

FIT

Database Module Documentation

© 2006 by Alcea Technologies Inc.

Version History

Version	Date	Description	Changed by	Approved by
1.0	15/01/2004	Document Creation	Chris Justus	Peter Worthing
1.1	18/02/2004	Added MS SQL 2000	Peter Worthing	Chris Justus
1.2	01/06/2004	Added Oracle	Peter Worthing	Chris Justus
1.3	05/07/2004	Added special consideration headings and MS SQL notes.	Peter Worthing	Chris Justus
1.4	21/07/2004	Added generic MySQL info	Peter Worthing	Chris Justus
1.5	12/08/2004	Added info for upgrading JRE Added fbt_project_def table to schemas	Peter Worthing	Chris Justus
1.6	20/10/2004	Added FBT_CHART_DEF and FBT_REPORT_DEF for Version 5.0	Peter Worthing	Chris Justus
1.7	21/12/2005	Added schema changes for Version 6.0 Fixed DB URL typo in 3.5.2	Peter Worthing	Chris Justus
1.8	29/01/2006	Changed BIGINT type to Float for decimal types	Peter Worthing	Chris Justus
2.0	18/09/2006	Release of FIT 7.0	Peter Worthing	Chris Justus
2.1	09/03/2007	Addition of TS_ATTACHMENT table	Peter Worthing	Jon Simpson
2.2	20/07/2007	Added MS SQL 2005 Instructions - renamed from Enterprise to Database Module	Peter Worthing	Jon Simpson
2.3	24/08/2007	New installer notes	Peter Worthing	Jon Simpson
2.4	15/07/2008	Addition of Dashboard Tables And changes to Oracle JDBC Driver	Peter Worthing	Jon Simpson
2.5	07/07/2009	Release of FIT 7.10 Schema Changes	Peter Worthing	Jon Simpson

Overview	5
1.1 Database Module	5
1.2 Satisfaction Guaranteed!	5
2 Database Configuration	6
2.1 Table Creation.....	6
2.2 Table Indexes	7
2.3 Driver Configuration.....	7
2.4 Request a Valid Database Key.....	7
2.5 Connect to Database	7
2.6 Upgrade your Java Runtime Environment.....	8
2.7 Migrate to Database System	9
3 MySQL Installation	11
3.1 Special Considerations.....	11
3.2 Creating Tables	11
3.3 Configuring JDBC Driver.....	17
3.4 Request a Valid Registration Key.....	18
3.5 Connecting to database	18
3.6 Upgrade your Java Runtime Environment.....	19
3.7 Migrating System.....	19
4 MS SQL 2000 Server Installation.....	20
4.1 Special Considerations.....	20
4.2 Creating Tables	21
4.3 Configuring JDBC Driver.....	28
4.4 Request a Valid Registration Key.....	28
4.5 Connecting to database	28
4.6 Upgrade your Java Runtime Environment.....	29
4.7 Migrating System.....	29
5 MS SQL 2005 Server Installation.....	30
5.1 Special Considerations.....	30
5.2 Creating Tables	31
5.3 Configuring JDBC Driver.....	38
5.4 Request a Valid Registration Key.....	38
5.5 Connecting to database	38
5.6 Upgrade your Java Runtime Environment.....	39
5.7 Migrating System.....	39
6 Oracle Installation.....	40
6.1 Special Considerations.....	40
6.2 Creating Tables	40
6.3 Configuring Driver.....	46
6.4 Request a Valid Registration Key.....	47
6.5 Connecting to database	47
6.6 Upgrade your Java Runtime Environment.....	48
6.7 Migrating System.....	48
7 Frequently Asked Questions	49
7.1 Invalid Database URL.....	49

7.2	ClassNotFoundException	49
7.3	Invalid Database Driver	49
7.4	ClassNotFoundException	50
7.5	java.sql.SQLException: [Microsoft][SQLServer 2000 Driver for JDBC]Can't start a cloned connection while in manual transaction mode.	50
7.6	java.sql.SQLExceptionUUUUU: [Microsoft][SQLServer 2000 Driver for JDBC]The year, 5, is outside the range allowed by the SQL Server.....	51
7.7	No suitable Driver.....	51
7.8	Cannot Start a Cloned Connection While in Manual Transaction Mode	51

Overview

1.1 Database Module

FIT was created as a stand alone solution for quick and painless installations on customer sites. The base model incorporates a built-in web server and flat file storage system which allows it to be installed anywhere in just minutes.

As the base model of FIT has matured, it has gained widespread acceptance as a robust and flexible tool. However, we have acknowledged that a flat file system is not always the desired method for data storage. In addition, there are a number of third party tools available with database extensions for graphing and reporting.

We have developed an Database Module that allows database storage on any JDBC compliant database.

1.2 Satisfaction Guaranteed!



We are committed to a long-term relationship with all of our customers, including continuous support and regular upgrades to our product. The FIT Database Module is backed by the same 100% money back guarantee as the rest of our products. If you are not completely satisfied with our product after 30 days, we will refund your money.

2 Database Configuration

This document assumes that a database has been installed and is ready to be configured. If you are not comfortable with database operations, you will need to consult your database administrator. You will need the following:

- database installation
- JDBC Driver
- DB URL, DB Driver Name, DB username, DB password

Immediately following, we list the general steps for configuring a database for FIT and migrating your data. In addition, we have dedicated chapters with specific instructions for configuring with MySQL, SQL Server, and Oracle.

2.1 Table Creation

Create a database schema with the appropriate tables to store system data. The tables needed are as follows:

```
TABLE FBT_ATTACHMENT – stores individual attachment details.
TABLE FBT_ATTACHMENT_VERSION – stores attachment version info.
TABLE FBT_ATTACHMENT_FOLDER – stores attachment folder info.
TABLE FBT_BUG_STRUCT – stores individual issue details.
TABLE FBT_BUG_STRUCT_ENTRY – stores historical entries to all issues.
TABLE FBT_BUG_USER_FIELD – stores current details for custom fields.
TABLE FBT_BUG_USER_FIELD_ENTRY – stores historical entries for custom fields.
TABLE FBT_CFG_DEF – stores settings for system configurations.
TABLE FBT_CHART_DEF – stores saved charts.
TABLE FBT_COLORCODE_DEF – stores the definitions for main menu color coding.
TABLE FBT_DASHBOARD – stores dashboard settings
TABLE FBT_DASHBOARD_COMPONENT – stores dashboard component settings
TABLE FBT_EVENT_DEF – stores event definitions.
TABLE FBT_FLD_DEF – stores custom field definitions.
TABLE FBT_FLT_DEF – stores filter definitions.
TABLE FBT_GROUP_DEF – stores group definitions.
TABLE FBT_HS_DEF – stores hierarchy definitions.
TABLE FBT_MAILRULE_DEF - stores mail rule definitions.
TABLE FBT_PROJECT_DEF – stores details for advanced projects.
TABLE FBT_REPORT_DEF – stores saved reports.
TABLE FBT_RET_DEF - stores return message definitions.
TABLE FBT_USER_DEF – stores user profile definitions.
TABLE FBT_WF_DEF – stores workflow definitions.
TABLE FIT_OBJECT – stores generic fit objects such as those for FTS module
TABLE TS_ATTACHMENT – stores attachments for TestSpec (TS).
TABLE FBT_ATTACHMENT_VERSION – stores TS attachment version info.
TABLE FBT_ATTACHMENT_FOLDER – stores TS attachment folder info.
```

See following detailed sections for more detailed schema explanations on specific databases.

2.2 Table Indexes

Add the following indexes to your schema to speed up access times:

```
CREATE INDEX FBSE_FSID ON FBT_BUG_STRUCT_ENTRY (FBT_STRUCT_ID);
CREATE INDEX FBUF_FSID ON FBT_BUG_USER_FIELD (FBT_STRUCT_ID);
CREATE INDEX FBUFE_FSID ON FBT_BUG_USER_FIELD_ENTRY (FBT_STRUCT_ID);
CREATE INDEX FBUFE_FSEID ON FBT_BUG_USER_FIELD_ENTRY
(FBT_STRUCT_ENTRY_ID);
```

2.3 Driver Configuration

Configure JDBC driver for FIT to connect to database. Connections are made to the database using the JDBC protocol. In order to make this connection a proper driver must be installed. You will need to find the driver specific to your system and database if it is not already installed. Doing search for “JDBC driver” on the home site for the given database usually finds drivers quite easily. In each case, there is often more than one type of driver. See your Database section for at least one example.

2.4 Request a Valid Database Key

Make sure that you have received a valid registration key with the Database Module enabled. If you are running it correctly, you will see a green check mark in the modules tab of the Admin->Track Administration menu. You will also then be able to proceed to the Database Administration menu. If either of these conditions is not valid, you must obtain the package from us and restart the system using this valid reg.bin file instead.

2.5 Connect to Database

Make a connection from FIT to your database using the “Admin->Database Administration” menu.

DB Driver:	<input type="text"/>	(ex: com.microsoft.jdbc.sqlserver.SQLServerDriver)
DB URL:	<input type="text"/>	(ex: jdbc:sqlserver://localhost:1433;DatabaseName=fbt)
DB User:	<input type="text"/>	

DB Password:

Fill in the 4 connection fields and try the “Save and Test” button. If you are successful, you will see an OK message, similar to the following:

```
OK:
Product Name: MySQL
Product Version: 4.1.12a-nt-max
Driver: 3.1
DriverName: MySQL-AB JDBC Driver
DriverVersion: mysql-connector-java-3.1.12 ( $Date: 2005-11-17
15:53:48 +0100 (Thu, 17 Nov 2005) $, $Revision$ )
```

If you see a NullPointerException, you were not able to make a good connection and will have to sort out the problem. A full description of the Exception is available in the console or the “Admin->Exceptions” menu.

If you see an exception such as “java.lang.ClassNotFoundException”, then you should read the next section.

2.6 Upgrade your Java Runtime Environment

If you see an exception such as “**java.lang.ClassNotFoundException**”, you may need to upgrade your Java Runtime Environment (JRE). You are encouraged to download the latest JRE from Sun. You can check what version you are running in the Admin->Advanced->System Info menu. On the System Properties tab, you will find a line similar to this:

```
java.version=1.5.0_06
```

FIT has a configuration file in the installation directory which tells it which JRE to use (in case there are multiple instances on your system). Your FIT download may have included a “jre” directory, which was downloaded with the product, or it may reference a common JRE directory elsewhere on your file system.

Your installation will include one of two files, depending on when you installed the product.

If you have a FIT.lax configuration file, the JRE is configured on a line similar to one of the following:

```
lax.nl.current.vm=jre\\bin\\jre.exe      (uses the JRE installed with FIT)
lax.nl.current.vm=java.exe             (uses the current system JRE)
lax.nl.current.vm=c:\\Program Files\\Java\\bin\\java (direct JRE reference)
```

If you do not have a FIT.lax file, you will want to look for the FIT.ini and FITService.ini configuration files (always modify both), where the JRE is configured on a line similar to:

JRE Path=c:/Program Files/Fit/jre/bin/java.exe

After you modify the respective line, simply restart FIT and check the System Info, to make sure things are running correctly.

2.7 Migrate to Database System

The other buttons on the Database Administration page are for migrating a system from file storage to DB storage (and the reverse). Even if you have a fresh system, you will need to use the migration tool to make sure that the configuration settings are properly copied to the database. You will need to use the three operation buttons in sequence and check the “Admin->Exceptions” log to make sure there were no driver problems.

