, unpacking, and installing Eclipse. How can I use this as the basis for a small downloadable program to be used by novice computer users?
- Only
- As my work currently is split between java and flash, it would be nice to see better Flex2 (MXML and ActionScript) support without the need of an expensive adobe plug-in.
- Excellent platform that reduces our need to build underlying foundation code.
- no experience.

- Im currently not aware of the 'Company' whereabouts. In our team we will decide on using the RCP very soon, but Swing is not written off yet.
- We are not using Eclipse.
- ERCP is very powerfull, but it is not simple.
- We do plan to build RIA applications using Canoo ULC. If the need arises to build rich client applications we will consider both RCP and the .NET platform.
- We have our own framework for building rich client applications.
- no comment.
- It's great.

Appendix X (QUESTION 22): Which tools frameworks are currently in use within your organization?

(Answers to "Other, please specify")

- Access Development
- APEX
- AS400 SEU
- bitweaver
- Borland C++ (BDS)
- Borland Delphi
- CodeWarrior
- Codewright for general source editing
- custom MVC
- Delphi
- Delphi
- Delphi
- Delphi / Borland
- Delphi Win32 and Delphi.Net
- Dreamweaver
- dreamweaver,notepad++,scite
- Emacs
- Emacs
- emacs + gnu toolchain
- emacs, kdevelop
- Hibernate, Spring
- Homebrew
- home-made
- IBM iSeries SEU
- Informix Based development
- JCreator Pro
- JDeveloper
- JDeveloper
- jDeveloper
- JDeveloper
- JDeveloper
- JDeveloper
- KDevelop
- KDevelop
- mainframe based
- MPLAB IDE and tools
- none of the above
- Open Edge
- Oracle
- Oracle Forms
- Oracle JDeveloper

- Oracle JDeveloper
- Oracle JDeveloper, ADF
- os/400
- others unknown
- PowerBuilder
- proprietary
- QT
- RAD
- Rhapsody 6.1
- Ruby on Rails
- Servoy
- Source Insight
- Sun Studio 11
- Sybase PowerBuilder
- TI Code Composer Studio
- Unix tools stack
- various
- vi/sccs
- Visual DataFlex
- Visual SlickEdit
- Wind River Tornado
- WindRiver Tornado
- WSAD
- XCode
- Xcode
- Xcode from Apple
- XCode, CodeBlocks
- zOS/CICS/COBOL

Appendix XI (QUESTION 26): Any final comments about Eclipse?

(Open-ended question)

- Have developers that are aware of it, wanting to use it, and are told that it's not going to happen here.
- Eclipse seems to be running very slow some times when anti virus is on.
- Nice idea, but until it can perform as well as native windows forms apps, weak.
- Love it...
- Can I get a free non-expiring evaluation copy ?
- It's a great platform.
- I don't like setting it up on systems.
- great work, make a default black background configuration (all text is color on black, opposite of the white background version - a la Borland)
- A nice toolchest, but really cluttered, slow and complex
- I used to work with Eclipse and it was awesome and I appreciate the development the platform has taken over the past couple years. I hope to get a chance to use it in the near future. I have personally been monitoring its progress (but not monitoring it for my current job.)
- it's a good tool, but not my preferred
- Needs to go away.
- Lets not destroy an open source base by promoting only commercially based extensions
- I know the Java guys use it, but as they're pretty much a group to themselves, don't know how they use it and with what extras. The non-Java developers don't use it.
- It would be nice if the Mac OSX support wasn't so patchy for Intel (ie CDT only has PPC binaries)
- Looks like I better find out more about Eclipse!
- Eclipse enables us to get developers up and running with a minimum of effort.
- Keep up the good work and keep it open source please.
- I'm lovin it!!!!
- one of the best things that have happened on the Java platform
- Eclipse is causing eclipse on other commerical and open source/free ides in terms of user base and popularity.
- Great platform!
- Keep up the good work!
- Really nice product to use, just needed to adapt from NetBeans and missed the integrated GUI developer
- I think this tool is great but as I found out I cannot think in 'IBM-style'
- Not sure 3.2.1/JRE1.5 is quite as stable as 3.0/JRE1.4.2 Eclipse never used to crash. It is only when you stop using eclipse and go back to something else you realise how good eclipse actually is.
- All the best!

- Keep the fantastic work.
- Continue the good work.
- Who cares?
- I've used NetBeans for years and Visual Cafe before that. What really attracted me to Eclipse was the community that has grown up around the platform. As a result, integration with new frameworks and tools is delivered through this community much more quickly than with NetBeans, and I don't have to fear that Eclipse will fade away any time soon (Visual Cafe).
- I wish I had designed it or conceived it... its that good
- your survey is limiting for me as we have over 4000 developers and 14000 employees. The survey questions are too focused for this size of organization.
- It's great!
- Wish I had more time to learn to use it effectively.
- I found it easy to use & intuitive. My code was up & running on the embedded target in less than one day.
- The best think that has happen in my professional life as tools go!
- Please make it work well, fast and small. Plug-in compatibility between open source and commercial is crucial. Let everyone play. make intelligent use of the menus and toolbars based on perspective selected. If I'm not doing any PHP coding and PHP perspective is not open, I don't want to see its tool button and menu items. (this is just an example)
- Startup time and memory usage MUST improve... at least in a Windows environment. Also, better support for startup options (i.e. parameters) should be provided.
- Love it, keep up the good work!
- We're primarily a .NET2.0 organization delivering a highly-scaled application to millions of students. It's nice that there is such a large infrastructure around Eclipse, but it still feels like a science project to put together a robust development environment.
- It's a great java editor but fell short somewhere in the WST/deployment to websphere area.
- Needs support for .NET and Mono.
- I love it and hope it continues to evolve!
- Not my IDE of choice
- compatibility management and upgrades need to be fixed. java performance management is very poor web integration is very very poor (netbeans is hugely better, and eclipse has not caught up to where NB was 3 years ago IMHO)
- Great IDE!
- Great platform We plan to keep using it for a long time
- Is there an Eclipse plug-in allowing a developer to build wizards in a web application (i.e. modal dialog box behaviour) ?
- I use eclipse as the required IDE in courses that I teach in introductory programming and programmer testing. We teach test-first programming to first-year programming students. It is challenging, but Eclipse makes it MUCH easier for the students and for me. Similarly for the advanced testing course.
- Keep up the great work.

