



FLM Installation Guide

Release 270

Version History

Version	Date	Author(s)	Reason for update
1	01/10/2007	Andy Gardner	Initial version
2		Andy Gardner	Update for CMS / User group / FLM Portal
3		Emily Burfoot	Screenshots added
4	02/06/2008	Mitchell Hagerty	271 Updates

Table of Contents

Introduction	3
1.1 Prerequisites for installing FLM	3
1.2 Java Prerequisites.....	3
1.3 Preparation for installing FLM	4
2 Performing the installation of FLM	5
2.1 SAINT	5
2.2 SGEN.....	5
3 Post Installation of FLM	6
3.1 Adobe ReaderRights Credential	6
3.2 Namespace /FLM/ and Software Component FLM	6
3.3 Packages /FLM/CUST and ZFLM.....	6
3.3.1 /FLM/CUST	6
3.3.2 ZFLM	7
3.4 Content Management Physical Tables	7
3.5 Number Range /FLM/TID.....	7
3.6 SICF Settings	8
3.7 Define Content Repositories.....	8
3.8 Define and Assigne Content Categories.....	10
3.9 Link Document Types	11
4 Setting Up Authorisation Checks For FLM Forms	11
4.1 Overview	12
4.2 Authorization Fields	12
4.3 Authorization Object	13
4.4 Roles	15
4.4.1 Creating an FLM Admin Role	15
4.4.2 Creating Form User Roles.....	18
4.5 User Masters Roles	19
4.6 Offline User	21
4.7 FLM Developer User Role and Authorisations	21
5 Deploying the FLM Portal	24
5.1 Introduction	21
5.2 Enterprise Archive	22
5.2.1 SDM Procedure.....	22
5.2.2 Java Connectors.....	22

Introduction

This document describes the installation process for Forms Lifecycle Manager release 260.

1.1 Prerequisites for installing FLM

- Current versions of kernel, tp and R3trans

Make sure that your system contains the current versions of kernel, tp and R3trans.

- Current SPAM / SAINT Update

Make sure that you have imported the most recent SPAM / SAINT Update into your system. Compare the short text of the last SPAM / SAINT Update you imported with that of the SPAM / SAINT Update in the SAP Service Marketplace. If the version of the SPAM / SAINT Update in the SAP Service Marketplace is more recent, import it.

- Before starting the installation, read the following SAP Notes and documentation:

Description	Note
Problems w/add-on inst/upgrade to SAP NW 7.0s ABAP	822380
Known Problems with Support Packages in SAP NW 2004s AS ABAP	822379

- The following software components are required in your system:

SAP_ABA	700
SAP_BASIS	700

1.2 Java Prerequisites

For the on-line scenario you also require a Java Component, the "FLM Portal", which resides on the SAP WebAS Java stack 700.

1.3 Preparation for installing FLM

- Mount the Add-On Installation CD.
- Unpack the archive FLM_INST_<VER>.SAR in the subdirectory DATA using the command:

```
sapcar -xvf FLM_INST_<VER>.SAR
```

- From the resulting EPS/IN directory, copy the E5Q0020246584_*.PAT file into your development system EPS inbox directory:

```
//sapmnt/trans/EPS/in
```

- Ensure that the permissions of the file are set for read/write access for the SAP O/S users.

For more information about this, see the online documentation for Add-On Installation Tool. To do this, choose the help function in the application toolbar and navigate to *Online Docu* → *Loading Installation Packages*.

2 Performing the installation of FLM

This section describes the installation procedure for the software.

2.1 SAINT

1. Log on to your SAP system as client 000 and as a user that has SAP_ALL authorization. Do not use the user SAP* or DDIC.
2. Start the installation or upgrade using Add-On Installation Tool (transaction SAINT):

For more information about this, see the online documentation for Add-On Installation Tool. To do this, choose the help function on the toolbar.

2.2 SGEN

At this point you should compile FLM using the SGEN transaction.

1. Transaction SGE N.
2. Select the first option: Generate All Objects of Selected - Continue
3. Select for Parallel Generation: Leave default. - Continue
4. Start Job Directly.
5. Generation speed depends on your system but should take under 10 minutes.

