
ODBC 3.0
Michael Pizzo
Developer Division
Group Program
Manager
Microsoft Corporation

Agenda

- Goals of ODBC 3.0
- ODBC 3.0 New Features
- ODBC in the OLE world
- Summary

Goals of ODBC 3.0

To provide an efficient API, suitable for native access to Relational SQL Data

- Alignment with the Standards
- Enhanced application model
  - Performance
  - Functionality
  - Developer requests
  - Support for OLE DB

ODBC 3.0 Features

- Descriptors
  - Item-by-item binding/describe of columns/parameters
  - Extensible
- Diagnostics
  - Extensible diagnostic records
  - Non-destructive retrieval
- Binding Enhancements
  - Out-of-Line binding
  - Bind offsets
  - Individual Descriptor Fields

ODBC 3.0 Features (cont.)

- Locators
  - Specification/Retrieval of large data
- Unicode support
  - Unicode versions of APIs
  - Support for Unicode data
- Variable Length Bookmarks
  - More natural (Primary Key)
  - Allows Fetch Offset

ODBC 3.0 Features (cont.)

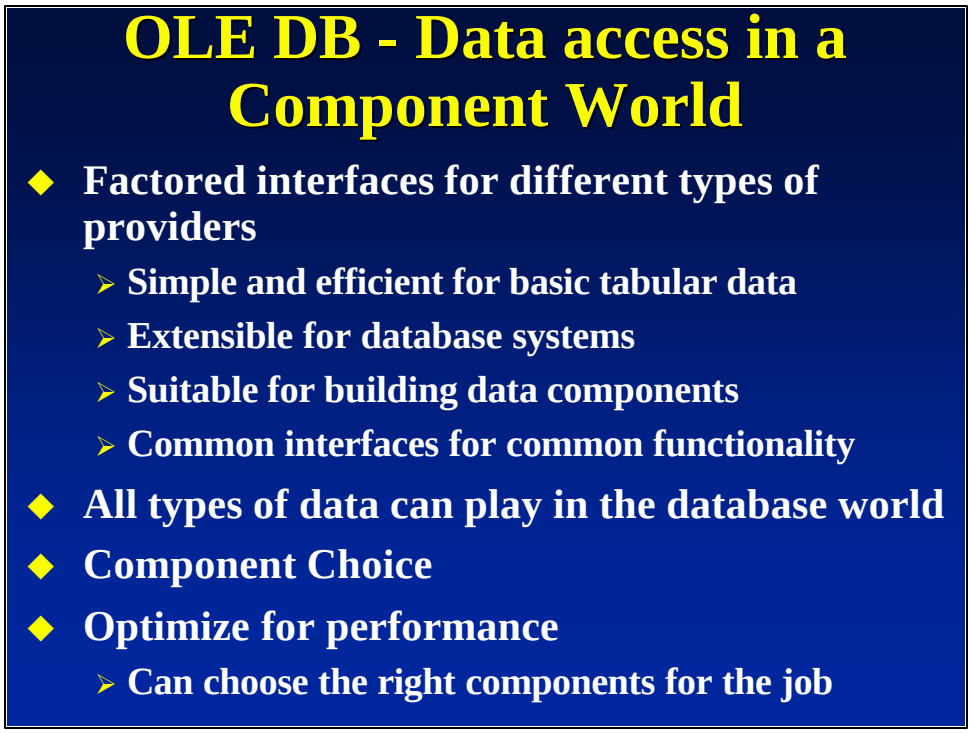
- Position by Search Condition
  - Positions within result set
- Nested Result Sets
  - Support for computed rows
  - Support for
- Batches
  - GetInfos to describe Rowcount, processing semantics

ODBC in the OLE World

- OLE - a Component World
- OLE DB - Data Access in a Component World
- OLE DB provider for ODBC data

OLE - a Component World

- Object-oriented
  - Re-useable objects provide common services
- Common interfaces
  - Objects support common interfaces for sharing data
  - Consistent programming model across different services
- System support
  - System components provide mechanisms for object instantiation and marshaling



•OLE DB will define common data objects and interfaces used by all data providers, consumers, and components.

It's the foundation of the Microsoft Component Database Architecture.

It will be suitable for simple data providers (such as the file system) to efficiently expose Rowset, robust enough to add service-layer components for more sophisticated data manipulation and navigation.

It will provide rich functionality to expose advanced capabilities of more sophisticated data stores.

It will be extensible. Tools and database vendors will be able to add their own unique extensions; the COM environment and the design of OLE DB will make this straightforward and smooth.

•OLE DB interfaces will provide the following functionality to any data store or data component.

Rowsets, Queries, Notification, Concurrency, Security, Transaction, DDL, Schema

•OLE DB will be efficient and will provide the most fundamental way to access data within a COM environment.

It will be a low-level API that directly exposes common characteristics of relational and non-relational data.

