Developing Web Applications for Microsoft SQL Server Databases - What you need to know

Presented by Frances Peake of Proctor & Peake, Inc.

ATEC2008 Conference Session Description

Alpha Five’s web components simplify working with SQL databases, but what do you really need to know to get your web application up and running? From getting started with your connection string and building a grid component, to specific challenges such as data validation and working with complex relationships, this session will prepare you for the task. You will learn about Alpha Five tools and tricks for generating SQL statements to use in your web applications. This session is for Alpha Five users who are new to working with SQL databases, for SQL users who are new to Alpha Five, and for the curious!

Table of Contents

Setup ............................................................................................................................... 1
What to know about Alpha Five and Microsoft SQL Server.......................................... 2
Preparing for your project............................................................................................... 3
  Know your tools.......................................................................................................... 3
  Know your Database Administrator ........................................................................... 3
  Know your database.................................................................................................... 3
Creating a connection in the Alpha Five Database Explorer.......................................... 9
Creating your web application...................................................................................... 15
  Create your first grid component .............................................................................. 15
  Create a grid component based on a SQL query....................................................... 20
  Create an updatable grid ........................................................................................... 28
  Create a dialog component to insert a record ............................................................ 35
  Create a report based on a SQL query ...................................................................... 39
  Create a dialog to print the report ............................................................................. 45
SQL Terminology ......................................................................................................... 51
Resources ......................................................................................................................52

Setup

The exercises and examples in this document use Alpha Five Version 9, Microsoft SQL Server 2005 and Microsoft SQL Server Management Studio.

The sample Alpha Five database ATEC2008_SQLWebApps.adb is provided with this presentation. The Microsoft SQL Server sample database AdventureWorksLT can be downloaded free. For instructions on how to obtain it and a free edition of Microsoft SQL
Server, please refer to the document “ATEC2008_SQLServerSampleDatabaseSetup” which is also provided with this presentation.

**What to know about Alpha Five and Microsoft SQL Server**

There are a number of reasons to develop an Alpha Five web application with a SQL Server database back end.

- The data is already in a Microsoft SQL Server database.
  
  Perhaps you are developing an application for an existing database. The Alpha Five application may be the only way that users will access the data, or your application may be providing access to some part of a larger database.

- The requirements demand that Microsoft SQL Server be used.

  Microsoft SQL Server is the standard enterprise database for many big organizations and high-demand applications. It has a reputation for reliability, security and scalability. Microsoft SQL Server can handle large numbers of records and large numbers of users.

- You want to take advantage of the power and functionality that Microsoft SQL Server provides.

If you are an Alpha Five user who has been working exclusively with Alpha Five native tables (DBF files) then the line between the *database* and the *application*, or the *back end* and the *front end*, may not be very clear. That is because Alpha Five can be used to create both.

With Alpha Five you can create databases and manage data. You can also create forms, reports, menus and dialog boxes to run on the desktop. You can create web components, pages and navigation to run in a web browser.

If you are creating an Alpha Five web application for a Microsoft SQL Server database, you will need to think about that line between the database and application. In simple terms, the *database* is the container where the data resides. The *application* is the program that allows users to interact with the data. In reality, Microsoft SQL Server does more than just hold data. It does the job of managing the relationships among tables, enforcing data integrity and optimizing access by large numbers of users to a large amount of data.
Preparing for your project

Know your tools

If you are new to developing web applications in Alpha Five, take the time to understand what Alpha Five is and what it is capable of. Alpha Five web applications run on the Alpha Five Application Server. Know the requirements for hosting your application.

If you are an Alpha Five user who is new to Microsoft SQL Server, take the time to understand what Microsoft SQL Server is and what it is capable of. Know the requirements to set up and maintain the database.

Know your Database Administrator

Who is responsible for creating and maintaining the SQL Server database? It is critical that you establish a relationship with the Database Administrator before you start development. You will need their support throughout the project and afterward. With Microsoft SQL Server you cannot just “set it and forget it.”

Make sure that the Database Administrator knows what you are planning to do and keep them informed. Find out what requirements, concerns and limitations they have. Share yours with them. Agree on where your responsibility ends and theirs begins.

Know your database

Get to know the database for which you are building the web application. What are the tables, their fields and their relationships? What are the primary keys? What are the constraints?

These simple details are very important. Knowing your database will help you avoid frustration during the development process and prevent unfriendly SQL error messages from being displayed to your users.

EXERCISE 1

Explore the database in the Microsoft SQL Server Management Studio.

