About The Intranet Database Assistant (IDBA)

The Intranet Database Assistant is a high performance personal web database server product for Windows® 95. It allows you to host HTML pages with dynamic database content using ODBC names and a unique extension to HTML.

In a peer-to-peer environment, IDBA presents a very powerful remote database access tool, ideal for workgroup/intranet applications where individuals have the capability to interactively share HTML and database content with each other.

Outstanding benefits of the Intranet Database Assistant:

- Free personal web server with ODBC database access for licensed Windows ® 95 machines.
- Easy and uniquely powerful HTML extensions for dynamic database access and user interaction.
- Multiple database queries on the same page.
- Generates standard HTML types, filling them with database data.
- Additional functionality for form variables.
- Server persistence of data between HTML pages.
- Supports MS-Exchange "inbox" email and faxing.
- User defined error reporting.

HTML 3.0 generated types currently supported are;

- TABLE
- OLIST
- ULIST
- MENU
- SELECT

In addition to these types, IDBA supports;

- MODIFY
- SPREAD

This mechanism of dynamically generating data filled types, removes a great deal of work from the HTML page designer. Enabling the designer to focus his/her efforts on producing good HTML pages without having to worry about learning a new language, or creating other unnecessary script files, which ultimately slow down the most efficient of web servers. The only requirement is that the designer understand SQL so that efficient and correct queries are generated for the appropriate HTML type objects.

Even if invalid queries occur, IDBA supports error tags which allow you to output an error message in HTML text, or even jump to another HTML file in your local LAN environment.

Powerful features of the Intranet Database Assistant;

- It can substitute HTML form variables that are generated by a POST or GET in the ACTION page, to either generate dynamic queries or simply display variable values. This enables IDBA to support server

persistence simply through HTML [not browser dependent].

- It supports the ability to email or directly fax data through a comment tag in your HTML file. This capability is supported through your Windows ® 95 MS- Exchange “inbox”.

- It supports multiple queries within the same HTML page

IDBA Syntax

IDBA Syntax is always contained within HTML comment tags, so that if a page using the syntax is displayed using another web server it appears just as a comment.

Within the comment tags, the remaining syntax is just an **action-key** with its related **argument-tags** that are separated by the "::" specifier.

The ending syntax is denoted by the keyword "End>".

IDBA Syntax is:

<! *Action-Key*:: ….*Argument-Tags* ::End>

*Action-Key*s are:

- Query- Instructs IDBA to query a database using argument tags.
- Send- Instructs IDBA to send a message using argument tags.
- Data- Instructs IDBA to display a memory based table data cell.

Remarks

The syntax for the DATA action-key is different in that it does not have an "End" keyword and its argument-tags are represented more in a function call style.

IDBA supports the substitution of a HTML form variable key, which can be used to generate dynamic queries or simply to display variable values. See VAR for more details.

[Please note that, although IDBA is distributed free, it is licensed and the HTML extensions (API) supported to access database data and email/fax are **strictly** protected by international copyright laws.]

Hints

Always pack IDBA syntax tightly, avoid unnecessary white space which may lead to parsing errors or slower response times.

IDBA syntax can be written over multiple lines for larger argument entries.

If IDBA doesn't seem to be parsing your HTML file correctly, then try to break up the argument tags into separate lines, as shown in the example below. [Note: that by doing this, your response time might be affected. However, your syntax is more readable !.]

Example:

```
<!Query::
  DSN=NorthWind::
  SQL=Select * from Employees::
  TYPE=TABLE::
```

```
End>
```

Query [*Action-Key***]**

Instructs IDBA to query a database using the required *Argument-Tags*:

- DSN - ODBC data source name.
- SQL - SQL syntax.
- TYPE - HTML/IDBA generated type.

Optional *Argument-Tags* are:

- OK - display text if query succeeds.
- FAIL - display text if query fails.

Minimal query syntax is of the form:

<!Query::DSN=*xxx*::SQL=*xxx*::TYPE=*xxx*::End>

Send [*Action-Key***]**

Instructs IDBA to send a fax or email using the required *Argument-Tags*:

- TO - recipient.
- SUBJECT - title of message.
- MESSAGE - message content.

Optional *Argument-Tags* are:

- OK - display text if send succeeds.
- FAIL - display text if send fails.