3 Post Installation of FLM

Installation should be done on the Client that will be using FLM so the objects created below transport properly to QA and Customer Systems. We suggest creating a new transport when prompted named "FLM <VER> Installation <CLNT>".

3.1 Adobe ReaderRights Credential

In order for ADS to be able to render pdf forms which are interactive, it is required to install into each Java stack an 'Adobe Credential for ReaderRights'. This is a piece of Adobe intellectual property delivered via the SAP OSS system in the form a small file. It is installed into the WebAS java stack and then configured using the 'Visual Administrator'. Only one Credential is required per customer.

The process is described in SAP Note 736902.

3.2 Namespace /FLM/ and Software Component FLM

During the process of forms creation the FLM framework generates Data dictionary objects in the namespace /FLM/. Therefore this namespace must be set to 'modifiable'. Process:

1. Transaction SE03 [ess-ee-zero-three].
2. Choose 'Set System Change Option'.
3. Navigate in the 'Software Component' Table to FLM.
4. Choose 'Modifiable'.
5. Navigate in the 'Namespace/Name range' Table to /FLM/.
6. Choose 'Modifiable'.
7. Save the settings.

3.3 Packages /FLM/CUST and ZFLM

Two packages are important for the maintenance of the FLM system, /FLM/CUST and ZFLM.

3.3.1 /FLM/CUST

This package is used to hold and transport the business logic associated with customer forms and is delivered empty. Certain aspects of form business logic has to be generated by the FLM framework in the /FLM/ namespace, and hence an /FLM/ package can only be used to hold these objects.

After the initial installation, this package should be repaired in transaction SE21 to change the transport layer to the customer's standard transport layer, such that objects in this package can be successfully migrated through the customer landscape.

3.3.2 ZFLM

This package is used to hold objects that cannot be maintained in the /FLM/ namespace, eg authorisation objects, such that they can be migrated through the customer's landscape. The package is created in transaction SE21.

6. Transaction SE21.
7. Package will be ZFLM and select Create.
8. Short Description: FLM: Authorization Objects
9. Application Component: CA
10. Software Component: HOME
11. Transport Layer: Customer Specific, Default SAP.
12. Save

3.4 *Content Management Physical Tables*

FLM uses the 'Content Management System' [CMS] or 'Content Server' to store form and form templates. The setup of CMS is SAP standard and described fully in the IMG documentation accordingly.

The standard FLM installation scenario makes use of SAP DDIC physical tables as the final repository for CMS data, and under those circumstances it is necessary to create physical table for that purpose as follows:

13. Transaction SE11.
14. Select the radiobutton 'Tables' and enter table SDOKCONT1.
15. Copy this table to /FLM/SDOKCONT1, assigning package /FLM/CUST and workbench request as required.
16. Activate this table.

Ensure that the technical settings of the table match the expected size of the table, and adjust accordingly.

FLM supports any CMS repository types, consult the SAP documentation for a discussion of the advantages and disadvantages of the different repository types that can be utilized with this system.

The remaining FLM table configuration behind the CMS is described in the Implementation guide.

3.5 *Number Range /FLM/TID*

The FLM Transport ID (/FLM/TID) Number Range Object is delivered with FLM and a new interval range must be setup in it.

1. Transaction SNRO [es-en-ar-oh].
2. Enter number range object name (/FLM/TID).
3. Press Number Range button.
4. Press change Intervals.
5. Create an Interval. E.g 01 100000000 - 1999999999 - 1000000000
6. Press save.

3.6 SICF Settings

FLM makes use of a business server pages transaction in order to allow FLM form URLs to be sent as part of the reminder functionality in form routing. The BSP application must be activated in transaction SICF. Proceed as follows:

1. Transaction SICF.
2. Enter *FLM* into Service Name field
3. Press Execute
4. Right-click and select Activate all FLM nodes
5. Choose to Activate this node and all child nodes

Please note that there are certain standard BSP service nodes that must be activated as a prerequisite for using any BSP application; please consult the SAP documentation for details.

3.7 Define Content Repositories

In this Activity you can define content repositories to which contents can be moved.

In Knowledge Warehouse three different types of content repositories can be used, we suggest using the SAP System Database and we will highlight the steps for it. Possibilities include:

HTTP Content Server

Documents of different types are generally stored in an external Content Server with SAP HTTP Interface (SAP Content Server).

