

# JScript Object Structures v1.0

Last Updated November 19, 1998

This document is intended for use by developers who wish to understand how the JavaScript objects that correspond to each NFX component work, or wish to modify the JavaScript objects. Each ASP component for NetObjects Fusion consists of two parts - a Java-based NetObjects Fusion API component and a server-side JScript object. This document outlines the structure of each JScript object that corresponds to a NetObjects Fusion API component.

## MSDBConnection

Properties:		
String	databaseType	Database type ("INFORMIX", "ORACLE", "ODBC")
String	databaseServer	Name of the database server or ODBC service
String	globalConn	"true" = use initial page for a global connection for entire application, "false" = one connection per page
String	name	Name of DBConnect object
String	ODBCDatabaseType	ODBC type ("MS Access", "MS SQL Server", "Other")
String	password	Password used to log into the database
String	username	Username used to log into the database

Methods:	
MSDBConnection	Constructor.
connect	Tests for a database connection. If not already connected, attempts to make a connection to the database. Returns <b>true</b> if a connection is established, otherwise it returns <b>false</b> .
connected	Attempts to determine the state of the connection. If the connection exists then this function returns <b>true</b> . Otherwise, <b>false</b> is returned.
disconnect	Releases the connection to the database
emitProperties	Writes out all properties of the object to the page on which the object exists.
render	Outputs necessary HTML to render the object on an HTML page. Writes out hidden form fields to the page

## MSDBDynaField

Properties:		
Boolean	bold	Is text bold?
String	dataField	The field from which to get the DBDynaField value
String	dataType	The data type of the field from which to get the DBDynaField value
String	defaultValue	The default value for the DBDynaField if the Data Field property is not specified
String	font	Text font name
String	fontSize	Text font size
String	fontColor	Text font color

Boolean	isEditable	Flag to store whether the DBDynaField is editable or not. This determines if the render method writes an input text box or plain HTML text
Boolean	italic	Is text italic?
int	maxLength	The maximum length of characters of DBDynaField
String	name	Name of DBDynaField component
String	queryComponentName	Name of Query component
Boolean	underline	Is text underlined?
String	useQuery	Use previous query? String "true" == use previous query, string "false" == use this query
int	visibleLength	The visible length of the DBDynaField
int	visibleHeight	The visible height of the DBDynaField in lines. Text area fields are used where line count > 1.

### Methods:

MSDBDynaField	Constructor.
EmitProperties	Writes out all properties of the object to the page on which the object exists
GetColumnValue	Gets a value from a recordset field
Render	Outputs necessary HTML to render the object on an HTML pag

### MSDBList

### Properties:

int	borderSize	Table border size
int	cellPadding	Table cell padding
int	cellSpacing	Table cell spacing
Boolean	dataBold	Is data text bold?
String	dataFont	Data text font name
Boolean	dataItalic	Is data text italic?
int	dataSize	Data text size
int	dataTextColor	Data text color
Boolean	dataUnderline	Is data text underlined?
int	fieldCount	Count of fields to display (must be less than number of fields selected in query<not-enforced>)
String[]	fieldNames	Name of fields (must be selected from available fields in query component<not-enforced>)
String[]	fieldLabels	Column labels for each field
Boolean	hyperlinkError	Flag to determine whether or not a hyperlinked field is actually present or not. <b>false</b> indicates that no error exists, <b>true</b> indicates that an error exists, e.g. the user specified a hyperlink field that is not in the query
String	hyperlinkField	Field on which the hyperlink is placed (Field should be listed in fieldNames list<enforced>))
int	hyperlinkPage	Page to which the link jumps
int	keyFieldCount	Count of key fields (must be entered after and less than field count<enforced>)
String[]	keyFieldNames	Name of key fields (must be selected from field names list after defined<enforced>)
String[]	keyFieldTypes	Data types of the key fields
Boolean	labelBold	Is label text bold?

String	labelFont	Label text font name
Boolean	labelItalic	Is label text italic?
int	labelSize	Label text size
int	labelTextColor	Label text color
Boolean	labelUnderline	Is label text underlined?
int	maxRecords	Maximum number of records to display at one time
String	name	Name of List component
String	navComponent	Name of the Navigator component, if one exists, that is responsible for navigating the list's records.
String	queryComponent	Name of Query component
int	tableWidth	Overall width of the table in pixels

<b>Methods:</b>	
MSDBList	Constructor
checkFieldErrors	Checks for the presence of expected fields, key fields and hyperlink fields in query.
emitProperties	Writes out all properties of the object to the page on which the object exists
emitTableBody	Writes out data, possibly successful
emitTableFooter	Closes table, always successful
emitTableHeader	Opens table, sets formatting, outputs table column labels, always successful
emitTableRow	Writes out a formatted data row as a row in the HTML table
render	Outputs necessary HTML to render the object on an HTML page. Calls emitTableHeader, emitTableBody, emitTableFooter.

