

Microsoft Office Access

This paper is addressed to information workers, business decision makers, and IT managers, to assist in evaluating how Microsoft® Office Access can help meet the data management needs of their organizations. The paper relays descriptions from customers of how they have found Office Access valuable, and it summarizes the product's key capabilities and benefits. In addition, the paper describes several investments Microsoft is making in the future of the product and its direction.

June 2006



Table of Contents

Executive Summary	3
Meeting the Varied Database Needs of an Organization	4
Broad Market Success	4
From Lists to Relational Databases	6
Microsoft Office Access, SQL Server, Visual Studio, and the .NET Framework	7
Case Study: Tracking Product Inspections with Microsoft Office Access	8
How Office Access Empowers Information Workers	9
Native and Linked Tables	9
Tables in Office Access 2007	10
Data Entry Forms	11
Reports	12
Queries	13
Macros	15
Modules	16
Office Access as a Developer Tool	17
Case Study: Developing Applications Quickly with Microsoft Office Access	17
Microsoft Office Access and Enterprise IT	18
Case Study: Streamlining Application Development	18
Case Study: Using Access to Manage the Small Databases of a Large Company	21
Access 2007 and Windows SharePoint Services	22
Revision History	22
Permission Setting	22
Recycle Bin	22
Office Access is a Strategic Microsoft Product	23



Executive Summary

Microsoft® Office Access has achieved broad success by enabling information workers to perform data-tracking tasks that otherwise would require the time and expertise of professional programmers and database administrators. In addition, professional developers have found that Office Access can save time and meet business requirements for many applications. IT departments can take advantage of Access capabilities by providing support and management of the Access applications being developed in their organizations. Microsoft is making significant investments in Microsoft Office Access 2007, redesigning the user interface and interactive design tools to make them even more approachable to entry level Access users. Integration with Microsoft Windows® SharePoint® Services will make Office Access 2007 a great collaboration tool, while enabling the data to be stored on enterprise servers for better manageability.

Meeting the Varied Database Needs of an Organization

Relational database technology has had an immense impact on business in the few decades since it was first developed. The vast majority of business transactions are now recorded, tracked, and analyzed using data stored in relational systems. However, enterprise-level database applications, such as sales accounting, inventory control, and human resources management, represent only a portion of any organization's total information management requirements.

One way to categorize the data-tracking needs of an organization is according to the number of people who work with the data. Some information is used only by individuals, some is used by small work groups, some by entire departments, and some by larger cross-departmental groups within the organization or in the wider community of partners and customers. The need for data security often varies with the size of the group affected, although some individual or small-group applications can require stringent security.

Another key factor to consider is the impact of an interruption in the availability of the data. Some data needs to be reliably available at all times, and some data can intermittently become unavailable without significantly impacting the mission of the organization.

Microsoft Office Access can play a useful role in meeting a wide range of data-tracking requirements. In some cases, the scalability, availability, reliability, or security requirements may demand a server-based engine such as Microsoft SQL Server™. However, even when the data is managed by a non-Access engine, Access forms, reports, and queries can still be useful and safe for creating user interfaces with the data.

Broad Market Success

By meeting the needs of a large and diverse population of users, Microsoft Office Access has achieved greater acceptance in the marketplace than any other database product. Information workers with little or no formal training in programming or database design have been successful in using the wizards and graphical tools in Access to develop useful database applications.