Structure Storage System

Structures are stored in the Structure Storage System of the SAP System.

Database

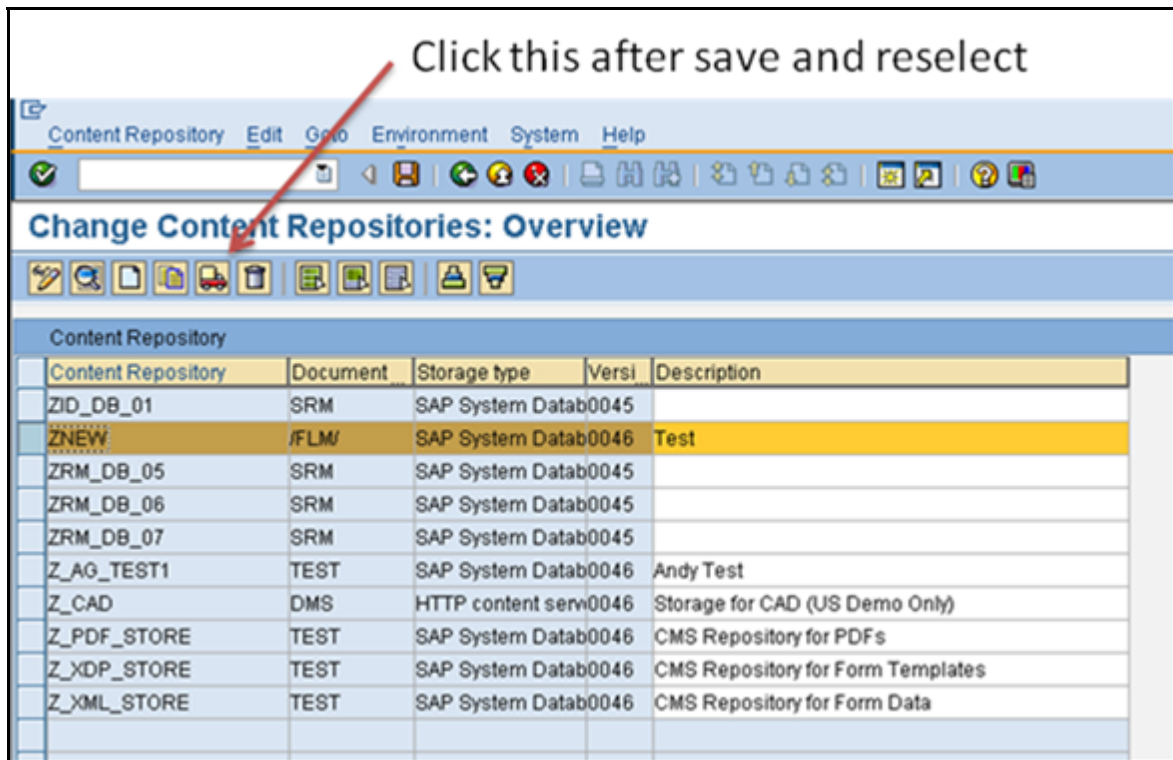
Document administration data, macro files and other utilities needed to use the Knowledge Warehouse are stored in the SAP System Database.

You can find further information on content repositories in the SAP Knowledge Warehouse Installation Guide and in the SAP Library under Basis - Basis Services - SAP Knowledge Provider - Content Management Service - Concepts.

Setup the Content Repository to use the SAP System Database.

1. Transaction OAC0 (OH-A-C-Zero).
2. Change - Create
3. Content Rep. : We suggest "ZFLM0001"
4. Description: FLM: Content Repository
5. Document Area: We suggest /FLM/ or Forms Lifecycle Manager
6. Storage type: SAP System Database
7. Version No. : 0046
8. Contents Table: /FLM/SDOKCONT1
9. Save

If you encounter problems with entering data at this step please see SAP NOTE: 1163925



3.8 Define and Assign Content Categories

In this activity you assign the storage categories of documents managed in the Knowledge Warehouse to content repositories (in the Knowledge Provider these are also called content categories).

Follow the steps below to create a new storage category and assign it to the content repository.

