

? : (Conditional) (SSIS Expression)

SQL Server 2012 | 4 out of 11 rated this helpful

Returns one of two expressions based on the evaluation of a Boolean expression. If the Boolean expression evaluates to TRUE, then the first expression is evaluated and the result is the expression result. If the Boolean expression evaluates to FALSE then the second expression is evaluated and its result is the expression result.

▲ Syntax

```
boolean_expression?expression1:expression2
```

▲ Arguments

boolean_expression

Is any valid expression that evaluates to TRUE, FALSE, or NULL.

expression1

Is any valid expression.

expression2

Is any valid expression.

▲ Result Types

The data type of *expression1* or *expression2*.

▲ Remarks

If the *boolean_expression* evaluates to NULL, the expression result is NULL. If a selected expression, either *expression1* or *expression2* is NULL, the result is NULL. If a selected expression is not NULL, but the one not selected is NULL, the result is the value of the selected expression.

If *expression1* and *expression2* have the same data type, the result is that data type. The following additional rules apply to result types:

- The DT_TEXT data type requires that *expression1* and *expression2* have the same code page.
- The length of a result with the DT_BYTES data type is the length of the longer argument.

The expression set, *expression1* and *expression2*, must evaluate to valid data types and follow one of these rules:

- **Numeric** Both *expression1* and *expression2* must be a numeric data type. The intersection of the data types must be a numeric data type as specified in the rules about the implicit numeric conversions that the expression evaluator performs. The intersection of the two numeric data types cannot be null. For more information, see [Integration Services Data Types in Expressions](#).
- **String** Both *expression1* and *expression2* must be a string data type: DT_STR or DT_WSTR. The two expressions can evaluate to different string data types. The result has the DT_WSTR data type with a length of the longer argument.
- **Date, Time, or Date/Time** Both *expression1* and *expression2* must evaluate to one of the following data types: DT_DBDATE, DT_DATE, DT_DBTIME, DT_DBTIME2, DT_DBTIMESTAMP, DT_DBTIMESTAMP2, DT_DBTIMESTAMPOFFSET, or DT_FILETIME.

Note

The system does not support comparisons between an expression that evaluates to a time data type and an expression that evaluates to either a date or a date/time data type. The system generates an error.

When comparing the expressions, the system applies the following conversion rules in the order listed:

- When the two expressions evaluate to the same data type, a comparison of that data type is performed.
- If one expression is a DT_DBTIMESTAMPOFFSET data type, the other expression is implicitly converted to DT_DBTIMESTAMPOFFSET and a DT_DBTIMESTAMPOFFSET comparison is performed. For more information, see [Integration Services Data Types in Expressions](#).
- If one expression is a DT_DBTIMESTAMP2 data type, the other expression is implicitly converted to DT_DBTIMESTAMP2 and a DT_DBTIMESTAMP2 comparison is performed.
- If one expression is a DT_DBTIME2 data type, the other expression is implicitly converted to DT_DBTIME2, and a DT_DBTIME2 comparison is performed.
- If one expression is of a type other than DT_DBTIMESTAMPOFFSET, DT_DBTIMESTAMP2, or DT_DBTIME2, the expressions are converted to the DT_DBTIMESTAMP data type before they are compared.

When comparing the expressions, the system makes the following assumptions:

- If each expression is a data type that includes fractional seconds, the system assumes that the data type with the least number of digits for fractional seconds has seven for

that the data type with the least number of digits for fractional seconds has zeros for the remaining digits.

- If each expression is a date data type, but only one has a time zone offset, the system assumes that the date data type without the time zone offset is in Coordinated Universal Time (UTC).

For more information about data types, see [Integration Services Data Types](#).

▲ Expression Examples

This example shows an expression that conditionally evaluates to **savannah** or **unknown**.

```
@AnimalName == "Elephant"? "savannah": "unknown"
```

This example shows an expression that references a **ListPrice** column. **ListPrice** has the DT_CY data type. The expression conditionally multiplies **ListPrice** by .2 or .1.

```
ListPrice < 350.00 ? ListPrice * .2 : ListPrice * .1
```

▲ See Also

Concepts

[Operator Precedence and Associativity](#)

Other Resources

[Operators \(SSIS Expression\)](#)

Community Additions

© 2013 Microsoft. All rights reserved.