

# Adding Custom Site Settings

## Document History

Version	Date	Name	Comments
1.0	2012-07-21	Mark Anderson	Initial Version

# Contents

Introduction .....	1
SQL.....	2
Creating the Dummy Application .....	2
Adding Settings .....	2
Inserting Multiple Settings .....	5
Retrieving Settings .....	6

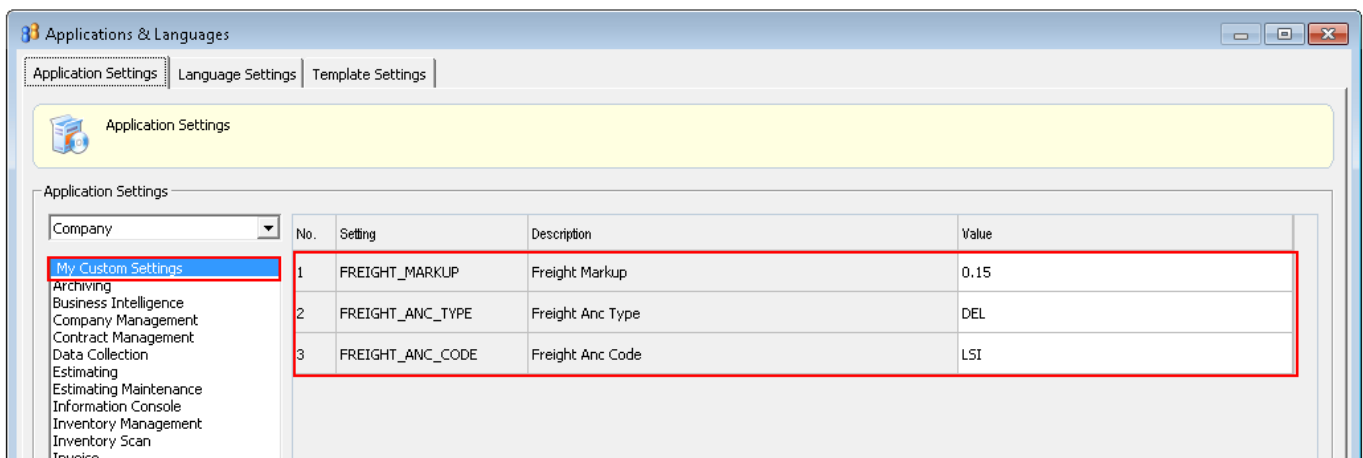
## Introduction

Quite often site-wide variables are required for reporting, custom integrations, or other purposes (e.g. Freight markups). (Customer-specific settings can be stored in the CRM custom fields.)

These often get hard-coded in reports and stored procedures. A better practice is to create a custom table for maintaining these in a single location, so that dozens of stored procedures or reports don't need to be changed when a value changes. That said, this method is also limited, as it requires the user who maintains these to have access to SQL Server Management Studio (or some other UI). For example, the CFO (FD) wants to be able to control the freight markup without going to IT to change it.

In Technique, it's fairly simple to add a set of custom settings that can be edited through InfoConsole. This involves creating a dummy app (whose ID must be less than 150) and then adding settings. The result is a set of app settings specific to a site.

Below is an example:



The screenshot shows the 'Applications & Languages' window with the 'Application Settings' tab selected. A table of settings is displayed for the 'My Custom Settings' application. The table has four columns: 'No.', 'Setting', 'Description', and 'Value'. Three settings are listed:

No.	Setting	Description	Value
1	FREIGHT_MARKUP	Freight Markup	0.15
2	FREIGHT Anc TYPE	Freight Anc Type	DEL
3	FREIGHT Anc CODE	Freight Anc Code	LSI

## SQL

### Creating the Dummy Application

The following statement will create the dummy app, which will cause a new item to show up in application settings in InfoConsole (once a setting is added):

```
-- Create Dummy App for the Settings (run once only)
INSERT INTO [QA_Customers].[dbo].[ApplicationDetails]
( [AppID]
, [AppName]
, [AppVersion]
, [AppVersionDate]
, [AppStartUp]
, [AppConsole]
, [Version] )
VALUES
( 149 -- Must be <150 (Otherwise InfoConsole crashes)
, ' My Customs Settings' -- Start with spave to get to top of list
, 1.0
, NULL
, 'Dummy'
, 0
, 0 )
```

