

MARCH 2005

# Development Techniques for Creating Analytic Applications

By Wayne W. Eckerson

## SPONSOR EXECUTIVE SUMMARY



TDWI REPORT SERIES

Co-Sponsored by



A 101communications Publication

### About the Author



WAYNE ECKERSON is the director of research for The Data Warehousing Institute (TDWI), the leading provider of high-quality, in-depth education and research for data warehousing and business intelligence professionals worldwide. Eckerson oversees TDWI's research services and coordinates its BI Strategies education program.

As an industry analyst, Eckerson has written and spoken on data warehousing and business intelligence subjects since 1995. He has published in-depth reports and articles on a variety of data warehousing and business intelligence topics. In addition, Eckerson has delivered presentations at industry conferences, user group meetings, and vendor seminars. He also consults with vendor and user firms.

Prior to joining TDWI in 2000, Eckerson was a senior consultant at the Patricia Seybold Group in Boston, where he started the firm's data warehousing and business intelligence practice. He can be reached at [weckerson@tdwi.org](mailto:weckerson@tdwi.org).

### About the TDWI Report Series

The TDWI Report Series is designed to educate technical and business professionals about emerging issues in business intelligence (BI). TDWI's in-depth reports offer objective, vendor-neutral research consisting of interviews with practitioners and industry experts and a survey of BI professionals worldwide. TDWI in-depth reports are sponsored by vendors who collectively wish to evangelize a BI discipline or emerging technology.

### About TDWI

The Data Warehousing Institute (TDWI), a division of 101communications LLC, is the premier provider of in-depth, high-quality education and training in the business intelligence and data warehousing industry. TDWI offers quarterly educational conferences, regional seminars, onsite training, professional membership, leadership awards, print and online publications, and a public and private (Members-only) Web site.

### Report Sponsors

The research for this report was made possible by the following sponsors: ADVIZOR Solutions, arcplan, Business Objects SA, Microsoft Corporation, MicroStrategy, ProClarity Corporation, and SAP America Inc.

Development Techniques for Creating Analytic Applications

By Wayne W. Eckerson

**Table of Contents for Complete Report**

*(To download the full report, go to [www.tdwi.org/research/](http://www.tdwi.org/research/) and click on "Report Series.")*

About the Author. . . . . 2

About the TDWI Report Series. . . . . 2

Acknowledgements . . . . . 2

About TDWI . . . . . 2

Executive Summary . . . . . 3

Research Methodology . . . . . 4

Overview of Analytic Applications . . . . . 5

    Characteristics of an Analytic Application . . . . . 5

    Profile of Analytic Applications . . . . . 6

Development Approaches. . . . . 10

    Build Versus Buy . . . . . 10

    Architectural Layers . . . . . 11

    The Spectrum of Development Techniques. . . . . 12

Development Techniques in Practice . . . . . 16

    Customizing Analytic Applications . . . . . 20

    Development Processes . . . . . 23

Analytic Development Environments . . . . . 26

    Vendor Offerings. . . . . 26

Using ADEs to Build Dashboards and Scorecards. . . . . 32

Conclusion . . . . . 35

Appendix Outline. . . . . 36

### Executive Summary

**Analytic Applications.** An analytic application is not just a set of randomly created reports in a person's inbox or "My Reports" folder. An analytic application contains business logic that steps users through a series of interactive reports so they can access, analyze, and take action to optimize the activities in a specific business domain. Although some organizations currently deploy dozens of analytic applications to hundreds of users, most organizations in our survey are still inching their way toward maturity in this space.

**Customization Challenges.** Building intuitive analytic applications is not easy. Organizations spend too much time customizing and extending commercial products to meet user requirements. On average, organizations customize 33 percent of the final analytic application using mostly SQL and other hand-written code. The total process takes 7.5 months on average, too much time to meet fast-changing user needs.

In practice, organizations are most likely to build analytic applications around a BI tool, but then add substantial amounts of custom code (mostly SQL) to customize and extend the application. Most organizations also customize the BI tool itself, focusing on the GUI, calculations, and navigational elements. Developers spend significant time customizing ETL mappings and data models in packaged applications. Developers, who are mainly IT staff and application programmers, make frequent changes to analytic applications, and power users are often enlisted to change the front-end environment.

