**Heap Structures**

SQL Server 2008 Books Online (November 2009)

A heap is a table without a clustered index. Heaps have one row in [sys.partitions](http://msdn.microsoft.com/en-us/library/ms175012.aspx), with **index\_id** = 0 for each partition used by the heap. By default, a heap has a single partition. When a heap has multiple partitions, each partition has a heap structure that contains the data for that specific partition. For example, if a heap has four partitions, there are four heap structures; one in each partition.

Depending on the data types in the heap, each heap structure will have one or more allocation units to store and manage the data for a specific partition. At a minimum, each heap will have one IN\_ROW\_DATA allocation unit per partition. The heap will also have one LOB\_DATA allocation unit per partition, if it contains large object (LOB) columns. It will also have one ROW\_OVERFLOW\_DATA allocation unit per partition, if it contains variable length columns that exceed the 8,060 byte row size limit. For more information about allocation units, see [Table and Index Organization](http://msdn.microsoft.com/en-us/library/ms189051.aspx).

The column **first\_iam\_page** in the **sys.system\_internals\_allocation\_units** system view points to the first IAM page in the chain of IAM pages that manage the space allocated to the heap in a specific partition. SQL Server uses the IAM pages to move through the heap. The data pages and the rows within them are not in any specific order and are not linked. The only logical connection between data pages is the information recorded in the IAM pages.

|  |
| --- |
| **ms188270.note(en-us,SQL.100).gifImportant:** |
| The **sys.system\_internals\_allocation\_units** system view is reserved for Microsoft SQL Server internal use only. Future compatibility is not guaranteed. |

Table scans or serial reads of a heap can be performed by scanning the IAM pages to find the extents that are holding pages for the heap. Because the IAM represents extents in the same order that they exist in the data files, this means that serial heap scans progress sequentially through each file. Using the IAM pages to set the scan sequence also means that rows from the heap are not typically returned in the order in which they were inserted.

The following illustration shows how the SQL Server Database Engine uses IAM pages to retrieve data rows in a single partition heap.