1. Open the Microsoft SQL Server Management Studio and connect to the server where you attached the AdventureWorks database.

2. In the Object Explorer, expand the AdventureWorks database to see its tables.
3. Expand the Product table to see its columns (fields). You can quickly see the following information about them:

- The column names, such as ProductID, Name, ListPrice, Size.
- Primary keys (PK) and foreign keys (FK). The ProductID is the primary key (PK) and that ProductCategoryID and ProductModelID are foreign keys.
- The column type, such as int (numeric - integer), nvarchar (character) and datetime. Microsoft SQL Server has more data types than Alpha Five does.
- Whether nulls are allowed in the column. Microsoft SQL Server distinguishes between null (no value assigned) and empty. Alpha Five does not distinguish between null and empty, or null and zero for numeric fields.
4. Use View Dependencies to learn how the Product table is related to other tables in the database.
5. For another view of the Product table’s relationships, open Keys and right-click a foreign key (FK). Choose modify to open the Foreign Key Relationships window.
6. The easiest way to explore Unique Constraints (AK) is by opening Indexes and right-clicking on the index to choose Properties.

7. To explore Check Constraints (CK), right-click on the constraint and choose Modify.

   The list price must be a value greater than or equal to zero.
8. Alternatively, you can script the constraint to the Query Editor Window. This is for study only. Do not execute the query.
9. To explore Default Constraints (DK), right-click the constraint and choose Script Constraint As > CREATE To > New Query Editor Window.

The default value for the ModifiedDate column is today’s date.

![Diagram of Microsoft SQL Server Management Studio](image)

Right-click the default constraint, Choose Script Constraint As > CREATE To > New Query Editor Window.

10. You can explore other database objects such as Triggers, Functions and Stored Procedures the same way, by scripting them to the Query Editor Window.

In this exercise, you:
- Used the Microsoft SQL Server Management Studio to explore a database

**Creating a connection in the Alpha Five Database Explorer**

The Database Explorer is an excellent place to get started in Alpha Five. With the Database Explorer you can view the tables and query the data.

To use the Database Explorer you must first connect to your Microsoft SQL Server database. Alpha Five will help you build the connection string. The information required to connect to a particular database, such as the server, username and password, will vary. If you are connecting to an existing database, then the SQL Database Administrator will be able to provide you with the connection information.

By creating and saving a Named Connection String, you can make your SQL connection available throughout Alpha Five. For example, the named connection string can be used to provide data to web components and reports.

Using Named Connection Strings makes your application flexible and portable. If you need to change the connection – perhaps to a different copy of the data or to change a password - you need only change it in one place. As long as you have used the Named Connection String throughout your application instead of the string itself, Alpha Five will be able to find the database.
EXERCISE 2

In this exercise you will create a named connection string “AdvWorks” that will connect to the AdventureWorks sample database. This will allow you to explore the database in Alpha Five’s Database Explorer and use it in your web application.

1. Open Alpha Five. Open the database ATEC2008_SQLWebApps.adb.

2. From the Control Panel choose Tools > External Databases > Database Explorer.

3. Add a new database using a named connection string.
4. Click the Edit saved connections link. Create a new connection named “AdvWorks”.

In Database Explorer, click the “Add new database to the Explorer” button. In the Connection String window, choose “Use a named connection string”.

Click Edit saved connections. The AlphaDAO Connections window will open. Click the new button to open the New AlphaDAO Connection window. Type in the Connection Name “AdvWorks”.

Connection Name: AdvWorks
Connection String:
5. Click the Build button to open the Create SQL Connection String window. Make
the following selections.

Connection Type: SQLServer
Server: (the server to which you attached AdventureWorksLT)
Database: AdventureWorksLT

6. Click Test Connection. If the connection succeeded then click OK. Close the
windows until you return to the Connection String window. Choose the
AdvWorks named connection string that you just created.
Click OK. The AdventureWorks database will now appear in the Database Explorer. Note the similarity with the Microsoft SQL Server Management Studio’s Object Explorer.

Expand the Product table to see its columns (fields). You can quickly see the following information about them:

- The column names, such as ProductID, Name, ListPrice, Size.
- Primary keys (PK). The ProductID is the primary key.
- The column type. Alpha Five has different data types, such as N (numeric) and T (date/time), than Microsoft SQL Server has. Alpha Five equivalent field types are shown.
- “Not null” indicates that nulls are not allowed in the column.
7. To preview the data from the Database Explorer, right-click on the SalesLT.Product table and choose Show Data in Table.
The Database Explorer does not provide all the details that the Microsoft SQL Server Management Studio does, but it is a convenient tool for the Alpha Five web application developer.

