

# BeanExplorer™ White Paper



What is BeanExplorer?  
How does BeanExplorer cut development time?  
How does it work?  
Where it can be used?  
System requirements  
BeanExplorer 1.2 Feature Matrix

Download a free evaluation version: [www.itcsoftware.com](http://www.itcsoftware.com)

Distributed by:

ITC  
Software

ITC Software  
Tel 978 287 4855  
[www.itcsoftware.com](http://www.itcsoftware.com)  
[info@itcsoftware.com](mailto:info@itcsoftware.com)

## What is BeanExplorer?

BeanExplorer™ is a tool that dramatically reduces the costs of implementing data processing applications. The product consists of two parts:

- The Methodology to develop applications, and an
- Extensive framework to implement data-driven applications

## How does BeanExplorer cut development time?

BeanExplorer helps to achieve tremendous time savings in the development of user interfaces (UI), necessary to process large amounts of data. BeanExplorer's unparalleled ability to generate rich sets of UI's with almost no programming effort is unique and currently not found in any other development tool. Importantly, the BeanExplorer can generate UI's for new data still unknown during the development process. BeanExplorer allows you to effortlessly adapt your IT structures under the pressures of today's rapidly changing business requirements.

To illustrate this in more detail, let's consider the typical stages of a software development project (summarizing most software development methodologies):

*Business/Requirements/Domain Analysis* – BeanExplorer's approach to build applications from definitions of data saves an enormous amount of analyst time. This includes defining how the data needs to be processed and what screen, printing and web forms must be implemented to process this. Sometimes preliminary analysis time can be simply omitted, i.e. programmers can start coding immediately and generate frequent releases (a trivial exercise with BeanExplorer) and obtain immediate feedback from users and customers gradually improving the delivery system.

*System Development Architecture* - BeanExplorer provides standard blocks for most parts of an application's graphical user interface (GUI), so that the architect can primarily focus on the application logic or "model". BeanExplorer encourages the use of a component-based design and Model-View-Controller design pattern in GUI application development and provides a ready solution for the View and Controller part – using a set of powerful *Property Inspectors*. This approach allows project teams to focus mainly on the custom application logic, thus **cutting the work on system architecture by 30-70%**, depending on the application.

*Coding* – Since the development of UI's is generally the most time consuming part of GUI application development, BeanExplorer's ability of generating UI's can **save up to 90% of the time required for coding**.

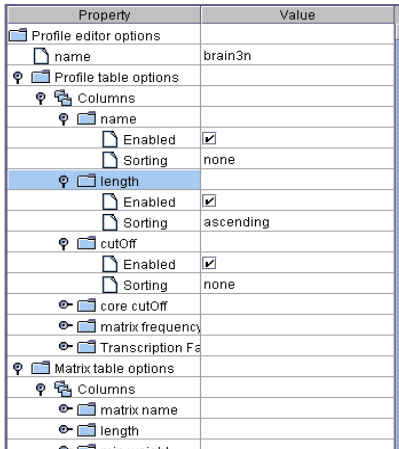
*Testing* – Historically, functional testing is a time consuming, expensive and complex task, especially in environments which require frequent changes. Since the BeanExplorer UI's are generated automatically, the amount of work required for **functional testing is decreased by up to 90%**. The testing team can focus on the development of regression tests, helping the software to meet challenging business requirements.

BeanExplorer fully utilizes an incremental approach to software development. In any business environment the IT team must perform substantial work to develop a basic version - even one with minimal functionality - of the software product. (Specifically work on analysis, modeling/prototyping, etc.) In contrast, BeanExplorer offers a different and far more effective approach: It's pre-existing software components are capable of accommodating your business model at run time – all you need to do is to gradually add features defined by the requirements of your business and competitive structure.