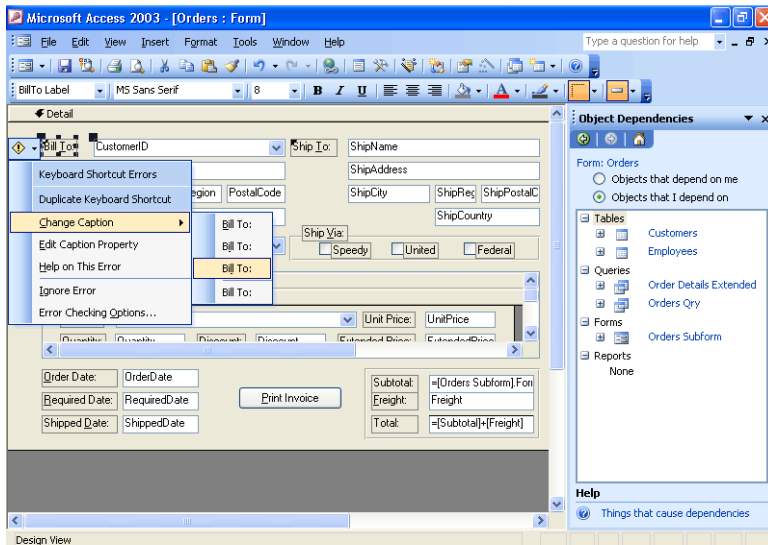


Figure 1. Order form in Microsoft Office Access 2003

Over 1.4 million Access database templates were downloaded from the Microsoft Office Online Web site, www.microsoft.com/office, in 2004. The MSDN® Office Developer Center Access portal logged more than 3 million page views and downloads in 2004, averaging approximately 80,000 unique users per month.

An independent Web site, www.utteraccess.com, which provides forums supporting Access, has registered over 90,000 members and has logged over 800,000 posts. On an average day, the Utteraccess.com site records over 500,000 page views, 100 new registered users, and 1,000 new posts.

A study of small and midsize companies conducted by Crestwood Associates in August 2004 found Access to be the leading database program, with 22 percent of respondents reporting that they used applications built with Microsoft Office Access. This study found that Access was used more than twice as much as the second-place product.

From Lists to Relational Databases

Automated spreadsheet applications such as the Microsoft Office Excel® spreadsheet software have existed since the beginning of the personal computer to meet the very common need for information workers to visualize and manipulate tabular data. Modern Web-based collaborative applications such as Windows SharePoint Services also support the creation of custom lists that can be created and maintained using browser-based interfaces. Office Excel provides more extensive calculation and charting capabilities, while Windows SharePoint Services makes it easier to share data across teams. Neither product, however, provides the advantages of true relational database storage. For example, set-based relational queries or bulk updates are not directly supported.

As a list-based application becomes more complex, the need emerges to transfer the data to normalized relational tables that can then be queried and manipulated using Structured Query Language (SQL).

When lists evolve to relational data, Office Access enables information workers to quickly establish a database application to track the information with increased efficiency. Several dozen database templates are available from Office Online to provide models for solving common data-tracking tasks. In addition, Access includes wizards and friendly user interface features to ease the creation of relational data structures and the queries, forms, and reports needed to work with the data.

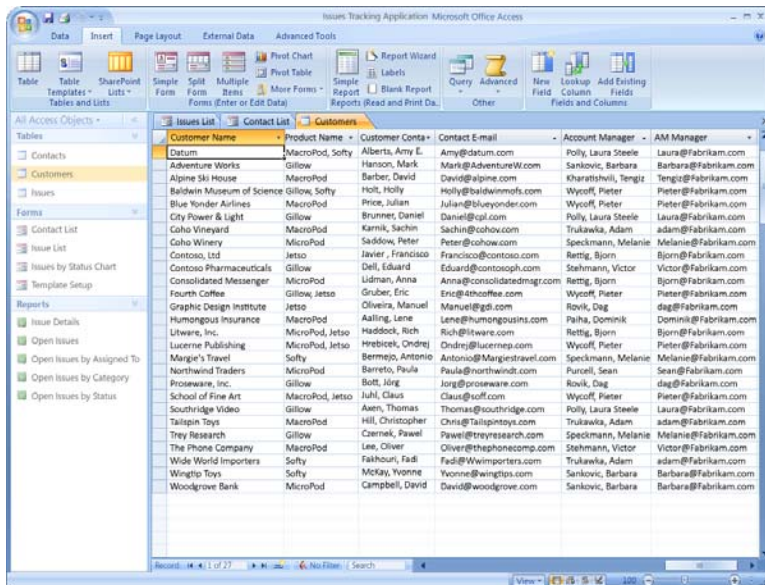


Figure 2. Microsoft Office Fluent™ user interface for Office Access 2007

Microsoft Office Access 2007 has been completely redesigned since Office Access 2003 to further improve the ease of use for information workers who have no coding expertise and very limited database knowledge. The Microsoft Office Fluent™ user interface is results-oriented and context-sensitive to expose the features the user needs at the right time. Furthermore, new pre-built solutions (templates) are available and are easily accessible at startup.

Microsoft Office Access, SQL Server, Visual Studio, and the .NET Framework

Microsoft offers a range of products that enable organizations to solve problems by entering, manipulating, and querying relational data. These products include the database engines that directly store and retrieve data, as well as application development tools that enable users to interact with the database engines.

