

# Utility Function: QueryAddRowData()

Posted At : January 8, 2009 5:30 AM | Posted By : Jon Hartmann

Related Categories: Utility Function

This one probably isn't original, but I wanted to be able to add whole rows to query sets in a simple manner. The built in functions let you add new columns in a simple way, but not new rows. That's kind of weird, since I'd think that people would want to add rows more often than the columns. QueryAddRowData() is designed to be flexible in its allowed inputs, working with structs, arrays, and lists.

```
<cffunction name="QueryAddRowData" access="public" output="false" returntype="void">
    <cfargument name="query" type="query" required="true" />
    <cfargument name="data" type="any" required="true" />
    <cfargument name="delimiter" type="string" required="false" default="," />

    <cfset var local = StructNew() />

    <cfif NOT (IsArray(arguments.data) OR (IsStruct(arguments.data) AND NOT isObject(arguments.data)) OR IsSimpleValue(arguments.data))>
        <cfthrow errorcode="QueryAddRowData" message="Invalid data for argument data. Valid types are array, struct, and string." />
    </cfif>

    <cfset QueryAddRow(arguments.query) />

    <cfif IsArray(arguments.data)>
        <cfset local.columns = ListToArray(arguments.query.columnList) />
        <cfloop from="1" to="#ArrayLen(local.columns)#" index="x">
            <cfset QuerySetCell(arguments.query, local.columns[x], arguments.data[x]) />
        </cfloop>

    <cfelseif IsStruct(arguments.data) AND NOT isObject(arguments.data)>
        <cfloop collection="#arguments.data#" item="column">
            <cfset QuerySetCell(arguments.query, column, arguments.data[column]) />
        </cfloop>

    <cfelseif IsSimpleValue(arguments.data)>
        <cfset local.data = ListToArray(arguments.data, arguments.delimiter) />
        <cfset local.columns = ListToArray(arguments.query.columnList) />
        <cfloop from="1" to="#ArrayLen(local.columns)#" index="x">
            <cfset QuerySetCell(arguments.query, local.columns[x], local.data[x]) />
        </cfloop>
    </cfif>

    <cfreturn />
</cffunction>
```

Please note that with arrays and lists, the data must be ordered as it would be in the Query.columnList (alphabetically).