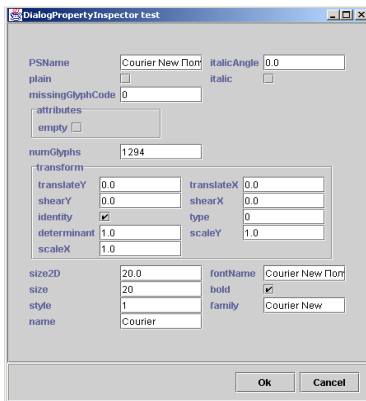
BeanExplorer's approaches are based on an Object Oriented (OO) paradigm which leverages your investments to implement modular and distributed systems.

## How does it work?

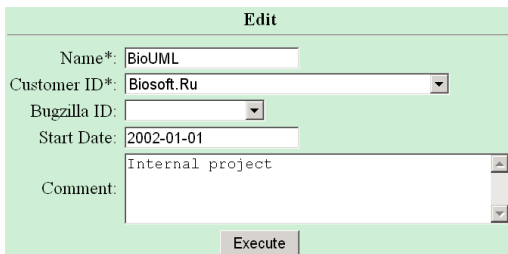
BeanExplorer operates on top of BeanHelpers - an extension of the standard [JavaBeans™ technology](#). This is *the only component architecture you should consider if you're developing for the Java platform*, according to Sun Microsystems. Components are self-contained, reusable software units that can be visually composed into composite components, applets, applications, and servlets using visual application builder tools.



The Classic Property Inspector allows developers to build conventional tree based interfaces which are used to implement dialogs, where the



processing of many parameters is required and the exact list of parameters is not necessarily defined.



JavaBean components are known as **Beans**. BeanHelpers add extra functionality to JavaBeans which are necessary to build modern, feature-rich applications with a consistent, data driven interface.

Once application developers have implemented all the necessary logic in the form of JavaBeans (with or without BeanHelper extensions), BeanExplorer handles the rest – it extracts meta information available on JavaBean components and uses it to generate various UI's for standalone and/or web-based applications. Conceptually, UI's are based on the rich set of *Property Inspectors*.

### BeanHelper's features

- Dynamic (calculated at run time) property attributes
- Composite properties
- Array properties
- Components with dynamic set of properties
- JDBC to JavaBeans mappers

	name	length	cutOff	core cutOff	matrix frequency	Transcription Factor
0	ISTTK69_01	8	0.8	0.8	0.7	Itk 69K
1	VSEVI1_06	9	0.8	0.8	0.1	Est-1
2	IKCF2II_02	9	0.8	0.8	0.8	CF2-II
3	IKCF2II_01	9	0.8	0.8	2.4	CF2-II
4	YVDMYR_01	10	0.921357	0.7	1.0	Y-Mdb
5	IKR_01	10	0.8	0.8	3.4	Kt
6	IKR_01	10	0.8	0.8	3.9	Hb
7	VRSPI_01	10	0.8	0.8	10.6	Sp1
8	VAMTOD_01	12	0.983229	0.7	0.0	MtdD
9	IKTE_01	12	0.8	0.8	0.6	Fig
10	VDATE_01	14	0.8	0.8	1.1	ATF
11	VSEIK1_02	14	0.8	0.8	4.7	Etk-1
12	VRSUF_02	14	0.229337	0.7	45.3	USF1
13	VSEF_01	15	0.8	0.8	17.2	KFEA
14	VYANACREB_01	15	0.229337	0.7	17.2	KFEA
15	PRO2_01	16	0.8	0.8	0.0	Onaque-2
16	VRMFE2_01	16	0.8	0.8	0.0	MFE-2
17	VRSRFC4_01	16	0.8	0.8	0.0	RSRFC4
18	VSEIK1_01	16	0.8	0.8	0.3	Etk-1
19	IKPEF_01	16	0.8	0.8	4.6	Drd
20	EBREPCAR1_01	17	0.8	0.8	0.4	repressor of CAR1
21	IKET4A_01	17	0.8	0.8	1.7	E74A
22	VRA4_01	18	0.8	0.8	0.3	AP-4
23	VACMTR_01	18	0.8	0.8	7.4	c-Mdb
24	VACMTR_01	18	0.8	0.8	6.3	c-Mdb

The Tabular Property Inspector allows developers to present data in a tabular format where they can be browsed, sorted, modified and much more.

