
FM Toolkit Installation Instructions

Dean Evans & Associates, Inc.



Copyright

Copyright © 2012 Dean Evans & Associates, Inc. All rights reserved.

No part of this document may be reproduced, transmitted or stored in a retrieval system in any form, electronic or mechanical, for purposes other than the personal use of the original purchaser except by written permission of Dean Evans & Associates, Inc.

Dean Evans & Associates, Inc.
6465 Greenwood Plaza Boulevard
Suite 600
Centennial, CO 80111

This document may be copied for use by individuals employed by the purchaser.

Information in this document is subject to change without notice.

EMS, the EMS logo and EMS Regics are registered trademarks of Dean Evans & Associates, Inc. Event Management Systems, EMS Enterprise, EMS Client, EMS Lite, EMS Campus, EMS Workplace, EMS Legal, EMS District, EMS Master Calendar and Virtual EMS are trademarks of Dean Evans & Associates, Inc. Microsoft, Windows and Outlook are registered trademarks and SQL Server is a trademark of Microsoft Corporation. Lotus Notes and Domino are registered trademarks of IBM Corporation. Other products, brands and trademarks are property of their respective owners/companies.

FM Toolkit Installation Instructions

INTRODUCTION4

CUSTOMER SUPPORT.....4

OBTAINING THE FM TOOLKIT COMPONENTS4

FM TOOLKIT OVERVIEW.....4

THE EMS_STAGING DATABASE.....4

MAINTAINING ROOM RECORDS IN EMS5

Adding New Records in EMS.....6

Updates to Existing Records in EMS.....6

Inactive Room Records6

INSTALLING AND EXECUTING THE FM TOOLKIT6

INSTALLATION.....6

EXECUTION6

Introduction

EMS contains an inventory of buildings and rooms which is the basis for meeting and event reservations. Facilities management (FM) systems are also built upon an inventory of buildings and rooms. The FM Toolkit provides a means of synchronizing the space inventory between these two systems.

This document lists the steps you must take to install the FM Toolkit in your environment. If you are unsure whether or not your organization is licensed for the FM Toolkit, or if you would like to learn more about it, please contact your Account Executive.

Customer Support

Unlimited toll-free customer support is available to EMS users who have a current Annual Service Agreement (ASA). Please contact Customer Support, or a member of the Client Services Team if you are working with one, if you have any questions or encounter any issues implementing the FM Toolkit.

Email:	support@dea.com
Web:	www.dea.com
Phone:	(800) 288-4565
Fax:	(303) 796-7429

Please note that support for the FM Toolkit extends only to installing the EMS components outlined below. For help identifying the appropriate FM data source within your environment or defining/creating a process to import data into the *EMS_Staging* database, please consult your internal IT Department.

Obtaining the FM Toolkit Components

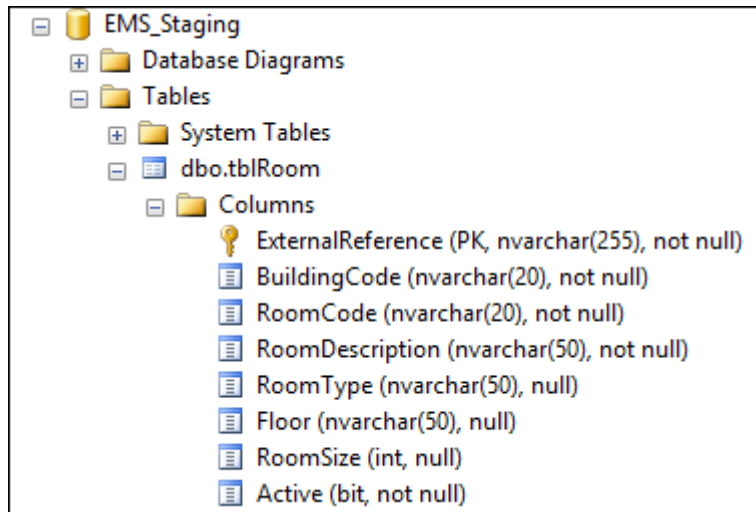
Please contact your Client Services Consultant to obtain the following components:

- **FMTK Staging DB Install.sql** – SQL file required to create the staging database – *EMS_Staging*.
- **FMTK Update Script.sql** – SQL file required to create the stored procedure *FMTK_Update_Rooms* within your EMS database.

FM Toolkit Overview

The EMS_Staging Database

The *EMS_Staging* database (created by **FMTK Staging DB Install.sql**) is used to stage room records before they are imported into EMS.