TIP: A hidden gem of the Database Explorer is the SQL Query Genie, which you can open by right-clicking on the table and choose Open Query Builder. You will use the SQL Query Genie in the Grid Builder in an exercise later in this session.

In this exercise, you:
- Connected to a Microsoft SQL Server database
- Created a Named Connection String
- Used the Alpha Five Database Explorer to explore a database

Creating your web application

With Alpha Five you can create web applications based on Alpha Five (dbf) databases and non-Alpha Five “SQL” data sources such Access, Oracle, MySQL and Microsoft SQL Server. The benefits of developing SQL-based web applications with Alpha Five come from Alpha’s ease-of-use features:

- Web Component Builders
- Query Builder Genie

In the following exercises you will learn how to create web components and queries based on Microsoft SQL Server data. Not all features of the Web Projects Control Panel will be covered; rather, the exercises will focus on tools and skills that you will need when developing web applications for Microsoft SQL Server databases.

Create your first grid component

Grid components connect directly to a database. They can be used to search, sort, display, add and update data.

EXERCISE 3

You will create a non-updatable grid to display records from the Customer table.

1. Open Alpha Five. Open the database ATEC2008_SQLWebApps.adb.

2. From the Control Panel’s toolbar, choose Web Projects.

4. Click Next in the New File window. Then choose the Grid component type.
5. Click OK to open the Grid Builder. Leave the Component Type as is and choose Grid > Data Source.

6. Choose Grid > Query (AlphaDAO). Make the following selections.

   Connection Type: Used Named Connection String
   Connection Name: AdvWorks

   You created the Named Connection String “AdvWorks” in EXERCISE 1.
7. Click the Connect button. Select the Product table.

8. Choose Grid > Fields. Select the following fields.

- CustomerID
- FirstName
- LastName
- CompanyName
- EmailAddress
- Phone
9. Choose File > Save. Save the grid with the name “grd_Cust”. The grid will appear in the Preview window. For a live preview in your browser, click the Browser button on the Web Projects toolbar. The grid component will open in your browser.

![Image of grid component](image_url)

10. Select the EmailAddress column. Change its Field Properties as follows.

   - Control type: Link
   - Link address type: Field value is an email address
   - Column heading: E-mail

   Save the grid component to preview the changes.
11. Close the Grid Builder window to return to the Web Projects Control Panel.

In this exercise, you:
- Created a grid component based on a single table.

**Create a grid component based on a SQL query**

**EXERCISE 4**

In this exercise you will create a grid component to display the customer’s company name and the address of the customer’s main office.

The grid component will be based on a SQL Select Statement. You will use the SQL Builder to join three tables: Customer, Customer Address and Address. Then you will use the SQL Query Genie to complete the query by filtering on the “Main Office” address type and sorting on company name.

1. Begin by creating a new Grid Component and connecting to the Named Connection String “AdvWorks”. To do this, repeat steps 1 through 6 from the previous exercise.

2. This time, instead of basing the grid on a table, choose “SQL Select Statement” and then click the Edit SQL button.
3. In the SQL Builder window, click the Tables in Query button. Select the Customer table.

4. Click OK. The Define Join window will open. In the Define Join window, select the Customer Address table.
5. Click OK to add the table. Click the Add table button again to add the Address table.
6. Click OK to add the table and return to the Define Join window. You will need to fix the link between the Customer Address table and Address table. The link field should be AddressID.

7. Click the Apply Edits button. Then click OK to return to the SQL Builder window.

8. Click the Fields in Query button and select the CompanyName field from the Customer table. You will add more fields in the SQL Query Builder Genie.
9. Click OK to return to the SQL Builder window. Select the SQL Text tab.

10. On the SQL Text tab, click the SQL Query Builder link to open the SQL Genie.
11. In the SQL Genie click the Columns tab. You will create a calculated field that combines the city, state and zip code.

The expression is:

```
alltrim(SalesLT_Address.City) + ", " + alltrim(SalesLT_Address.StateProvince) + 
" " + SalesLT_Address.PostalCode
```
Notice that you can write the expression using Alpha Five’s syntax and built-in functions. The SQL Genie will convert it Portable SQL, which will in turn be automatically converted to Microsoft SQL Server syntax.