Microsoft Office Access uses a database engine that can store data directly and that can also mediate storage and retrieval of data using other database engines, including Microsoft SQL Server. The Access application development tools can be used either with the Access database engine or directly with the SQL Server database engine.



For most database applications that are used only by individuals or small workgroups, Microsoft Office Access and its database engine can provide valuable solutions while keeping the cost of development low. Users can often solve data-tracking problems using Access, without requiring professional technical support. However, professional design reviews and occasional consultation can significantly improve the quality of Access solutions. For applications requiring greater scalability, availability, or security, SQL Server is an alternative to the built-in database engine of Access for data storage, although Access can still be used to develop user interface objects.

For applications that require a multi-tiered or service oriented architecture, or that require an HTML-based user interface, the Microsoft Visual Studio® development system and the Microsoft .NET Framework may be preferred. The .NET Framework is an integral component of the Microsoft Windows operating system that provides a programming model and runtime for Web services, Web applications, and smart client applications.

Office Access allows problems to be solved relatively quickly and effectively by users who understand the problems well, but who often do not have the resources or the time to develop a more extensively architected and engineered solution.

Case Study: Tracking Product Inspections with Microsoft Office Access

Ken Stewart is the Chief Financial Officer at a company that supplies linens to restaurants and medical clinics. The company has about 150 employees, four of whom work in the accounting department tracking the company's data. Until two years ago, quality control inspectors were using Excel spreadsheets to record and tally the results of their daily product inspections. They were creating new spreadsheets for each day's data, recording the number of rejected products and the operators who had worked on them. The accounting department found these spreadsheets hard to summarize into useful reports and noticed that it was easy for the inspectors to make invalid entries in the spreadsheets.

Stewart decided to try using Microsoft Office Access to create a better system, and although he had no prior programming experience, he was able to put together an Access application that solved the problems the accounting department was having. The solution is still in use today. He created

data entry forms that helped users ensure that all the required data was entered and formatted properly, and he built reports that summarized the data in useful ways.

Stewart used less than an hour of a consultant's time to help him organize the data properly into several related tables, and he found that he could create the queries, forms, and reports he needed by using the Access wizards and then modifying a few properties in the graphical designers. When Stewart got stuck trying to do something he couldn't find explained in the Access help topics, he used Internet searches to locate examples from others who had solved similar problems. He is now considering setting up another Access application to help with the company's hiring process.

How Office Access Empowers Information Workers

Microsoft Office Access provides a highly integrated set of tools that can be used together to create tables for data storage, links to external data sources, queries for retrieving and manipulating data, forms for data entry, reports for aggregating and analyzing data, macros to automate common tasks, and Microsoft Visual Basic® for Applications (VBA) code for creating advanced customizations. The following sections give an overview of how these Access objects enable users to meet their business requirements, and how Microsoft is enhancing these objects for Office Access 2007.

Native and Linked Tables

Access supports two fundamentally different ways of creating database tables. Users can create new tables in an Access database, or they can create links to existing data. These links appear and behave just like native tables. This support for creation of linked tables is key to the tremendous ease with which Access users can navigate, combine, and update information from multiple existing data sources, while still being able to create and maintain their own new data containers.

The Access database engine can work with its own native tables or with tables it links to using Open Database Connectivity (ODBC) or installable index sequential access method (ISAM) drivers. ODBC provides connectivity to all the common relational database servers, including Microsoft SQL Server, Oracle, IBM DB2, Informix, and Sybase.



Access project files, also called ADPs or .adp files, provide an alternative to using the Access database engine by allowing an Access application to connect directly to the tables in a SQL Server database. ADPs will continue to be supported in Office Access 2007. To take maximum advantage of the new features in Office Access 2007, Microsoft recommends the use of linked tables to connect to SQL Server data.

When Access was introduced, installable ISAM drivers were used primarily for linking to the prevailing desktop databases—Paradox, dBase and FoxPro—or to data in Lotus and Excel spreadsheets. Since then, drivers were added to connect to Microsoft Exchange Server data and to HTML tables. In addition, ODBC support helps connect to many corporate database systems, such as accounting systems, Human Resources (HR), sales, and more.

