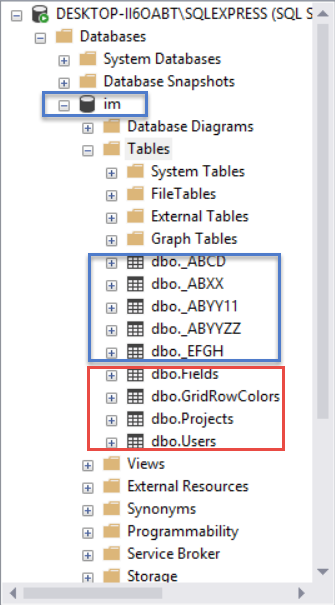
**IssueTracker Phase 1 Design Notes**

In this document, we will discuss:

1. Database
2. Logging On
3. Building project tree
4. Column\_Name to FieldName Mapping
5. What fields go to the Issue Grid
6. Coloring the rows on the Gird
7. What fields go to the Preview Pane
8. Cursors and Project Name Label
9. **Database**

We use SQL Server or SQL Server Express as the data store. The data.sql will create a test database named ‘im’ with 9 tables:



The top 5 tables are project tables (application data), the bottom 4 tables are overhead tables used by the program internally. We will discuss the usage of these 4 bottom tables throughout this document:

[im].[dbo].[Users] // User ID table

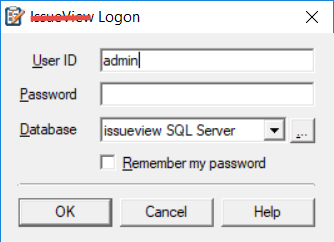
[im].[dbo].[Projects] // Project information table

[im].[dbo].[Fields] // FieldName cross-reference table

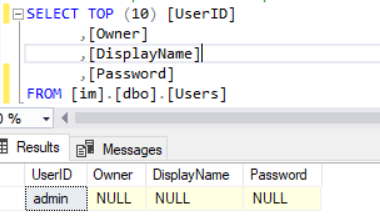
[im].[dbo].[GridRowColors] // Row Color rules table

1. **Logging On**

To log onto IssueTracker, the user must have an ID and a password. In Phase 1, only ‘admin’ is allowed and no password is needed.



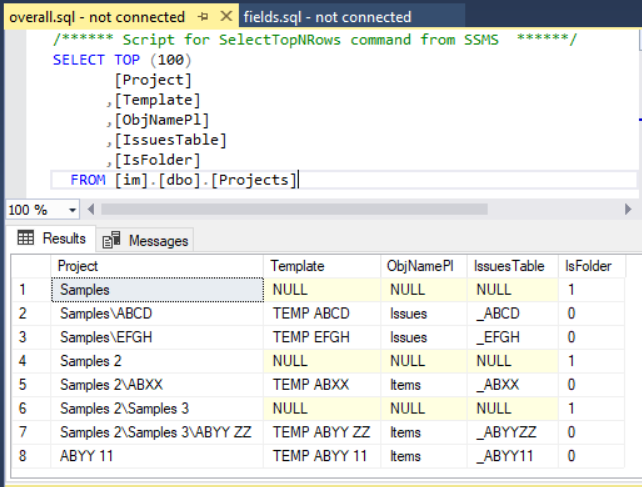
The ID entered by the user will be used to check against the [dbo].[Users] table entries for a match.



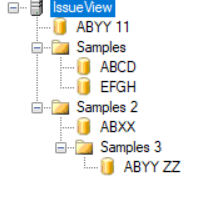
1. **Building the Project Tree**

There are many fields defined in the [dbo].[Projects] table. We are listing 5 fields that will be used in Phase 1:

* Project: Path information for the tree node
* Template: Template name used by the project
* ObjNamePl: Issue type for this project, it could be Items, Tasks, or Issues etc.
* IsFolder: Is this entry a folder or a project
* IssueTable: Table name used to implement the project



Using Project and IsFolder fields, the project tree can be built:



1. **Column\_Name to FieldName Mapping**

In IssueTracker, we will use a more user-friendly display name to represent the default Column\_Name. We call this display name FieldName. As an example, we prefer using ‘Updated By’ over ‘UpdatedBy’. The FieldNames are defined in a centralized table [dbo].[Fields] for all projects. To get the FieldName for a given Column\_Name, we need to go thru a lookup process. The lookup process is a four-step process:

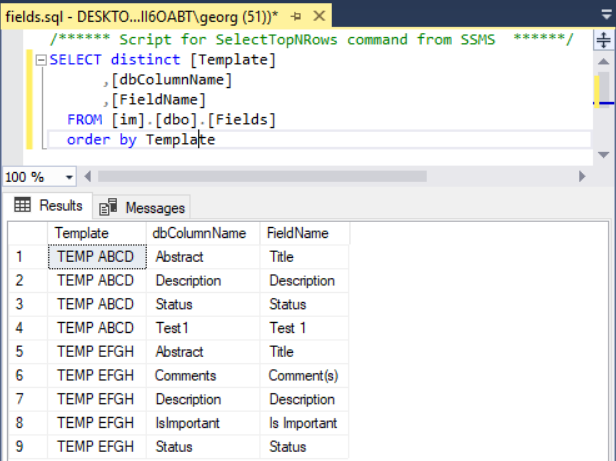
**Step 1**: If the Column\_Name is one of the three system generated fields, listed below, we will exit the lookup process because they are internal fields which are not suitable for displaying:

1. KeyID
2. Issue
3. LinkHidden

**Step 2**: If the Column\_Name is one of the eight system generated fields, listed below, we will map it using the table below:

|  |  |
| --- | --- |
| Column\_Name | FieldName |
| ID | ID |
| Owner | Owner |
| CreateDate | Created |
| LastUpdate | Updated |
| UpdatedBy | Updated By |
| Attachments | Use Paper clip icon |
| Links | Use Link icon |
| SCMFiles | SCMFiles |

**Step 3**: If we reach this step, the Column\_Name is a user-defined field which is suitable for displaying. The association relationship between Column\_Name and FieldName is defined in [dbo].[Fields] table:



We will use an example to illustrate how to find the FieldName for Column\_Name Test1 of table [dbo].[\_EFGH]:

First, we need to find the template used by \_EFGH table:

**SELECT [Template]**

**FROM [im].[dbo].[Projects]**

**WHERE IssueTable = ‘\_EFGH’**

**Return 🡪 ‘TEMP EFGH’**

then using the returned template name and Column\_Name to find FieldName:

**SELECT DISTINCT [FieldName]**

**FROM [im].[dbo].[Fields]**

**WHERE Template = ‘TEMP EFGH’ AND dbColumnName = ‘Test1’**

**Returns 🡪 ‘Test 1’**

**Step 4**: If the returned FieldName contains a ‘&’, we will strip it off. As an example, we will convert ‘Stat&us’ to ‘Status’ or ‘Tes&t 1’ to ‘Test 1’.

1. **What fields go to the Issue Grid**

Because long text fields are not suitable to display in the Grid, we will first filter out those fields with DATA\_TYPE = ‘ntext’ from the entire Column\_Name list:

**SELECT COLUMN\_NAME**

**FROM INFORMATION\_SCHEMA.COLUMNS**

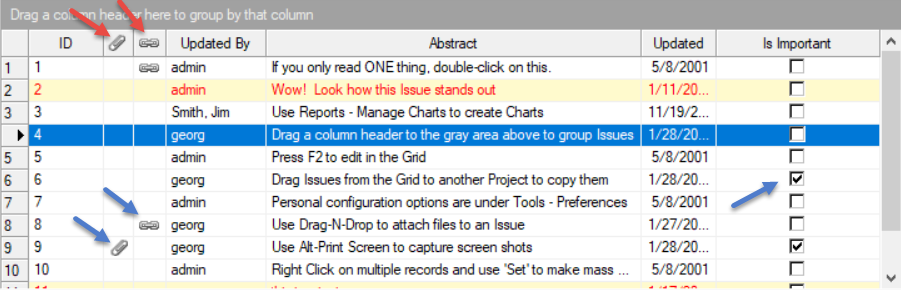
**WHERE TABLE\_NAME = issuetablename AND DATA\_TYPE <> ‘ntext’**

Then, we will send the returned list to undergo the lookup process described in item #4. The resulting fields are candidates suitable for displaying on the Grid. The user can use the Grid Column selection box to select the fields that he/she wants to display.

