<https://docs.microsoft.com/en-us/sql/relational-databases/data-compression/unicode-compression-implementation>

# Unicode Compression Implementation

2017-3-14 1 min to read Contributors

* [BYHAM](https://github.com/BYHAM)

* [Craig Guyer](https://github.com/craigg-msft)

**THIS TOPIC APPLIES TO:** yesSQL Server (starting with 2016)yesAzure SQL DatabasenoAzure SQL Data Warehouse noParallel Data Warehouse

SQL Server uses an implementation of the Standard Compression Scheme for Unicode (SCSU) algorithm to compress Unicode values that are stored in row or page compressed objects. For these compressed objects, Unicode compression is automatic for **nchar(n)** and **nvarchar(n)** columns. The Database Engine stores Unicode data as 2 bytes, regardless of locale. This is known as UCS-2 encoding. For some locales, the implementation of SCSU compression in SQL Server can save up to 50 percent in storage space.

## Supported Data Types

Unicode compression supports the fixed-length **nchar(n)** and **nvarchar(n)** data types. Data values that are stored off row or in **nvarchar(max)** columns are not compressed.

Note

Unicode compression is not supported for **nvarchar(max)** data even if it is stored in row. However, this data type can still benefit from page compression.

## Upgrading from Earlier Versions of SQL Server

When a SQL Server database is upgraded to SQL Server 2017, Unicode compression–related changes are not made to any database object, compressed or uncompressed. After the database is upgraded, objects are affected as follows:

* If the object is not compressed, no changes are made and the object continues to function as it did previously.
* Row- or page-compressed objects continue to function as they did previously. Uncompressed data remains in uncompressed form until its value is updated.
* New rows that are inserted into a row- or page-compressed table are compressed using Unicode compression.

Note

To take full advantage of the benefits of Unicode compression, the object must be rebuilt with page or row compression.

## How Unicode Compression Affects Data Storage

When an index is created or rebuilt or when a value is changed in a table that was compressed with row or page compression, the affected index or value is stored compressed only if its compressed size is less than its current size. This prevents rows in a table or index from increasing in size because of Unicode compression.

The storage space that compression saves depends on the characteristics of the data that is being compressed and the locale of the data. The following table lists the space savings that can be achieved for several locales.

| Locale | Compression percent |
| --- | --- |
| English | 50% |
| German | 50% |
| Hindi | 50% |
| Turkish | 48% |
| Vietnamese | 39% |
| Japanese | 15% |

## See Also

[Data Compression](https://docs.microsoft.com/en-us/sql/relational-databases/data-compression/data-compression)  
[sp\_estimate\_data\_compression\_savings (Transact-SQL)](https://docs.microsoft.com/en-us/sql/relational-databases/system-stored-procedures/sp-estimate-data-compression-savings-transact-sql)  
[Page Compression Implementation](https://docs.microsoft.com/en-us/sql/relational-databases/data-compression/page-compression-implementation)  
[sys.dm\_db\_persisted\_sku\_features (Transact-SQL)](https://docs.microsoft.com/en-us/sql/relational-databases/system-dynamic-management-views/sys-dm-db-persisted-sku-features-transact-sql)