**Buy and Extend Using ADEs.** BI vendors have recognized the need to deliver "buy and extend" capabilities. Most are starting to deliver analytic development environments (ADEs), which expose BI components in a graphical drag-and-drop development interface. ADEs are the analytic complement of integrated development environments (IDEs), which are used to build transaction applications. ADEs promise to accelerate development time and reduce costs by allowing power users in every workgroup to tailor an analytic template for their immediate colleagues.

There are a wide range of ADEs on the market today. Pure ADEs, like those from arcplan, Business Objects, Microsoft, ProClarity, and SAP, give developers almost unlimited control over the look and feel of an application and the way users navigate through it. Newer report authoring tools, like those from MicroStrategy and ADVIZOR Solutions, give report developers greater flexibility to create a range of reports or dashboards to meet unique requirements of a broad range of users.

**ADEs for Dashboards.** Many ADEs are now used to build customized dashboards and scorecards, which are quickly becoming the most popular way for users to navigate analytical information. While most dashboards today are strategic in nature and enterprise in scale, the number and type of users supported indicate that we are still in the early stages of dashboard deployments.

# Overview of Analytic Applications

## Characteristics of an Analytic Application

Business intelligence (BI) professionals throughout the world have one thing in common: no matter how they design or architect their systems, the end result is an analytic application. Yet, this phrase is vague and generally misunderstood by most data warehousing and business intelligence (DW/BI) professionals and their business counterparts.

This confusion exists partly because we, as an industry, tend to focus on the tools, technologies, and architectures that we use to create analytic applications, rather than their output. But if we've made the term ambiguous through lack of attention, we've also bastardized it by giving it a multiplicity of definitions. Although no definition will satisfy every constituency and industry pundit, we use the following in this report:

*An **analytic application** consists of a series of logically integrated, interactive reports, including dashboards and scorecards, that enable a wide range of users to access, analyze, and act on integrated information in the context of the business processes and tasks that they manage in a given domain, such as sales, service, or operations.*

**The Definition of an Analytic Application Has Always Been Vague**

## Build Versus Buy

There is a range of approaches for creating analytic applications. On one end of the spectrum—the “buy” side—organizations purchase packaged analytic applications that require minimal customization and little or no coding across a range of functionality. On the other end of the spectrum—the “build” side—programmers write the entire application from scratch using custom code. In between these two poles are hybrid options that blend both packaged and custom approaches.

**Pros and Cons.** Many organizations seek to purchase tools or packages to standardize their software investments, accelerate deployment, and reduce total cost of ownership. Unfortunately, as we shall see, many companies end up over-customizing the commercial software, undermining its potential to reduce costs and accelerate deployment. And although building applications from scratch provides complete flexibility, it is expensive to keep developers on staff to maintain and extend an application that may already exist commercially.<sup>1</sup>

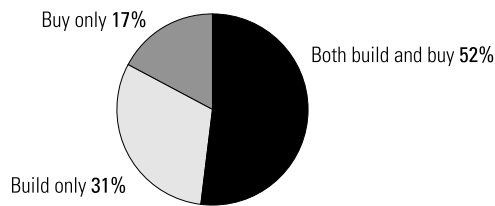
**The Need to Customize.** Today, no single development approach delivers a complete analytic application out of the box. Even packaged analytic applications supply only 60 to 80 percent of requisite functionality depending on users' requirements, the organization's existing infrastructure, and the data sources used. Thus, organizations need to figure out ways to customize existing BI packages and tools using a variety of “build” approaches.

**Buy and Extend.** It's no surprise, then, that the majority of respondents to our survey (52 percent) said that their organizations prefer to both build and buy application components—the “buy and extend” approach. (See illustration ?) Here, organizations purchase a BI tool or analytic package and then customize it.

**“Buy and Extend” Is the Most Common Development Approach**

<sup>1</sup> We obtained the first number (320) from a question focused only on dashboards and scorecards, which are a specific type of analytic application. The 528 number comes from preliminary results in TDWI's 2005 *Business Intelligence Salary, Roles, and Teams Report*, published in February 2005.

### “Does Your Group Prefer to Build or Buy Its Analytic Applications?”



*Illustration 9. Based on 473 respondents.*

The remaining 48 percent of respondents took a hard line, preferring to either build or buy analytic applications. Nearly twice as many of these “hard-liners” prefer to build rather than buy (31 percent to 17 percent). (See illustration 9) This two-to-one ratio mirrors the findings in our 2002 study, *The Rise of Analytic Applications: Build or Buy?*

While the industry is beginning to coalesce around a buy-and-extend approach to creating analytic applications, some organizations are still committed to build-only or buy-only methodologies.

