



InterBase

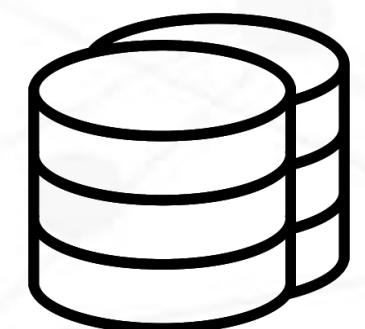
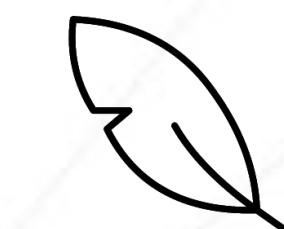
Getting Started...





What is InterBase?

InterBase® is a **full-featured, high performance, ultrafast**, encryptable, scalable, relational and embeddable multi-platform SQL database with commercial-grade **data security, disaster recovery** and **change synchronization** for developers who are looking to embed a **low-cost, zero-admin, secure** database into their cross-platform connected applications. Global companies across a range of industries have a competitive advantage with InterBase® through advanced **data protection, security, speed and performance**.

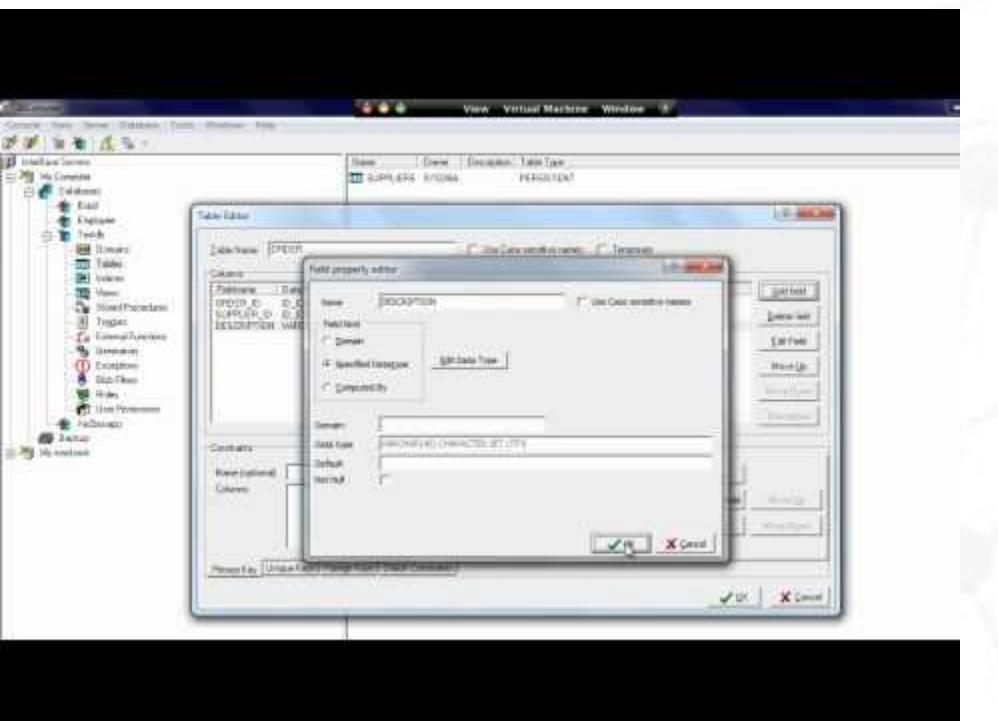
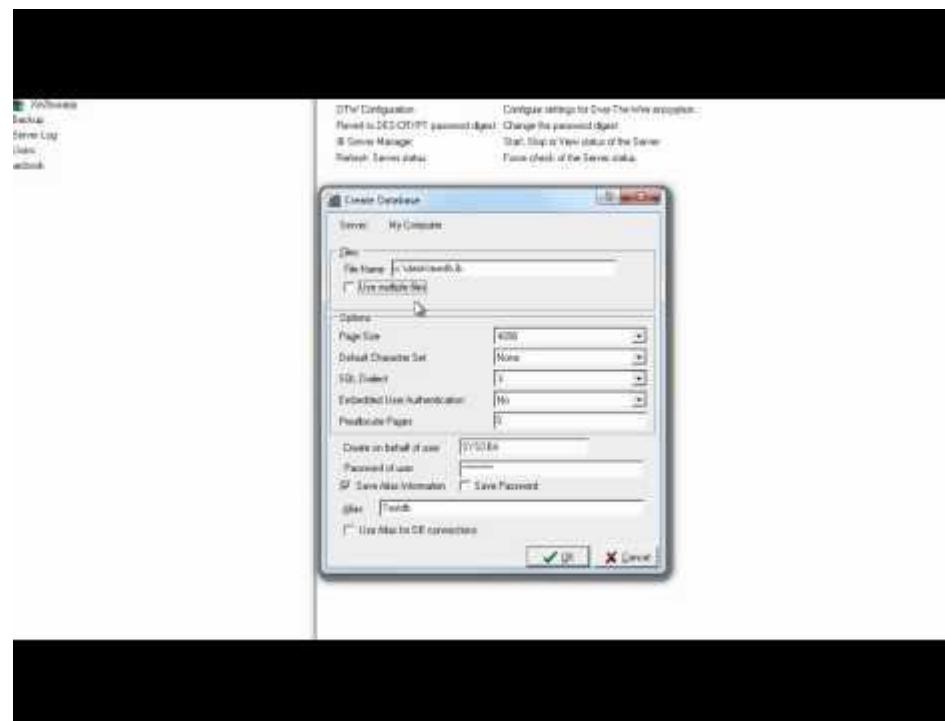


