Abstract
This paper presents an overview of Microsoft SQLServer for Windows CE. This is a database engine designed for mobile and embedded applications. The focus of the presentation is on:

- connectivity and integration with backend SQLServer databases (replication and synchronization),
- the application development framework (integration with common development tools and languages),
- SQL features and functionality.

1 The Need for a Database on Windows CE

We have entered an era of pervasive computing. Computing platforms extend from servers and desktops to every conceivable form of electronic device. Important emerging “small-scale” platforms include handheld devices and PDAs, cellular phones, embedded controllers, and smartcards. As a better understanding of these platforms has emerged, it has become clear that any form of pervasive computing across these devices will have the following characteristics:

- The devices are rarely standalone. Usually, they operate as part of a larger computing infrastructure ranging from backend servers to desktops to the devices themselves.

- The uniformity of application use (and application development) across the computing infrastructure is critical for enhancing the user (or developer) experience and really enhancing the platform.

- Most applications on small devices are data-centric, and involve local caching or replication of data.

Windows CE is Microsoft’s operating system platform for small-scale mobile and embedded devices. It provides a wide variety of operating system features (similar in flavor to the familiar Windows API), with specialized customization features to minimize footprint. It runs on a large number of hardware platforms (more than 100 different CPUs), and can be shipped in a large variety of configurations (using a componentized architecture that permits the addition or removal of individual components). Hardware vendors customize Windows CE for different categories of devices. For personal devices (like PDAs), the graphical and user interface components are included, whereas they are typically excluded for embedded devices.

There is clearly a need for a robust data management solution for the Windows CE platform. It should be possible to store, retrieve, and manipulate data on the local device, and copy or replicate data from backend servers. These data manipulations should be as seamless as possible, retaining the abstractions of data access that application developers are familiar with. There are a wide variety of applications that require such database support. These may be broadly categorized into:

- Enterprise applications: these include sales force automation, order entry, health care applications, and financial service applications.
- Personal applications: these include PIM tools, travel services, web-based channels, entertainment and games, etc.

Application developers use traditional desktop development tools like Visual Basic, which have extensions that support development for the Windows CE platform. In both categories of applications, there is need for a local queryable store that can be synchronized with a backend server. In the case of enterprise applications, the backend is usually a database server.

2 Overview of SQLServer for Windows CE

SQL Server for Windows CE (SSCE) is Microsoft’s relational database for Windows CE. Since Windows CE is a platform for small devices (with limited memory sizes), SSCE may be termed a “small-footprint” database engine. The constraint on program size has lead to interesting tradeoffs in choices of features to include or exclude. In this section, we describe the architecture of the engine. There are three main components:

- The Storage Engine (SE): this component provides a relational store, including support for a variety of ANSI SQL data types, indexes, and transactions.
- The Query Processor (QP): this component provides a SQL parser, a query optimizer, an execution engine for SQL queries, and support for cursors.
c) The Replication module (REPL): this component provides the ability to replicate data with a backend SQL Server database.

Each of these components is supported as a dynamically loaded library, thereby enabling applications to avoid extra footprint when it is unnecessary. Additionally, there are two components that allow application developers to access SSCE database in a convenient and familiar paradigm. They are:

a) OLEDB: the features of SSCE can be accessed from a C or C++ program through the established OLEDB programming APIs (actually, a subset of OLEDB called OLEDB-CE)

b) ADO: the features of SSCE can be accessed through ADO (actually, a subset of ADO called ADO-CE) from a program in any language including Visual Basic.

The picture below (Figure 1) presents the high-level architectural diagram of the system.

SSCE runs on Windows CE P/PC’s (Palm-size PCs), H/PC’s (Handheld PCs), as well as embedded systems running Windows CE OS version 2.11 (Birch) and higher. It runs on all Windows CE supported CPUs (some of the most popular being SH3/4, SA1100, MIPS 3900/4000 series). SSCE is scheduled for release to the Internet in Q3 2000

3 Connectivity

There are two mechanisms available for connectivity to backend servers. If the device has a persistent network connection, any of the well-known Microsoft data access mechanisms (ADO, OLE-DB, RDO, etc.) can be used to either directly manipulate server data or to bring it over to the client device and manipulate it locally. If the device only has occasional connectivity, the replication mechanism is used. SSCE supports “merge replication” based upon SQL Server 2000 merge replication. Merge replication is ideally suited to portable devices because it allows data to be updated independently on the portable device and the server. The data can later be merged when the device is connected to the server.