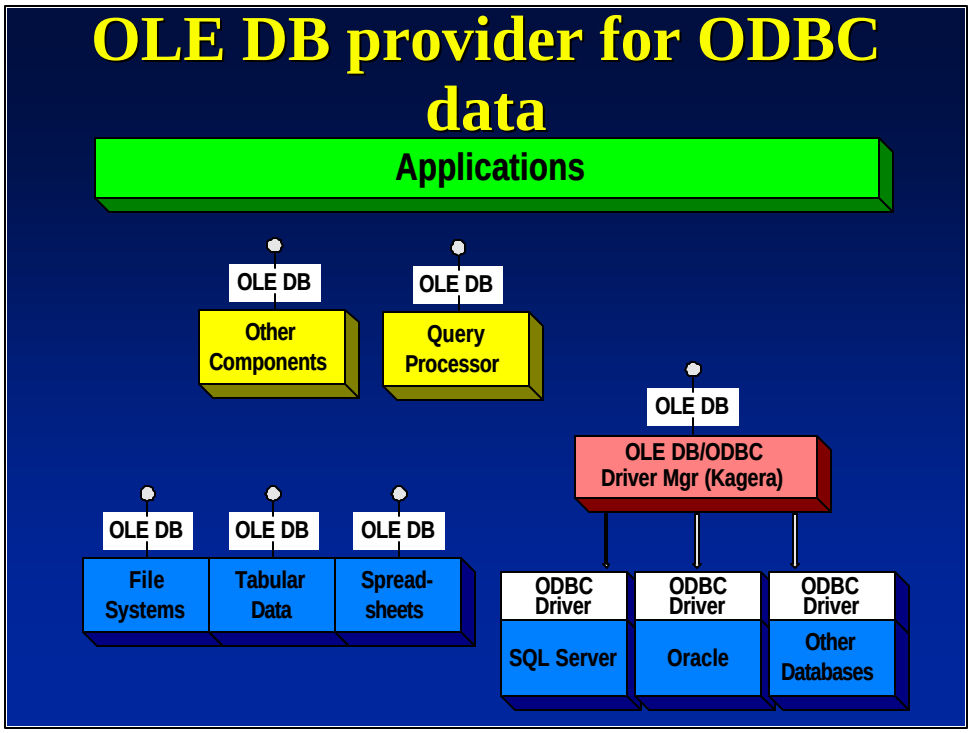
Initially, it may not be suitable for macro-type programmers dealing with high-level object abstractions. A higher-level language abstraction layered on top of OLE DB (such as DAO) will be built/provided.

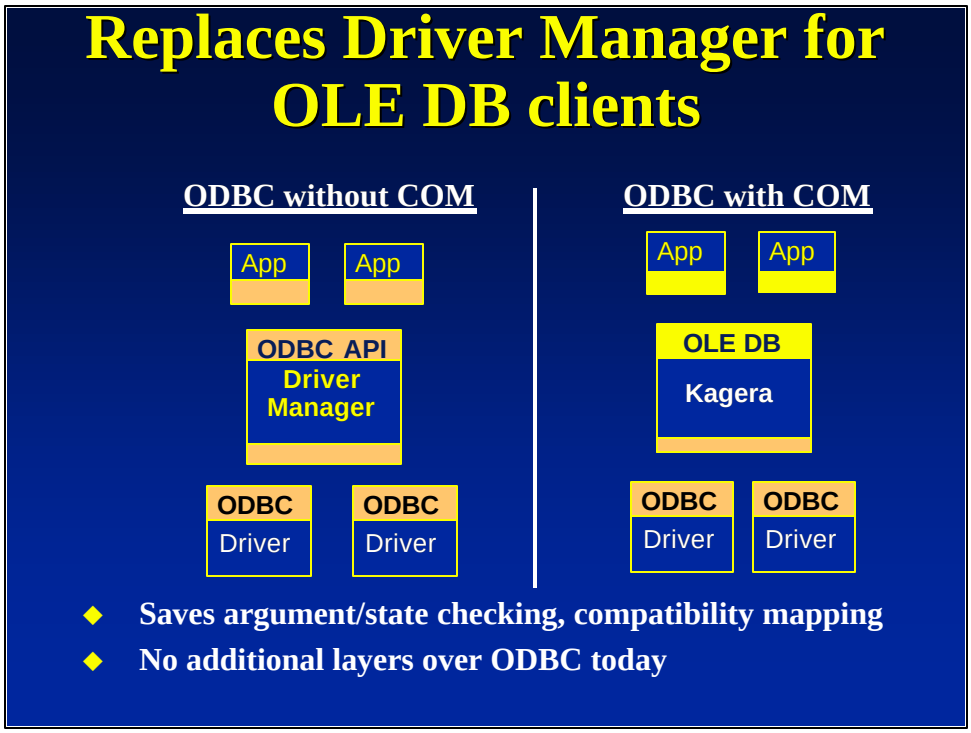
Non-relational data stores (data without order or query capabilities) within that environment can directly implement OLE DB interfaces to access that data.

Database vendors that require the Standards-compliant call-level interface for access to SQL data will continue to use ODBC. To that end, OLE DB will be designed to work well with ODBC, and ODBC will add features and functionality to better support OLE DB.

OLE DB provider for ODBC data (Kagera)

- Implements OLE DB interfaces over ODBC drivers
- Opens a new market of consumers of ODBC data
- Works with ODBC 2.x
- Works best with ODBC 3.0
  - OLE DB leverages concepts from ODBC 3.0
  - Separate compatibility components


OLE DB provider for ODBC data
Applications
OLE DB
Other Components
OLE DB
Query Processor
OLE DB
OLE DB/ODBC Driver Mgr (Kagera)
OLE DB
File Systems
OLE DB
Tabular Data
OLE DB
Spread-sheets
ODBC Driver
SQL Server
ODBC Driver
Oracle
ODBC Driver
Other Databases



- ***Same ODBC Drivers serve both OLE DB and ODBC applications***
- ***No additional layers in either environment***

Summary

- ODBC 3.0
  - Alignment with the Standards
  - Numerous enhancements
- OLE DB
  - Component architecture
    - Common interfaces to common functionality
  - Access to *all* types of data
    - Doesn't require SQL engine
  - ODBC 3.0 provides access to SQL data