| Secure | Self-reliant | Resilient | Lightweight | Flexible | Efficient | Compliant |
|---|---|--|--|--|--|--|
| Cross-Platform Encryption Separate Security Login Reduced Exposure to Risk Role-Based User Security Encrypted Backups | Independent Operation Near-Zero Maintenance Rapid Crash Recovery Disaster Recovery | Live Backups Distinguished Data Dumps Fast Restores Write-Ahead Logging Point-in-Time Recovery | Small Footprint Fast Install Faster Data | Multi-Language Support Cross-Platform Support Server Data Storage Local Data Storage Single On-Disk Format | Log-Less Replication Data Change Subscriptions ChangeViews Data Tracking Role-Based User Security Early Fetch Data Deltas | ANSI SQL Stored Procedures and Triggers SQL-Based Performance Monitoring Open-Standards-Based Database Connectivity Multi-Gen Architecture/MVCC Event Alerts Customizable Generators |





Creating a Database & Tables with InterBase



Creating a New Database with InterBase

In this tutorial, learn how to create a new blank database using IBConsole. Learn about the different options you can potentially choose and why the default options are typically the best to use.

Creating Tables Smartly with InterBase Domains

In this tutorial (following on from the previous tutorial where we created a blank database) we learn how to add tables and introduce the idea of InterBase Domains to help manage field types consistently across the database, speeding up database administration and assisting table wide consistency in your data model.

Introducing Temporary Tables

Global Temporary Tables are tables in the database that support temporary data storage for the life of the connection, or the transaction (as defined by the user). These are great for short term data storage, as the example in the video shows.





Triggers, Generators, and Stored Procedures

Example Trigger

```
SET TERM ^;

CREATE TRIGGER "SUPPLIERS_AUTOID" FOR "SUPPLIERS"
ACTIVE BEFORE INSERT POSITION 0
AS
BEGIN
  IF (NEW.id IS NULL) THEN
    NEW.id = GEN_ID(gen_NextID, 1);
END
^
SET TERM ;^
```

Example Generator with Trigger.

```
CREATE GENERATOR gen_NextID;

SET GENERATOR gen_NextID TO 1;

CREATE TRIGGER suppliers_autoid FOR suppliers
BEFORE INSERT AS
BEGIN
  IF (NEW.id IS NULL) THEN
    NEW.id = GEN_ID(gen_NextID, 1);
END;
```

The screenshot shows the InterBase SQL IDE interface. A window titled 'Stored Procedure' is open, displaying the following SQL code:

```
CREATE PROCEDURE SP_TEST
  BEGIN
    SELECT department_id, department_name
    FROM departments;
  END;
```

The interface includes a tree view on the left showing database objects like 'EMPLOYEE', 'DEPARTMENT', and 'EMPLOYEE'. The status bar at the bottom indicates '1 row(s) listed'.

Catching Data Changes - Introduction to Triggers

Triggers help you centralize business logic into the database by notifying you of changes to data either BEFORE or AFTER it happens, ensuring you can run checks and changes in a consistent manner (no matter who/or what program is connected to the database). In this session, we introduce Triggers, something that we will combine over the coming videos with other InterBase features to enable extended business logic in action.