Developers are likely to use SSCE replication in one or more of the following three ways.
1. Read-Only Replication
2. Data Capture and Upload
3. Replicate, Update, and Synchronize

Most applications will use these replication alternatives in combination. For example, a sales support application might use “read only replication” to download a price list to the device. The price list could be updated on the device but the changes are never sent to the server. It might use “data capture and upload” to capture new orders on the device. These orders would then be uploaded to the server. It might use “replicate, update and synchronize” to download customer information to the device. This customer information could be updated on the device and the resulting changes could be uploaded to the server.

The management of replication alternatives happens at the database server. The database administrator can use “horizontal” and “vertical” filters to define and maintain unique subsets of data for different clients. Horizontal filtering can be used to replicate a subset of the rows in a published table. For example, each sales person might only receive the customer records for the accounts they service. Vertical filtering can be used to replicate a subset of the columns in a published table. For example, a vertical filter could be used to eliminate large text or image columns that may be unnecessary for certain subscribers. Both horizontal and vertical filtering can be used in combination. Further, in the case of conflicts, resolution is typically effected at the server, using established SQL Server 2000 conflict resolution mechanisms. This is important because in many situations, there may not be a physical user associated with each device to manually resolve synchronization conflicts.

Application programs written in Visual Basic CE or Visual C++ CE use the SSCE Merge ActiveX Control to programmatically control replication. The application uses the control to subscribe to a publication and to

![Figure 1: Architecture of SSCE](image-url)
initially download the subscription to the Windows CE device. The application can later use the control to synchronize the subscription on the Windows CE device with the SQL Server publication on the server.

In the most general form, there are four stages to SSCE Merge replication.

1. The database administrator creates a publication containing SQL Server 2000 data that is eligible for replication. A publication is simply a collection of articles. An article is a table that is eligible for replication.

2. Once the database administrator defines the publication, a Windows CE application can subscribe to the publication using the SSCE Merge Control. The data in the publication is downloaded from the publisher to the subscriber during the subscription process. The download creates a subscription on the Windows CE device.

3. The subscription on the Windows CE device can then be updated by applications running on that device.

4. Periodically, the updates made to the replica are sent to the publisher where they are merged into the publisher’s database. Similarly, changes made at the publisher since the initial download or most recent merge, are sent to the Windows CE device where they are merged into the replica. These periodic merge operations keep the data on the server and the Windows CE device synchronized.

SSCE merge replication is Internet-based. Windows CE devices perform synchronization by establishing an HTTP connection to SQL Server through IIS (web server). Using IIS allows the SSCE replication to take advantage of the authentication and authorization services of IIS and to communicate with a SQL Server system located behind a firewall or proxy server. SSCE replication can be performed over a wide variety of transports including wired and wireless LANs, wired and wireless modems, serial links, and infrared ports. The communication protocol is well suited for wireless transports. Compression is used to reduce the amount of transmitted data. Encryption is used to safeguard sensitive data.

4 Application Development
A guiding principle in the development of SSCE has been to minimize changes to the development experience of application builders. There are many third-party developers who use Visual Basic and Visual C/C++ to build database applications, using ADO and OLE-DB for database access. Ideally, these developers should be able to use the same paradigms for application development. Further, SSCE is part of the SQL Server product family, and extends the SQL Server database solution to a new category of platforms. Clearly, the feature set and behavior of SSCE needs to be a subset of SQL Server to promote a uniform user experience.

Towards this end, the SSCE product provides the following features:

a) The SQL features and syntax supported are a strict subset of SQL Server 2000.

b) Data access is supported through OLEDB-CE and ADO-CE (both subsets of the corresponding data access interfaces of SQL Server 2000).

c) Replication is controlled through an Active-X control.

d) Bi-directional merge replication works seamlessly with a SQL Server 2000 backend.

5 Query Processing and Storage
SSCE provides a transactional data store that can either be directly accessed (through OLEDB-CE or ADO-CE) or can be manipulated by SQL commands. The SQL functionality supported is a subset of ANSI SQL-92, and the SQL supported by SQL Server 2000. Several of the components of the system arise directly from the SQL Server codebase, while others have been built specifically for the Windows CE platform. Importantly, the lexicographic and parsing components are common to both products, leading to similar syntactic behavior.

Since footprint needs to be minimized, SSCE does not utilize a full-function cost-based query optimizer in its initial version, although this option is not precluded in the system design. To the contrary, since this is an emerging market where the needs of applications could change significantly and quickly, SSCE is designed to be extended easily in a variety of different ways – with new SQL features, with new optimization or execution algorithms, or with customization capabilities. These design hooks will be utilized in subsequent releases to tailor the product to the varying demands of the handheld and embedded devices. The initial release provides a powerful yet compact database engine that caters to a wide variety of enterprise and personal information management applications.