



# TMS INTRAWEB QUERY BUILDER DEVELOPERS QUICK START GUIDE

## Contents

---

Supported Delphi, C++Builder and IntraWeb versions .....	3
Installation .....	4
Uninstall any previous versions, trials or separate components .....	4
Unzip the IntraWeb Query Builder file into a new folder .....	4
Setting library paths .....	4
Install packages .....	4
Installing the component pack in multiple development environments .....	6
Installing additional database interfaces .....	6
Updates .....	7
Support .....	8
Website login .....	9
License agreement .....	10
TMS IntraWeb Query Builder single developer license .....	10
TMS IntraWeb Query Builder site license .....	10
Termination of license .....	12
Getting started with IntraWeb Query Builder .....	13
Using IntraWeb Query Builder to modify predefined SQL command queries .....	14
Using IntraWeb Query Builder with the MetaSQL-based setup .....	15
Setup of TTIWQueryBuilder user interface .....	16
IntraWeb Query Builder with databases without included interface component .....	17

## Supported Delphi, C++Builder and IntraWeb versions

---

The TMS IntraWeb Query Builder supports following development environments:

Delphi 2009 (Professional, Enterprise, Architect)  
Delphi 2010 (Professional, Enterprise, Architect)  
Delphi XE (Professional, Enterprise, Architect)  
Delphi XE2 (Professional, Enterprise, Architect)  
Delphi XE3 (Professional, Enterprise, Architect)  
Delphi XE4 (Professional, Enterprise, Architect)  
Delphi XE5 (Professional, Enterprise, Architect)  
Delphi XE6 (Professional, Enterprise, Architect)

C++Builder 2009 (Professional, Enterprise, Architect)  
C++Builder 2010 (Professional, Enterprise, Architect)  
C++Builder XE (Professional, Enterprise, Architect)  
C++Builder XE2 (Professional, Enterprise, Architect)  
C++Builder XE3 (Professional, Enterprise, Architect)  
C++Builder XE4 (Professional, Enterprise, Architect)  
C++Builder XE5 (Professional, Enterprise, Architect)  
C++Builder XE6 (Professional, Enterprise, Architect)

The TMS IntraWeb Query Builder can be simultaneously installed and used on multiple development environments on a single machine.

TMS IntraWeb Query Builder requires that IntraWeb 10 or higher is installed in Delphi or C++Builder. IntraWeb can be obtained from <http://www.atozedsoftware.com>  
The trial version requires that the default version included with Delphi 2009, 2010, XE, XE2, XE3, XE4, XE5, XE6 is installed.

Supports SQL syntax for Microsoft Access, Microsoft SQL Server, Oracle, MySQL, Nexus, Interbase, IBO, SQLDirect, DBISAM and Local BDE

## Installation

---

### [Uninstall any previous versions, trials or separate components](#)

Make sure to first uninstall any possible previous version of the IntraWeb Query Builder. A correct uninstall means:

- Remove the package through Component, Install packages ..., Remove
- Delete ALL component DCU, OBJ, HPP, PAS, BPL, BPI files
- Remove the path from the library path via Tools, Environment options, Library, Library path

### [Unzip the IntraWeb Query Builder file into a new folder](#)

*For the evaluation version:*

*iwqs.zip:*

- Delphi 2009,2010,XE,XE2,XE3,XE4,XE5,XE6
- C++Builder 2009,2010,XE,XE2,XE3,XE4,XE5,XE6 files

*For the registered version:*

*Iwqsreg.zip:*

- Delphi 2009,2010,XE,XE2,XE3,XE4,XE5,XE6
- C++Builder 2009,2010,XE,XE2,XE3,XE4,XE5,XE6 files

### [Setting library paths](#)

Under Tools, Environment Options, Library, Library path, add following directories to the library path:

```
{IWQB}\Source  
{IWQB}\Common
```

where {IWQB} is the directory where the distribution has been unzipped.