Once you have identified the FM data source within your organization that will be used to create and maintain your EMS room records, you will be required to define/create a process to import room records into the *tblRoom* table within *EMS_Staging*.

Maintaining Room Records in EMS

The *FMTK_Update_Room* stored procedure (created by **FMTK Update Script.sql**) contains all of the logic to create your EMS room data. Records are imported into EMS from the *EMS_Staging* database according to the specifications outlined in the table below:

<i>EMS_Staging.tblRoom</i> Field	<i>EMS.tblRoom</i> Field	Notes
ExternalReference	ExternalReference	Required. Primary Key. Unique identifier that links a room record in your FM system with a room record in EMS.
BuildingCode	BuildingID	BuildingCode is matched against <i>tblBuilding.BuildingCode</i> to derive the correct Building ID to be stored in <i>tblRoom.BuildingID</i> . If BuildingCode can't be matched in <i>tblBuilding</i> , the room record WILL NOT be created in EMS.
RoomCode	Room	Required
RoomDescription	Description	Required
RoomType	RoomType	RoomType is matched against <i>tblCategoryGroup.Description</i> (Room Type table) to derive the correct Room Type ID to be stored in <i>tblRoom.RoomType</i> . If the room type can't be matched in <i>tblCategoryGroup</i> , a new room type record WILL be created.
Floor	FloorID	Floor is matched against <i>tblFloor.Description</i> to derive the correct Floor ID to be stored in <i>tblRoom.FloorID</i> . If the floor can't be matched in <i>tblFloor</i> , a new floor record WILL be created.
RoomSize	RoomSize	Optional
Active	Active	Active = 1. Inactive = 0.

The FM Toolkit uses the *ExternalReference* field in *EMS_Staging.tblRoom* and the *tblRoom.ExternalReference* field in the EMS database to determine whether a record needs to be added, updated, or inactivated.

Adding New Records in EMS

Rooms: If the *ExternalReference* value in *EMS_Staging.tblRoom* is not found on an existing room record in EMS, a new room will be created.

Important Note: Existing EMS clients wanting to implement the FM Toolkit must “synchronize” existing EMS rooms before activating the toolkit by adding a valid *ExternalReference* value in *tblRoom*. **Ignoring this step will result in duplicate records.**

Room Types: If the *RoomType* value in *EMS_Staging.tblRoom* is not found in the existing list of room types in EMS, a new room type will be created.

Floors: If the *Floor* value for a particular room record in *EMS_Staging.tblRoom* is not found in the existing list of floors in EMS, a new floor will be created.

Buildings: The FM Toolkit does not add or maintain building records in EMS.

Updates to Existing Records in EMS

If the *ExternalReference* value in *EMS_Staging.tblRoom* is found on an existing room record in EMS, the room will be updated with any changes.

Inactive Room Records

A room will be inactivated if one of the two conditions outlined below is met:

- 1) If an existing room record in EMS does not exist in *EMS_Staging.tblRoom* (based on *ExternalReference*).
- 2) If a room record's **Active** status in *EMS_Staging.tblRoom* = 0.

Note: Room records within EMS with an empty External Reference field are excluded from processing. Therefore, rooms that are not included in your data feed can be added and maintained manually within EMS and will not be impacted by the FM Toolkit.

Installing and Executing the FM Toolkit

Installation

The following steps should be performed on the MS SQL Server that hosts your EMS database.

1. Obtain the SQL files outlined in the [Obtaining the FM Toolkit Components](#) section above.
2. Using *Microsoft SQL Server Management Studio* execute the **FMTK Staging DB Install.sql** script against your **master** database. This will create the *EMS_Staging* database containing one table named *tblRoom*.
3. Using *Microsoft SQL Server Management Studio* execute the **FMTK Update Script.sql** script against your **EMS** database. This will create a stored procedure in your EMS database named *FMTK_Update_Room*.

Execution

1. Import records from your organization's facility management system into the *tblRoom* table within the *EMS_Staging* database.

Important Note: Existing EMS clients wanting to implement the FM Toolkit must “synchronize” existing EMS rooms before activating the toolkit by adding a valid *ExternalReference* value in *tblRoom*. **Ignoring this step will result in duplicate records.**

2. Using *Microsoft SQL Server Management Studio* execute the *FMTK_Update_Room* stored procedure.

Note: Step 2 (and Step 1 with assistance from your IT Department) can be scheduled to execute automatically using a SQL Job. Please refer to your MS SQL Server documentation for more information on creating and scheduling a SQL Job.