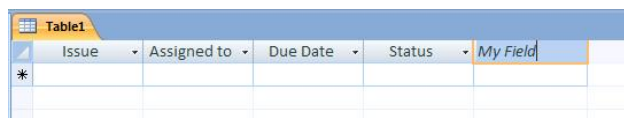
In Access 2003, a significant advance occurred when Microsoft introduced drivers that connect to data stored in Windows SharePoint Services. This was significant because these drivers call a component that sends and receives SOAP messages to the Web Services interfaces in Windows SharePoint Services. By enabling users to link to SharePoint lists, Access allows queries, forms, and reports to treat SharePoint lists as if they were standard relational tables.

In Office Access 2007, Access will move from using the Jet database engine (currently part of Windows) to a version that will be completely under the control of the Access development team. The standard Jet database engine will continue to be available for applications that need it, but the new Access database engine—which will be 100 percent backward compatible with Jet—will add Access-specific features. These features will include new capabilities that will enhance compatibility between Access and Windows SharePoint Services. In addition to linking to data in Windows SharePoint Services as they do now, Access users will be able to take this data offline in a local Access database, work with the data locally, and then reconnect to the SharePoint site to upload any changes. Any conflicting data changes made by other users will be handled by a conflict resolution wizard.

Tables in Office Access 2007

Office Access 2007 will make the creation of native Access tables more intuitive, especially for users who are familiar with Excel. In fact, users will be able to copy and paste a range of cells from

Excel into Access and have the table schema inferred automatically. Office Access 2007 will also enhance the grid-like datasheet view with new filtering, sorting, and totaling features.



	Issue	Assigned to	Due Date	Status	My Field
*					

Figure 3. Excel-like experience for Access tables

Data Entry Forms

Access forms are unique in the way they intuitively support data operations. Record selectors, navigation buttons, multicolumn combo boxes and list boxes, as well as a rich set of data-oriented controls, properties, and events, make it very fast and easy for information workers and developers to create full-featured user interfaces for data entry. Unlike other forms packages that focus on easing the creation of Win32® Windows applications and controls, with data binding requiring separate objects, Access forms provide an abstraction layer that is directly related to the underlying data.

Office Access 2007 features a new what-you-see-is-what-you-get (WYSIWYG) forms design interface. Users can design and modify the form layout in real time with live data on the screen—and preview the form as they build it. The new release will also introduce a new split view in which one part of the form will show a series of rows and the other part will show all the details from the selected row.

Office Access 2007 will include new controls for entering and displaying rich text and for selection of dates from a drop-down calendar. It also introduces a new method of data collection based on integration with the Microsoft Office InfoPath® information gathering program or HTML forms sent out in e-mails. When completed forms are received by return e-mail messages, the data is automatically entered into Access tables.

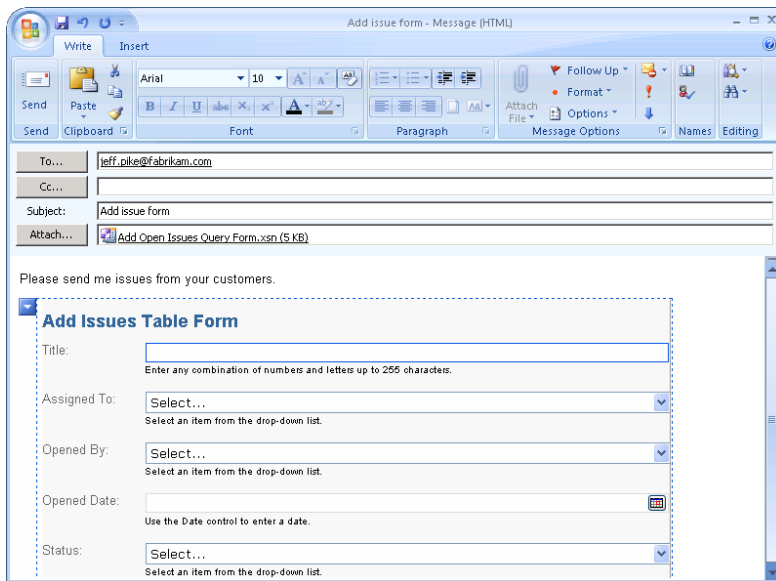


Figure 4. E-mail data collection

Reports

Perhaps the most well-known feature in Access is its wizard-supported reporting engine. Access uses a “banded” reporting model that enables easy creation of reports with nested levels of grouping, sorting, and totaling. The report designer is similar to the form designer, with graphical tools that allow reports to be created by dragging and dropping fields from the selected data source.