## Spectrum of Development Techniques

Below is a list of development techniques that organizations use to create analytic applications. Illustration 12 shows the extent to which each can deliver a complete analytic application. In the next section we will examine several of the more prominent techniques in detail.

1. **Packaged Analytic Applications** deliver 60 to 80 percent of a complete analytic application out of the box, usually in a specific business domain.
2. **Packaged Data Marts** provide an ETL tool, source adapters, a target data model, and source-to-target mappings for specific source systems.
3. **BI Starter Kits** are templates in a BI tool that contain style sheets, metrics, reports, and some business logic tailored to a business domain.
4. **Microsoft Office Tools** include Microsoft Excel, PowerPoint, and Access.
5. **BI Tools** provide query, reporting, and analysis tools.
6. **Analytic Development Environments (ADEs)** are graphical, point-and-click development environments for rapidly building analytic applications.
7. **Scripting** includes lightweight programs that enable developers to quickly customize a BI tool or package’s look and feel or functionality.
8. **Portal Integration Kits** include APIs and code samples that make it easier to embed BI reports and controls into a third-party portal.
9. **BI Software Development Kits (SDKs)** are a set of documents that describe a BI tool’s API and how to use it.
10. **Custom Code** is a programming language used to build or modify applications, such as 3GL, 4GL, Web scripts, or SQL.

- 11. **Modeling Tools** create conceptual, logical, and physical models for a data warehouse or data mart.
- 12. **ETL** includes extract, transform, and load routines or tools to populate data warehouses and data marts.

Major Application Elements

|                       | Front End | Mid-Tier                             |                     | Back End            |                |            |
|-----------------------|-----------|--------------------------------------|---------------------|---------------------|----------------|------------|
| Developer Involvement | Low       | Packaged Analytic Applications       |                     |                     |                |            |
|                       |           | BI Starter Kits                      |                     | Packaged Data Marts |                |            |
|                       |           | BI Tools                             |                     |                     |                |            |
|                       |           | MS Office Tools                      |                     |                     |                |            |
|                       |           | Analytic Development Environment     |                     |                     | Modeling Tool  | ETL Tool   |
|                       |           | Portal Kits                          |                     |                     |                |            |
|                       |           | Web Scripts                          |                     |                     |                |            |
|                       | High      | BI Software Development Kit          |                     |                     | DDL            | ETL Code   |
|                       |           | Custom Code & Web Application Server |                     |                     |                |            |
|                       |           | Look & Feel                          | Reports & Analytics | Analytic Server     | Business Logic | Data Model |

**Major Application Elements**

Illustration 12. No development technique supports all major elements required to deliver an analytic application.

**Application Completeness.** An analytic application comprises many elements. At a high level, these include: 1) look and feel, 2) reports and analytics, 3) analytic server, 4) business logic, 5) data model, and 6) source mappings. These elements must be either built or purchased; even if purchased, developers will need to modify or extend them to meet business requirements.

Illustration 12 above shows the degree to which each development technique just listed supports these six application elements. Clearly, no technique alone creates an analytic application, although packaged applications and custom coding go the farthest toward delivering comprehensive functionality.

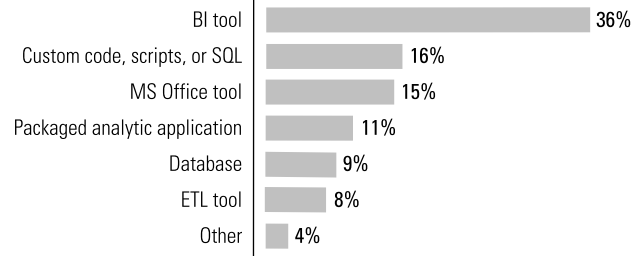
## Development Techniques in Practice

**Primary Technique.** Given the spectrum of development techniques for building analytic applications, which techniques or combination of techniques do organizations employ in practice?