### [Install packages](#)

In Delphi or C++Builder, select from the menu File, Open and browse for the appropriate package file:

```
amsql2009.dpk : Delphi 2009 & C++Builder 2009 (start BDS 2009 with all personalities to install)  
amsql2010.dpk : Delphi 2010 & C++Builder 2010 (start CDS 2010 with all personalities to install)  
amsql2011.dpk : Delphi XE & C++Builder XE (start DS XE with all personalities to install)  
amsqlXE2.dpk : Delphi XE2 & C++Builder XE2 (start DS XE2 with all personalities to install)  
amsqlXE3.dpk : Delphi XE3 & C++Builder XE3 (start DS XE3 with all personalities to install)  
amsqlXE4.dpk : Delphi XE4 & C++Builder XE4 (start DS XE4 with all personalities to install)  
amsqlXE5.dpk : Delphi XE5 & C++Builder XE5 (start DS XE5 with all personalities to install)  
amsqlXE6.dpk : Delphi XE6 & C++Builder XE6 (start DS XE6 with all personalities to install)
```

After opening the package file, choose Compile and after compilation, select Install. The MetaSQL support components will appear on the component palette.

In Delphi or C++Builder, select from the menu File, Open and browse for the appropriate package file:

#### For IntraWeb 10.x

lw10qb2009.dpk : Delphi 2009  
lw10qb2010.dpk : Delphi 2010

#### For IntraWeb 11.x

lw11qb2009.dpk : Delphi 2009  
lw11qb2010.dpk : Delphi 2010  
lw11qb2011.dpk : Delphi XE

#### For IntraWeb 12.x

lw11qb2009.dpk : Delphi 2009  
lw11qb2010.dpk : Delphi 2010  
lw11qb2011.dpk : Delphi XE  
lw12qbXE2.dpk : Delphi XE2  
lw12qbXE3.dpk : Delphi XE3  
lw12qbXE4.dpk : Delphi XE4

#### For IntraWeb 14.0.0

lw14qbXE3.dpk : Delphi XE3  
lw14qbXE4.dpk : Delphi XE4  
lw14qbXE5.dpk : Delphi XE5  
lw14qbXE6.dpk : Delphi XE6

#### For IntraWeb 14.0.x

lw14qb2009.dpk : Delphi 2009  
lw14qb2010.dpk : Delphi 2010  
lw14qb2011.dpk : Delphi XE  
lw14qbXE2.dpk : Delphi XE2  
lw14qbXE3upd.dpk : Delphi XE3  
lw14qbXE4upd.dpk : Delphi XE4  
lw14qbXE5upd.dpk : Delphi XE5

After opening the package file, choose Compile and after compilation, select Install. The MetaSQL support components will appear on the component palette.

### Installing the component pack in multiple development environments

The recommended procedure to install the component pack in different Delphi or C++Builder environments on a single machine is to unzip the IntraWeb Query Builder into a different folder for each Delphi or C++Builder version and to use as such a different library path for each Delphi or C++Builder version.

### Installing additional database interfaces

Packages for installing additional database interfaces are included in the file WQDBSUP.ZIP

## Updates

---

The IntraWeb Query Builder registered version comes with a full version cycle of free updates. A full version cycle means that from version x.y to version x+1.y, the updates are free. For example, if a registration starts at version v2.5, updates are free till version v3.5. The latest date of the updates can be seen after login on our website (see next section). This date reflects the exact file date of the latest update.

## Support

---

IntraWeb Query Builder registered users have priority support on a special email address that is sent in a registration confirmation email. Other questions can be sent to [help@tmssoftware.com](mailto:help@tmssoftware.com). Registered users have also access to newsgroups & support forums. The required information can be found after login on our website with your registration email & code.



## Website login

---

Registered users of the TMS IntraWeb Query Builder receive a code with which it is possible to login on the website to obtain:

- Free updates for a full version cycle of the components.
- Discount offers on other products
- Access to additional documentation
- Access to samples projects
- Change registration email
- Information for peer to peer support newsgroup access

Login on the website is done with:

- Email with which you registered the TMS IntraWeb Query Builder
- Code that is sent by email with the first registered version

Direct link to login:

<http://www.tmssoftware.com/site/login.asp>

### Important notes

Keep this email and code in a safe place.

If for some reason, your registration email address changes, the only way to have this updated is by sending email to TMS software to request for a change. The email **must** be sent from the original email address and specify the new address to change to. Email address changes are manually checked. If an email change is requested for a single developer license, changes to email addresses belonging to other persons will not be allowed.

Make sure to use an email account that can handle file attachments up to 1.5MB, that there is no filtering on ZIP attachments, contents of ZIP attachments and that there is no spam filter or other filters that would block the receipt of emails with updates.