Office Access 2007 will debut dramatic improvements to the report design experience by introducing a completely new Layout view for reports (and for forms, too). In Layout view, the live data will appear while the report is being designed, and it will be much easier to move and resize blocks of data, with the results immediately visible. Selecting data for the report will also be easier, with related tables becoming available for selection automatically after an initial table is chosen. The user interface for specifying grouping and sorting will be significantly improved as well.



Figure 5. WYSIWYG report editing

Another innovation in Office Access 2007 will be the ability to re-run a wizard to modify an existing report instead of requiring users to start from scratch each time they run the wizard. Microsoft expects that many new Access users will discover the power of Access reporting and will quickly become proficient.



Figure 6. Group, sort, and total pane

Queries

As a true relational database management system, Access includes a query processor that efficiently executes commands written in Structured Query Language (SQL). In addition, Access provides a variety of wizards and graphical tools that enable untrained information workers to

construct and make use of these queries. In fact, fully functional applications handling data entry and reporting can be created by users with no knowledge of SQL. A graphical drag-and-drop query designer allows users to combine data from multiple tables, including linked tables, and from other saved queries. By setting properties or running wizards, users can easily create complex queries that calculate totals and averages, apply multiple criteria and sorts, pivot data, find top values or unmatched values, and perform updates, inserts, and deletions. In addition, users can create parameterized queries that use criteria based on values supplied at run time, including values automatically picked up from Access forms.

Access builds and executes many types of queries in the background without users even realizing that queries are being created. For example, reports run multiple queries to create nested groupings, subtotals, and totals. Forms automatically generate update, insert, and delete queries as data operations are performed. Sub-forms automatically filter the displayed records or add missing values to new records, based on data displayed in the main form in which the sub-form has been embedded. Lookup Wizards create queries that display related data from another table when data in the main table is viewed. These are all tasks that require skilled query construction in other development environments.

By enabling users to view the SQL code that is created by the wizards and graphical tools, Access also provides excellent opportunities for learning SQL. For experienced SQL coders, advanced techniques such as outer joins and correlated sub-queries are fully supported.

In Office Access 2007, new processing capabilities combined with new options in the Lookup Wizard will enable users to easily display and update data based on many-to-many joins without needing to understand the underlying table and query structures. For example, users will be able to assign multiple people to a task or multiple tasks to a person after creating only two tables, one for people and one for tasks. In the background, Access will automatically create the additional junction table needed to store pairings of people and tasks and the queries needed to retrieve and store the data. A new drop-down list control with check boxes permitting multiple selections will facilitate data entry. In addition, users will easily be able to create append-only columns that allow them to add a series of date-stamped comments to what appears to be a single memo field. The multiple entries will be stored in the background, in a separate table from which they can be filtered and sorted as

needed. A similar new data structure will be used to store file streams efficiently as one or more attachments to database rows.

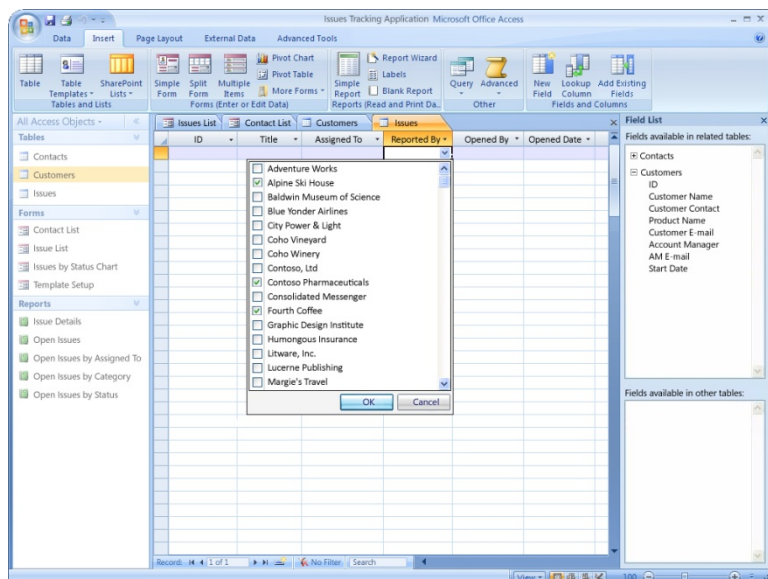


Figure 7. Multi-value fields

Macros

Access macros are created in a declarative programming environment that does not require writing any code. Access provides a set of parameterized macro actions for performing common operations, such as opening forms and reports, executing queries, finding records, or importing and exporting data. Users build macros by selecting from lists of options, and Access macros can include conditional branching logic. Macros can be set to run automatically in response to a wide range of user behaviors.