Not surprisingly, the largest percentage of respondents (36 percent) selected BI tools as the primary technique for developing their analytic applications. Less than half as many selected custom code (16 percent), Microsoft Office tools (15 percent), and packaged analytic applications (11 percent). (See Illustration 13)

**BI Tools Are the Starting Point**

## Foundation of Analytic Applications



**Custom Coding Is the Most Prevalent Technique**

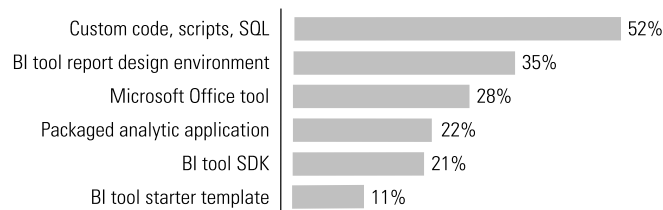
*Illustration 13. Respondents were asked to select the item that their group prefers to use as the basis for its analytic applications. Based on 473 respondents.*

**All Techniques.** Next, we asked respondents to select all the development techniques they used to create their single largest or “primary” analytic application.<sup>2</sup> Their answers would tell us which techniques organizations employ most often, as well as the combination of techniques in use.

Although organizations prefer to build their primary analytic application with BI tools, on the whole, they use custom coding more than any other development technique. More than half (52 percent) of respondents use custom code when building an analytic application. Next, BI tools were selected by nearly the same percentage as in the previous question (35 percent), followed by Microsoft Office tools (28 percent), packaged analytic applications (22 percent), BI tool SDKs (21 percent), and starter templates (11 percent).

**Profile of an Analytic Application.** Extrapolating from this data, the “typical” analytic application consists of a BI tool that has been customized using custom code, scripts, or SQL, is written partially against the SDK, and supports Excel as an alternative front end. (See illustration ?)

## Development Techniques Used in Primary Analytic Applications



**Most Companies Build Analytic Applications Around a BI Tool, Which They Supplement with Custom Code**

*Illustration 14. Techniques that groups used to build their primary analytic application. Respondents could select multiple choices. Based on 473 respondents.*

<sup>2</sup> Asking respondents to answer questions about the single largest application managed by their group provides more accurate data than asking them to estimate organization-wide averages.

**Customization Practices.** In practice, organizations are most likely to build analytic applications around a BI tool, but then add substantial amounts of custom code (mostly SQL) to customize and extend the application. Most organizations also customize the BI tool itself, focusing on the GUI, calculations, and navigational elements. Developers also spend significant time customizing ETL mappings and data models in packaged applications. Developers—mainly IT staff and application programmers—make frequent changes to analytic applications, while power users are often enlisted to change the front-end environment.

## Analytic Development Environments

There is an emerging category of tools that make it faster and easier to create custom analytic applications. TDWI calls this new toolset an analytic development environment, or ADE.

An ADE is the analytic counterpart to the integrated development environment, or IDE, which developers have used for years to build operational applications. Examples of the more popular IDEs today are Microsoft Visual Studio.NET, Borland's JBuilder, Eclipse, IBM's WebSphere Studio, and BEA's WebLogic Workshop, to name a few. ADEs are the spiritual heir to IDEs, both in functionality and name.

**A Promising Future.** ADEs promise to accelerate the development of custom-built analytic applications as well as make it easier and faster to customize packaged analytic applications. An ADE enables developers to drag and drop analytic components onto a screen to rapidly create analytic applications. More than report designers, ADEs give developers precise control over the look and feel, functionality, and workflow of an application.

As a result, developers will soon be using ADEs to “buy and extend” analytic applications. Starting with either a packaged analytic application or a BI tool, they will be able to use an ADE to quickly customize the last 20 to 40 percent of the front end.

In fact, the drag-and-drop nature of ADEs will further shift development responsibilities from IT developers to power users in the field. With an ADE, a power user can easily modify a packaged analytic application, flesh out a report definition, or create a new application or report from scratch (once IT has established data connections and BI query objects). Thus, ADEs will once and for all get the IT staff out of the business of creating reports so they can focus on what they are best at: building robust data architectures and abstraction layers for end users.

ADE tools will also accelerate the trend toward rapid prototyping. Developers and power users can use an ADE tool in a joint application design session to get immediate feedback from users on data, application screens, metrics, and report designs. This iterative process results in better designed applications that are delivered more rapidly. Many vendors are shipping ADEs for specific applications to facilitate rapid prototyping. For example, many dashboard and scorecard solutions are ADEs.