## License agreement

---

The IntraWeb Query Builder is available with two licensing schemes, a single developer license and a site license. The details of the two license types are below:

### TMS IntraWeb Query Builder single developer license

The single developer license of the component gives you the right to:

- Using the component for development of applications or any type of software module in general by a single developer within the company holding the license.
- Sell any commercial compiled application with the control, published by the company holding the license
- Make modifications to the source code of component for own use.
- Use the component and source code on all development systems used by the developer assigned by the company holding the license.
- Request future versions of the component at any time either through the web or by email for a full version cycle of the component. After the free update version cycle has expired, we cannot be held responsible to keep backup versions of old versions.
- Access to priority email support by the single developer assigned by the company holding the license.
- Sell any number of applications in any quantity without any additional run-time fees or royalties required.

The license agreement prevents you from:

- Distributing parts or full source code of any component from TMS software.
- Using parts or full source code of components from the TMS software for creating any type of other components that are distributed or sold with or without source code.
- Changing the source code of any component from TMS software and sell or distribute this as a modified product.
- Creating a descendant compiled product such as OCX or ActiveX control and sell or distribute this as a product.
- Using the control in applications sold with different publisher name than the company holding the license.
- Transfer the license to any other developer than the original registered developer
- Using the components by multiple developers in the company holding the license

### TMS IntraWeb Query Builder site license

The site license of the component gives you the right to:

- Using the component for development of applications or any type of software module in general by a single developer within the company holding the license.
- Sell any commercial compiled application with the control, published by the company holding the license
- Make modifications to the source code of component for own use.

- Use the component and source code on all development systems used by the developer assigned by the company holding the license.
- Request future versions of the component at any time either through the web or by email for a full version cycle of the component. After the free update version cycle has expired, we cannot be held responsible to keep backup versions of old versions.
- Access to priority email support by the single developer assigned by the company holding the license.
- Sell any number of applications in any quantity without any additional run-time fees or royalties required.
- Change at any time the number of developers using the TMS software components within the company holding the license.
- Notify TMS software at any time to allow new developers within the company to access the priority email support.
- Allow any number of developers within the company holding the license to access the web based interface for obtaining product updates.

The site license agreement prevents you from:

- Distributing parts or full source code of any component from TMS software.
- Using parts or full source code of components from the TMS software for creating any type of other components that are distributed or sold with or without source code.
- Changing the source code of any component from TMS software and sell or distribute this as a modified product.
- Creating a descendant compiled product such as OCX or ActiveX control and sell or distribute this as a product.
- Using the control in applications sold with different publisher name than the company holding the license.
- Transfer the license to any other developer not working for the company holding the license.

## Termination of license

---

The license agreement terminates immediately after violation of any of the terms and conditions described. A termination of the license means that the company has no longer any rights to use the components for development, sell applications using the components, obtain free updates of the components and is no longer entitled to email support or any other form of support.

The company or developer holding the license is responsible for respecting the terms and conditions of the license agreement and shall thus make sure that no other person has access to the TMS Components to use these for any purposes that violate the license agreement.

## Getting started with IntraWeb Query Builder

---

Drop a MetaSQL database component and TTIWQueryBuilder on the form for the database that you want to use. For example, to use BDE, the TatBDEDataBase component is provided. Connect the TatBDEDataBase (or other database component) to the TTIWQueryBuilder.Database property.

Put your database query component on the form and connect it to a datasource. Connect the datasource to the TTIWQueryBuilder.DataSource property. Setup a simple query such as:

```
SELECT * FROM mytable
```

Set the TTIWSQLBuilder.SQLSyntax property according to the database you're working with.

With default settings, you are ready to start using the TTIWQueryBuilder. Through the browser, you can select all fields from the query, set conditions and sort order. To retrieve the SQL statement generated by the IntraWeb Query Builder, use the TTIWSQLBuilder.SQLCommand property. To have TTIWSQLBuilder update your dataset, call TTIWSQLBuilder.UpdateDataSet;

## Using IntraWeb Query Builder to modify predefined SQL command queries

---

In the getting started example, the query to start from was as simple as:

```
SELECT * FROM mytable
```

The predefined query determines what the user can change / set at run time with the Query Builder. With the above SQL statement, the user can for example select all fields, add conditions on all fields and change the sort order for all fields in the table “mytable”. In many cases, the fields that the user has access to for creating queries is more limited (user might not have access to all fields, restricted fields, key fields etc...) or the fields come from multiple tables with a join setup by the programmer that the user is not aware of. Handling these cases is as simple as presetting the query with the selectable fields and predefined joins.