For a migration you will use :

Copy Files to DB

Enable

Delete System File Data (**optional)

** if you wish to free up the original flat files used

A successful migration will show you some informative message in the Exceptions log file, similar to the following (indicating how many entries were made to the database):

```
migrate attachments (.att) --> 2
migrate issues (.bug) --> 19
migrate archived attachments / issues (.old) --> 0
migrate config files (.cfg) --> 18
migrate charts (.cs) --> 0
migrate color codes (.ccs) --> 1
migrate dashboard components (.dc) -> 8
migrate dashboards (.dash) -> 1
migrate event files (.es) --> 2
migrate custom fields (.fld) --> 4
migrate filters (.flt) --> 8
migrate groups (.grp) --> 3
migrate hierarchies (.hs) --> 2
migrate projects (.prj) --> 2
migrate rules (.rul) --> 2
migrate reports (.rs) --> 0
migrate return messages (.ret) --> 0
migrate users (.use) --> 22
migrate workflow (.wf) --> 2
```

If you see any exceptions, your driver is probably incompatible and you need to contact us right away.

3 MySQL Installation

3.1 Special Considerations

By default, MySQL is configured to store a maximum of 2M for a field. Therefore, to compensate for larger attachments (FBT allows up to 6M), a special packet parameter must be used when starting up the database.

Here is the command we use to start things up.

```
mysqld-max-nt --max_allowed_packet=16M --standalone --console
```

- max_allowed_packet allows it to handle queries containing up to 16M of data...

3.2 Creating Tables

The following set of commands will create a valid FIT schema on a MySQL database. Make sure that you are in the correct database before executing the commands.

-- Schema in MySQL syntax

-- also available here :

http://fittrackingsolutions.com/misc/FIT_MYSQL.SQL

```
-- main table for holding attachment info (ie: A1.att)
```

```
CREATE TABLE FBT_ATTACHMENT (  
    CONTEXT_ID INT,  
    FBT_ATTACHMENT_ID BIGINT,  
    FBT_STRUCT_ID BIGINT,  
    FBT_CREATED_BY VARCHAR(255),  
    FBT_ATTACH_FILENAME VARCHAR(255),  
    FBT_ORIGINAL_FILENAME VARCHAR(255),  
    FBT_CONTENT_TYPE VARCHAR(255),  
    FBT_ATTACHMENT_DATE DATETIME,  
    FBT_COMMENTS VARCHAR(255),  
    FBT_VERSION VARCHAR(255),  
    FBT_FOLDER_ID BIGINT,  
    FBT_ATTACHMENT_VERSION_ID BIGINT,  
    FBT_CONTENT LONGBLOB,  
    FBT_ARCHIVED INT,  
    PRIMARY KEY (CONTEXT_ID,FBT_ATTACHMENT_ID)  
);
```

```
-- table for holding attachment version info (ie: A1.att)
```

```
CREATE TABLE FBT_ATTACHMENT_VERSION (  
    CONTEXT_ID INT,  
    FBT_ATTACHMENT_ID BIGINT,  
    FBT_STRUCT_ID BIGINT,
```

```
FBT_CREATED_BY VARCHAR(255),
FBT_ATTACH_FILENAME VARCHAR(255),
FBT_ORIGINAL_FILENAME VARCHAR(255),
FBT_CONTENT_TYPE VARCHAR(255),
FBT_ATTACHMENT_DATE DATETIME,
FBT_COMMENTS VARCHAR(255),
FBT_VERSION VARCHAR(255),
FBT_FOLDER_ID BIGINT,
FBT_ATTACHMENT_VERSION_ID BIGINT,
FBT_CONTENT LONGBLOB,
FBT_ARCHIVED INT,
PRIMARY KEY (CONTEXT_ID,FBT_ATTACHMENT_VERSION_ID)
);

-- table for holding attachment folder info (ie: f1.fol)
CREATE TABLE FBT_ATTACHMENT_FOLDER (
    CONTEXT_ID INT,
    FBT_FOLDER_ID BIGINT,
    FBT_STRUCT_ID BIGINT,
    FBT_GLOBAL INT,
    FBT_CREATED_BY VARCHAR(255),
    FBT_FOLDER_DATE DATETIME,
    FBT_FOLDER_NAME VARCHAR(255),
    FBT_PARENT_FOLDER_ID BIGINT,
    FBT_ARCHIVED INT,
    PRIMARY KEY (CONTEXT_ID,FBT_FOLDER_ID)
);

-- main table for bug details (ie: b1.bug field details)
CREATE TABLE FBT_BUG_STRUCT (
    CONTEXT_ID INT,
    FBT_STRUCT_ID BIGINT,
    FBT_SUBJECT VARCHAR(255),
    FBT_ASSIGNEDTO VARCHAR(255),
    FBT_STATUS VARCHAR(255),
    FBT_PRIORITY INT,
    FBT_PROJECT VARCHAR(255),
    FBT_AREA VARCHAR(255),
    FBT_ENVIRONMENT VARCHAR(255),
    FBT_VERSION VARCHAR(255),
    FBT_ENTEREDBY VARCHAR(255),
    FBT_NOTIFYPYLIST TEXT,
    FBT_PARENT_ID BIGINT,
    FBT_REQUESTED_DUE_DATE DATETIME,
    FBT_ACTUAL_COMPLETION_DATE DATETIME,
    FBT_ESTIMATED_HOURS FLOAT,
    FBT_ACTUAL_HOURS FLOAT,
    FBT_PERCENT_COMPLETE FLOAT,
    FBT_ARCHIVED INT,
    FBT_DIRTY INT,
    PRIMARY KEY (CONTEXT_ID,FBT_STRUCT_ID)
);

-- table for bug history entries (ie: b1.bug history entries)
CREATE TABLE FBT_BUG_STRUCT_ENTRY (
    CONTEXT_ID INT,
```

```
FBT_STRUCT_ENTRY_ID BIGINT,  
FBT_STRUCT_ID BIGINT,  
FBT_WHO VARCHAR(255),  
FBT_DATE_WHEN DATETIME,  
FBT_STATUS VARCHAR(255),  
FBT_ASSIGNEDTO VARCHAR(255),  
FBT_DESCRIPTION TEXT,  
FBT_SUBJECT VARCHAR(255),  
FBT_PRIORITY INT,  
FBT_PROJECT VARCHAR(255),  
FBT_AREA VARCHAR(255),  
FBT_ENVIRONMENT VARCHAR(255),  
FBT_VERSION VARCHAR(255),  
FBT_NOTIFYLIST VARCHAR(255),  
FBT_PARENT_ID BIGINT,  
FBT_REQUESTED_DUE_DATE DATETIME,  
FBT_ACTUAL_COMPLETION_DATE DATETIME,  
FBT_ESTIMATED_HOURS FLOAT,  
FBT_ACTUAL_HOURS FLOAT,  
FBT_PERCENT_COMPLETE FLOAT,  
FBT_ATTACHMENTS VARCHAR(255),  
FBT_NOTES VARCHAR(255),  
PRIMARY KEY (CONTEXT_ID,FBT_STRUCT_ENTRY_ID)  
);  
  
-- main table for holding bug custom field details (ie bl.bug custom  
details)  
CREATE TABLE FBT_BUG_USER_FIELD (  
    CONTEXT_ID INT,  
    FBT_USER_FIELD_ID BIGINT,  
    FBT_STRUCT_ID BIGINT,  
    FBT_FIELD_ID INT,  
    FBT_STRING_VALUE TEXT,  
    FBT_DATE_VALUE DATETIME,  
    FBT_NUMBER_VALUE FLOAT,  
    PRIMARY KEY (CONTEXT_ID,FBT_USER_FIELD_ID)  
);  
  
-- table for custom fields of bug history entries (ie: bl.bug custom  
history entries)  
CREATE TABLE FBT_BUG_USER_FIELD_ENTRY (  
    CONTEXT_ID INT,  
    FBT_USER_FIELD_ENTRY_ID BIGINT,  
    FBT_STRUCT_ID BIGINT,  
    FBT_STRUCT_ENTRY_ID BIGINT,  
    FBT_FIELD_ID INT,  
    FBT_STRING_VALUE TEXT,  
    FBT_DATE_VALUE DATETIME,  
    FBT_NUMBER_VALUE FLOAT,  
    PRIMARY KEY (CONTEXT_ID,FBT_USER_FIELD_ENTRY_ID)  
);  
  
-- config files (ie: NAME.cfg)  
CREATE TABLE FBT_CFG_DEF (  
    CONTEXT_ID INT,  
    FBT_CFG_ID BIGINT,  
    FBT_CFG_NAME VARCHAR(255),
```

```
    FBT_CFG_INFO TEXT,
    FBT_CFG_DUPLICATE INT,
    PRIMARY KEY (CONTEXT_ID,FBT_CFG_ID)
);

-- saved charts
CREATE TABLE FBT_CHART_DEF (
    CONTEXT_ID INT,
    FBT_CHART_ID BIGINT,
    FBT_CHART_NAME VARCHAR(255),
    FBT_CHART_INFO TEXT,
    PRIMARY KEY (CONTEXT_ID,FBT_CHART_ID)
);

-- saved color codes
CREATE TABLE FBT_COLORCODE_DEF (
    CONTEXT_ID INT,
    FBT_COLORCODE_ID BIGINT,
    FBT_COLORCODE_FIELD VARCHAR(255),
    FBT_COLORCODE_INFO TEXT,
    PRIMARY KEY (CONTEXT_ID,FBT_COLORCODE_ID)
);

-- new in Version 7.08
-- store dashboard settings
CREATE TABLE FBT_DASHBOARD (
    FBT_DASHBOARD_ID INT,
    FBT_LOGINID VARCHAR(255),
    FBT_DASHBOARD_INFO TEXT,
    PRIMARY KEY (FBT_DASHBOARD_ID)
);

-- new in Version 7.08
-- store dashboard component settings
CREATE TABLE FBT_DASHBOARD_COMPONENT (
    FBT_DASHBOARD_COMPONENT_ID INT,
    FBT_COMPONENT_INFO TEXT,
    PRIMARY KEY (FBT_DASHBOARD_COMPONENT_ID)
);

-- event definitions (ie: e1.es)
CREATE TABLE FBT_EVENT_DEF (
    CONTEXT_ID INT,
    FBT_EVENT_ID BIGINT,
    FBT_EVENT_NAME VARCHAR(255),
    FBT_EVENT_INFO TEXT,
    PRIMARY KEY (CONTEXT_ID,FBT_EVENT_ID)
);

-- defines custom field files (ie: F1.fld)
CREATE TABLE FBT_FLD_DEF (
    CONTEXT_ID INT,
    FBT_FLD_ID BIGINT,
    FBT_FLD_NAME VARCHAR(255),
    FBT_FLD_INFO TEXT,
    PRIMARY KEY (CONTEXT_ID,FBT_FLD_ID)
);
```

```
-- defines filter files (ie: F1.flt)
CREATE TABLE FBT_FLT_DEF (
    CONTEXT_ID INT,
    FBT_FLT_ID BIGINT,
    FBT_FLT_NAME VARCHAR(255),
    FBT_FLT_INFO TEXT,
    PRIMARY KEY (CONTEXT_ID,FBT_FLT_ID)
);

-- group definitions (ie: G1.grp)
CREATE TABLE FBT_GROUP_DEF (
    CONTEXT_ID INT,
    FBT_GROUP_ID BIGINT,
    FBT_GROUP_NAME VARCHAR(255),
    FBT_GROUP_INFO TEXT,
    PRIMARY KEY (CONTEXT_ID,FBT_GROUP_ID)
);

-- hierarchy definitions (ie: 1.hs)
CREATE TABLE FBT_HS_DEF (
    CONTEXT_ID INT,
    FBT_HS_CHILD_ID BIGINT,
    FBT_HS_INFO TEXT,
    PRIMARY KEY (CONTEXT_ID,FBT_HS_CHILD_ID)
);

-- mail rules (ie: mr1.rul)
CREATE TABLE FBT_MAILRULE_DEF (
    CONTEXT_ID INT,
    FBT_MAILRULE_ID BIGINT,
    FBT_MAILRULE_NAME VARCHAR(255),
    FBT_MAILRULE_INFO TEXT,
    PRIMARY KEY (CONTEXT_ID,FBT_MAILRULE_ID)
);

-- project definitions (ie: P1.prj)
CREATE TABLE FBT_PROJECT_DEF (
    CONTEXT_ID INT,
    FBT_PROJECT_ID BIGINT,
    FBT_PROJECT_NAME VARCHAR(255),
    FBT_PROJECT_INFO TEXT,
    PRIMARY KEY (CONTEXT_ID,FBT_PROJECT_ID)
);

-- saved reports
CREATE TABLE FBT_REPORT_DEF (
    CONTEXT_ID INT,
    FBT_REPORT_ID BIGINT,
    FBT_REPORT_NAME VARCHAR(255),
    FBT_REPORT_INFO TEXT,
    PRIMARY KEY (CONTEXT_ID,FBT_REPORT_ID)
);

-- mail rule return messages (ie: rml.ret)
CREATE TABLE FBT_RET_DEF (
```

```
CONTEXT_ID INT,
FBT_RET_ID BIGINT,
FBT_RET_NAME VARCHAR(255),
FBT_RET_INFO TEXT,
PRIMARY KEY (CONTEXT_ID,FBT_RET_ID)
);

-- user definitions (ie: U1.use)
CREATE TABLE FBT_USER_DEF (
CONTEXT_ID INT,
FBT_USER_ID BIGINT,
FBT_LOGINID VARCHAR(255),
FBT_USER_INFO TEXT,
PRIMARY KEY (CONTEXT_ID,FBT_USER_ID)
);

-- workflow definitions (ie: W1.wf)
CREATE TABLE FBT_WF_DEF (
CONTEXT_ID INT,
FBT_WF_ID BIGINT,
FBT_WF_NAME VARCHAR(255),
FBT_WF_INFO TEXT,
PRIMARY KEY (CONTEXT_ID,FBT_WF_ID)
);

-- table used for generic objects (ie: FTS)
CREATE TABLE FIT_OBJECT (
CONTEXT_ID INT,
SYSTEM_ID INT,
FIT_OBJECT_ID BIGINT,
CLASSNAME VARCHAR(255),
OBJECT_DATA TEXT,
FIT_ARCHIVED INT,
PRIMARY KEY (CONTEXT_ID,SYSTEM_ID,FIT_OBJECT_ID)
);

-- New in Version 7.04
-- table for holding TestSpec attachment info (ie: tsA1.tsa)
CREATE TABLE TS_ATTACHMENT (
CONTEXT_ID INT,
TS_ATTACHMENT_ID BIGINT,
TS_STRUCT_ID BIGINT,
TS_CREATED_BY VARCHAR(255),
TS_ATTACH_FILENAME VARCHAR(255),
TS_ORIGINAL_FILENAME VARCHAR(255),
TS_CONTENT_TYPE VARCHAR(255),
TS_ATTACHMENT_DATE DATETIME,
TS_COMMENTS VARCHAR(255),
TS_CONTENT LONGBLOB,
TS_ARCHIVED INT,
PRIMARY KEY (CONTEXT_ID,TS_ATTACHMENT_ID)
);

-- table for holding attachment version info (ie: A1.att)
CREATE TABLE TS_ATTACHMENT_VERSION (
CONTEXT_ID INT,
TS_ATTACHMENT_ID BIGINT,
```

```
TS_STRUCT_ID BIGINT,
TS_CREATED_BY VARCHAR(255),
TS_ATTACH_FILENAME VARCHAR(255),
TS_ORIGINAL_FILENAME VARCHAR(255),
TS_CONTENT_TYPE VARCHAR(255),
TS_ATTACHMENT_DATE DATETIME,
TS_COMMENTS VARCHAR(255),
TS_VERSION VARCHAR(255),
TS_FOLDER_ID BIGINT,
TS_ATTACHMENT_VERSION_ID BIGINT,
TS_CONTENT LONGBLOB,
TS_ARCHIVED INT,
PRIMARY KEY (CONTEXT_ID,TS_ATTACHMENT_VERSION_ID)
);

-- table for holding attachment folder info (ie: f1.fol)
CREATE TABLE TS_ATTACHMENT_FOLDER (
CONTEXT_ID INT,
TS_FOLDER_ID BIGINT,
TS_STRUCT_ID BIGINT,
TS_GLOBAL INT,
TS_CREATED_BY VARCHAR(255),
TS_FOLDER_DATE DATETIME,
TS_FOLDER_NAME VARCHAR(255),
TS_PARENT_FOLDER_ID BIGINT,
TS_ARCHIVED INT,
PRIMARY KEY (CONTEXT_ID,TS_FOLDER_ID)
);

CREATE INDEX FBSE_FSID ON FBT_BUG_STRUCT_ENTRY
(FBT_STRUCT_ID,CONTEXT_ID);
CREATE INDEX FBUF_FSID ON FBT_BUG_USER_FIELD
(FBT_STRUCT_ID,CONTEXT_ID);
CREATE INDEX FBUFE_FSID ON FBT_BUG_USER_FIELD_ENTRY
(FBT_STRUCT_ID,CONTEXT_ID);
CREATE INDEX FBUFE_FSEID ON FBT_BUG_USER_FIELD_ENTRY
(FBT_STRUCT_ENTRY_ID,CONTEXT_ID);
```