**Service-Oriented Architecture.** The real power behind ADEs comes from the fact that vendors have componentized the functionality of their BI tools. In the past, vendors hard-wired presentation, logic, and data functionality together. But the advent of object-oriented programming and service-oriented architectures has enabled vendors to open up their products, componentizing functionality within a services-oriented framework. The upshot is that ADEs enable developers to create multiple instances of components, store them centrally, and reuse them in other applications. This is a much more efficient way of creating and extending applications.

**ADEs Are the Analytic Complement to IDEs**

**ADEs Will Get IT Out of the Report Creation Business**

**Reusable, Object-Oriented Components Power ADEs**

Clearly, BI vendors have recognized the need to deliver “buy and extend” capabilities. Most are starting to deliver ADEs or ADE-like capabilities. In both cases, the tools provide an easier-to-use authoring environment, which is helping to finally move development out of the hands of professional developers and into the hands of power users and business analysts.

## Using ADEs to Build Dashboards and Scorecards

Rapid prototyping using ADEs will soon become the predominant method for building dashboards and scorecards. Already, many BI vendors now offer specialized ADEs for creating dashboards. (See the full report’s Appendix for a list of criteria to evaluate vendor dashboard solutions: [www.tdwi.org/adeapp](http://www.tdwi.org/adeapp))

**Definition.** A dashboard or scorecard is a graphical display that compares performance against predefined goals. Most people use these two terms interchangeably, although there is a subtle difference. A dashboard records actual performance or behavior—like an automobile dashboard—while a scorecard measures that performance against objectives or goals. In other words, a dashboard tells you how you are doing, while a scorecard tells how well you’re doing. We will use the term “dashboard” from now on to refer to both types of analysis.

**Popularity.** Dashboards are increasingly popular; a majority of our survey respondents said their group uses a dashboard as its primary analytic application (31 percent) or has deployed one elsewhere (28 percent). Another 24 percent are currently developing a dashboard or scorecard. Thus, almost three-quarters of respondents either have a dashboard or scorecard or are developing one.

**Dashboard Tools.** Most organizations created their dashboards from BI tools (41 percent), followed by custom code (22 percent) and Microsoft Office (13 percent). Only 17 percent have purchased a packaged dashboard solution from a vendor. We expect this percentage to rise in the next few years as more vendors provide robust dashboards solutions.

Most dashboards support a range of functionality. In general, dashboards have three levels: they let users drill down from high-level graphical indicators (72 percent) to a table or chart (82 percent) that can be filtered to show different views of data (71 percent) to detailed transaction data (54 percent). (See Illustration 34)

---

### Dashboard Functionality

|                  |   |  |
|------------------|---|--|
| Dashboard Levels | 1 | Graphical Indicators <b>72%</b><br><i>(Dials, gauges, speedometers, maps)</i>                |
|                  | 2 | Interactive Tables/Charts <b>82%</b><br><i>(OLAP, parameterized reports, linked reports)</i> |
|                  | 3 | Transaction Detail <b>54%</b><br><i>(Contained in a data warehouse or legacy systems)</i>    |

---

*Illustration 34. Percentage of respondents who support each level. Based on 240 respondents.*

Advanced dashboard implementations make it easy for executives to add or modify metrics. However, this ability for end users to modify the dashboards is not overwhelmingly employed, although power users are enlisted to update dashboards in one-third of organizations. Currently, the IT and

**A Dashboard Tells  
How You Are Doing, a  
Scorecard, How Well**

application development departments are the most likely candidates to update the dashboard to reflect new requirements. (See illustration 35)

Who Updates the Dashboard?

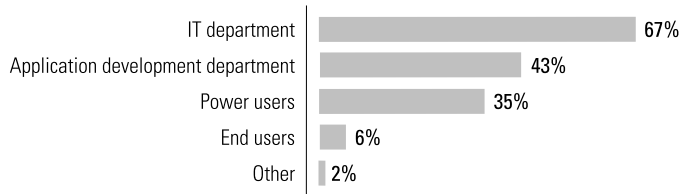


Illustration 35. Respondents could select multiple answers. Based on 240 responses.

**Summary.** Dashboards are quickly becoming the primary interface to business intelligence information because they conform to the way the majority of users wish to access, analyze, and act on information. While most dashboards today are strategic in nature and enterprise in scale, the number and type of users supported indicate that we are still in the early stages of dashboard deployments in most organizations.