12. Click OK to insert the expression. This will create a new column. Give the column the Alias “cCityStZip”. The “c” prefix will remind you that this is a calculated field.

13. Create the filter to select records where Address Type is “Main Office”.

![Diagram of SQL Genie interface showing the steps to create a calculated field and filter records]

For the Value/Expression, select “Main Office.”
14. Click the Order tab. Set the order for the query to “CompanyName”. Then click the Execute Query button to preview.

15. Click OK to close the SQL Genie, then click OK to close the SQL Builder.
16. Choose File > Save to save the grid component with the name grd_CustAddress. Close the grid component to return to the Web Components Control Panel.

In this exercise, you:
- Learned to use the SQL Builder and the SQL Query Genie
- Built a query based on more than one table
- Created a calculated field

Create an updatable grid

Alpha Five grid components make it easy to add and edit records in a Microsoft SQL Server database. Knowing your database – its tables, fields, primary keys, foreign keys and constraints – is the most important ingredient to making updatable grids successful. You must validate data before it is submitted so that users will not be exposed to unfriendly SQL error messages like the one shown below.
EXERCISE 5

In this exercise, you will add Validation Rules to a pre-written grid component. By exploring the Product table in the Microsoft SQL Server Management Studio, you will have learned that some columns (fields) are populated by the database, such as the ProductID and ModifiedDate. You will have also learned which columns do not allow nulls, which have uniqueness constraints and which have check constraints.

The grid component’s Detail View will allow users to update the following columns. Many of the columns require validation.

Name – required, must be unique
ProductNumber – required, must be unique
StandardCost – required, must be >= 0.00
ListPrice - required, must be >= 0.00
Size
ProductCategoryID
SellStartDate - required

1. From the Web Projects Control Panel, choose Web Components. Select the component “grd_ProdUpdate” and click the Edit button on the toolbar.

2. To see what the grd_ProdUpdate component does, click the Browser button on the toolbar. If prompted, start the Application Server. The grid component will open in your web browser.
Click the New Product link. Enter the new product information as follows.

Product Number: AA-1
Product Name: Test product
Cost: 10
List Price: 15
Size: S
Category: Accessories
Sell Start Date: (leave blank – will fill with today’s date)
3. Click the Add button, then click the Close link. You will return to the grid. The new product record will appear.

4. Now, return to the Grid Builder. The grd_ProdUpdate grid contains an updatable Detail View. This is where you will go to add Validation Rules.
5. Choose Detail View > Fields. Select the ListPrice field. Click the Validation Rules button.

6. Add Validation Rules for the ListPrice as follows:

   - Enable rules: (checked)
   - Require value: (checked)
   - Custom error message: You must enter a List Price.
   - Validation expression type: Simple expression
   - Must be: Greater than or equal
   - Value: 0
   - Custom error message: List Price must be greater than or equal to zero.
7. Click OK to return to the Field Properties. Save the component by choosing File > Save.

8. Click the Browser button on the toolbar to open the component in your web browser again. This time, click the New Product link and leave the Detail View blank.

Click the Add button to see the custom error messages. Note: The error message template can be customized to hide the column name.

9. Return to the grd_ProdUpdate grid in the Grid Builder. The grid contains pre-written Validation Rules for the other columns that require them.

The ProductNumber column has a uniqueness constraint in the Microsoft SQL Server database. A Cross-file Validation rule is used to prevent the user from submitting a duplicate Product Number.

Select the Product Number field. Open the Validation Rules window.
10. Click OK to close the Field Validation window.

11. The SellStartDate column is required. The grd_ProdUpdate component takes a different approach to prevent this column from being left blank.

Before a new record is submitted, it checks to see if the user has filled in a date. If they have not, then today’s date will be entered automatically.

IF DataSubmitted.SellStartDate = "" THEN
    DataSubmitted.SellStartDate = date()
END IF

Note that although the SellStartDate field is date/time type, the DataSubmitted.SellStartDate variable is character type.
TIP: The Insert button lists the DataSubmitted variables, which you can insert into the code.

TIP: To show line numbers, right click in the Code Editor and choose “Line Numbers”.


In this exercise, you:
- Added Validation Rules to an updatable grid component
- Saw how grid validations that can prevent unfriendly SQL errors

Create a dialog component to insert a record

Grid components connect directly to a database. Dialog components do not. Dialog components have some features that grid components do not. For example, in a dialog component you can easily initialize the value of a control. You can see results of calculations before submitting the form.