3.3 Configuring JDBC Driver

At the time of testing, the following URL was used to obtain the “MySQL Connector/J” JDBC MySQL driver.

<http://dev.mysql.com/downloads/connector/j/3.0.html>

This driver contains a jar file similar to mysql-connector-java-3.0.8-stable-bin.jar which needs to be copied into the FBT installation directory where it can be referenced.

Your installation will include one of two files, depending on when you installed the product.

If you have a FIT.lax configuration file, you need add this file to the lax classpath as follows:

```
# LAX.CLASS.PATH
# -----
# the Java classpath necessary to run this application
# Can be separated by colons (Mac OS/Unix) or semicolons (Windows)
lax.class.path=fit.jar;lax.jar;mysql-connector-java-3.1.12-bin.jar
```

If you do not have a FIT.lax file, you will want to look for the FIT.ini and FITService.ini configuration files (always modify both), where the JRE is configured on a line similar to:

```
[Class Path]
Class Path=C:\Program Files\FIT\fit.jar;C:\Program Files\FIT\mysql-connector-java-3.1.12-
bin.jar
```

Finally, you may simply have a Fit script which you can modify similar to :
Java -cp "fit.jar;mysql-connector-java-3.1.12-bin.jar" Fit

After you modify the correct location, simply restart FIT and check the System Info, to make sure the driver is listed as part of the "java.class.path", on the System Properties tab

3.4 Request a Valid Registration Key

Please see generic instructions in section 2.4.

3.5 Connecting to database

All database configuration is handled in the Database Administration menu, which is available from the Advanced section in the Admin Menu of the Enterprise Solution software. If you can not see this menu, then you are probably using a normal FIT installation and have not received a properly activated registration key.

Using the JDBC driver in the jar file from section 3.3.2 (ie: mysql-connector-java-3.1.12-bin.jar) , you would use the following information to connect to the database. (Note that the machine and port are included in the DB URL, as they are on a separate machine)

```
DB Driver:      com.mysql.jdbc.Driver
DB URL:         jdbc:mysql://optiplex:3306/fbt
DB User:        fbt
DB Password:    ***
```

After the correct values are entered, you can press the “Save and Test” button to check them. You should receive something like the following:

```
OK:
Product Name: MySQL
Product Version: 4.0.18-max-nt
Driver: 3.0
DriverName: MySQL-AB JDBC Driver
DriverVersion: mysql-connector-java-3.0.14-production ( $Date: 2004/04/24 15:49:43 $,
$Revision: 1.27.2.39 $ )
```

3.6 Upgrade your Java Runtime Environment

Please see Section 2.6 for generic instructions.

3.7 Migrating System

Please see Section 2.7 for generic instructions.

4 MS SQL 2000 Server Installation

4.1 Special Considerations

In case you are new to using this database, this is how we set things up during testing.

A : SQL 2000 Setup

Open up the Enterprise Manager

browse to Console Root -> Microsoft SQL Servers -> SQL Server Group -> [machine]
(Windows NT)

If you want a separate Database/Catalog, create one in the Databases area.

Right click on New Database and then enter the new name. I used FBT for my system.

Under the Security area, right click on Logins to create a new user (if desired)

Under the properties for that user, grant the database access for the new database.

Then choose "Database Creators" and "Bulk Insert Administrators" from the roles tab

Then select the Database that you want this user to use under the Defaults... Database section.

You then need to go back to the Database properties and select "Create Table" under the Permissions tab.

B: Schema Setup

Open up an "SQL Query Analyser" and connect to the database using the new user that you created.

Then execute the schema that is detailed in section 4.2 of the document or open this file and execute:

```
http://www.fittrackingsolutions.com/misc/FIT\_MSSQL.SQL
```

This should create the tables just fine and you can see them under the TABLES section for your Database in the Enterprise manager.

Then you can connect from FIT and execute the "Copy Files to DB" option.

C: Authentication

One other thing:

By default, I think SQL Server 2000 only accepts windows authenticated users.

To allow the SQL Server users, do the following:

Right click on [machine] and select Properties.

Under the Security tab, make sure "SQL Server and Windows" is checked for Authentication.

You will have to restart SQL 2000 if you make this change.