## ***MSDBNav***

<b>Properties:</b>		
String []	detailArray	Contains values of the key fields
boolean	detailPage	<b>true</b> / <b>false</b> indicating whether navigation is on a detail page.
int	keyFieldCount	Count of key fields
String []	keyFieldNames	Name of key fields
String []	keyFieldTypes	Data types of key fields
String	lastRequestString	String to be appended to the URL query string of the <i>previous</i> button
String	name	Name of query component
String	nextImage	URL for the image of the Next button
String	nextRequestString	String to be appended to the URL query string of the <i>next</i> button
String	numRecords	Number of records to move on next or previous
String	pageName	URL of the current page
String	previousImage	URL for the image of the Previous button
String	queryComponent	Name of Query component
String	returnImage	URL for the image of the Return to List button
String	returnPageName	URL of the page to return to (List Page)
String	returnPageURLString	
Boolean	validRecord	True if the current record is valid within the recordset; otherwise, false

<b>Methods:</b>	
MSDBNav	Constructor

createBaseCursor	Creates a recordset on the entire, original data set indicated by the queryComponent property
createRequestString	
emitProperties	Writes out all properties of the object to the page on which the object exists
findMatch	Loops through the data set defined by the baseCursor to find the previous, current, and next records in the data set
keyFieldMatch	Determines if the record pointed to by the baseCursor in the looping process of findMatch is the record being navigated to
render	Outputs necessary HTML to render the object on an HTML page
generateString	Generates string used by the query to determine how to advance the recordset
setDetailArray	Populates detail array with key field values

## MSDBQuery

<b>Properties:</b>		
Object	cursor	An ASP recordset object
String	from	From portion of SQL
Boolean	isCursorInitialized	<b>true</b> / <b>false</b> Flag indicating whether the recordset had been advanced to the first record
Boolean	isNavigating	Whether or not we are currently in the midst of navigating a recordset
Boolean	listActive	Flag to determine if the page that this DBQuery component exists within was arrived at from a page containing a hyperlinked DBList object. Used in conjunction with "usePreviousQuery" to determine if the object should use it's own query parameters or those of the DBQuery object from the previous page that contained a DBList object. (Gives the appearance that user is navigating the rows displayed in the previous list).
String	matchAll	% or blank
String	matchParm	= or like
String	matchStart	% or blank
String	matchType	"exact match", "contains", "starts with"
String	name	Name of query component
String	orderBy	Order By portion of SQL
String	select	Select portion of SQL
String	sql	Current sql statement
String	where	Where portion of SQL
Object	newReq	Object
String	oldURLString	Holds the old URLString from a previous
String	URLString	URLString needed by DBNav to navigate back to a previous list page containing a DBList component
Boolean	usePreviousQuery	Whether query should use it's own SQL string or that of the query on the previous page.
Number	cursorPosition	The ordinal position of the cursor before navigation begins on the current page.

<b>Methods:</b>	
MSDBQuery	Constructor
buildSQL	Builds SQL, adding passed constraints to the where clause and concatenating the 4 parts
buildWhere	Builds the SQL "where" clause from user entered property and key fields passed in via the Request object
closeCursor	Performs checks for null and closes recordset object
emitProperties	Writes out all properties of the object to the page on which the object exists.
getCurrentCursor	Returns either the current recordset if it exists or a new, initialized recordset (calling openCursor and initializeCursor)
getQueryParams	Resets the object's query parameters (select, from, where, order by) to those stored in client object from a previous query if "Use Previous Query" is set to <b>true</b> . Otherwise, copies it's query parameters to the client object's select, from, where and order by properties for potential use by some other DBQuery object on another page
getURLStr	Return value of old URL string from a previous list page
initializeCursor	Checks the isCursorInitialized property. If False moves recordset to next

	record and sets property to <b>true</b>
isCursorEnd	Detects end of record set
isCursorStart	Detect beginning of record set
isNavigating	Determines whether or not we are currently in the midst of navigating a recordset
moveNext	Advances the cursor forward the number of records indicated. Returns the number of records actually moved. Returns 0 if at EOF
movePrevious	Advances the cursor backwards the number of records indicated. Returns the number of records actually moved. Returns 0 if at BOF
openCursor	Creates a new recordset
setCursor	Replaces the current cursor for both this.cursor and Session(this.name) with the recordset object passed into the method
setURLStr	Stores value of old URL string from a previous list page

## ***MSDBUpdate***

<b>Properties:</b>		
String	action	
String	errorURL	The URL of the error page
String	name	The name of the component
int	fieldCount	The number of fields to be modified
String []	fieldNames	The names of the fields to be modified
String []	fieldDataTypes	The data types of the fields to be modified
String []	fieldValues	The new values that will be used to modify an existing row
int	primaryKeyFieldCount	The number of primary key fields in the table to be updated
String []	primaryKeyFieldNames	The names of the primary key fields in the table to be updated
String []	primaryKeyFieldValues	The values of the primary key fields in the table to be updated
String []	primaryKeyFieldTypes	The data types of the primary key fields in the table to be updated
String	queryComponent	Name of query component from which to gather key field values
String	tableName	Name of table to update
String	successURL	The URL of the success page

<b>Methods:</b>	
MSDBUpdate	Constructor
buildAddSQL	Builds the SQL statement for add actions
buildDeleteSQL	Builds the SQL statement for delete actions
buildModifySQL	Builds the SQL statement for modify actions
buildWhereSQL	Builds the SQL where clause for modify actions, based upon the primaryKeyFieldValues
emitProperties	Writes out all properties of the object to the page on which the object exists
execute	Executes the SQL statement to perform the add, modify or delete action
getValues	Gets the key field values and populates the primaryKeyFieldValues array
render	Outputs necessary HTML to render the object on an HTML page
setDatabaseTypes	Used by the form handler to read in database types sent over by hidden form fields. Uses the values to determine the database type so it can properly perform an update against a specific database type