Minimal query syntax is of the form:

<!Send::TO=*xxx*::SUBJECT=*xxx*::MESSAGE=*xxx*::End>

TYPE [*Argument-Tag***]**

TYPE is a **required** *Argument-Tag* for the Query *Action-Key*. It specifies an output HTML object that replaces the IDBA query syntax in an HTML file.

The HTML object that is generated, is populated with data that returns through a successful SQL query. The format of the return data is important for the TYPE of object that will be generated. For example, To display a HTML table type, the query should return a table of data.

IDBA Supported TYPES are:

- TYPE=TABLE::
- TYPE=OLIST::
- TYPE=ULIST::
- TYPE=MENU::
- TYPE=SELECT::
- TYPE=MODIFY::
- TYPE=SPREAD::

TABLE [*Query*-*TYPE*]

This TYPE generates a HTML equivalent of the <TABLE> object, where each column name in the result set is included using the <TH> tag.

This TYPE can also have the following variation;

- TYPE=TABLE(*vName*,*vBorder*)::
- TYPE=TABLE(*vName*)::

vName - specifies a variable name to use for the <CAPTION> tag of a table.
VBorder = 1, specifies a table with a border. *VBorder* = 0, specifies a table with no border.

Remarks

Multiple table objects can be generated as a result of one SQL query.
If the default TYPE=TABLE:: is used, then NO caption is generated.
The default for *vBorder* in TYPE=TABLE:: and TYPE=TABLE(vName):: is 1.

Hints

To specify a table with no caption and no border use TYPE=TABLE(,0):: remember to use the comma to separate the arguments.

OLIST [*Query-TYPE*]

This TYPE generates a HTML equivalent of the <OL> object, where each row in the result set is displayed using the <LI> tag.

Remarks

Ideally ensure that the SQL query result set has only one column. If it has more than one, then each column data value in the same row will be joined together to form one string with the <OL> tag. No separators are included between data cells.

ULIST [*Query-TYPE*]

This TYPE generates a HTML equivalent of the <UL> object, where each row in the result set is displayed using the <LI> tag.

Remarks

Ideally ensure that the SQL query result set has only one column. If it has more than one, then each column data value in the same row will be joined together to form one string with the <UL> tag. No separators are included between data cells.

MENU [*Query-TYPE*]

This TYPE generates a HTML equivalent of the <LI> object, where each row in the result set is displayed using the <MENU> tag.

Remarks

Ideally ensure that the SQL query result set has only one column. If it has more than one, then each column data value in the same row will be joined together to form one string with the <MENU> tag. No separators are included between data cells.

SELECT [*Query-TYPE*]

This TYPE generates a HTML equivalent of the <SELECT> object, where each row in the result set is displayed using the <OPTION> tag.

This TYPE can also have the following variations;

- TYPE=SELECT(*vName*,*vDefault*)::
- TYPE=SELECT(*vName*)::

vName - specifies a variable name to use for a selected value for a form. It has the effect of generating a <SELECT NAME="*vName*"> tag instead of a simple <SELECT> tag.

vDefault - specifies a default selection to use. In this case the data cell value which matches *vDefault* generates a <OPTION SELECTED> instead of a plain <OPTION> tag.

Remarks

Ideally ensure that the SQL query result set has only one column. If it has more than one, then each column data value in the same row will be joined together to form one string with the <OPTION> tag. No separators are included between data cells.

The SELECT object should only be used within <FORM … ACTION=""> tag specifiers, as its primary purpose is to allow the user to select a variable which is used in the action HTML page.

MSELECT [*Query-TYPE*]

This TYPE generates a HTML equivalent of the <SELECT MULTIPLE> object, where each row in the result set is displayed using the <OPTION> tag.

This TYPE can also have the following variations;

- TYPE=MSELECT(*vName*,*vDefault*)::
- TYPE=MSELECT(*vName*)::

vName - specifies a variable name to use for a selected value for a form. It has the effect of generating a <SELECT NAME="*vName*" MULTIPLE> tag instead of a simple <SELECT MULTIPLE> tag.

vDefault - specifies a default selection to use. In this case the data cell value which matches *vDefault* generates a <OPTION SELECTED> instead of a plain <OPTION> tag.