### Adding Settings

The following statement adds a setting for Freight Markup:

```
-- Insert items
-- Get next free SettingID for this App and Site
DECLARE @SettingID AS INT
SET @SettingID = (
    SELECT
        MAX(SettingID) + 1
    FROM
        [QA_Customers].[dbo].[ApplicationSettings]
    WHERE
        AppID = 149
        AND SiteID = 1
)

INSERT INTO [QA_Customers].[dbo].[ApplicationSettings]
(
    [AppID]
, [SettingID]
, [SiteID]
, [Name]
, [Value]
, [Description] )
VALUES
(
    149
, @SettingID
, 1
, 'FREIGHT_MARKUP'
, '0.15'
, 'Freight Markup (enter decimal value)' )
```

The following statement adds a setting for the ancillary type for freight (used by a custom integration to automatically insert freight costs):

```
-- Get next free SettingID for this App and Site
DECLARE @SettingID AS INT
SET @SettingID = (
    SELECT
        MAX(SettingID) + 1
    FROM
        [QA_Customers].[dbo].[ApplicationSettings]
    WHERE
        AppID = 149
        AND SiteID = 1
)

INSERT INTO [QA_Customers].[dbo].[ApplicationSettings]
(
    [AppID]
, [SettingID]
, [SiteID]
, [Name]
, [Value]
, [Description] )
VALUES
(
    149
, @SettingID
, 1
, 'FREIGHT_ANC_TYPE'
, 'DEL'
, 'Freight Anc Type' )
```

The following statement adds a setting for the ancillary code for freight (used by a custom integration to automatically insert freight costs):

```
-- Get next free SettingID for this App and Site
DECLARE @SettingID AS INT
SET @SettingID = (
    SELECT
        MAX(SettingID) + 1
    FROM
        [QA_Customers].[dbo].[ApplicationSettings]
    WHERE
        AppID = 149
        AND SiteID = 1
)

INSERT INTO [QA_Customers].[dbo].[ApplicationSettings]
(
    [AppID]
, [SettingID]
, [SiteID]
, [Name]
, [Value]
, [Description] )
VALUES
(
    149
, @SettingID
, 1
, 'FREIGHT_ANC_CODE'
, 'DAS'
, 'Freight Anc Code' )
```

## Inserting Multiple Settings

The following statement inserts all three settings at once:

```
DECLARE @SettingID AS INT
SET @SettingID = (
    SELECT
        MAX(SettingID) + 1
    FROM
        [QA_Customers].[dbo].[ApplicationSettings]
    WHERE
        AppID = 149
        AND SiteID = 1
)
```

-- Remember to get/set new setting ID for each select statement

-- Make sure column order matches table

```
INSERT INTO [QA_Customers].[dbo].[ApplicationSettings]
SELECT
    149
    ,@SettingID
    ,1
    ,'FREIGHT_MARKUP'
    ,'0.15'
    ,'Freight Markup'
UNION
SELECT
    149
    ,@SettingID + 1
    ,1
    ,'FREIGHT_ANC_TYPE'
    ,'DEL'
    ,'Freight Anc Type'
UNION
SELECT
    149
    ,@SettingID + 2
    ,1
    ,'FREIGHT_ANC_CODE'
    ,'DAS'
    ,'Freight Anc Code'
```



## Retrieving Settings

Below is an example of retrieving the value of the Freight Markup setting:

```
-- Example of getting settings
-- Return 0 if setting not present or is NULL
-- Remember to cast to correct type as all settings are varchars
DECLARE @Msg AS VARCHAR(MAX)
DECLARE @FreightMarkup AS FLOAT
SET @FreightMarkup = (
    SELECT
        ISNULL(CAST(Value AS FLOAT), 0.0)
    FROM
        QA_Customers.dbo.ApplicationSettings
    WHERE
        AppID = 149
        AND SiteID = 1
        AND Name = 'FREIGHT_MARKUP'
)

-- Basic error handling
-- Check if we got a valid freight markup type
-- Log error but don't throw exception, as we have a default
IF @FreightMarkup = 0.0 OR @FreightMarkup IS NULL
    BEGIN
        SET @Msg = 'Freight Markup is 0.0. Ancillaries will be inserted with no
            markup.'
        RAISERROR (@Msg,10,1) WITH LOG -- Customize your error handling
    END
```