You can use dialog components to insert and update records in a Microsoft SQL Server database. Doing so, however, requires writing the INSERT or UPDATE statement.

Alpha Five’s Action Scripting genie will generate the code for you. Action Scripting is intended for desktop applications, but the code can be modified for use in web applications.

EXERCISE 6

In this exercise you will see how code that was generated by Action Scripting can be modified and used in a dialog component. The exercise uses a pre-written script and a pre-written dialog component.
1. On the Alpha Five Control Panel’s Code tab, select the script named “scpProdInsert”. Right-click on the script and choose Design.

2. In the Code Editor, click the Edit Action button to see what each action does. On the “Execute INSERT Query on SQL database” action, click the button for “Set column values”.

3. The column values will be set from the variables’ values. The SellStartDate will be set to today’s date.
4. Close the Script Genie and return to the Code Editor. Choose Code > Convert Action Script to Xbasic. A message will warn you that the conversion cannot be undone. Answer “Yes” to continue.

5. A message will warn you that the conversion to Xbasic cannot be undone. Answer “Yes” to continue. Choose File > Save As to save the converted Xbasic with the name “scpProdInsert_Xbasic”.
6. Open the Web Projects Control Panel. Edit the pre-written dialog component, “dlg_ProdInsert”. Note that the control names match the names of the variables in the Action Script that you just converted.

7. In the Dialog Builder, choose Form > Properties. Scroll down to the After Validate event.

8. The converted code requires a few modifications to work in a web application: Message boxes are commented out; all dialog component controls are saved to the “character” type.
9. Close the Form Events window and the Dialog Builder. You can try the “dlg_ProdInsert” component by previewing it to the browser. There are no validations when you submit the data so remember to fill all fields and enter a unique Product Name and Number each time.

You can preview the grd_ProdInsert component to search for the products you inserted.

In this exercise, you:
- Saw Action Scripting generate a SQL INSERT statement
- Converted an Action Script to Xbasic
- Learned how the converted code can be used in a dialog component

**Create a report based on a SQL query**

With Alpha Five you can create reports that are based on data from a Microsoft SQL Server database.

**EXERCISE 7**

In this exercise you will create a report based on AdventureWorks’ Product table. You will add an argument that will act as a filter variable.

1. Open the ATEC2008_SQLWebApps.adb database.

2. Go to Reports tab and click the New button to create a new report.

![Image of New Report window]

3. Click OK. Choose “Create ad hoc DataSource” then choose the Named Connection String “AdvWorks”.

![Image of creating a report]
4. Click Next. Choose “Custom SQL Select statement or stored procedure”. Then click the “Query Builder” button.
5. Click the “Add Table” button. Then select the ProductCategory table.

6. Click Next to continue to the Add Table window. Choose ProductCategoryID as the linking field on both sides of the join.

7. Click Finish to return to the SQL Genie. Choose the Columns tab. Select the following columns:

   From the Category table:
   ProductNumber
Name
ListPrice
ProductCategoryID

From the ProductCategory table:
Name – Give this column the alias “Category”

8. Choose the Properties tab. Click “Define Arguments” button.

9. In the Define/Edit Arguments window, click the “Add” button.

Argument name: ProdCatID
Type: Numeric
10. Click OK to return to the Define/Arguments window. Click OK again to return to the SQL Genie.

11. Choose the Filter tab. On the Filter tab, select the ProductCategoryID column, the “is equal to” operator and the ProdCatID argument. The genie will put the colon in front of the argument.

12. Click OK to return to the SQL Genie. Choose the Order tab and select the ProductNumber column.
13. Click OK to return to the Specify DataSource window. Click Finish.

14. You will be prompted for a value for the :ProdCatID argument and for the number of records. Enter “6” and “0” respectively. When prompted, use the Report Genie.

15. Click OK. The Quick Report Genie will open. Type in the title “Product List” and select the ProductNumber, Name and Listprice fields.
16. Click OK. In the Report Editor, choose File > Save. Save the report with the name “rptProdList”. Close the Report Editor.

In this exercise, you:
- Created a report that is based on a Microsoft SQL Server database
- Added an argument to filter the report

Create a dialog to print the report

You can use a dialog component to prompt the user for filter criteria and print the report. Alpha Five has a built-in print-to-PDF feature.

EXERCISE 8

In this exercise will create a dialog component that lets the user choose the product category from a dropdown box and prints the report to PDF.

1. Go to Web Projects Control Panel and create a new dialog component.

2. In the Dialog Builder, choose Form > Controls. Select the Dropdownbox control type. In the New Control window, enter the following.