Remarks

Ideally ensure that the SQL query result set has only one column. If it has more than one, then each column data value in the same row will be joined together to form one string with the <OPTION> tag. No separators are included between data cells.

MODIFY [*Query-TYPE*]

This TYPE is used to modify data within a particular database, and does not generate any output HTML object.

Remarks

To be used with any SQL query that modifies data within a database, such as;

INSERT, **UPDATE** or **DELETE** specifiers.

Use the "OK=" or "FAIL=" tags to specify any HTML output based on the respective success or failure of the query.

SPREAD [*Query-TYPE*]

This TYPE is used to retain a table in memory as a result of a SQL query, and does not generate any output HTML object.

Remarks

Will only retain one table in memory.

A table is retained in memory between SPREAD usage. That is, the table in memory is only valid upto the next SPREAD type used.

To access table data in memory use the DATA Action-*Key.*

DATA [*Action-Key***]**

This key is used to access the data cells of a table held in memory as a result of a Query that uses the SPREAD[Query -TYPE]. It will display the data accessed in your HTML file.

DATA has the format: <!DATA(*row,col*)>

row is a one based index into the rows of the table in memory.
col is a one based index into the columns of the table in memory.

Remarks

The data held in the memory table is only valid between SPREAD queries.

row = 0 will index the column headers of a table.
col = 0 is **illegal** and should not be used !

An "End>" specifier is not used, and is invalid for this *Action-Key.*
"OK=" and "FAIL=" tags are currently invalid for this *Action-Key*.

The DATA *Action-Key* can be used anywhere inside your HTML file.
Hint: for correct parsing always pack a data statement tightly. E.g. <!DATA(1,2)> has no whitespace in it.

DSN [Argument-Tag]

DSN is a **required** *Argument-Tag* for the Query *Action-Key*. It specifies the ODBC *datasourcename* that is used to identify a database to connect to.

Remarks

The *datasourcename* is case-sensitive and must be exactly the same as specified in your local ODBC configuration program [See: ControlPanel on your desktop].

The *datasourcename* used in a query **must** be added to the "IDBA Sources" list. To confirm this, select the "configure" menu option from IDBA's view menu, and select the "DataSource" tab.

SQL [*Argument-Tag*]

SQL is a **required** *Argument-Tag* for the Query *Action-Key*. It specifies the SQL query to pass through to a database.

Remarks

Can be any valid SQL syntax.
Must provide appropriate result for the [Query-TYPE] being used.

Hint: To keep parsing to a minimum for performance, place complex queries in stored procedures or views and access them using the SQL tag.

e.g. SQL=execute MyStoredProc(x1,x2)::

OK [*Argument-Tag***]**

OK is an **optional** *Argument-Tag* for the Query and Send *Action-Keys*. It specifies what text to output if the IDBA action is successful.

The JUMP *Action-Tag* can be used to display data in another file. In this case, the *Argument-Tag* is written as:

OK=JUMP(*relPathAndFileName*)::

relPathAndFileName - is a variable that specifies a relative path and filename to load.

E.g: OK=JUMP(c:\html\errors\ok.html):: [NOTE: Do not use URL locators like http://…]

Remarks

Can be any text, including HTML specifiers.

FAIL [*Argument-Tag***]**

FAIL is an **optional** *Argument-Tag* for the Query and Send *Action-Keys*. It specifies what text to output if the IDBA action failed.

The JUMP *Action-Tag* can be used to display data in another file. In this case, the *Argument-Tag* is written as:

FAIL=JUMP(*relPathAndFileName*)::

relPathAndFileName - is a variable that specifies a relative path and filename to load.

E.g: FAIL=JUMP(c:\html\errors\fail.html):: [NOTE: Do not use URL locators like http://…]

Remarks

Can be any text, including HTML specifiers.

TO [*Argument-Tag***]**

TO is a **required** *Argument-Tag* for the Send *Action-Key*. It specifies the recipient to whom the message will be sent.

Remarks

Currently supports MS-Exchange "inbox" format. For example;

Email: TO=anonymous@somwhere.com
Fax: TO=FAX:1-503-968-7278

SUBJECT [*Argument-Tag***]**

SUBJECT is a **required** *Argument-Tag* for the Send *Action-Key*. It specifies the subject title of the message being sent.