4.2 Creating Tables

```
-- Schema in MS SQL Server syntax
-- also available here :
http://www.fittrackingsolutions.com/misc/FIT_MSSQL.SQL
```

```
begin tran fbtsetup;
```

```
-- main table for holding attachment info (ie: A1.att)
```

```
CREATE TABLE FBT_ATTACHMENT (
    CONTEXT_ID INT,
    FBT_ATTACHMENT_ID BIGINT,
    FBT_STRUCT_ID BIGINT,
    FBT_CREATED_BY VARCHAR(255),
    FBT_ATTACH_FILENAME VARCHAR(255),
    FBT_ORIGINAL_FILENAME VARCHAR(255),
    FBT_CONTENT_TYPE VARCHAR(255),
    FBT_ATTACHMENT_DATE DATETIME,
    FBT_COMMENTS VARCHAR(255),
    FBT_VERSION VARCHAR(255),
    FBT_FOLDER_ID BIGINT,
    FBT_ATTACHMENT_VERSION_ID BIGINT,
    FBT_CONTENT IMAGE,
    FBT_ARCHIVED INT,
    PRIMARY KEY (CONTEXT_ID,FBT_ATTACHMENT_ID)
);
```

```
-- table for holding attachment version info (ie: A1.att)
```

```
CREATE TABLE FBT_ATTACHMENT_VERSION (
    CONTEXT_ID INT,
    FBT_ATTACHMENT_ID BIGINT,
    FBT_STRUCT_ID BIGINT,
    FBT_CREATED_BY VARCHAR(255),
    FBT_ATTACH_FILENAME VARCHAR(255),
    FBT_ORIGINAL_FILENAME VARCHAR(255),
    FBT_CONTENT_TYPE VARCHAR(255),
    FBT_ATTACHMENT_DATE DATETIME,
    FBT_COMMENTS VARCHAR(255),
    FBT_VERSION VARCHAR(255),
    FBT_FOLDER_ID BIGINT,
    FBT_ATTACHMENT_VERSION_ID BIGINT,
    FBT_CONTENT IMAGE,
    FBT_ARCHIVED INT,
    PRIMARY KEY (CONTEXT_ID,FBT_ATTACHMENT_VERSION_ID)
);
```

```
-- table for holding attachment folder info (ie: f1.fol)
```

```
CREATE TABLE FBT_ATTACHMENT_FOLDER (
    CONTEXT_ID INT,
    FBT_FOLDER_ID BIGINT,
    FBT_STRUCT_ID BIGINT,
    FBT_GLOBAL INT,
    FBT_CREATED_BY VARCHAR(255),
```

```
    FBT_FOLDER_DATE DATETIME,
    FBT_FOLDER_NAME VARCHAR(255),
    FBT_PARENT_FOLDER_ID BIGINT,
    FBT_ARCHIVED INT,
    PRIMARY KEY (CONTEXT_ID,FBT_FOLDER_ID)
);

-- main table for bug details (ie: bl.bug field details)
CREATE TABLE FBT_BUG_STRUCT (
    CONTEXT_ID INT,
    FBT_STRUCT_ID BIGINT,
    FBT_SUBJECT VARCHAR(255),
    FBT_ASSIGNEDTO VARCHAR(255),
    FBT_STATUS VARCHAR(255),
    FBT_PRIORITY INT,
    FBT_PROJECT VARCHAR(255),
    FBT_AREA VARCHAR(255),
    FBT_ENVIRONMENT VARCHAR(255),
    FBT_VERSION VARCHAR(255),
    FBT_ENTEREDBY VARCHAR(255),
    FBT_NOTIFYLIST TEXT,
    FBT_PARENT_ID BIGINT,
    FBT_REQUESTED_DUE_DATE DATETIME,
    FBT_ACTUAL_COMPLETION_DATE DATETIME,
    FBT_ESTIMATED_HOURS FLOAT,
    FBT_ACTUAL_HOURS FLOAT,
    FBT_PERCENT_COMPLETE FLOAT,
    FBT_ARCHIVED INT,
    FBT_DIRTY INT,
    PRIMARY KEY (CONTEXT_ID,FBT_STRUCT_ID)
);

-- table for bug history entries (ie: bl.bug history entries)
CREATE TABLE FBT_BUG_STRUCT_ENTRY (
    CONTEXT_ID INT,
    FBT_STRUCT_ENTRY_ID BIGINT,
    FBT_STRUCT_ID BIGINT,
    FBT_WHO VARCHAR(255),
    FBT_DATE_WHEN DATETIME,
    FBT_STATUS VARCHAR(255),
    FBT_ASSIGNEDTO VARCHAR(255),
    FBT_DESCRIPTION TEXT,
    FBT_SUBJECT VARCHAR(255),
    FBT_PRIORITY INT,
    FBT_PROJECT VARCHAR(255),
    FBT_AREA VARCHAR(255),
    FBT_ENVIRONMENT VARCHAR(255),
    FBT_VERSION VARCHAR(255),
    FBT_NOTIFYLIST VARCHAR(255),
    FBT_PARENT_ID BIGINT,
    FBT_REQUESTED_DUE_DATE DATETIME,
    FBT_ACTUAL_COMPLETION_DATE DATETIME,
    FBT_ESTIMATED_HOURS FLOAT,
    FBT_ACTUAL_HOURS FLOAT,
    FBT_PERCENT_COMPLETE FLOAT,
    FBT_ATTACHMENTS VARCHAR(255),
```

```
    FBT_NOTES VARCHAR(255),
    PRIMARY KEY (CONTEXT_ID,FBT_STRUCT_ENTRY_ID)
);

-- main table for holding bug custom field details (ie bl.bug custom
details)
CREATE TABLE FBT_BUG_USER_FIELD (
    CONTEXT_ID INT,
    FBT_USER_FIELD_ID BIGINT,
    FBT_STRUCT_ID BIGINT,
    FBT_FIELD_ID INT,
    FBT_STRING_VALUE TEXT,
    FBT_DATE_VALUE DATETIME,
    FBT_NUMBER_VALUE FLOAT,
    PRIMARY KEY (CONTEXT_ID,FBT_USER_FIELD_ID)
);

-- table for custom fields of bug history entries (ie: bl.bug custom
history entries)
CREATE TABLE FBT_BUG_USER_FIELD_ENTRY (
    CONTEXT_ID INT,
    FBT_USER_FIELD_ENTRY_ID BIGINT,
    FBT_STRUCT_ID BIGINT,
    FBT_STRUCT_ENTRY_ID BIGINT,
    FBT_FIELD_ID INT,
    FBT_STRING_VALUE TEXT,
    FBT_DATE_VALUE DATETIME,
    FBT_NUMBER_VALUE FLOAT,
    PRIMARY KEY (CONTEXT_ID,FBT_USER_FIELD_ENTRY_ID)
);

-- config files (ie: NAME.cfg)
CREATE TABLE FBT_CFG_DEF (
    CONTEXT_ID INT,
    FBT_CFG_ID BIGINT,
    FBT_CFG_NAME VARCHAR(40),
    FBT_CFG_INFO TEXT,
    FBT_CFG_DUPLICATE INT,
    PRIMARY KEY (CONTEXT_ID,FBT_CFG_ID)
);

-- saved charts
CREATE TABLE FBT_CHART_DEF (
    CONTEXT_ID INT,
    FBT_CHART_ID BIGINT,
    FBT_CHART_NAME VARCHAR(255),
    FBT_CHART_INFO TEXT,
    PRIMARY KEY (CONTEXT_ID,FBT_CHART_ID)
);

-- saved color codes
CREATE TABLE FBT_COLORCODE_DEF (
    CONTEXT_ID INT,
    FBT_COLORCODE_ID BIGINT,
    FBT_COLORCODE_FIELD VARCHAR(255),
    FBT_COLORCODE_INFO TEXT,
    PRIMARY KEY (CONTEXT_ID,FBT_COLORCODE_ID)
```

```
);

-- new in Version 7.08
-- store dashboard settings
CREATE TABLE FBT_DASHBOARD (
  FBT_DASHBOARD_ID INT,
  FBT_LOGINID VARCHAR(255),
  FBT_DASHBOARD_INFO TEXT,
  PRIMARY KEY (FBT_DASHBOARD_ID)
);

-- new in Version 7.08
-- store dashboard component settings
CREATE TABLE FBT_DASHBOARD_COMPONENT (
  FBT_DASHBOARD_COMPONENT_ID INT,
  FBT_COMPONENT_INFO TEXT,
  PRIMARY KEY (FBT_DASHBOARD_COMPONENT_ID)
);

-- event definitions (ie: e1.es)
CREATE TABLE FBT_EVENT_DEF (
  CONTEXT_ID INT,
  FBT_EVENT_ID BIGINT,
  FBT_EVENT_NAME VARCHAR(255),
  FBT_EVENT_INFO TEXT,
  PRIMARY KEY (CONTEXT_ID,FBT_EVENT_ID)
);

-- defines custom field files (ie: F1.fld)
CREATE TABLE FBT_FLD_DEF (
  CONTEXT_ID INT,
  FBT_FLD_ID BIGINT,
  FBT_FLD_NAME VARCHAR(255),
  FBT_FLD_INFO TEXT,
  PRIMARY KEY (CONTEXT_ID,FBT_FLD_ID)
);

-- defines filter files (ie: F1.flt)
CREATE TABLE FBT_FLT_DEF (
  CONTEXT_ID INT,
  FBT_FLT_ID BIGINT,
  FBT_FLT_NAME VARCHAR(255),
  FBT_FLT_INFO TEXT,
  PRIMARY KEY (CONTEXT_ID,FBT_FLT_ID)
);

-- group definitions (ie: G1.grp)
CREATE TABLE FBT_GROUP_DEF (
  CONTEXT_ID INT,
  FBT_GROUP_ID BIGINT,
  FBT_GROUP_NAME VARCHAR(255),
  FBT_GROUP_INFO TEXT,
  PRIMARY KEY (CONTEXT_ID,FBT_GROUP_ID)
);

-- hierarchy definitions (ie: 1.hs)
```

```
CREATE TABLE FBT_HS_DEF (
    CONTEXT_ID INT,
    FBT_HS_CHILD_ID BIGINT,
    FBT_HS_INFO TEXT,
    PRIMARY KEY (CONTEXT_ID,FBT_HS_CHILD_ID)
);

-- mail rules (ie: mr1.rul)
CREATE TABLE FBT_MAILRULE_DEF (
    CONTEXT_ID INT,
    FBT_MAILRULE_ID BIGINT,
    FBT_MAILRULE_NAME VARCHAR(255),
    FBT_MAILRULE_INFO TEXT,
    PRIMARY KEY (CONTEXT_ID,FBT_MAILRULE_ID)
);

-- project definitions (ie: P1.prj)
CREATE TABLE FBT_PROJECT_DEF (
    CONTEXT_ID INT,
    FBT_PROJECT_ID BIGINT,
    FBT_PROJECT_NAME VARCHAR(255),
    FBT_PROJECT_INFO TEXT,
    PRIMARY KEY (CONTEXT_ID,FBT_PROJECT_ID)
);

-- saved reports
CREATE TABLE FBT_REPORT_DEF (
    CONTEXT_ID INT,
    FBT_REPORT_ID BIGINT,
    FBT_REPORT_NAME VARCHAR(255),
    FBT_REPORT_INFO TEXT,
    PRIMARY KEY (CONTEXT_ID,FBT_REPORT_ID)
);

-- mail rule return messages (ie: rml.ret)
CREATE TABLE FBT_RET_DEF (
    CONTEXT_ID INT,
    FBT_RET_ID BIGINT,
    FBT_RET_NAME VARCHAR(255),
    FBT_RET_INFO TEXT,
    PRIMARY KEY (CONTEXT_ID,FBT_RET_ID)
);

-- user definitions (ie: U1.use)
CREATE TABLE FBT_USER_DEF (
    CONTEXT_ID INT,
    FBT_USER_ID BIGINT,
    FBT_LOGINID VARCHAR(255),
    FBT_USER_INFO TEXT,
    PRIMARY KEY (CONTEXT_ID,FBT_USER_ID)
);

-- workflow definitions (ie: W1.wf)
CREATE TABLE FBT_WF_DEF (
    CONTEXT_ID INT,
    FBT_WF_ID BIGINT,
    FBT_WF_NAME VARCHAR(255),
```

```
        FBT_WF_INFO TEXT,
        PRIMARY KEY (CONTEXT_ID,FBT_WF_ID)
);

-- table used for generic objects (ie: FTS)
CREATE TABLE FIT_OBJECT (
    CONTEXT_ID INT,
    SYSTEM_ID INT,
    FIT_OBJECT_ID BIGINT,
    CLASSNAME VARCHAR(255),
    OBJECT_DATA TEXT,
    FIT_ARCHIVED INT,
    PRIMARY KEY (CONTEXT_ID,SYSTEM_ID,FIT_OBJECT_ID)
);

-- New in Version 7.04
-- main table for holding attachment info (ie: tsA1.tsa)
CREATE TABLE TS_ATTACHMENT (
    CONTEXT_ID INT,
    TS_ATTACHMENT_ID BIGINT,
    TS_STRUCT_ID BIGINT,
    TS_CREATED_BY VARCHAR(255),
    TS_ATTACH_FILENAME VARCHAR(255),
    TS_ORIGINAL_FILENAME VARCHAR(255),
    TS_CONTENT_TYPE VARCHAR(255),
    TS_ATTACHMENT_DATE DATETIME,
    TS_COMMENTS VARCHAR(255),
    TS_CONTENT IMAGE,
    TS_ARCHIVED INT,
    PRIMARY KEY (CONTEXT_ID,TS_ATTACHMENT_ID)
);

-- table for holding attachment version info (ie: A1.att)
CREATE TABLE TS_ATTACHMENT_VERSION (
    CONTEXT_ID INT,
    TS_ATTACHMENT_ID BIGINT,
    TS_STRUCT_ID BIGINT,
    TS_CREATED_BY VARCHAR(255),
    TS_ATTACH_FILENAME VARCHAR(255),
    TS_ORIGINAL_FILENAME VARCHAR(255),
    TS_CONTENT_TYPE VARCHAR(255),
    TS_ATTACHMENT_DATE DATETIME,
    TS_COMMENTS VARCHAR(255),
    TS_VERSION VARCHAR(255),
    TS_FOLDER_ID BIGINT,
    TS_ATTACHMENT_VERSION_ID BIGINT,
    TS_CONTENT IMAGE,
    TS_ARCHIVED INT,
    PRIMARY KEY (CONTEXT_ID,TS_ATTACHMENT_VERSION_ID)
);

-- table for holding attachment folder info (ie: f1.fol)
CREATE TABLE TS_ATTACHMENT_FOLDER (
    CONTEXT_ID INT,
    TS_FOLDER_ID BIGINT,
    TS_STRUCT_ID BIGINT,
    TS_GLOBAL INT,
```

```
TS_CREATED_BY VARCHAR(255),
TS_FOLDER_DATE DATETIME,
TS_FOLDER_NAME VARCHAR(255),
TS_PARENT_FOLDER_ID BIGINT,
TS_ARCHIVED INT,
PRIMARY KEY (CONTEXT_ID,TS_FOLDER_ID)
);

CREATE INDEX FBSE_FSID ON FBT_BUG_STRUCT_ENTRY (FBT_STRUCT_ID);
CREATE INDEX FBUF_FSID ON FBT_BUG_USER_FIELD (FBT_STRUCT_ID);
CREATE INDEX FBUFE_FSID ON FBT_BUG_USER_FIELD_ENTRY (FBT_STRUCT_ID);
CREATE INDEX FBUFE_FSEID ON FBT_BUG_USER_FIELD_ENTRY
(FBT_STRUCT_ENTRY_ID);

commit tran fbtsetup;
```

4.3 Configuring JDBC Driver

This section illustrates how to configure and use JDBC drivers for use with MS SQL.