In Office Access 2007, macros will be significantly enhanced with features that previously were available only when programming with VBA code. These enhancements include better support for error handling and debugging, as well as the ability to assign values to variables.

It will be possible to build full-featured Access applications that run even with all VBA code disabled and which, therefore, can be safely e-mailed or downloaded. In fact none of the templates that will

ship with Office Access 2007 will contain VBA code. This code-free option will be supported by new macro actions and also by features in the Office Fluent user interface. For example, it will be possible to configure a combo box to allow users to add new items to the drop-down list by automatically opening a designated dialog form and then requerying the list when a new item is entered—all without needing to create any VBA code.

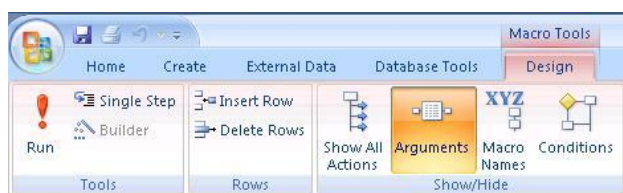


Figure 8. Macro design in Access 2007

Modules

For advanced customization, Access fully supports the use of VBA functions and subroutines. Functions can be called from Access queries to perform complex data processing, and VBA event procedures can run automatically in response to user actions. In many cases, even non-technical Access users can create useful code by running Access wizards or by copying and pasting from the many examples available in books, magazines, and online forums.

The primary application programming interface (API) for working with the Microsoft Jet database engine from code is Data Access Object (DAO). In Office Access 2007, new objects, properties, and methods will be added to DAO to support the new features in the Access database engine. For example, multi-valued fields created using the new Lookup Wizard will be accessible from code as DAO recordsets.

In Office Access 2007, programmers will also have the option of enhancing their Access applications with add-ins and smart panes based on managed Microsoft .NET code. A primary interop assembly (PIA) for Access, as well as one for DAO, will enable managed code running in these add-ins and smart panes to manipulate Access user interface objects and Access data.

Office Access as a Developer Tool

Access is most notable in how it enables information workers to build applications that otherwise would require a professional developer. However, Access is also used by many professional developers, including many who are also trained to use more advanced programming tools, such as Visual Studio. These developers, and their clients, have found that for many database applications, Access can get the job done faster and therefore less expensively, while fully meeting the requirements.

Access is not suitable for every task. For example, Access provides only very limited tools for building Web applications. Also, Access does not easily support the development of multi-tiered or service-oriented application architectures composed of loosely coupled components, or applications that need to be built and maintained by large teams of programmers. The typical Access application is created by a single developer and is architected as a relatively monolithic solution that intermixes user interfaces, business logic, and data access. Applications that must support large populations of users, or that require very high levels of reliability or security are usually not good candidates for development in Access. However a very large number of business applications do not have these requirements.

Case Study: Developing Applications Quickly with Microsoft Office Access

John Brennecke is Senior Project Manager in the Customer Service and Distribution department of a biotechnology company with over 8,000 employees. Brennecke's department is responsible for monitoring corporate and customer inventories to ensure that patients have an uninterrupted supply of the company's products. Brennecke described why the developers in his department often turn to Access:

"Access is used for projects with little or no budget—'drive-by' projects that can be developed by one person in no more than a few weeks. These smaller projects are able to be approved by lower- or mid-level managers without the usual lengthy approval process. We can respond to requests for quick-turnaround reporting or process-monitoring. Usually with minimal red-tape and no extensive design phase."



Brennecke's department uses Access to link to the company's Oracle data warehouse and to Electronic Data Interchange (EDI) data received from customers. Using VBA, staff members also import data from external sources that are exposed over Web services. In addition, Brennecke values the ability of Access to exchange data with other Office applications. The department often uses Access applications to export data to Excel to create charts for executives. Department members can also automatically create e-mail messages in the Microsoft Office Outlook® messaging and collaboration client that are sent to customers.

When asked how the role of Access at his company has evolved, Brennecke reports, "Access has become more popular in recent years. Everyone is realizing how fast application development can occur with Access and how cleanly the finished app performs. Other departments are beginning to take notice and are asking to hire Access consultants for their projects."

