

# MS SQL's Casting Causes Consternation

Posted At : October 2, 2006 6:02 PM | Posted By : Mark Kruger

Related Categories: MS SQL Server

If you are using MSSQL Server an Cfqueryparam, you should be aware of the old *implicit datatype conversion*. This helpful feature uses an order of precedence to accommodate code that is written to pass in a sql\_variant data type. To put it another way, lets say you are comparing a column of the type "varchar" to the the number 1 or 2. If it was me, I would write "WHERE myCol = '1' OR myCol = '2'" - or perhaps use an "IN" clause. This code will always work. It is saying "compare a character '2' to the character column myCol". Since they are both of the same type there is no conflict - but what happens if you mismatch the type?

## Sometimes Mismatching Works

Let's say you create a "type" that you set to an int and use a type of 1 or 2. Your where clause is going to say "WHERE myType = 2". Let's also suppose that after the fact you reconsider what you might want to put in the "type" column so you change the type to a string. Your original query still works - why? It turns out that MS SQL helps you out and simply converts your column to a type of int before comparing it.

*Did you catch that?* A query that looks like this:

```
<cfquery name="test" datasource="blah">
  SELECT *
  FROM    myItems
  WHERE myType = <cfqueryparam cfsqltype="CF_SQL_INTEGER" value="1">
</cfquery>
```

Or without the cfqueryparams...

```
<cfquery name="test" datasource="blah">
  SELECT *
  FROM    myItems
  WHERE myType = 1
</cfquery>
```

will not do exactly what you expect. It's NOT going to CAST your variable as a character. Nope - MSSQL will assume you knew what you were doing. Instead it will CAST the column AS an INT. So under the hood, the MSSQL statement will look like this:

```
<cfquery name="test" datasource="blah">
  SELECT *
  FROM    myItems
  WHERE CAST(myType AS INT) = 1
</cfquery>
```

## The Catch

The problem is that this will actually work *as long as there are only castable values in the column*. In other words, as long as 1,2,3 and 4 are stored in the column the query will run. As soon as you put an "A" or some other alpha string in the column the query will error out. This is why it is important to get the cfqueryparam data type correct in the first place. Such an error could lie hidden during development but rear it's ugly head after launch as new values are put into the column.