At the time of testing, the following URL was used to obtain Microsoft's JDBC SQLServerDriver.

<http://www.microsoft.com/downloads/results.aspx?productID=&freetext=jdbc+driver&DisplayLang=en>

This driver will install to a directory on your machine. This directory will include a "lib" directory where the actually driver files are located.

Your installation will include one of two files, depending on when you installed the product.

If you have a FIT.lax configuration file, you need add this file to the lax classpath as follows:

```
# LAX.CLASS.PATH
# -----
# the Java classpath necessary to run this application
# Can be separated by colons (Mac OS/Unix) or semicolons (Windows)
lax.class.path=fit.jar;lax.jar;msutil.jar;mssqlserver.jar;msbase.jar
```

If you do not have a FIT.lax file, you will want to look for the FIT.ini and FITService.ini configuration files (always modify both), where the JRE is configured on a line similar to:

```
[Class Path]
Class Path=C:\Program Files\FIT\fit.jar;C:\Program Files\FIT\msutil.jar; C:\Program Files\FIT\mssqlserver.jar; C:\Program Files\FIT\msbase.jar
```

Finally, you may simply have a Fit script which you can modify similar to :

```
Java -cp "fit.jar;msutil.jar;mssqlserver.jar;msbase.jar" Fit
```

After you modify the correct location, simply restart FIT and check the System Info, to make sure the driver is listed as part of the "java.class.path", on the System Properties tab

4.4 Request a Valid Registration Key

Please see section 2.4 for generic instructions.

4.5 Connecting to database

All database configuration is handled in the Database Administration menu, which is available from the Advanced section in the Admin Menu of the Enterprise Solution software. If you can not see this menu, then you are probably not using a properly activated key.

Using the JDBC driver in 3 jar files from section 4.3.2 (msbase.jar, mssqlserver.jar, and msutil.jar) , you would use the following information to connect to the database. (Note that the machine and port are included in the DB URL, as they are on a separate machine)

```
DB Driver:          com.microsoft.jdbc.sqlserver.SQLServerDriver
DB URL:             jdbc:microsoft:sqlserver://optiplex:1433;DatabaseName=myDB;
                    SelectMethod=Cursor
DB User:           fbt
DB Password:       ****
```

After the correct values are entered, you can press the “Save and Test” button to check them. You should receive something like the following:

```
OK:
Product Name: Microsoft SQL Server
Product Version: Microsoft SQL Server 2000 - 8.00.194 (Intel
X86)
  Aug  6 2000 00:57:48
  Copyright (c) 1988-2000 Microsoft Corporation
  Developer Edition on Windows NT 5.1 (Build 2600: Service Pack 2)

Driver: 2.2
DriverName: SQLServer
DriverVersion: 2.2.0037
```

4.6 Upgrade your Java Runtime Environment

Please see Section 2.6 for generic instructions.

4.7 Migrating System

Please see Section 2.7 for generic instructions.

5 MS SQL 2005 Server Installation

5.1 Special Considerations

In case you are new to using this database, this is how we set things up during testing.

A : SQL 2005 Setup

Open up the Enterprise Manager

browse to Console Root -> Microsoft SQL Servers -> SQL Server Group -> [machine]
(Windows NT)

If you want a separate Database/Catalog, create one in the Databases area.

Right click on New Database and then enter the new name. I used FBT for my system.

Under the Security area, right click on Logins to create a new user (if desired)

Under the properties for that user, grant the database access for the new database.

Then choose "Database Creators" and "Bulk Insert Administrators" from the roles tab

Then select the Database that you want this user to use under the Defaults... Database section.

You then need to go back to the Database properties and select "Create Table" under the Permissions tab.

B: Schema Setup

Open up an "SQL Query Analyser" and connect to the database using the new user that you created.

Then execute the schema that is detailed in section 4.2 of the document or open this file and execute:

```
http://www.fittrackingsolutions.com/misc/FIT\_MSSQL.SQL
```

This should create the tables just fine and you can see them under the TABLES section for your Database in the Enterprise manager.

Then you can connect from FIT and execute the "Copy Files to DB" option.

C: Authentication

One other thing:

By default, I think SQL Server 2005 only accepts windows authenticated users.

To allow the SQL Server users, do the following:

Right click on [machine] and select Properties.

Under the Security tab, make sure "SQL Server and Windows" is checked for Authentication. (Mixed Mode)

You will have to restart SQL 2005 if you make this change.