- It takes significant effort to train on the platform and to understand the variety of plug-ins and their effectiveness. It would help if there was a core such as web applications where you could get training updates or developers notebook type of thing. For example, if I want to use Ruby on Rails is there anything in Eclipse, how do I find out and how does it work. With PHP is it worth learning how to use it in eclipse or should I just use a text editor.
- Have not fully convinced everyone in our organisation to use Eclipse but we are getting there. Have only recently become aware of the plethora of Eclipse extensions and are looking forward to investigating their usage.
- I'm a fan of open source, but for my IDE, I want what's best, not what's cheapest. I was skeptical when we started using Eclipse, but I now feel that not only is it cheaper than the alternatives, but It is higher quality as well. I also love the flexibility that we have with Eclipse. I'll always have a place in my heart for IntelliJ idea, but I'll probably never purchase a newer version... Eclipse has just gotten to good to justify anything else. We need C#/mono/.net development tools for Eclipse, then we could develop solely on GNU/Linux, and get rid of Visual Studio.
- great tool
- Reduce visual clutter. Better version management of jar files. Shrink base download to under 1MB and depend more on the built in updating features.
- As I understand it, Eclipse is written in Java, so I'd expect it to be slower than native tools. Also, we have no cross-platform needs. So while there's an intellectual interest in an open-source IDE, it really has no relevance to us.
- Good, but might be better. Netbeans is a better choice these days.
- You get what y0u pay for.
- I think it is great. I was acquired by IBM in mid 2001, and when Eclipse was open sourced, I figured it would be a good career move to become familiar with it. I have worked with Symantec, JBuilder, NetBeans, and Eclipse, each in turn. I enjoyed all but Symantec. And each of the first two seemed antiquated or cumbersome after using a subsequent IDE. Probably what you are used to! NetBeans and Eclipse are just different, perhaps due to the base of SWT versus Swing.
- Thank you!
- Good environment - beats even emacs in complexity and resource hogging!
- Eclipse seems to be gathering an increasingly large support base.
- Good stuff.
- All around great product!
- I believe it should be possible to improve the memory footprint. Switching perspective could mean discarding items/views not in use.
- Screen looks very good :)
- Never used the software and I have never been interested in the software. I have only heard of it in relation to Java 'programming' and since I am not interested nor ever will I be interested in Java 'programming' I have never checked out how the software would be useful.

- It's way to bloated. If I wouldn't have to use it because of company policies, I simply would shun it. SWT is also a big negative. On any platform other than Windows, Swing is preferable.
- only solution that makes sense
- Keep up the good work.
- I think it has an important place in the industry. We were hoping that Borland would build a best-in-class IDE, but that obviously wasn't in the cards. In general, even when working with Java, we would prefer something else, but it's what we've got.
- We are not using Eclipse.
- It looks like it can do a lot currently using Dreamweaver for web development, but if it integrates SQL and PHP pogramming I will consider it. Looks like something I should know about.
- It is a great platform, but we need to show it better. By instance, in Brazil people think it is only a IDE, and I believe around the world too. We need to show the powerfull of Eclipse.
- We all love it! ;)
- Just about the best thing since sliced bread. Not joking.
- I expect us to be moving to Eclipse from JBuilder in the next 12-18 months.
- What I've seen is great, so far. I hope I have more time to explore the plug-ins that are available and to learn more about Eclipse.
- I blogged about our experiences with it back in '04, at http://www.oreillynet.com/onjava/blog/2004/06/ive_been_eclipsed.html
- great tool
- With my limited knowledge of Eclipse as an individual, I can realize how good and helpful it can be for our development team.
- As noted above, I am a very new user of Eclipse, and have a lot to learn, but so far I am impressed with the quality of the product.
- Who or what is Eclipse? Does this company have web development software?
- Keep up the good work!
- + Brilliant concept; + maybe most responsive Java app I've ever used; + wonderfully cross-platform. - would like to be able to integrate the to-do list w/ my 'real' to-do list (Outlook; Palm; etc.) - would like a decent, WYSWIG Java GUI (Swing; SWT;
- Big, slow, and unwieldy. Yet it has pretty much taken over. Go figure.
- I believe Eclipse will be one of the key enablers that will drastically alter the landscape of how software is developed. To me it feels like the days when backplane busses (i.e. VME) provided a new way of developing computing platforms by lowering cost and speeding/opening up the combinations of ways that computing hosts and peripherals could be aggregated. And when the IBM PC provided a new way of developing software solutions by lowering costs and speeding/opening up the combinations of ways that software applications could be hosted.