Creating Unique Values Using Generators

Discover Generators with an example that will combine with a trigger to create unique values for a record as it's inserted into the database. Generators are unique Integer values that are specific to each database. You can use Generators from negative Integer to positive Integer to get unique values for use in, for example, primary keys. Using Generators is a positive thing as they are managed by the database to ensure their uniqueness. You can have multiple Generators on any database

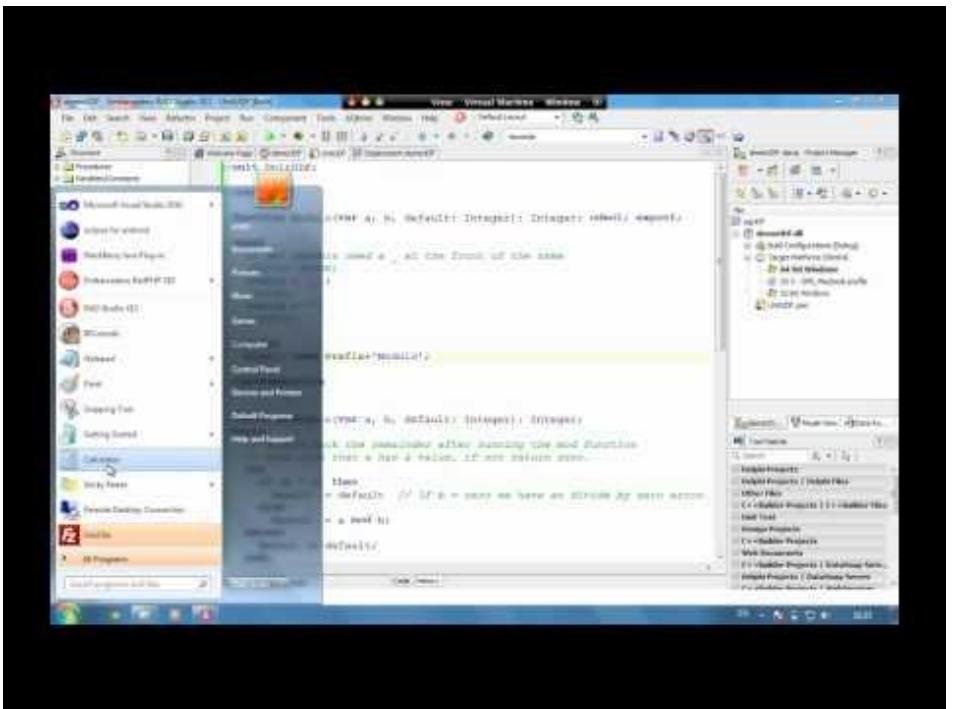
Introduction to Stored Procedures

InterBase allows you to centralize business logic into the server for both updating and selecting data. A core element that makes this possible is the use of Stored Procedures. Stored procedures also help reduce network traffic and can also open up additional options with user security. In this tutorial we will introduce the concept of stored procedures and explore some examples from the example Employee database.