5.2 Creating Tables

```
-- Schema in MS SQL Server syntax
-- also available here :
http://www.fittrackingsolutions.com/misc/FIT_MSSQL.SQL
```

```
begin tran fbtsetup;
```

```
-- main table for holding attachment info (ie: A1.att)
```

```
CREATE TABLE FBT_ATTACHMENT (
    CONTEXT_ID INT,
    FBT_ATTACHMENT_ID BIGINT,
    FBT_STRUCT_ID BIGINT,
    FBT_CREATED_BY VARCHAR(255),
    FBT_ATTACH_FILENAME VARCHAR(255),
    FBT_ORIGINAL_FILENAME VARCHAR(255),
    FBT_CONTENT_TYPE VARCHAR(255),
    FBT_ATTACHMENT_DATE DATETIME,
    FBT_COMMENTS VARCHAR(255),
    FBT_VERSION VARCHAR(255),
    FBT_FOLDER_ID BIGINT,
    FBT_ATTACHMENT_VERSION_ID BIGINT,
    FBT_CONTENT IMAGE,
    FBT_ARCHIVED INT,
    PRIMARY KEY (CONTEXT_ID,FBT_ATTACHMENT_ID)
);
```

```
-- table for holding attachment version info (ie: A1.att)
```

```
CREATE TABLE FBT_ATTACHMENT_VERSION (
    CONTEXT_ID INT,
    FBT_ATTACHMENT_ID BIGINT,
    FBT_STRUCT_ID BIGINT,
    FBT_CREATED_BY VARCHAR(255),
    FBT_ATTACH_FILENAME VARCHAR(255),
    FBT_ORIGINAL_FILENAME VARCHAR(255),
    FBT_CONTENT_TYPE VARCHAR(255),
    FBT_ATTACHMENT_DATE DATETIME,
    FBT_COMMENTS VARCHAR(255),
    FBT_VERSION VARCHAR(255),
    FBT_FOLDER_ID BIGINT,
    FBT_ATTACHMENT_VERSION_ID BIGINT,
    FBT_CONTENT IMAGE,
    FBT_ARCHIVED INT,
    PRIMARY KEY (CONTEXT_ID,FBT_ATTACHMENT_VERSION_ID)
);
```

```
-- table for holding attachment folder info (ie: f1.fol)
```

```
CREATE TABLE FBT_ATTACHMENT_FOLDER (
    CONTEXT_ID INT,
    FBT_FOLDER_ID BIGINT,
    FBT_STRUCT_ID BIGINT,
    FBT_GLOBAL INT,
    FBT_CREATED_BY VARCHAR(255),
    FBT_FOLDER_DATE DATETIME,
```

```
    FBT_FOLDER_NAME VARCHAR(255),
    FBT_PARENT_FOLDER_ID BIGINT,
    FBT_ARCHIVED INT,
    PRIMARY KEY (CONTEXT_ID,FBT_FOLDER_ID)
);

-- main table for bug details (ie: bl.bug field details)
CREATE TABLE FBT_BUG_STRUCT (
    CONTEXT_ID INT,
    FBT_STRUCT_ID BIGINT,
    FBT_SUBJECT VARCHAR(255),
    FBT_ASSIGNEDTO VARCHAR(255),
    FBT_STATUS VARCHAR(255),
    FBT_PRIORITY INT,
    FBT_PROJECT VARCHAR(255),
    FBT_AREA VARCHAR(255),
    FBT_ENVIRONMENT VARCHAR(255),
    FBT_VERSION VARCHAR(255),
    FBT_ENTEREDBY VARCHAR(255),
    FBT_NOTIFYPYLIST TEXT,
    FBT_PARENT_ID BIGINT,
    FBT_REQUESTED_DUE_DATE DATETIME,
    FBT_ACTUAL_COMPLETION_DATE DATETIME,
    FBT_ESTIMATED_HOURS FLOAT,
    FBT_ACTUAL_HOURS FLOAT,
    FBT_PERCENT_COMPLETE FLOAT,
    FBT_ARCHIVED INT,
    FBT_DIRTY INT,
    PRIMARY KEY (CONTEXT_ID,FBT_STRUCT_ID)
);

-- table for bug history entries (ie: bl.bug history entries)
CREATE TABLE FBT_BUG_STRUCT_ENTRY (
    CONTEXT_ID INT,
    FBT_STRUCT_ENTRY_ID BIGINT,
    FBT_STRUCT_ID BIGINT,
    FBT_WHO VARCHAR(255),
    FBT_DATE_WHEN DATETIME,
    FBT_STATUS VARCHAR(255),
    FBT_ASSIGNEDTO VARCHAR(255),
    FBT_DESCRIPTION TEXT,
    FBT_SUBJECT VARCHAR(255),
    FBT_PRIORITY INT,
    FBT_PROJECT VARCHAR(255),
    FBT_AREA VARCHAR(255),
    FBT_ENVIRONMENT VARCHAR(255),
    FBT_VERSION VARCHAR(255),
    FBT_NOTIFYPYLIST VARCHAR(255),
    FBT_PARENT_ID BIGINT,
    FBT_REQUESTED_DUE_DATE DATETIME,
    FBT_ACTUAL_COMPLETION_DATE DATETIME,
    FBT_ESTIMATED_HOURS FLOAT,
    FBT_ACTUAL_HOURS FLOAT,
    FBT_PERCENT_COMPLETE FLOAT,
    FBT_ATTACHMENTS VARCHAR(255),
    FBT_NOTES VARCHAR(255),
```

```
        PRIMARY KEY (CONTEXT_ID,FBT_STRUCT_ENTRY_ID)
);

-- main table for holding bug custom field details (ie bl.bug custom
details)
CREATE TABLE FBT_BUG_USER_FIELD (
    CONTEXT_ID INT,
    FBT_USER_FIELD_ID BIGINT,
    FBT_STRUCT_ID BIGINT,
    FBT_FIELD_ID INT,
    FBT_STRING_VALUE TEXT,
    FBT_DATE_VALUE DATETIME,
    FBT_NUMBER_VALUE FLOAT,
    PRIMARY KEY (CONTEXT_ID,FBT_USER_FIELD_ID)
);

-- table for custom fields of bug history entries (ie: bl.bug custom
history entries)
CREATE TABLE FBT_BUG_USER_FIELD_ENTRY (
    CONTEXT_ID INT,
    FBT_USER_FIELD_ENTRY_ID BIGINT,
    FBT_STRUCT_ID BIGINT,
    FBT_STRUCT_ENTRY_ID BIGINT,
    FBT_FIELD_ID INT,
    FBT_STRING_VALUE TEXT,
    FBT_DATE_VALUE DATETIME,
    FBT_NUMBER_VALUE FLOAT,
    PRIMARY KEY (CONTEXT_ID,FBT_USER_FIELD_ENTRY_ID)
);

-- config files (ie: NAME.cfg)
CREATE TABLE FBT_CFG_DEF (
    CONTEXT_ID INT,
    FBT_CFG_ID BIGINT,
    FBT_CFG_NAME VARCHAR(40),
    FBT_CFG_INFO TEXT,
    FBT_CFG_DUPLICATE INT,
    PRIMARY KEY (CONTEXT_ID,FBT_CFG_ID)
);

-- saved charts
CREATE TABLE FBT_CHART_DEF (
    CONTEXT_ID INT,
    FBT_CHART_ID BIGINT,
    FBT_CHART_NAME VARCHAR(255),
    FBT_CHART_INFO TEXT,
    PRIMARY KEY (CONTEXT_ID,FBT_CHART_ID)
);

-- saved color codes
CREATE TABLE FBT_COLORCODE_DEF (
    CONTEXT_ID INT,
    FBT_COLORCODE_ID BIGINT,
    FBT_COLORCODE_FIELD VARCHAR(255),
    FBT_COLORCODE_INFO TEXT,
    PRIMARY KEY (CONTEXT_ID,FBT_COLORCODE_ID)
);
```

```
-- new in Version 7.08
-- store dashboard settings
CREATE TABLE FBT_DASHBOARD (
  FBT_DASHBOARD_ID INT,
  FBT_LOGINID VARCHAR(255),
  FBT_DASHBOARD_INFO TEXT,
  PRIMARY KEY (FBT_DASHBOARD_ID)
);

-- new in Version 7.08
-- store dashboard component settings
CREATE TABLE FBT_DASHBOARD_COMPONENT (
  FBT_DASHBOARD_COMPONENT_ID INT,
  FBT_COMPONENT_INFO TEXT,
  PRIMARY KEY (FBT_DASHBOARD_COMPONENT_ID)
);

-- event definitions (ie: e1.es)
CREATE TABLE FBT_EVENT_DEF (
  CONTEXT_ID INT,
  FBT_EVENT_ID BIGINT,
  FBT_EVENT_NAME VARCHAR(255),
  FBT_EVENT_INFO TEXT,
  PRIMARY KEY (CONTEXT_ID,FBT_EVENT_ID)
);

-- defines custom field files (ie: F1.fld)
CREATE TABLE FBT_FLD_DEF (
  CONTEXT_ID INT,
  FBT_FLD_ID BIGINT,
  FBT_FLD_NAME VARCHAR(255),
  FBT_FLD_INFO TEXT,
  PRIMARY KEY (CONTEXT_ID,FBT_FLD_ID)
);

-- defines filter files (ie: F1.flt)
CREATE TABLE FBT_FLT_DEF (
  CONTEXT_ID INT,
  FBT_FLT_ID BIGINT,
  FBT_FLT_NAME VARCHAR(255),
  FBT_FLT_INFO TEXT,
  PRIMARY KEY (CONTEXT_ID,FBT_FLT_ID)
);

-- group definitions (ie: G1.grp)
CREATE TABLE FBT_GROUP_DEF (
  CONTEXT_ID INT,
  FBT_GROUP_ID BIGINT,
  FBT_GROUP_NAME VARCHAR(255),
  FBT_GROUP_INFO TEXT,
  PRIMARY KEY (CONTEXT_ID,FBT_GROUP_ID)
);

-- hierarchy definitions (ie: 1.hs)
CREATE TABLE FBT_HS_DEF (
```

```
CONTEXT_ID INT,
FBT_HS_CHILD_ID BIGINT,
FBT_HS_INFO TEXT,
PRIMARY KEY (CONTEXT_ID,FBT_HS_CHILD_ID)
);

-- mail rules (ie: mr1.rul)
CREATE TABLE FBT_MAILRULE_DEF (
CONTEXT_ID INT,
FBT_MAILRULE_ID BIGINT,
FBT_MAILRULE_NAME VARCHAR(255),
FBT_MAILRULE_INFO TEXT,
PRIMARY KEY (CONTEXT_ID,FBT_MAILRULE_ID)
);

-- project definitions (ie: P1.prj)
CREATE TABLE FBT_PROJECT_DEF (
CONTEXT_ID INT,
FBT_PROJECT_ID BIGINT,
FBT_PROJECT_NAME VARCHAR(255),
FBT_PROJECT_INFO TEXT,
PRIMARY KEY (CONTEXT_ID,FBT_PROJECT_ID)
);

-- saved reports
CREATE TABLE FBT_REPORT_DEF (
CONTEXT_ID INT,
FBT_REPORT_ID BIGINT,
FBT_REPORT_NAME VARCHAR(255),
FBT_REPORT_INFO TEXT,
PRIMARY KEY (CONTEXT_ID,FBT_REPORT_ID)
);

-- mail rule return messages (ie: rml.ret)
CREATE TABLE FBT_RET_DEF (
CONTEXT_ID INT,
FBT_RET_ID BIGINT,
FBT_RET_NAME VARCHAR(255),
FBT_RET_INFO TEXT,
PRIMARY KEY (CONTEXT_ID,FBT_RET_ID)
);

-- user definitions (ie: U1.use)
CREATE TABLE FBT_USER_DEF (
CONTEXT_ID INT,
FBT_USER_ID BIGINT,
FBT_LOGINID VARCHAR(255),
FBT_USER_INFO TEXT,
PRIMARY KEY (CONTEXT_ID,FBT_USER_ID)
);

-- workflow definitions (ie: W1.wf)
CREATE TABLE FBT_WF_DEF (
CONTEXT_ID INT,
FBT_WF_ID BIGINT,
FBT_WF_NAME VARCHAR(255),
FBT_WF_INFO TEXT,
```

```
        PRIMARY KEY (CONTEXT_ID,FBT_WF_ID)
);

-- table used for generic objects (ie: FTS)
CREATE TABLE FIT_OBJECT (
    CONTEXT_ID INT,
    SYSTEM_ID INT,
    FIT_OBJECT_ID BIGINT,
    CLASSNAME VARCHAR(255),
    OBJECT_DATA TEXT,
    FIT_ARCHIVED INT,
    PRIMARY KEY (CONTEXT_ID,SYSTEM_ID,FIT_OBJECT_ID)
);

-- New in Version 7.04
-- main table for holding attachment info (ie: tsA1.tsa)
CREATE TABLE TS_ATTACHMENT (
    CONTEXT_ID INT,
    TS_ATTACHMENT_ID BIGINT,
    TS_STRUCT_ID BIGINT,
    TS_CREATED_BY VARCHAR(255),
    TS_ATTACH_FILENAME VARCHAR(255),
    TS_ORIGINAL_FILENAME VARCHAR(255),
    TS_CONTENT_TYPE VARCHAR(255),
    TS_ATTACHMENT_DATE DATETIME,
    TS_COMMENTS VARCHAR(255),
    TS_CONTENT IMAGE,
    TS_ARCHIVED INT,
    PRIMARY KEY (CONTEXT_ID,TS_ATTACHMENT_ID)
);

-- table for holding attachment version info (ie: A1.att)
CREATE TABLE TS_ATTACHMENT_VERSION (
    CONTEXT_ID INT,
    TS_ATTACHMENT_ID BIGINT,
    TS_STRUCT_ID BIGINT,
    TS_CREATED_BY VARCHAR(255),
    TS_ATTACH_FILENAME VARCHAR(255),
    TS_ORIGINAL_FILENAME VARCHAR(255),
    TS_CONTENT_TYPE VARCHAR(255),
    TS_ATTACHMENT_DATE DATETIME,
    TS_COMMENTS VARCHAR(255),
    TS_VERSION VARCHAR(255),
    TS_FOLDER_ID BIGINT,
    TS_ATTACHMENT_VERSION_ID BIGINT,
    TS_CONTENT IMAGE,
    TS_ARCHIVED INT,
    PRIMARY KEY (CONTEXT_ID,TS_ATTACHMENT_VERSION_ID)
);

-- table for holding attachment folder info (ie: f1.fol)
CREATE TABLE TS_ATTACHMENT_FOLDER (
    CONTEXT_ID INT,
    TS_FOLDER_ID BIGINT,
    TS_STRUCT_ID BIGINT,
    TS_GLOBAL INT,
    TS_CREATED_BY VARCHAR(255),
```

```
TS_FOLDER_DATE DATETIME,  
TS_FOLDER_NAME VARCHAR(255),  
TS_PARENT_FOLDER_ID BIGINT,  
TS_ARCHIVED INT,  
PRIMARY KEY (CONTEXT_ID,TS_FOLDER_ID)  
);  
  
CREATE INDEX FBSE_FSID ON FBT_BUG_STRUCT_ENTRY (FBT_STRUCT_ID);  
CREATE INDEX FBUF_FSID ON FBT_BUG_USER_FIELD (FBT_STRUCT_ID);  
CREATE INDEX FBUFE_FSID ON FBT_BUG_USER_FIELD_ENTRY (FBT_STRUCT_ID);  
CREATE INDEX FBUFE_FSEID ON FBT_BUG_USER_FIELD_ENTRY  
(FBT_STRUCT_ENTRY_ID);  
  
commit tran fbtsetup;
```

5.3 Configuring JDBC Driver

This section illustrates how to configure and use JDBC drivers for use with MS SQL 2005.

At the time of testing, the following URL was used to obtain Microsoft's JDBC SQLServerDriver.

<http://msdn2.microsoft.com/en-ca/data/aa937724.aspx>

You can find the sqljdbc.jar driver in this package, once it is installed and it must be copied into the installation directory for FIT.

If you have a FIT.lax configuration file, you need add this file to the lax classpath as follows:

```
# LAX.CLASS.PATH
# -----
# the Java classpath necessary to run this application
# Can be separated by colons (Mac OS/Unix) or semicolons (Windows)
lax.class.path=fit.jar;lax.jar;sqljdbc.jar
```

If you do not have a FIT.lax file, you will want to look for the FIT.ini and FITService.ini configuration files (always modify both), where the JRE is configured on a line similar to:

```
[Class Path]
Class Path=C:\Program Files\FIT\fit.jar;C:\Program Files\FIT\sqljdbc.jar
```

Finally, you may simply have a Fit script which you can modify similar to :

```
Java -cp "fit.jar;sqljdbc.jar" Fit
```

After you modify the correct location, simply restart FIT and check the System Info, to make sure the driver is listed as part of the "java.class.path", on the System Properties tab

5.4 Request a Valid Registration Key

Please see section 2.4 for generic instructions.

5.5 Connecting to database

All database configuration is handled in the Database Administration menu, which is available from the Advanced section in the Admin Menu of the Enterprise Solution software. If you can not see this menu, then you are probably not using a properly activated key.

Using the JDBC driver in 3 jar files from section 4.3.2 (msbase.jar, mssqlserver.jar, and msutil.jar) , you would use the following information to connect to the database. (Note that the machine and port are included in the DB URL, as they are on a separate machine)

DB Driver: com.microsoft.sqlserver.jdbc.SQLServerDriver
DB URL: jdbc:sqlserver://optiplex:1433
DB User: fbt
DB Password: ****

After the correct values are entered, you can press the “Save and Test” button to check them. You should receive something like the following:

OK:
Product Name: Microsoft SQL Server
Product Version: 8.00.194
Driver: 1.1
DriverName: Microsoft SQL Server 2005 JDBC Driver
DriverVersion: 1.1.1501.101

5.6 Upgrade your Java Runtime Environment

Please see Section 2.6 for generic instructions.