Remarks

Can be any text.

MESSAGE [*Argument-Tag***]**

MESSAGE is a **required** *Argument-Tag* for the Send *Action-Key*. It specifies the message content of the message being sent.

Remarks

Can be any text.

VAR [*Form-Key-Variable*]

The VAR *Form-Key-Variable* instructs IDBA to replace a variable name by its value as received in a POST or GET action.

The following is valid syntax:

VAR(*vName*) - replaces vName with its value in HTML text.
VARQ(*vName*) - replace vName with its value enclosed with quotes (useful for SQL queries).

Remarks

VAR statements can be placed anywhere in an HTML file.
vName has to be a valid form variable, that is received by an action page that implements the VAR statement, in the format of "vName=value".

e.g. in form.html

```
<FORM ACTION="action.html">
Enter Details:<INPUT TYPE=hidden NAME="vTitle" VALUE="Mr. Z">
</FORM>
```

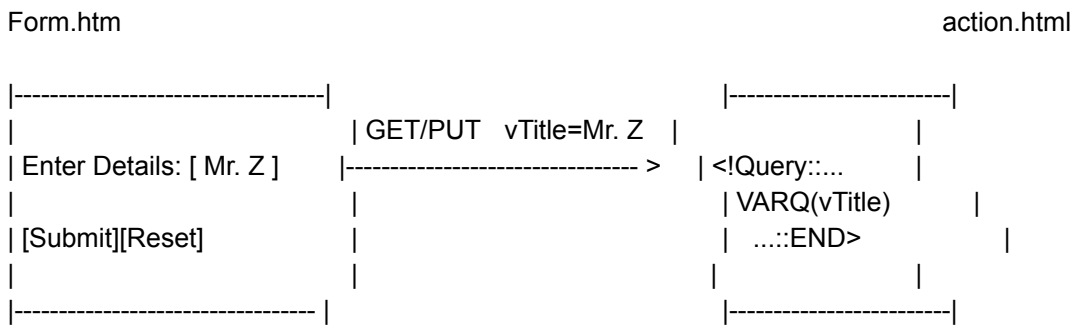
and in action.html

```
VAR(vTitle)
VARQ(vTitle)
```

when executed, the output shown when action.html gets loaded is;

Mr. Z
'Mr. Z'

This example is illustrated in more detail with the diagram below;



For More Information

Please contact us for more information at

Intranet 2001, Inc.
9220 S.W. Barbur Blvd #119-282
Portland, OR 97291

WWW: **http://www.inet2001.com**
Email : **info@inet2001.com**

Intranet 2001, Inc

Intranet 2001, Inc. has been founded to help corporations build up intranet solutions which tie seamlessly with their internet efforts. The goal is to help companies migrate from current legacy systems to I.net based solutions.

- Develop state of the art modular products.
- Work with partners to implement solutions based on our products.
- Co-develop legacy data access modules which tie into our products.

This enables corporations to build intranet solutions that interact with their host systems.

IDBA Requirements

IDBA will run as a personal web server on any (Intel based) PC that runs Windows 95.

To use IDBA Query syntax to access backend databases on your Win95 platform, IDBA requires the following;

1. ODBC must be installed. You can check this by looking for the "ODBC32" icon in Control Panel. If this ICON does not exist, then please refer to your Windows help guides to install ODBC and the appropriate drivers for your database.
2. Once you have ODBC32 installed, add your datasource names to it. Then activate and configure IDBA

To use IDBA Send syntax for email or fax purposes, IDBA requires MS-Exchange "inbox" (v4 or greater) to be installed and active on your Win95 platform. If you have it installed on your desktop, activate it and minimize the application.

To use the sample HTML files that come with IDBA, you need to have MS-Access and the sample database "Northwind.mdb" (that comes with MS-Access) installed on your PC.

Getting Started with the Sample HTML files

1. Please ensure that you have ODBC32, MS-Access and NorthWind.mdb installed on your system. [See IDBA requirements]

2. Run the executable IDBA.EXE found in the \Program Files\Intranet 2001\idba\program folder.
This can be done by clicking on the "Start" button found on the Win95 taskbar, then selecting the following menus;

Programs -> Intranet 2001 -> Intranet Database Assistant

If IDBA appears in your Win95 ICONTRAY then RightMouseButton click on the icon, select "Open" or "Configuration" in the menu, and either the application or the configuration dialog will appear.