### Companies: Not categorized

#	User	Name	E-mail address	Cat	Msgs
1	zha	Academy IT	academy@it.ru	Check	1
2	zha	ACS LTD Telecommunication Bureau & Internet	null	Check	0
3	zha	ARNVIND INVEST	contact.arnvind@mail...	Check	1
4	zha	Balbes Consultants	null	Check	0
5	zha	Bausch & Lomb Pharmaceuticals Division	null	Check	0
6	zha	BC Digital Inc	null	Check	0
7	fedor	Biosoft Ru	info@biosoft.ru	Check	0
8	zha	Center for Financial Technologies, Inc	null	Check	0
9	zha	Cognia Corporation	david_rubin@cognia.c...	Check	0
10	zha	DesertRain Lab	null	Check	0
11	igev	Diadema	manager@diadema.ru	Check	0
12	zha	Elecard Inc	webmaster@elecard.co...	Check	1
13	zha	Faudon Design	null	Check	8
14	zha	Global Digital Markets Ltd	info@gdmkts.com	Check	0
15	zha	H3 Pharma, Inc.	feedback@h3pharma.co...	Check	1
16	zha	Institute of Biology, Ufa Research Center	null	Check	0
17	igev	Interlink Ltd	info@interlink.nsk.s...	Check	0
18	zha	IT-Carenet.com	mail@it-carenet.co...	Check	118
19	zha	KTI VT	null	Check	0
20	zha	Laboratory of Bioinformatics	null	Check	0

Result Pages: 1 2 [Next >>] [Last]

☐ Check all records

The BeanExplorer applies the same concept of Property Inspectors to web interfaces thus simplifying the implementation of enterprise level applications.

## Where it can be used?

Typical applications of the BeanExplorer can be:

- Customer Relationship Management (CRM) systems
- Enterprise Resource Planning (ERP) systems
- Scientific applications which require researchers to deal with many parameters applied to scientific data.

Examples of modules that can be quickly implemented using BeanExplorer are:

- “Thick” and “thin” client applications
- Web interfaces to your data stored in the corporate repositories/data warehouses
- Navigation interfaces for huge amounts of heterogeneous data
- Interfaces **to** legacy applications (by presenting them as JavaBeans)

## System requirements

- Intel Pentium II 266 MHz or faster
- 64MB available RAM, 128MB recommended
- 10MB of available disk space
- Video SVGA, 800x600 or higher
- Sun Java2™ SDK version 1.3 and higher
- MS Internet Explorer 5.0 (or greater), or Netscape Navigator 4.6 (or greater)

## BeanExplorer 1.1 Feature Matrix

Feature	Limited	Standard	Enterprise
BeanHelpers framework – Java library for defining JavaBeans with extended functionality	■	■	■
JDBC result sets to dynamic beans mapper		■	■
BeanInfo source code generator		■	■
Graphical BeanInfo Editor		■	■
“Classic” PropertyInspector	■	■	■
Dialog-based PropertyInspector	■	■	■
Tabular Property Inspector		■	■
Web Form PropertyInspector			■
WebTable PropertyInspector			■
Tutorial	■	■	■
Tutorial for EJB components		■	■
Introspector of database metadata			■
SQL code library for defining additional metadata in most popular DBMS			■
SQL query builder			■
WebServices support			■
Native multi-user support			■
Access Control Editor which is used to assign permissions for accessing corporate data			■

You can find further information about BeanExplorer along with a downloadable evaluation version at [www.itcsoftware.com](http://www.itcsoftware.com).