Example:

```
Select c.custno, c.company from customer c, orders o  
where c.custno = o.custno order by c.company
```

In this example, the user will only be able to select CustNo and Company fields and it will appear as if the fields come from a single table but effectively coming from the Customer and Orders table. When using the IntraWeb Query Builder to setup a query, this will ‘extend’ the preset query with the additional criteria set by the user. Configuring TTIWQueryBuilder to use such preset query can be done by setting the SQL command in the attached query dataset and calling

```
TTIWQueryBuilder. UpdateMetaSQLFromDataset;
```

## Using IntraWeb Query Builder with the MetaSQL-based setup

---

Instead of using a SQL command to predefine what the user can see or do in the IntraWeb Query Builder, this can also be done visually at design-time with the MetaSQL editor. Selecting the MetaSQL property shows the design time editor which has more extensive and advanced control over the settings of the IntraWeb Query Builder. With the MetaSQL editor, tables can be selected, table aliases can be set, fields can be added including defining field aliases, field expressions, joins can be defined as well as group, conditions and sort sorted. After configuring the dataset, the user will be able to manipulate it with the IntraWeb Query Builder. The configuration of MetaSQL can be saved and restored with:

```
TTIWQueryBuilder.SaveToFile(settingsfilename);
```

and

```
TTIWQueryBuilder.LoadFromFile(settingsfilename);
```

Note that an important advantage of a MetaSQL setup is that it is database independent. You can configure MetaSQL with a given database, save the settings and change the database afterwards. By setting the correct `TTIWQueryBuilder.SQLSyntax` for your database, MetaSQL will always generate the correct SQL according to the used database.

## Setup of TTIWQueryBuilder user interface

---

By default, the IntraWeb Query Builder allows to select fields, conditions and sort order for a dataset. Properties control whether these three capabilities are available in the user interface:

ConditionSelectorVisible: Boolean;  
FieldSelectorVisible: Boolean;  
SortOrderSelectorVisible: Boolean;

The various color settings of the TTIWQueryBuilder interface are available under the Display property.

Together with basic field selection, it is also possible to select field functions. Whether this capability is available to the user or not is set with the property FieldFunctionsVisible: Boolean;



## IntraWeb Query Builder with databases without included interface component

---

You can use TatCustomDatabase component to manually retrieve tables and field names and pass the names to visual query component. Just drop a TatCustomDatabase component and set event handlers for all its events. The example below shows how to use TatCustomDatabase to support BDE components (although it is already supported by the specific TatBDEDatabase component). You must have a TDatabase component in the form named Database1 which points to the DBDEMOS database.

```

procedure TForm1.atCustomDatabase1RetrieveFieldNameListEvent (
  const ATableName: String; const AList: TStrings);
var
  table: TTable;
  c: integer;
begin
  table:=TTable.Create(nil);
  try
    table.DatabaseName := 'DBDEMOS';
    table.TableName := ATableName;
    table.FieldDefs.Update;
    for c := 0 to table.FieldDefs.Count-1 do
      AList.AddObject(
        table.FieldDefs[c].Name, // FieldName
        TObject(Ord(table.FieldDefs[c].DataType)) ); // DataType
    finally
      table.Free;
    end;
  end;

procedure TForm1.atCustomDatabase1RetrieveTablenameListEvent (
  const AList: TStrings);
begin
  Database1.GetTableNames(AList);
end;

procedure TForm1.atCustomDatabase1OpenQueryEvent(ASql: String;
  var ADataset: TDataSet);
begin
  ADataset := TQuery.Create(Self);
  with TQuery(ADataset) do
    try
      DatabaseName := 'DBDEMOS';
      Sql.Text := ASql;
      Open;
    except
      ADataset.Free;
      ADataset := nil;
      raise;
    end;
  end;

```

```
procedure TForm1.atCustomDatabase1WriteSqlPropertyEvent (ADataset: TDataSet;  
    ASql: String);  
begin  
    if ADataset is TQuery then  
        TQuery(ADataset).SQL.Text := ASQL;  
end;
```