3. If you have already have a web server running on your PC, then its port number address might confict with the IDBA's port address. If this is the case, then change IDBA's port number to another value (e.g. 40) by selecting the following menus on IDBA;

View -> Configuration -> Server

and change "Listen on HTTP Port:" default (80) to 40 (or any other reasonable value).
Select "OK" when done.

4. To try the samples, you specify a "Data Source Name" for the NorthWind.mdb sample that comes with MS-Access. This can be done by clicking on the "Start" button found on the Win95 taskbar, then selecting the following menus;

Settings -> Control Panel

- Double click on the "ODBC32" icon.
- Click on the "Add" button.
- In the "Add Data Source" dialog, select the Microsoft Access driver (*.mdb) in the ODBC drivers list box. Then click on "OK".
- In the ODBC MS-Access setup dialog box, Enter "NorthWind" in the DataSourceName edit box, and select "Northwind.mdb" using the Database->"Select" button. Click on "OK" and then click on "Close" on the DataSources dialog.

Next, select *View -> Configuration -> DataSources* on IDBA.

Select "NorthWind" in the ODBC Sources list, and then click on the "Add" button. This should add the NorthWind datasourcename to the IDBA sources list. Click on "OK".

5. Run your Internet browser and load "QueryForm.htm" by entering "http://127.0.0.1/" in the www address/location text entry field.

[Note: if you change IDBA's port number, the string becomes;
http://127.0.0.1:40/
where 40 specifies the port number (as in step 3 above)]

At this point you should be able to submit queries, and modify the HTML files to see what IDBA can do.

6. If you want to set up a default root directory to store your own HTML files, where anyone in your inet environment can access these files using your machine name. Then select the following menus on IDBA;

View -> Configuration -> HTML Files

And change the default directory and index file to point to your own default directory using the ‘‘Browse’’ button.

[Warning: If you change the default directory, the sample HTML files will not run, unless an absolute path to them is specified in your browser.]

Examples

Here are some examples that show how you can get IDBA to generate automatic HTML types in your HTML source files. The examples are based on the Northwind sample database that comes with MS-Access.

1. For a HTML table object, all that is required is;

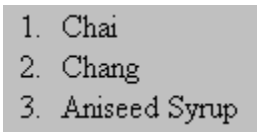
<!**Query**::**DSN**=NorthWind::
SQL=*SELECT ProductName, QuantityPerUnit, UnitPrice, UnitsInStock*
FROM Suppliers, Products WHERE Suppliers.SupplierID = Products.SupplierID
AND CompanyName = 'Exotic Liquids'::
TYPE=TABLE::
End>

Chang	24 - 12 oz bottles	19.0000	17
Aniseed Syrup	12 - 550 ml bottles	10.0000	13

2. For a HTML ordered list object, all that is required is;

<!**Query**::**DSN**=NorthWind::
SQL=*SELECT ProductName*
FROM Suppliers, Products WHERE Suppliers.SupplierID = Products.SupplierID
AND CompanyName = 'Exotic Liquids'::
TYPE=OLIST::
End>

To generate



1. Chai
2. Chang
3. Aniseed Syrup

3. For a HTML unordered list object, all that is required is;

<!**Query**::**DSN**=NorthWind::
SQL=*SELECT ProductName*
FROM Suppliers, Products WHERE Suppliers.SupplierID = Products.SupplierID
AND CompanyName = 'Exotic Liquids'::
TYPE=ULIST::
End>

To generate

- Chai
- Chang
- Aniseed Syrup

4. For a HTML select object, all that is required is;

<!**Query**::**DSN**=NorthWind::
SQL=*SELECT CompanyName FROM Suppliers*::
OK=<STRONG>Query Succeeded !!!!</STRONG>::
FAIL=Query Failed !!!!::
TYPE=SELECT(vName)::
End>