Conclusion

The goal of a BI professional is to deliver an effective analytic application, one that makes it easy for a range of users to access, analyze, and act on information tailored to their business processes and domain. The best analytic applications contain navigational logic that steps users through the process of analyzing and acting on data. But building intuitive analytic applications is not easy. Organizations spend an inordinate amount of time customizing and extending commercial products to meet user requirements.

Fortunately, help is on the way. Most BI vendors are componentizing their BI tools and exposing them through a graphical interface. TDWI calls these tools analytic development environments (ADEs), and they promise to greatly accelerate development time and lower costs. IT will lay the technical and data groundwork for these ADEs, including the semantic mapping to back-end data sources. They will then turn them over to power users, who will tailor the analytic application to users in their workgroups.

**The Future is Clear.** Many vendors have released their first-generation ADEs, and many are now tailoring them to dashboards and scorecards, where there is a heightened need to customize the interface, especially for executive-level scorecards. Next-generation BI tools will in fact become ADEs or will be embedded in packaged applications to facilitate customization. This next generation of BI tools will support users' desire to "buy and extend" existing tools and packages rather than start from scratch. We still have a long way to go, but the path to the future is now clear.

**BI Tools Are Morphing into ADEs**



ADVIZOR Solutions, Inc.  
1333 Butterfield Road, Suite 280  
Downers Grove, IL 60515 USA

+1.630.971.5250  
info@advizorsolutions.com  
advizorsolutions.com

## The New Era of Business Intelligence

In this report, TDWI validates the need for businesses to adopt bold new approaches that can dramatically improve the return on business intelligence investments. This new reality is the sweet spot that ADVIZOR Solutions® was founded to address. Namely, that new era technologies enable today's business people, whether managers, analysts or other knowledge workers, to derive extreme insight by exploring data through interactive reports and dashboards, and then publish their findings to Microsoft Office and the Internet with self-service business intelligence solutions.

### Answering the “Why?”

ADVIZOR customers have a unique vantage point to capitalize on several perfect storm market trends. Business intelligence and data warehouse investments have been underway for more than a decade, leading to an overall maturing of infrastructures that has opened the way for the widespread adoption of advanced reporting and analysis solutions across diverse industries.

During this same period, the proliferation of data warehousing and e-commerce has enabled businesses to amass unprecedented amounts of data, causing them to be data-rich but information-poor. This phenomenon has rendered obsolete many traditional forms of data analysis and reporting, which have focused on the “What?” whereas ADVIZOR answers the “Why?” and enables users to quickly analyze and present highly dimensional business data.

### Enabling Business Leaders to Succeed

ADVIZOR Solutions has become a leader in this new era by helping people make wise business decisions much more rapidly than with traditional business intelligence solutions. This emerging market is “visual” and enables people to see and understand information in new and ever more powerful ways.

To benefit from these powerful capabilities, business leaders must apply technology strategically rather than for incremental productivity gains, establish performance metrics that are critical to the business, make better fact-based decisions with today's technology versus relying on hundreds of static reports, and clearly communicate results in a clear and productive manner.

Powered by patented Visual Discovery™ technology, ADVIZOR is the result of 15 years of product development including a decade of Lucent/Bell Labs research. ADVIZOR is world-class in empowering people to display, interact with and understand their business information. These unique capabilities enable business people to:

- Quickly make fact-based decisions.
- Perform ad hoc analysis via point-and-click.
- Clearly communicate insights and decisions via Office or the Internet.
- Create powerful analytical dashboards and applications.
- Rapidly deploy enterprise-wide: analyze, publish, consume.

#### Top Five Reasons to Choose ADVIZOR

1. You need answers to the “Why?”
2. You're frustrated with reams of static reports.
3. You need summary through detail data.
4. You need to present your findings via Office or the Internet.
5. You don't have time to waste.

ADVIZOR's advanced reporting and analysis software allows greater numbers of people to quickly make accurate and informed decisions—moving from data to results in minutes.



5200 Southcenter Blvd.  
Suite 250  
Seattle, WA 98188  
Phone: 206.246.5059  
Fax: 206.246.5952  
E-mail: [info@tdwi.org](mailto:info@tdwi.org)  
[www.tdwi.org](http://www.tdwi.org)