



**PCD Awards 2005  
SUBMISSION FORM**

Category: **Extraordinary PCD Tools**

Deployment State: **In-Production**

Extraordinary PCD Tools are technologies (tools, products, protocol) that were not previously available commercially or otherwise. Innovative PCD Tools and Techniques are things that facilitate embedding support into other software, act as agents to deliver decision support at the time of need, integrate disparate end-user software, help anticipate and resolve performance challenges, use general inference techniques to eliminate complexity, reduce the need for conventional IT support, replace more conventional means of support, and the like. These can be new components of existing PCD Tools and Techniques if they are sufficiently unique.

Excellent examples of **Extraordinary Performance Centered Design Tools** are those that facilitate creating and integrating performance support environments, such as alternate interfaces, merging disparate systems (via interface integration, middleware, agents, and the like); capturing and disseminating business processes and best practices; improving business processes; embedding “wizards” and other productivity enhancers into existing systems; providing decision support and workflow support; providing resources to quickly troubleshoot problems, answer customer questions, and the like. Any tool that clearly stands head-and-shoulders above the rest in its category - or that defines a unique category - for enhancing the creation of EPSS / Performance-Centered Solutions is a candidate for an Extraordinary PCD Tool. Hardware and hardware devices that fit this description are also candidates.

---

**Entry Title**                    *Automation of business application software development*

**Submitted by:**                *Britesoft Corporation*

**Contact Name:**                *Jaya Sinnathamby*

**Phone:**                         *703-880-1618*

**E-mail:**                         [jaya@britesoftcorp.com](mailto:jaya@britesoftcorp.com)

**Address:**                        *2525 Trophy Lane  
Reston, VA 20191*

Logo:



Classification:

*Please classify your entry:*

- Comprehensive performance-centered web-based portal development tool
- Comprehensive performance-centered content/learning/knowledge management system development tool

***Specialty performance support tool that address one or more elements of the PCD lifecycle:***

- Process/workflow modeling and/or simulation
- User experience, interface development/generation, usability
- Content, information, knowledge engineering
- Real-time assembly of performance objects
- Other specialty tools that address PCD lifecycle element(s) (describe)
- Other (describe):

**Product /  
Component Name &  
Brief Description:**

*Provide the Extraordinary PCD Tool's name and a brief description (50 words maximum).*

**BriteWorks** - Imagine creating complex business applications such as CRM, ERP, or modernizing your legacy applications without writing a single line of code? Now deploy it on the intranet, Internet and mobile devices without having to deal with the complexities of J2EE.

That is what BriteWorks delivers.

**Vendor:**

*Provide vendor name, address, phone, and, email. Indicate "same as above" if this is the case.*

Same as above

**Operating Environment:**

*Describe the computer operating system, technical environment, system resource requirements, ancillary software if applicable) , etc.*

Any OS that supports a modern Java Virtual Machine (JVM) – Windows, Unix, Linux, I-Series, Z-Series, etc.

A Web Server such as Tomcat.

An application server is not required, though it maybe used for clustering.

CPU 2MHz, 1 GB RAM, minimum.

---

**Product/Component Detailed Description:**

*Provide a detailed description of your Extraordinary PCD Tool (400 words max). Explain how it helps to create performance-centered systems and what is unique about the tool.*

BriteWorks belongs to a genre of products that are based on the principles of model-driven development. It departs from that mold by bypassing code generation or the use of code skeletons completely. The model is the system and is executable instantly.

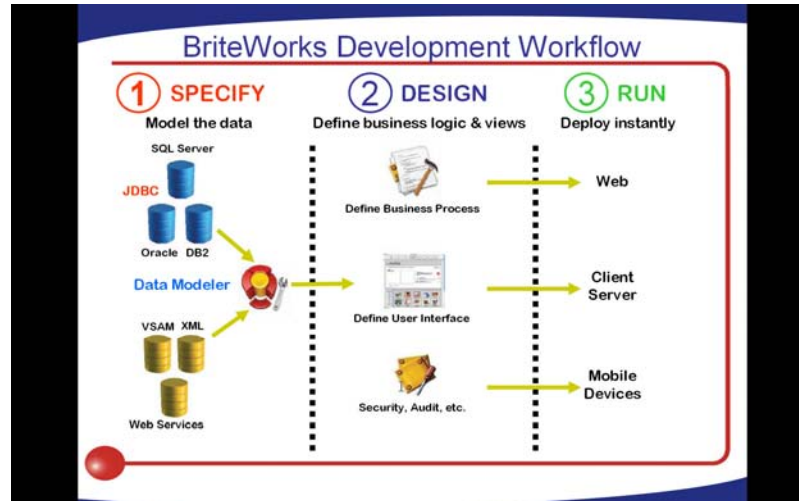
The core of BriteWorks is 'BriteWorks Server' which is a semi-complete application and also an application server in its own right. You tailor it to your specific needs using its complimentary development workbench, 'BriteWorks Studio'.