User Defined Functions and InterBase Exception



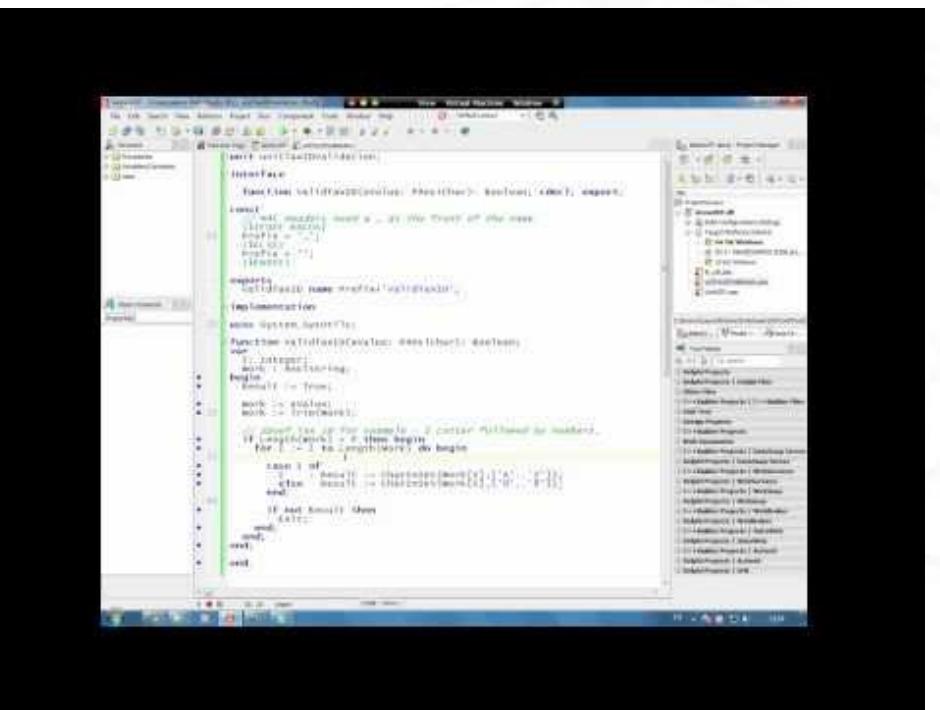
Expanding InterBase SQL

- An Introduction to User Defined

One of the great things about InterBase is that if you can't do something in SQL, you can always add in a User Defined Function (UDF) to help you make it possible! In this tutorial, we combine database programming with some Delphi development to expand the SQL syntax of InterBase using UDFs. Simply put, UDFs are libraries that contain quick, simple functions that calculate a result on data passed in.

Introducing InterBase Exceptions and validating your data!

InterBase has the ability to raise customized exceptions when something happens that you want to prevent. In this tutorial, we will learn how we can implement advanced business logic into a database to ensure that only valid data is entered. The example we follow ensures that data entered in a Tax ID field passed advanced check sum checking, and prevents the data being added if it fails the checks. Although you don't need to use UDF, this simple example is a natural follow on to the Triggers and UDF tutorials.





Registering InterBase for mobile



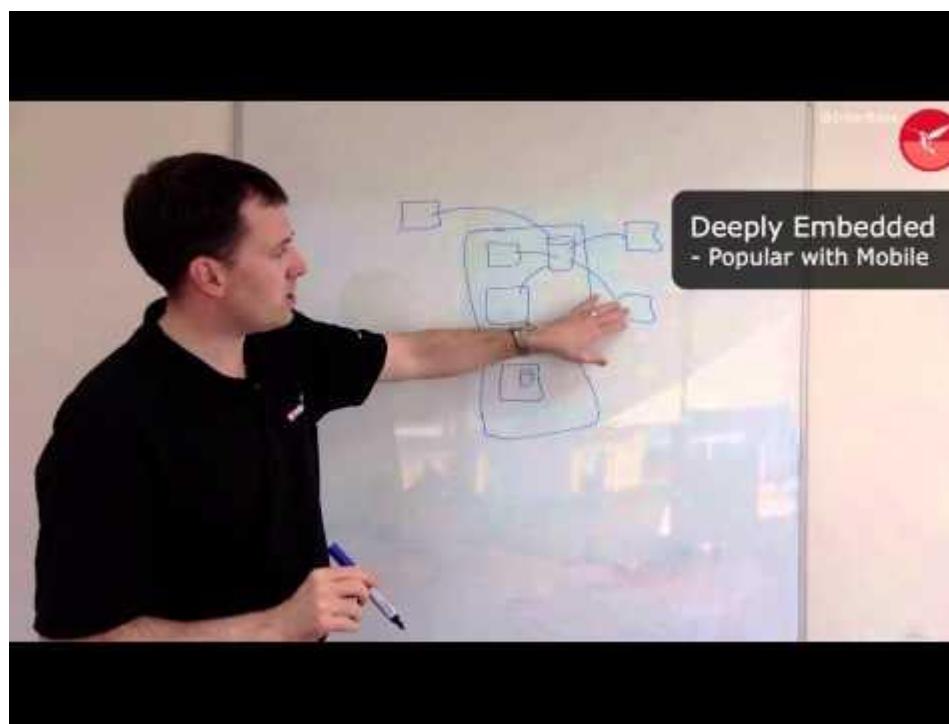
Registering InterBase for Mobile

Learn the steps to deploy your first mobile app using the InterBase ToGo or IBLite embeddable databases in apps built with RAD Studio, Delphi or C++Builder.





InterBase as an Embedded Database?



What does Embedded v Deeply Embedded mean?

InterBase can be deployed as a traditional SQL database server or as an embedded / deeply embedded server. But what does that mean and how does this relate to one another?

This video explores and explains these terms when discussing how InterBase can be deployed:

- With multiple clients connecting to the database server and to the databases on that server,
- Silently with your product
- Inside your product so only your product can use InterBase.