Microsoft Office Access and Enterprise IT

Case Study: Streamlining Application Development

Peter Mullen is Manager of Application Development in the corporate services department of a Fortune 500 insurance company with almost 50,000 employees and over 500 offices in 120 countries. Following a major acquisition, the company undertook a reorganization that involved centralizing its data management. Mullen, who had some experience working with Microsoft Office Access, started exploring the ways that Access database applications were being deployed across the enterprise. He was stunned to learn how pervasively Access was being used. Mullen says, "Access was all over the place doing so many things that it blew my mind. Once I became known as 'The Access Guy' there were people all over the company who wanted my time."

Not surprisingly, the quality of the Access applications Mullen discovered was quite variable. To Mullen's trained eye, some were in fact very poorly designed, and yet they were getting the job done. Mullen also found positions hidden in the payroll in the field that consisted essentially of supporting a small number of departmental Access applications.

Mullen and his manager put together a presentation for the Chief Information Officer (CIO), proposing that the IT department begin formally supporting Access development at the company. They outlined a strategy for centralizing and rationalizing management of the Access applications



that were evolving under the radar of their traditional portfolio management. The CIO saw the value in this proposal right away, and Mullen became the lead developer of a new group that was formed for this purpose. Mullen's group created and nurtured the use of naming standards, coding standards, and standard methodologies. They also were able to reduce headcount in the field as support activities were centralized.

Mullen reports that after about two years, "We had standard code; we had about 200 applications, and we were handling it. We started to rationalize our portfolio and really started to communicate exactly what we were doing to IT. This was the key point—that a lot of people in the organization started to see what we were really about. These were complicated problems being solved without spending a lot of money." As more people in the IT organization learned of the success of Mullen's group, he began to receive requests for "bolt-on applications, small department tracking projects, and proofs of concept. We hit a few of these and we had a real perceived value to the organization."

Mullen summarizes what he's learned about the optimal role of Access in the enterprise as follows:

"Access development in corporate organizations is simply a reality. People in the business are constantly looking for ways to add value to their customers and to free up their staff from mundane chores using automation that is inexpensive and easily accomplished. This is the strength of Access. The problem, from an IT perspective, is that the business comes to depend on products created without any IT involvement. Someone—usually an outsider or a low-level employee—becomes vital to the support of the product. So the question for IT becomes, 'Will we provide support for this process?' In my experience, providing an organized group that can create and support Access applications is a solid investment. It adds real and perceived value to the business and prevents a whole portfolio of applications from being developed under the radar of IT."

Mullen continues, "IT usually looks at all of the Access applications that are developed in the field as problems to be eliminated. I look at them and I see opportunity. Applications developed in the field are sometimes things that should never have been created; yet often they are products that are highly valued and fill a real need. By treating field applications (which most often means Access) with respect, and providing support and portfolio management, you can sometimes find patterns of demand in your organization and help drive real IT value back to the business."

Mullen is now working with the Architecture Team in his organization to identify candidates for migration to SQL Server and .NET applications. Mullen says, "Much of our portfolio will remain in Access, but some will migrate. There will always be a need for those quick and inexpensive solutions. The important thing is to keep track of these applications. They are the fingertips of your IT organization."

Market research by Microsoft indicates that fewer than 5 percent of all Access applications developed in large enterprises have requirements that justify moving the data from a Jet database to SQL Server, and fewer than 1 percent require migrating the forms, reports, and code to a .NET-based solution.

Access developers, administrators, and users now have a new toolkit to aid them in analyzing Access 97 databases for upgrade and conversion to Access 2003. Users can take advantage of the tools in the Access 2003 Conversion Toolkit to find and analyze databases in preparation for switching to Access 2003. For upgrading to the 2007 release, Microsoft plans to release the Microsoft Office Migration Planning Manager.



Figure 9. Access 2003 conversion toolkit

Case Study: Using Access to Manage the Small Databases of a Large Company

Dave Corbridge works as an Information Services Manager for an aerospace manufacturer with 14,000 employees. Corbridge reports, "About 10 years ago, we created an Access programming group within the IT department to help meet the needs of the company for small databases. As the needs grew, so did our little group until we had six people programming full time in Access. All of us came from the user community instead of from within IT. Our programs cover such things as keeping track of tickets sold for a company party, tracking the shelf life of various materials used in our processes, expense reporting, and printing acceptance tags for parts. Some of our applications incorporate other parts of the Office suite such as Word, Excel, or [Microsoft Office] PowerPoint®. A sister company that recently joined us has been using Access for several years to run queries and pull reports from accounting data stored in Oracle. We've considered moving away from Access to use .NET Windows Forms and SQL Server, but we have been slow to change because nothing has been able to compare with Access for quick development while providing a full-featured front end that is easy for our customers to use."