BriteWorks applications are built on the foundation of modeling behavior on the top of the domain model. Application development in BriteWorks is a 3-step process: 1) Model the data and business objects 2) Define the events and actions 3) Deploy.

**Step 1** - Introduce the domain model, in the form of the data model and business objects. BriteWorks has a very powerful data management facility that supports interaction with any JDBC RDBMS simultaneously and it provides a unified view of heterogeneous data sources without the developer having to concern himself with SQL. Web services are also handled naturally as another data source shielding the designer from its complexities. BriteWorks will then automatically instantiate the basic CRUD functionality via developer selected UI templates.

**Step 2** - The IDE provides a canvas for the developer to declare the business rules, design the menus and windows, and define the transaction flow using drag and drop. Most business 'data processing' systems are defined within BriteWorks without resorting to any handcrafted coding at all. But the system can be extended easily via 'plug-ins'.

**Step 3** – Deploy. Modeled system is executable instantly.



Because partial designs can be executed immediately without any programming effort, models become an instrument for thinking about and exploring requirements. What you get is a newfound freedom **from** choice rather than just freedom **of** choice. Prototyping becomes the norm and the prototypes are always robust enough to be turned into production systems that evolve as requirements are better understood and needs better defined.

Users are not pressured into committing to any particular approach, but can sit alongside designers to explore a variety of interaction opportunities. Menus and interfaces can be tailored and honed to support specific business cases/rules and to specific roles. Application design becomes fluid and the fear of becoming locked into a rigid, inflexible system gives way to highly pliable systems that can be easily adapted or changed completely to ever-changing needs.

**Product use:**

*List/describe who uses the tool and for what purpose. Include how long your tool has been available for general use (400 words max).*

BriteWorks has been available since October 2004 and has been in general availability since January 2005. Version 2.0 is scheduled for release in August 2005.

Conventionally, designers design things, engineers engineer things and builders build things. BriteWorks collapses these phases by following a descriptive rather than a prescriptive approach. A single model provides a cohesive entity, so nothing is lost in translation. BriteWorks is typically used by business analysts and developers familiar with data modeling. Power users and domain experts can be trained within a week to use it to develop sophisticated applications.

BriteWorks is a full-fledged development platform that can be used to create a variety of business applications limited only by creativity and imagination. Its ease of use and simplicity now empowers you to develop applications that are more complex than those that you would not have dared create before. Here are some examples:

- i. New applications from scratch
- ii. Modernizing legacy applications with a new Web and/or client/server interface
- iii. Extending functionality of legacy applications non-invasively
- iv. Composite applications – integrating fragmented user-interfaces into a unified interface
- v. Integrating disparate information silos
- vi. Consuming web services in an SOA environment