One point to note is that the program will display icons instead of data for three types of fields:

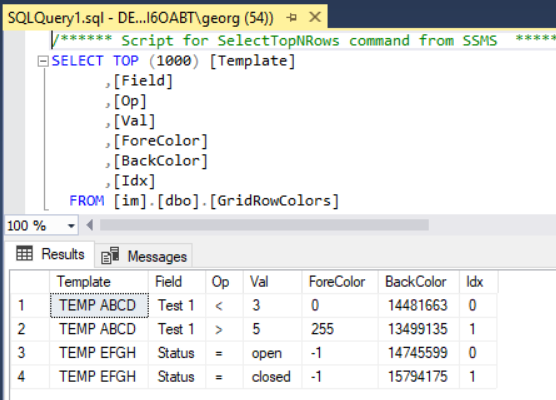
1. Attachments: display paper clip icon if the Attachments quantity > 0
2. Links: display link icon if the Links quantity number > 0
3. Fields where daya\_type = bit: display checked icon or unchecked icon

**Warning**: The data type of Links and Attachments could be declared as int [null], int [not null], float [null], or float [not null]. When testing the quantity of these fields, the code needs to be robust enough to handle all contingencies to avoid exception throwing. The same applies to the bit fields testing, the data type declaration for the field could be bit [null] or bit [not null].



1. **Coloring the rows on the Grid**

We will color the row as needed while populating the Grid. The rules to color the row are defined in [dbo].[GridRowColors]:



Here is an example showing how to apply the coloring rules (refer to the rules above):

If the row containing field ‘Test 1’ and the value of ‘Test 1’ is 2 which satisfies the (‘Test 1’ < 3) condition, then we will color the row foreground 0, background 14481663. Four comparison operators (Op) will be supported: equal to =, not equal <>, greater than >, and less than <.

1. **What fields go to the Preview Pane**

The preview pane contains a header and the body (see below picture). The Header is part of the preview pane, not a separate pane. The background of the header will be colored.

We will display 5 fields (ID, CreatedDate, UpdatedBy, LastUpdate, and Abstract) on the header. Yes, we will map them from Column\_Name to FieldNames before displaying.

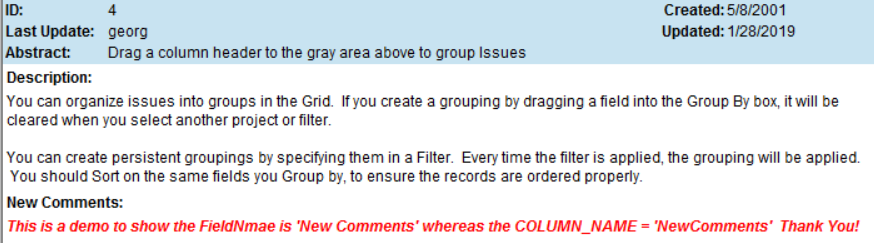
The body of the preview pane will display all fields with DATA\_TYPE = ‘ntext’ except the ‘History’ field:

**SELECT COLUMN\_NAME**

**FROM INFORMATION\_SCHEMA.COLUMNS**

**WHERE TABLE\_NAME = issuetablename AND DATA\_TYPE = ‘ntext’ AND COLUMN\_NAME <> ‘History’**

Yes, we will also map the Column\_Names to FieldNames for displaying, for example, from ‘NewComments’ to ‘New Comments’.



1. **Cursors and Project Name Label**

The cursor for the project tree or the issue Grid should display blue background color when the project name or the Grid row is clicked. The cursor color should change to light gray when the mouse is out of focus (as indicated by the blue arrows below).

The ‘Items in Knowledge Base’ label (as indicated by the red box) is derived from the ObjNamePl and Project fields defined in [dbo].[Projects] table.