5.7 Migrating System

Please see Section 2.7 for generic instructions.

6 Oracle Installation

6.1 Special Considerations

You will need to create a user, which is configured to use your newly created database where the tables are created (as explained in the next step).

6.2 Creating Tables

```
-- Schema in ORACLE syntax
-- also available here :
http://www.fittrackingsolutions.com/misc/FIT_ORACLE.SQL

-- Schema in ORACLE syntax
-- main table for holding attachment info (ie: A1.att)
CREATE TABLE FBT_ATTACHMENT (
    CONTEXT_ID INT,
    FBT_ATTACHMENT_ID NUMBER,
    FBT_STRUCT_ID NUMBER,
    FBT_CREATED_BY VARCHAR(255),
    FBT_ATTACH_FILENAME VARCHAR(255),
    FBT_ORIGINAL_FILENAME VARCHAR(255),
    FBT_CONTENT_TYPE VARCHAR(255),
    FBT_ATTACHMENT_DATE DATE,
    FBT_COMMENTS VARCHAR(255),
    FBT_VERSION VARCHAR(255),
    FBT_FOLDER_ID NUMBER,
    FBT_ATTACHMENT_VERSION_ID NUMBER,
    FBT_CONTENT LONG RAW,
    FBT_ARCHIVED INT,
    PRIMARY KEY (CONTEXT_ID, FBT_ATTACHMENT_ID)
);

-- table for holding attachment version info (ie: A1.att)
CREATE TABLE FBT_ATTACHMENT_VERSION (
    CONTEXT_ID INT,
    FBT_ATTACHMENT_ID NUMBER,
    FBT_STRUCT_ID NUMBER,
    FBT_CREATED_BY VARCHAR(255),
    FBT_ATTACH_FILENAME VARCHAR(255),
    FBT_ORIGINAL_FILENAME VARCHAR(255),
    FBT_CONTENT_TYPE VARCHAR(255),
    FBT_ATTACHMENT_DATE DATE,
    FBT_COMMENTS VARCHAR(255),
    FBT_VERSION VARCHAR(255),
    FBT_FOLDER_ID NUMBER,
    FBT_ATTACHMENT_VERSION_ID NUMBER,
    FBT_CONTENT LONG RAW,
    FBT_ARCHIVED INT,
    PRIMARY KEY (CONTEXT_ID, FBT_ATTACHMENT_VERSION_ID)
);
```

```
-- table for holding attachment folder info (ie: f1.fol)
CREATE TABLE FBT_ATTACHMENT_FOLDER (
    CONTEXT_ID INT,
    FBT_FOLDER_ID NUMBER,
    FBT_STRUCT_ID NUMBER,
    FBT_GLOBAL INT,
    FBT_CREATED_BY VARCHAR(255),
    FBT_FOLDER_DATE DATE,
    FBT_FOLDER_NAME VARCHAR(255),
    FBT_PARENT_FOLDER_ID NUMBER,
    FBT_ARCHIVED INT,
    PRIMARY KEY (CONTEXT_ID,FBT_FOLDER_ID)
);

-- main table for bug details (ie: b1.bug field details)
CREATE TABLE FBT_BUG_STRUCT (
    CONTEXT_ID INT,
    FBT_STRUCT_ID NUMBER,
    FBT_SUBJECT VARCHAR(255),
    FBT_ASSIGNEDTO VARCHAR(255),
    FBT_STATUS VARCHAR(255),
    FBT_PRIORITY INT,
    FBT_PROJECT VARCHAR(255),
    FBT_AREA VARCHAR(255),
    FBT_ENVIRONMENT VARCHAR(255),
    FBT_VERSION VARCHAR(255),
    FBT_ENTEREDBY VARCHAR(255),
    FBT_NOTIFYPYLIST LONG,
    FBT_PARENT_ID NUMBER,
    FBT_REQUESTED_DUE_DATE DATE,
    FBT_ACTUAL_COMPLETION_DATE DATE,
    FBT_ESTIMATED_HOURS NUMBER,
    FBT_ACTUAL_HOURS NUMBER,
    FBT_PERCENT_COMPLETE NUMBER,
    FBT_ARCHIVED INT,
    FBT_DIRTY INT,
    PRIMARY KEY (CONTEXT_ID,FBT_STRUCT_ID)
);

-- table for bug history entries (ie: b1.bug history entries)
CREATE TABLE FBT_BUG_STRUCT_ENTRY (
    CONTEXT_ID INT,
    FBT_STRUCT_ENTRY_ID NUMBER,
    FBT_STRUCT_ID NUMBER,
    FBT_WHO VARCHAR(255),
    FBT_DATE_WHEN DATE,
    FBT_STATUS VARCHAR(255),
    FBT_ASSIGNEDTO VARCHAR(255),
    FBT_DESCRIPTION LONG,
    FBT_SUBJECT VARCHAR(255),
    FBT_PRIORITY INT,
    FBT_PROJECT VARCHAR(255),
    FBT_AREA VARCHAR(255),
    FBT_ENVIRONMENT VARCHAR(255),
    FBT_VERSION VARCHAR(255),
    FBT_NOTIFYPYLIST VARCHAR(255),
```

```
    FBT_PARENT_ID NUMBER,
    FBT_REQUESTED_DUE_DATE DATE,
    FBT_ACTUAL_COMPLETION_DATE DATE,
    FBT_ESTIMATED_HOURS NUMBER,
    FBT_ACTUAL_HOURS NUMBER,
    FBT_PERCENT_COMPLETE NUMBER,
    FBT_ATTACHMENTS VARCHAR(255),
    FBT_NOTES VARCHAR(255),
    PRIMARY KEY (CONTEXT_ID,FBT_STRUCT_ENTRY_ID)
);

-- main table for holding bug custom field details (ie bl.bug custom
details)
CREATE TABLE FBT_BUG_USER_FIELD (
    CONTEXT_ID INT,
    FBT_USER_FIELD_ID NUMBER,
    FBT_STRUCT_ID NUMBER,
    FBT_FIELD_ID INT,
    FBT_STRING_VALUE LONG,
    FBT_DATE_VALUE DATE,
    FBT_NUMBER_VALUE NUMBER,
    PRIMARY KEY (CONTEXT_ID,FBT_USER_FIELD_ID)
);

-- table for custom fields of bug history entries (ie: bl.bug custom
history entries)
CREATE TABLE FBT_BUG_USER_FIELD_ENTRY (
    CONTEXT_ID INT,
    FBT_USER_FIELD_ENTRY_ID NUMBER,
    FBT_STRUCT_ID NUMBER,
    FBT_STRUCT_ENTRY_ID NUMBER,
    FBT_FIELD_ID INT,
    FBT_STRING_VALUE LONG,
    FBT_DATE_VALUE DATE,
    FBT_NUMBER_VALUE NUMBER,
    PRIMARY KEY (CONTEXT_ID,FBT_USER_FIELD_ENTRY_ID)
);

-- config files (ie: NAME.cfg)
CREATE TABLE FBT_CFG_DEF (
    CONTEXT_ID INT,
    FBT_CFG_ID NUMBER,
    FBT_CFG_NAME VARCHAR(40),
    FBT_CFG_INFO LONG,
    FBT_CFG_DUPLICATE INT,
    PRIMARY KEY (CONTEXT_ID,FBT_CFG_ID)
);

-- saved charts
CREATE TABLE FBT_CHART_DEF (
    CONTEXT_ID INT,
    FBT_CHART_ID NUMBER,
    FBT_CHART_NAME VARCHAR(255),
    FBT_CHART_INFO LONG,
    PRIMARY KEY (CONTEXT_ID,FBT_CHART_ID)
);
```

```
-- saved color codes
CREATE TABLE FBT_COLORCODE_DEF (
    CONTEXT_ID INT,
    FBT_COLORCODE_ID NUMBER,
    FBT_COLORCODE_FIELD VARCHAR(255),
    FBT_COLORCODE_INFO LONG,
    PRIMARY KEY (CONTEXT_ID,FBT_COLORCODE_ID)
);

-- new in Version 7.08
-- store dashboard settings
CREATE TABLE FBT_DASHBOARD (
    FBT_DASHBOARD_ID INT,
    FBT_LOGINID VARCHAR(255),
    FBT_DASHBOARD_INFO LONG,
    PRIMARY KEY (FBT_DASHBOARD_ID)
);

-- new in Version 7.08
-- store dashboard component settings
CREATE TABLE FBT_DASHBOARD_COMPONENT (
    FBT_DASHBOARD_COMPONENT_ID INT,
    FBT_COMPONENT_INFO LONG,
    PRIMARY KEY (FBT_DASHBOARD_COMPONENT_ID)
);

-- event definitions (ie: e1.es)
CREATE TABLE FBT_EVENT_DEF (
    CONTEXT_ID INT,
    FBT_EVENT_ID NUMBER,
    FBT_EVENT_NAME VARCHAR(255),
    FBT_EVENT_INFO LONG,
    PRIMARY KEY (CONTEXT_ID,FBT_EVENT_ID)
);

-- defines custom field files (ie: F1.fld)
CREATE TABLE FBT_FLD_DEF (
    CONTEXT_ID INT,
    FBT_FLD_ID NUMBER,
    FBT_FLD_NAME VARCHAR(255),
    FBT_FLD_INFO LONG,
    PRIMARY KEY (CONTEXT_ID,FBT_FLD_ID)
);

-- defines filter files (ie: F1.flt)
CREATE TABLE FBT_FLT_DEF (
    CONTEXT_ID INT,
    FBT_FLT_ID NUMBER,
    FBT_FLT_NAME VARCHAR(255),
    FBT_FLT_INFO LONG,
    PRIMARY KEY (CONTEXT_ID,FBT_FLT_ID)
);

-- group definitions (ie: G1.grp)
CREATE TABLE FBT_GROUP_DEF (
    CONTEXT_ID INT,
```

```
    FBT_GROUP_ID NUMBER,  
    FBT_GROUP_NAME VARCHAR(255),  
    FBT_GROUP_INFO LONG,  
    PRIMARY KEY (CONTEXT_ID,FBT_GROUP_ID)  
);  
  
-- hierarchy definitions (ie: 1.hs)  
CREATE TABLE FBT_HS_DEF (  
    CONTEXT_ID INT,  
    FBT_HS_CHILD_ID NUMBER,  
    FBT_HS_INFO LONG,  
    PRIMARY KEY (CONTEXT_ID,FBT_HS_CHILD_ID)  
);  
  
-- mail rules (ie: mr1.rul)  
CREATE TABLE FBT_MAILRULE_DEF (  
    CONTEXT_ID INT,  
    FBT_MAILRULE_ID NUMBER,  
    FBT_MAILRULE_NAME VARCHAR(255),  
    FBT_MAILRULE_INFO LONG,  
    PRIMARY KEY (CONTEXT_ID,FBT_MAILRULE_ID)  
);  
  
-- project definitions (ie: P1.prj)  
CREATE TABLE FBT_PROJECT_DEF (  
    CONTEXT_ID INT,  
    FBT_PROJECT_ID NUMBER,  
    FBT_PROJECT_NAME VARCHAR(255),  
    FBT_PROJECT_INFO LONG,  
    PRIMARY KEY (CONTEXT_ID,FBT_PROJECT_ID)  
);  
  
-- saved reports  
CREATE TABLE FBT_REPORT_DEF (  
    CONTEXT_ID INT,  
    FBT_REPORT_ID NUMBER,  
    FBT_REPORT_NAME VARCHAR(255),  
    FBT_REPORT_INFO LONG,  
    PRIMARY KEY (CONTEXT_ID,FBT_REPORT_ID)  
);  
  
-- mail rule return messages (ie: rml.ret)  
CREATE TABLE FBT_RET_DEF (  
    CONTEXT_ID INT,  
    FBT_RET_ID NUMBER,  
    FBT_RET_NAME VARCHAR(255),  
    FBT_RET_INFO LONG,  
    PRIMARY KEY (CONTEXT_ID,FBT_RET_ID)  
);  
  
-- user definitions (ie: U1.use)  
CREATE TABLE FBT_USER_DEF (  
    CONTEXT_ID INT,  
    FBT_USER_ID NUMBER,  
    FBT_LOGINID VARCHAR(255),  
    FBT_USER_INFO LONG,  
    PRIMARY KEY (CONTEXT_ID,FBT_USER_ID)
```

```
);

-- workflow definitions (ie: W1.wf)
CREATE TABLE FBT_WF_DEF (
    CONTEXT_ID INT,
    FBT_WF_ID NUMBER,
    FBT_WF_NAME VARCHAR(255),
    FBT_WF_INFO LONG,
    PRIMARY KEY (CONTEXT_ID,FBT_WF_ID)
);

-- table used for generic objects (ie: FTS)
CREATE TABLE FIT_OBJECT (
    CONTEXT_ID INT,
    SYSTEM_ID INT,
    FIT_OBJECT_ID NUMBER,
    CLASSNAME VARCHAR(255),
    OBJECT_DATA LONG,
    FIT_ARCHIVED INT,
    PRIMARY KEY (CONTEXT_ID,SYSTEM_ID,FIT_OBJECT_ID)
);

-- New in Version 7.04
-- main table for holding TestSpec attachment info (ie: tsA1.tsa)
CREATE TABLE TS_ATTACHMENT (
    CONTEXT_ID INT,
    TS_ATTACHMENT_ID NUMBER,
    TS_STRUCT_ID NUMBER,
    TS_CREATED_BY VARCHAR(255),
    TS_ATTACH_FILENAME VARCHAR(255),
    TS_ORIGINAL_FILENAME VARCHAR(255),
    TS_CONTENT_TYPE VARCHAR(255),
    TS_ATTACHMENT_DATE DATE,
    TS_COMMENTS VARCHAR(255),
    TS_CONTENT LONG RAW,
    TS_ARCHIVED INT,
    PRIMARY KEY (CONTEXT_ID,TS_ATTACHMENT_ID)
);

-- table for holding attachment version info (ie: A1.att)
CREATE TABLE TS_ATTACHMENT_VERSION (
    CONTEXT_ID INT,
    TS_ATTACHMENT_ID NUMBER,
    TS_STRUCT_ID NUMBER,
    TS_CREATED_BY VARCHAR(255),
    TS_ATTACH_FILENAME VARCHAR(255),
    TS_ORIGINAL_FILENAME VARCHAR(255),
    TS_CONTENT_TYPE VARCHAR(255),
    TS_ATTACHMENT_DATE DATE,
    TS_COMMENTS VARCHAR(255),
    TS_VERSION VARCHAR(255),
    TS_FOLDER_ID NUMBER,
    TS_ATTACHMENT_VERSION_ID NUMBER,
    TS_CONTENT LONG RAW,
    TS_ARCHIVED INT,
    PRIMARY KEY (CONTEXT_ID,TS_ATTACHMENT_VERSION_ID)
);
```

```
-- table for holding attachment folder info (ie: fl.fol)
CREATE TABLE TS_ATTACHMENT_FOLDER (
    CONTEXT_ID INT,
    TS_FOLDER_ID NUMBER,
    TS_STRUCT_ID NUMBER,
    TS_GLOBAL INT,
    TS_CREATED_BY VARCHAR(255),
    TS_FOLDER_DATE DATE,
    TS_FOLDER_NAME VARCHAR(255),
    TS_PARENT_FOLDER_ID NUMBER,
    TS_ARCHIVED INT,
    PRIMARY KEY (CONTEXT_ID,TS_FOLDER_ID)
);

CREATE INDEX FBSE_FSID ON FBT_BUG_STRUCT_ENTRY (FBT_STRUCT_ID);
CREATE INDEX FBUF_FSID ON FBT_BUG_USER_FIELD (FBT_STRUCT_ID);
CREATE INDEX FBUFE_FSID ON FBT_BUG_USER_FIELD_ENTRY (FBT_STRUCT_ID);
CREATE INDEX FBUFE_FSEID ON FBT_BUG_USER_FIELD_ENTRY
(FBT_STRUCT_ENTRY_ID);

commit;
```