   Name: vProdCatID
   Label: Category
   Type: Numeric
3. Click OK to return to the Dialog Builder. Click the button to edit DropDownBox choices.

4. Define the DropDownBox properties as follows.

   List type: Dynamic
   Data source type: Alpha DAO
   Use saved connection string: (checked)
   AlphaDAO Connection String: AdvWorks
5. Click the button to enter the AlphaDAO SQL SELECT statement.

   Table: ProductCategory
   Field to display: Name
   Field to store: ProductCategoryID

6. Click OK to return to the Dialog Builder. Choose Form > Properties. Scroll down to the After Validate event and click the button to edit.
7. The Form Events editor will open. Click the “Insert” button. Select Genies > Print Report. Press ENTER.

8. In the Print Report Genie window, select the report “rptProdList”. Then click the Add button to add a filter variable as follows.

   Variable Name: vProdCatID
   Variable Type: Numeric
9. Click OK to return to the Print Report Genie. Check the “Does the SQL statement use arguments?” checkbox. Click the “Add” button to add the following arguments.

Argument Name: ProdCatID
Argument Type: Numeric

The name and type of the argument must match the argument in the report’s data source.
Click OK to return to the Print Report Genie. Click OK again to return to the Form Events editor. Click OK a third time to return to the Dialog Builder.

10. Save the dialog component with the name “dlg_ProdList”. Then click the Browser button for a live preview of the dialog in your browser.

In this exercise, you:
- Created a dialog component
- Added a dropdown box with choices from a Microsoft SQL Server database
- Used the Print Report Genie to generate the code to filter and print a report
SQL Terminology

Constraint

Constraints are the built-in mechanism for enforcing data integrity. Using constraints is preferred to using triggers, rules, and defaults because built-in integrity features use much less overhead and perform faster than the ones you can create.

Source: http://www.mssqlcity.com/Articles/General/using_constraints.htm

Closest equivalents in Alpha Five: Field Rules, Field Validations

View

In SQL Server a view represents a virtual table. Just like a real table, a view consists of rows with columns, and you can retrieve data from a view (sometimes even update data in a view). The fields in the view’s virtual table are the fields of one or more real tables in the database.

You need to have a goal in mind when creating a view. There are a number of scenarios where you will want to look for a view as a solution.

- To hide the complexity of the underlying database schema, or customize the data and schema for a set of users.
- To control access to rows and columns of data.
- To aggregate data for performance.

Source: http://www.odetocode.com/articles/299.aspx

Closest equivalents in Alpha Five: Set, View

Function

User Defined Functions are compact pieces of Transact SQL code, which can accept parameters, and return either a value, or a table. They are saved as individual work units, and are created using standard SQL commands. Data transformation and reference value retrieval are common uses for functions.

Source: http://www.databasejournal.com/features/mssql/article.php/3348181

Closest equivalent in Alpha Five: Function

Stored Procedure

A stored procedure is an already written SQL statement that is saved in the database. If you find yourself using the same query over and over again, it would make sense to put it
into a stored procedure. When you put this SQL statement in a stored procedure, you can then run the stored procedure from the database's command environment using the exec command.


Closest equivalents in Alpha Five: Operation, Script

**Trigger**

A trigger is a database object that is attached to a table. In many aspects it is similar to a stored procedure. As a matter of fact, triggers are often referred to as a "special kind of stored procedure." The main difference between a trigger and a stored procedure is that the former is attached to a table and is only fired when an INSERT, UPDATE or DELETE occurs. You specify the modification action(s) that fire the trigger when it is created.


Closest equivalent in Alpha Five: Posting Field Rule

**Resources**

Video courses with instructor Mark Long: Microsoft SQL Server 2005 Database Administration and Microsoft SQL Server Database Programming from [www.VTC.com](http://www.VTC.com)

Microsoft SQL Server Books Online from [www.microsoft.com](http://www.microsoft.com)

Alpha Five Web Help, Video Library and Message Board at [www.alphasoftware.com](http://www.alphasoftware.com)

Book: Building Web Applications from [www.intellectualbusinesssolutions.com](http://www.intellectualbusinesssolutions.com)

Book: Web Applications Made Easy from [www.libertymanuals.com](http://www.libertymanuals.com)

**Questions? Comments?**

Contact Frances Peake at Proctor & Peake, Inc.
(305) 751-1181 [frances@proctorandpeake.com](mailto:frances@proctorandpeake.com)