Corbridge's department has considered the argument that users should be discouraged from creating Access applications and should instead work with the IT department to get professionally developed solutions. Corbridge gives four reasons why his department has decided to stick with Access:

"First, we don't have the manpower to do all of the small database work in the company. Many of these are one- or two-user databases. If the users can do it themselves, why tie up our limited resources?"

"Second, there are only a few applications that outgrow the users' abilities and need to be passed on to the IT department. These applications may not be normalized and the programming poorly done, but there still has been a lot of groundwork that we can take advantage of in moving the application to the next level. This is generally better than having to start with nothing."

"Third, when we need to bring additional people into the Access programming group, these people will probably come from our power users who have been using Access on their own. Allowing users to do the small stuff helps us 'grow' our future full-time Access programmers."

“And, fourth, if 97 percent of users’ databases fit a light and agile Access requirement, why tie up the big guns (our Java, .NET, and C programmers) to do these small jobs?”

Access 2007 and Windows SharePoint Services

With Access 2007, users can easily move data from a local database file to a server with Windows SharePoint Services, which the organization’s IT professionals can manage, secure, and back up according to the organization’s policies.

By running Office Access 2007 on a SharePoint site, users gain new data auditing capabilities.

Revision History

New functionality enables users to track records and see who created, edited, and deleted records. Users can also easily view when the information was modified.

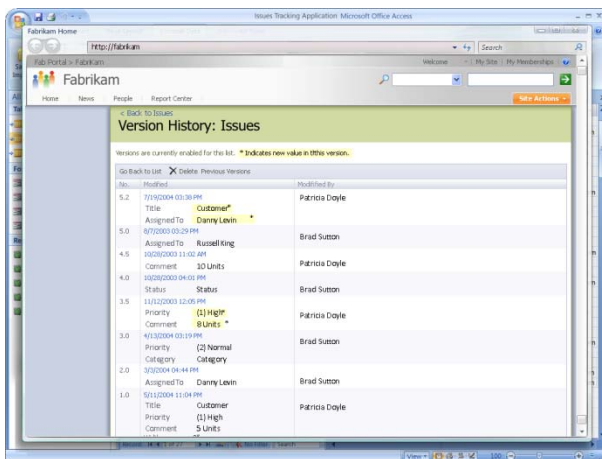


Figure 10. Office Access 2007 on a SharePoint site

Permission Setting

Using Windows SharePoint Services, users can manage who has access to data. Users can assign limited reading permissions or full editing rights.

Recycle Bin

Users can now recover data that was erroneously deleted thanks to the Recycle Bin feature in Windows SharePoint Services.



Office Access is a Strategic Microsoft Product

Microsoft believes strongly in the strategic importance of continued investment in Access and the Access database engine. Access addresses a class of users who would never use Visual Studio and developers who can help empower and support those users. The Access database engine, with its versatile ability to connect to multiple diverse data sources and its powerful client-side query processor, fills a need that is distinct from SQL Server and SQL Server Express. Also, as a tool for creating rich Windows-based forms and reports bound to Windows SharePoint Services data, Access can add significant value to the Web-based capabilities of Windows SharePoint Services. The Web services-based technology that Access uses to connect to Windows SharePoint Services is a promising harbinger of how Access will be able to add value to other emerging service-oriented data sources. For these reasons, Microsoft sees a bright future for Access and will continue to invest strongly in its continued growth.

To meet these new challenges, Microsoft Office has evolved from a suite of personal productivity products to a more comprehensive and integrated system of programs, servers, services, and solutions. Building on the familiar programs many people already know, the Microsoft Office system is a set of tools designed to work together to address a broad array of information worker problems. Microsoft is extending its commitment to ensure that individuals, their teams, and entire organizations can transform the full range of their data storage and data tracking into real business value.



This is a preliminary document and may be changed substantially prior to final commercial release of the software described herein.

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This white paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in, or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2006 Microsoft Corporation. All rights reserved.

With the exception of the case studies, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious. No association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

Microsoft, Excel, Fluent, InfoPath, MSDN, the Office logo, Outlook, PowerPoint, SharePoint, Visual Basic, Visual Studio, Win32, and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.