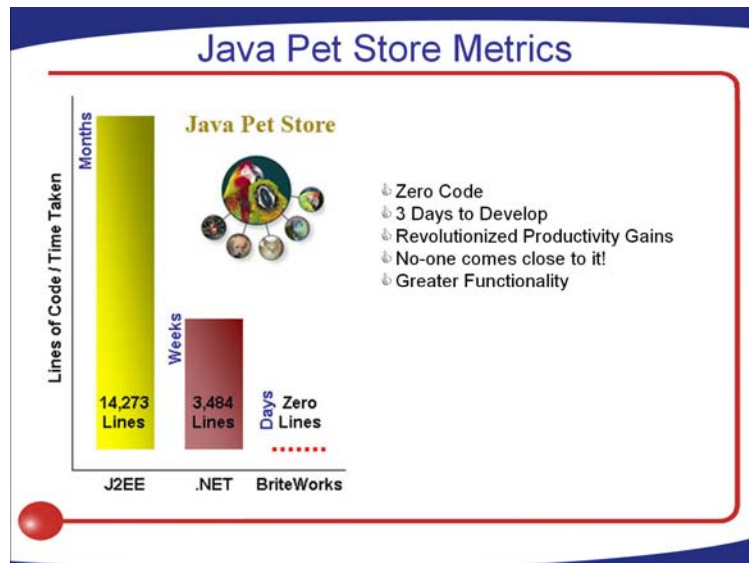
Many capabilities that are desirable, but considered to be a luxury are automatically available to BriteWorks applications:

- **Security** – seven levels of security, the most appealing of which is row / data level security
- **Auditing** – logging of any change (user definable) made to the application data by end users
- **Internationalization** – Automatic, context sensitive (i.e. left-to-right and right-to-left) runtime change in language. Includes double-byte. You can even create your own language if you so wish.
- **Import/Export** – facilities to define import and export formats, the invocation of import and export as well as a log of all such activities. Here, you can actually define almost any format, including those that have headers, details, trailers and associate with your database. You can import from Excel or CSV files.
- **Local Caching** – ability to 'download' part of the data you require, work offline (without network or internet connection) and then connect and synchronize your changes. This is useful for reducing bandwidth usage and avoids the need to go to the server for every action. Locally, the data will be stored in an embedded database to avoid tampering with the data.
- **User Options** – various preferences for end users, including setting default values for any window field, creating their own menus, look and feel etc.

**Deliverables provided and samples:**

*When PCD developers use the tool, exactly what do they produce; i.e., what is the deliverable (200 words max)? Provide access to at least one sample deliverable (preferably more than one) and describe the results achieved (i.e., how is performance supported and how is using such solutions better than alternatives (300 words max)?*

*Sample Application – comparing development effort with conventional IT solutions.*



*A number of capabilities make such results possible. Highlights:*

*BriteWorks frameworks include Ready Templates that reflect enterprise application needs based on how data is presented. No conventional coding is required.*

## Views/Forms/Windows Ready Templates

The screenshot displays the BriteSoft development environment. On the left is a 'Toolbox / Properties' panel for customizing the look and feel. The main workspace shows a form with a table of employee data:

EmployeeID	Last Name	First Name	Title	Title/Location	City
1	Anderson	John	Vice President, Sales	SA	800 W. Capital Way
2	Chandler	Andrew	Sales Representative	SA	11200 Lakeview Blvd
3	Deere	Robert	Product Manager	SA	11200 Lakeview Blvd
4	Fletcher	Michael	Product Manager	SA	11200 Lakeview Blvd
5	Grant	Frank	Product Manager	SA	11200 Lakeview Blvd
6	Haas	Michael	Product Manager	SA	11200 Lakeview Blvd
7	Johnson	Robert	Product Manager	SA	11200 Lakeview Blvd
8	Callahan	Leora	Product Manager	SA	11200 Lakeview Blvd
9	Chen	Thomas	Product Manager	SA	11200 Lakeview Blvd
10	Highmore	James	Product Manager	SA	11200 Lakeview Blvd

Annotations on the screenshot include:

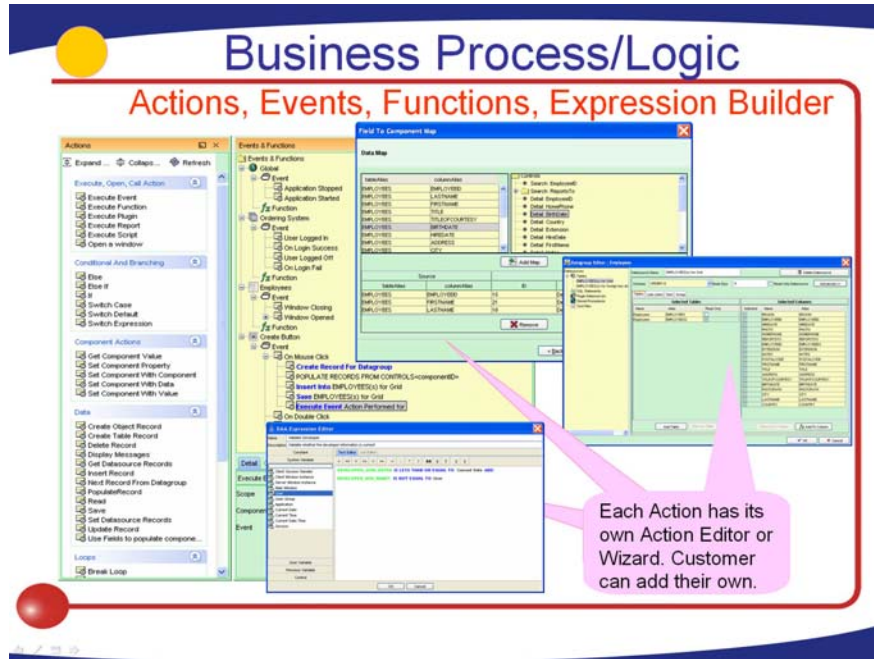
- Toolbox / Properties provides Look & Feel and customization** (pointing to the left sidebar).
- BriteSoft provides Template Library. Customer/Partner can extend library.** (pointing to the form design area).
- Components (Swing or Customized) for design capability** (pointing to the right sidebar).

Business logic is auto-documented and simple drag-and-drop metaphors are used to ensure that your solution reflects process flow – a critical element of PCD:

## Business Process/Logic Auto-Documents

The screenshot shows the BriteSoft development environment with an automatically generated flowchart diagram. The diagram illustrates a process flow with nodes and arrows. Annotations include:

- The diagram is generated automatically** (pointing to the flowchart).
- Drag & Drop Paradigm for Events/Actions** (pointing to the diagram's components).



**What difference does this tool make?**

*What difference does your tool/product/component make with respect to supporting performance; i.e. why should we care about it (300 words max)?*

Some of you maybe familiar with the concept of Model-View-Controller (MVC). Simply put, this is a best practice for decoupling the user interface from the data model. Data models are typically very stable, but business processes change. Even as this separation of concerns is hugely beneficial, handcrafting such solutions is not only tedious and laborious, but also complex and error-prone. Scripting languages are all the rage now, because they make interactivity more accessible, but they are still of the same mold. Factor in the needs of specific user groups and roles, the proliferation of new technologies in the form of displays and mobile devices, and you begin to see that handcrafted solutions simply do not scale.

BriteWorks provides a higher level of abstraction, as MVC is architected into the system at its foundation. The use of 'business objects' gives you the ability to reuse abstractions across many different views without having to be an expert in 'patterns' and or re-factoring code for re-usability.

With this approach, you can provide different groups of users task-specific interfaces that enhance their productivity. Users can also tailor the interface to their specific needs without compromising security. When business rules change, changes can be made centrally to the business objects and all processes that are impacted by the change will automatically inherit the changes, without having to manually code them.

Think of it as providing 'wizards' without having to handcraft them.

What else makes this tool extraordinary?

*Describe anything else that contributes to this being an extraordinary entry (200 words max). Delivery timescales, ease of implementation, support cost reduction, etc.*

BriteWorks brings an order of magnitude improvement in development productivity. BriteSoft has undertaken several projects using BriteWorks where our prototypes were deemed by the customer to be 80% of the solution, even as the competition was working on sizing the project!

The productivity gains achieved using BriteWorks are perhaps most easily understood when comparing the lines of code and time taken to complete the reference blueprint of the Java PetStore published by Sun. Whereas it took over 14,000 LOC in Java for the J2EE implementation and over 4,000 LOC in VB for the .Net implementation it took just 14 LOC in BriteWorks. Whereas the J2EE implementation and the .Net implementation were team efforts that took months and weeks, a junior developer took a mere 4 days using BriteWorks.

Maintenance is simplified, as it is done at the model level and at higher level of abstraction. The BriteWorks approach is descriptive rather than prescriptive and as such changes are highly centralized without the ripple effect of unintended consequences.

Perhaps the most compelling aspect of BriteWorks is the ability to toggle between execution and design mode instantly. Users experience instant gratification, though an unintended consequence is that users become more demanding!