6.3 Configuring Driver

This section illustrates how to configure and use JDBC drivers for use with Oracle. This driver is available as the classes111.jar file in the Ora90/jdbc/lib directory for an Oracle9 system (or equivalent).

You can then restart FIT with classes111.jar in the installation directory for FIT.

If you have a FIT.lax configuration file, you need add this file to the lax classpath as follows:

```
# LAX.CLASS.PATH
# -----
# the Java classpath necessary to run this application
# Can be separated by colons (Mac OS/Unix) or semicolons (Windows)
lax.class.path=fit.jar;lax.jar;classes111.jar
```

If you do not have a FIT.lax file, you will want to look for the FIT.ini and FITService.ini configuration files (always modify both), where the JRE is configured on a line similar to:

```
[Class Path]
Class Path=C:\Program Files\FIT\fit.jar;C:\Program Files\FIT\classes111.jar
```

Finally, you may simply have a Fit script which you can modify similar to :
Java -cp "fit.jar;**classes111.jar**" Fit

After you modify the correct location, simply restart FIT and check the System Info, to make sure the driver is listed as part of the "java.class.path", on the System Properties tab

NOTE: Java versions > 1.2 require you to download a specific oracle driver. The name will be "ojdbcX.jar" where X refers to the version of Java that you are using. For example, "ojdbc6.jar" is used for systems using a JVM 1.6

6.4 Request a Valid Registration Key

Please see section 2.4 for generic instructions.

6.5 Connecting to database

All database configuration is handled in the Database Administration menu, which is available from the Advanced section in the Admin Menu of the Enterprise Solution software. If you can not see this menu, then you are probably not using a properly activated registration key.

Using the JDBC driver in classes111.jar from section 5.3.2, you would use the following information to connect to the database. (Note that the machine and port are included in the DB URL, as they are on a separate machine)

DB Driver:	oracle.jdbc.driver.OracleDriver
DB URL:	jdbc:oracle:thin:@optiplex:1521:myDB
DB User:	pete
DB Password:	****

NOTE: As of driver "ojdbc6.jar" , the new driver class is "oracle.jdbc.OracleDriver" (ie: "driver" package is no longer used)

After the correct values are entered, you can press the "Save and Test" button to check them. You should receive something like the following:

```
OK:
Product Name: Oracle
Product Version: Oracle9i Enterprise Edition Release 9.0.1.1.1 - Production
With the Partitioning option
JServer Release 9.0.1.1.1 - Production
Driver: 9.0
DriverName: Oracle JDBC driver
```

DriverVersion: 9.0.1.1.0

6.6 Upgrade your Java Runtime Environment

Please see Section 2.6 for generic instructions.

6.7 Migrating System

Please see Section 2.6 for generic instructions.

7 Frequently Asked Questions

7.1 Invalid Database URL

java.sql.SQLException: [Microsoft][ODBC Driver Manager] Data source name not found and no default driver specified

- make sure that you have spelled the url correctly and that there are no extra spaces before/after the name

7.2 ClassNotFoundException

alcea.db.DatabaseHelper@43245a Database Connection NOT available

```
java.lang.ClassNotFoundException: com.mysql.jdbc.Driver
    at alcea.db.DatabaseHelper.if(Unknown Source)
    at alcea.db.DatabaseHelper.getConnection(Compiled Code)
    at alcea.db.DatabaseHelper.test(Unknown Source)
    at alcea.db.AdminDatabase.process(Compiled Code)
    at com.other.l.a(Unknown Source)
    at com.other.l.a(Unknown Source)
    at com.other.l.if(Compiled Code)
    at com.other.s.if(Unknown Source)
    at com.other.s.run(Compiled Code)
    at java.lang.Thread.run(Thread.java:459)
```

The default JRE shipped with the windows version of FBT is JRE1.1.8.

You need to upgrade the JRE directory in your installation directory to make sure the desired class can be found. This involves downloading the new JRE to this location and adjusting the FastIssueTrack.lax file so that the `lax.nl.current.vm` variable reflects the new JRE.

7.3 Invalid Database Driver

```
java.lang.IllegalArgumentException: sun.jdbc.odbc.JdbcOdbcDriver
    at alcea.db.BugDatabaseHelper.getConnection(BugDatabaseHelper.java:99)
    at alcea.db.BugDatabaseHelper.init(Compiled Code)
    at alcea.db.AdminDatabase.process(Compiled Code)
    at com.other.HttpHandler.processChain(HttpHandler.java:249)
    at com.other.HttpHandler.processChain(HttpHandler.java:205)
    at com.other.HttpHandler.process(Compiled Code)
    at com.other.ClientThread.handleClient(ClientThread.java:42)
    at com.other.ClientThread.run(Compiled Code)
    at java.lang.Thread.run(Thread.java:459)
```

- make sure that you have spelled the driver name correctly and that there are no extra spaces before/after the name

7.4 ClassNotFoundException

[Wed Jul 21 15:31:10 CDT 2004] - Exception java.lang.ClassNotFoundException:
com.mysql.jdbc.Driver

```
at java.net.URLClassLoader$1.run(Unknown Source)
at java.security.AccessController.doPrivileged(Native Method)
at java.net.URLClassLoader.findClass(Unknown Source)
at java.lang.ClassLoader.loadClass(Unknown Source)
at sun.misc.Launcher$AppClassLoader.loadClass(Unknown Source)
at java.lang.ClassLoader.loadClass(Unknown Source)
at java.lang.ClassLoader.loadClassInternal(Unknown Source)
at java.lang.Class.forName0(Native Method)
at java.lang.Class.forName(Unknown Source)
at alcea.db.DatabaseHelper.getConnection(Unknown Source)
at alcea.db.DatabaseHelper.test(Unknown Source)
at alcea.db.AdminDatabase.process(Unknown Source)
at com.other.k.a(Unknown Source)
at com.other.k.a(Unknown Source)
at com.other.k.if(Unknown Source)
at com.other.r.if(Unknown Source)
at com.other.r.run(Unknown Source)
at java.lang.Thread.run(Unknown Source)
```

- you probably have typed a path wrong, or you have not added the driver jar file to the FBT installation directory.

7.5 java.sql.SQLException: [Microsoft][SQLServer 2000 Driver for JDBC]Can't start a cloned connection while in manual transaction mode.

Some users have complained of this exception after upgrading to Version 6.0. The fix is to add 'SelectMethod=cursor;' to the DB URL:

DB Driver: com.microsoft.jdbc.sqlserver.SQLServerDriver

DB URL:

jdbc:microsoft:sqlserver://edmsqlp1:1433;SelectMethod=cursor;databaseName=FIT

7.6 java.sql.SQLExceptionUUUUU: [Microsoft][SQLServer 2000 Driver for JDBC]The year, 5, is outside the range allowed by the SQL Server.

You have an issue which contains a date field that can not be stored in MS SQL. You will have to locate the issue and fix the date prior to the migration. You should see a line like this in the exception trail indicating the troubled issue.

Problem with issue : 730

7.7 No suitable Driver

[Fri Jul 20 11:11:13 EDT 2007] - Exception

java.sql.SQLException: No suitable driver

at java.sql.DriverManager.getConnection(Unknown Source)

at java.sql.DriverManager.getConnection(Unknown Source)

at alcea.db.DatabaseHelper.forceReconnectIfRequired(DatabaseHelper.java:172)

...

This exception was seen when the “jdbc:” portion of the JDBC URL was left out.

7.8 Cannot Start a Cloned Connection While in Manual Transaction Mode

While using the Microsoft SQL Server 2000 Driver for JDBC, you may experience the following exception:

java.sql.SQLException: [Microsoft][SQLServer 2000 Driver for JDBC]Can't start a cloned connection while in manual transaction mode.

Simply add “;SelectMethod=Cursor” to the end of your JDBC URL