To generate

Note that select objects are best suited in html forms, otherwise the vName parameter has no meaning. To illustrate this in more detail, suppose your html form were to contain the following;

```
<FORM METHOD=POST ACTION="action.htm">
<BR>
Please Select a Supplier:
<!Query::DSN=NorthWind::
SQL=Select CompanyName from Suppliers::
TYPE=SELECT(vSupplier)::
OK=<INPUT TYPE="SUBMIT"><INPUT TYPE="RESET">::
FAIL=<BR><BR><STRONG>Query Failed !!!</STRONG>::
End>
</FORM>
```

Then, you could utilize the variable "vSupplier" in several different ways in your "action.htm" file, as follows;

```
The name you selected was VAR(vSupplier)
```

IDBA would generate: The name you selected was Exotic Suppliers

```
<!Query::DSN=NorthWind::
SQL=SELECT  ProductName, QuantityPerUnit, UnitPrice, UnitsInStock
FROM Suppliers, Products WHERE Suppliers.SupplierID = Products.SupplierID
AND CompanyName = VARQ(vSupplier)::
TYPE=TABLE::
End>
```

IDBA would generate the same table as seen in 1. Above.

5. For a spread object held in memory, all that is required is;

```
<!Query::DSN=NorthWind::
SQL=SELECT  ProductName, QuantityPerUnit, UnitPrice, UnitsInStock
FROM Suppliers, Products WHERE Suppliers.SupplierID = Products.SupplierID
AND CompanyName = 'Exotic Liquids'::
TYPE=SPREAD::
End>
<BR>
<BR>
<!DATA(0,1)> at row 1 is: <!DATA(1,1)>
```

IDBA would generate the following HTML:

ProductName at row 1 is: Chai

IDBA Configuration

IDBA comes configured with reasonable defaults that cover most scenarios at installation, to change IDBA's configuration just select the "Configure" menu option. This should result in a tab dialog appearing. The tabs that can be selected for configuring IDBA, are as follows;

Server - web server standard options.
HTML Files - where the web server should look for the default html file.
Log Files - where the server should output log data.
Status - what debug messages the server should output to its view.
DataSources - what ODBC datasource names the server will use to connect to databases.

Server

This page specifies server properties:

Listen on HTTP port

Determines the port number that this server will listen on for client requests. The default is 80.
If you have another server on the same machine or local network listening on port 80, you can change this value to a value that does not conflict with another servers port number.

Maximum clients allowed

Determines the maximum number of clients this server will service at any one time.

Check idle clients every *x* seconds

Does a check every x seconds to see if any idle clients are still connected. If they are, then the server disconnects them.

Enable Fax/Email

Check this box if you want this server to be able to process **Send** *Action-Key* statements in your HTML files.

WARNING: You must start MS-Exchange on the machine that this server is running on before checking the box and clicking “Ok”.

This feature is not currently supported under WinNT

HTML Files

This page specifies where the default HTML page is located. Every other HTML page is situated relative to the default HTML page.

To test the default location, type "**http:\\127.0.0.1**" in your browser window at the **location** prompt.

Base Path

Specifies the default path where the default html file is located.

Default HTML File

Specifies the default HTML file to load when the servers address is called.

Tag Message

Specifies the message that appears at the end of a HTML file requested through this server.

None- No tag message is output
Auto - Provides an automatic message "From the services of the Intranet Database Assistant"
File - Uses the text specified in the Tag File to output as the standard message.

Tag FileName: Specifies the filename for the above File option.

Log File

This page specifies whether to output server messages to a log file.

Enable - allows output to a log file.

Output To

Specifies which file to output server messages to.

Status

This page specifies which server status messages should be output to the servers view window.

- **Enable Status Output** - check this option for all server status output messages.
- **Enable Client Name Lookup** - check this option to view client request messages.
- **Enable Debug Output** - check this option to view server debug messages.

Data Sources

This page specifies the ODBC *datasourcenames* that this server will use for identifying SQL requests from HTML pages that it is servicing.

ODBC Sources

Specifies the ODBC *datasourcenames* that have been configured for the machine that this server is running on. To modify any of these names, use the ODBC32 application found in "Control Panel".

IDBA Sources

Specifies the ODBC *datasourcenames* that this server will use for identifying SQL requests from HTML pages that it is servicing. To add or remove datasourcenames, use the "Add" and "Remove" buttons.

NOTE:

1. Removing a datasourcename from IDBA sources does not become effective until the server is restarted.
2. IDBA sources should have at least one entry.