



Microsoft® SQL Server® Data Services

Your Data, Any Place, Any Time

Business Challenges

Organizations are seeing a confluence of factors meeting to create, manage and access data explosion.

1. The amount of digitally born information is increasing dramatically, enabled by new types of information, like digitization of data from digital imaging or video, to sensor information from RFID tags across your organizations.
2. Users want to use this information on any device and within applications that they use daily like Microsoft Office.
3. According to analysts, more than 70 percent of a typical IT budget is spent on infrastructure such as servers, operating systems, storage and networking. As the costs for hardware and software track downward, the costs of managing, refreshing and supporting the infrastructure increases.
4. New types of data-driven Web 2.0 applications are rising. Developers want to build interconnected database applications quickly and integrate end-to-end mash-ups with existing business without any IT complexity and infrastructure cost. They rather focus more on innovating with data and less on setting up infrastructure.

Microsoft® SQL Server® Data Services

Microsoft's Data Platform vision meets the needs of the evolving data explosion and the next generation of data-driven web applications with its new services offering in the cloud **called Microsoft® SQL Server™ Data Services (SSDS)**.

SQL Server Data Services (SSDS) is a highly scalable web facing data storage and query processing utility. Built on robust SQL Server database technology, these services provide high availability and security and support standards-based web protocols and interfaces (SOAP, REST) for rapid provisioning and ease of programming.

Businesses can store and access all types of data from birth to archival, using Microsoft SQL Server Data Services. Users can access information on any device, from the desktop to a mobile device.

Read on to learn how *SQL Server Data Services* deliver on the Microsoft Data Platform vision and meets the needs of the next generation of data-driven applications.

Key Features and Solution Benefits.

Application Agility for quick deployment

- Internet standard protocols and Interfaces (REST, SOAP).
- Flexible data model with no schema required.
- Simple text base query model.
- Easy to program to from any programming environment.

On-Demand Scalability

- Easy storage and access. Pay as you grow model.
- Scales as data grows.
- Web services for provisioning, deployment, and monitoring.

Business-Ready SLA

- Built on robust Microsoft SQL Server database and Windows server technologies.
- Store and manage multiple copies of the data for reliability and availability.
- Back up data stored in each data cluster. Geo-redundant data copies to ensure business continuity.
- Secure data access to help provide business confidentiality and privacy.

Solution Overview

Provisioning

Microsoft SSDS provisioning model is organized around the following hierarchy.

Customer { SSDS account (1..N) { Authority (1..N) { Container (0..N) { Entity (0..N)

- *Customers* are companies, organizations, individuals who want to use the SQL Server Data Services.
- *Accounts* are billing entities. Customers can open as many Microsoft SSDS accounts as they like. However, each account must use a unique LiveId. Each account has billing information associated with it.
- *Authorities* give billing entities a way to organize their usage for accounting, security and co-location purposes. All containers under a single authority are provisioned within the same data center. As such authorities are the unit of geo-scale and geo-location.
- *Containers* create contexts and scope for entity storage and query. For example, within its authorities, operations could choose to assign each member their own container, intended to contain a set of personal data for that member. Containers are the unit of consistency in the Microsoft SSDS service.
- *Entities* are the fundamental unit of storage in the system. For example, an individual member's jobs, educational institutions, contacts, recommendations, etc. could all be modeled as entities.

The figure below shows the relationship of authorities, containers, and entities.

Business Logic Layer	Definition	Purpose	Example (using Live Expo)
Authority	Set of containers	Organize containers for accounting, security, co-location	“Seattle”, “San Francisco”
Container	Set of entities	Organize entities for context, queries	“Autos for Sale,” “Services offered”
Entity	Initially a scalar property bag	Unit of storage	A particular “car for sale” ad as a set of field name, type, value triples

Data model

Microsoft SSDS supports a flexible data model with no schemas required. An entity is the smallest unit of consistency. It can be independently serialized, updated, etc.

Entities are “flat scalar property bags.” That is to say, entities have properties, properties have values, and those values are scalars. Every entity has service metadata properties as well as user defined properties.

There is no fixed typing of entities. From the point of view of the service, the set of properties associated with an entity are independent of the properties of any other entity. Further, the type of the value of a given property may vary from entity to entity. Any standardization of properties or property types across entities is under the control of the application developer.

The data types supported for properties are string, binary, Boolean, number, and date-time.

Data manipulation

The data manipulation model enables Create/Update/Delete operations on Authorities, Containers, and Entities. Microsoft SSDS will help enable the creation, modification, and deletion of Microsoft SSDS accounts and authorities via a Web site interface.

Data manipulation operations include:

- Creation and deletion of containers. There are no updatable container properties.
- Creation, replacement, and deletion of entities.
- Retrieval of a single container in a serialized format.
- Retrieval of a single entity in a serialized format.

Query language

Microsoft SSDS supports a text based query language. This language follows the LINQ pattern for C#. The query language is designed to enable simple filtering scenarios. Queries can be addressed to single authorities or single containers as per the following rules.

- An authority can be queried for the containers within it matching a specified condition. In this case, query scope is limited to the single authority to which the query is submitted.
- A container can be queried for all entities within that container matching a specified condition. In this case, query scope is limited to the single container to which the query is submitted.

The query language supports the retrieval of complete entities only. Selection conditions are defined by boolean combination of primitive comparison operations. Boolean operators to be supported are AND, OR, NOT. Comparison operators to be supported are <, >, <=, >=, !=, ==. Comparison operator semantics are those provided for the SQL variant type. All comparisons have the form (property OP constant_or_param).

For example, a query addressed to a container to retrieve all entities in that container having a “City” property equal to “Seattle” and a “State” property equal to “WA” would be written as follows:

```
from e in entities
where e["City"] == "Seattle" &&
      e["State"] == "WA"
select e
```

In this query ‘entities’ binds to the target container for the query.

Resource-based query

Microsoft SSDS also supports resource query. Resource query navigates a path to a single entity or container of entities and allows filtering of the leaf item(s) found there.

For example, the REST URI below refers to the specific container ChildrensBooksContainer1 beneath the authority mydomain.ssdsmicrosoft.com. The URI returns all the entities in that container.

```
http://mydomain.ssdsmicrosoft.com/ChildrensBooksContainer1
```

This REST URI refers to a specific entity resource (SomeBook) in the container identified by the URI above. The URI returns the single entity if it exists.

```
http://mydomain.ssdsmicrosoft.com/ChildrensBooksContainer1/SomeBook
```

Security

Security is at the account, authority and container levels. Accounts are secured by a Windows Live Id. Each authority is secured by a single “secret key” granting read/write access. Further, each container

within an authority is secured by its own single “secret key” granting read/write access. Optionally the contents of a container can be made read-accessible to the public.

API

The runtime aspects of the Microsoft SSDS service are made available via a set of web services. The primary service endpoint is RESTful service. Authorities, containers, and entities is directly accessible via URI addresses. Additionally, SOAP based endpoints is supported. One SOAP endpoint is supported for each authority. The primary wire format is XML at this time. We expect SSDS to support multiple protocols including AtomPub. A rich client library in C# or VB providing LINQ query support is also available.

Reliability, Availability, Security and Performance (RASP)

Microsoft SSDS is a distributed systems fabric hosting intrinsic SQL Server capabilities. Microsoft SSDS’s RASP properties derive fundamentally from it being built on top of the SQL Server Database technology.

For more information, please visit the site:

<http://www.microsoft.com/sql/dataservices/default.aspx>