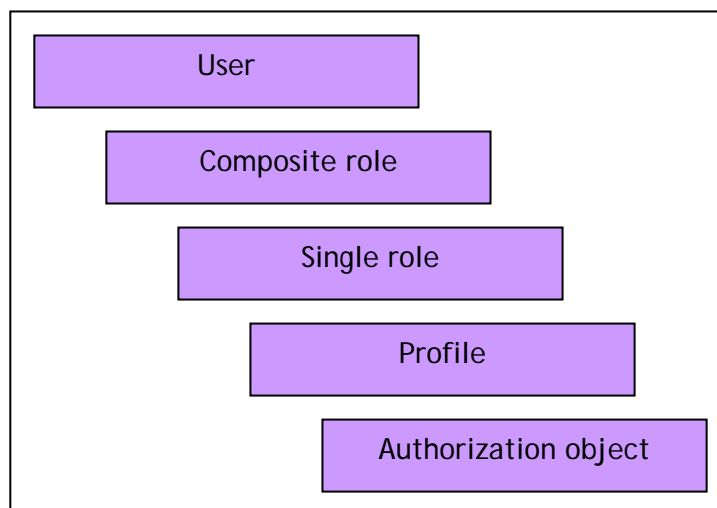
1. Transaction OACT (OH-A-C-T).
2. New Entries
3. Category: We suggest "ZFLM0001"
4. Description: FLM: Content Categories
5. Document Area: /FLM/
6. Content Repo: This is the repository created in the previous step, ZFLM0001
7. Save

4 Setting Up Authorisation Checks For FLM Forms

Unlike in previous releases of the software, FLM relies entirely on the user master record as defined in the ABAP stack. The UME of the java stack must have the ABAP stack set as it's data source in order to take advantage of this standard installation configuration.

Firstly, in order to control which users have access to which form categories and form types, an authorisation object must be created in the customer system and transported through the system landscape. Typically this would be in the customer namespace, we recommend for example, Z/FLM/0001.

4.1 Overview



In SAP, each user's authorization can be defined with reference to a composite role, single roles, and profiles. The object to be checked is an 'authorization object', and is made up of authorization fields that the system will need to use in the check. FLM is delivered with authorization checking code and the authorization fields, but the set-up of the users, roles, profiles and objects will be different in each implementation.

4.2 Authorization Fields

Three authorization fields are delivered as part of FLM:

- /FLM/CUST *Customer*
- /FLM/FTYPE *Form type*
- /FLM/FCAT *Form category*

These three authorization fields, along with the standard field ACTVT, must be included in the authorization object.

4.3 Authorization Object

FLM is not shipped with an authorization object. Part of the installation process involves the creation of authorization object 'z/flm/0001'.

Creating an Authorization Object

- Transaction SU21 or menu path 'Tools->ABAP Workbench->Development->Other tools->Authorization objects->Objects'
- Choose the pushbutton for create authorization object.
- Enter the following:
 - Object: 'z/flm/0001'
 - Text: 'FLM: Form posting authorization'
 - Class: 'AAAB'
 - Field name: /FLM/CUST
 - Field name: /FLM/FCAT
 - Field name: /FLM/FTYPE
 - Field name: ACTVT

Maintain the Authorization Objects

Regenerate SAP_ALL

Maintain authorization object

Object: Z/FLM/0001
 Text: FLM: Form posting authorization
 Class: AAAB Cross-application Authorization Objects
 Author: USER2

Authorization fields

Field name	Heading
/FLM/CUST	SFS: Customer Code
/FLM/FCAT	FLM: Form Category Code
/FLM/FTYPE	SFS: Form Type
ACTVT	Activity

Authorization Object Documentation

Create Object Documentation

Further authorization object settings

Conversion for authorization fields allowed

Permitted Activities

- o Select pushbutton to maintain the permitted activities.
- o Select options '01', '02', '03' and '10' only. Save,
- o Select pushbutton to create documentation, then save
- o Save the authorization object. You may get a message indicating it has already been saved, if so just cancel out of the maintenance window.
- o Finally we need to regenerate the SAP_ALL profile in order to include the new object. Use the pushbutton 'Re-generate SAP_ALL' on the top of the front screen for transaction RSU21_NEW.

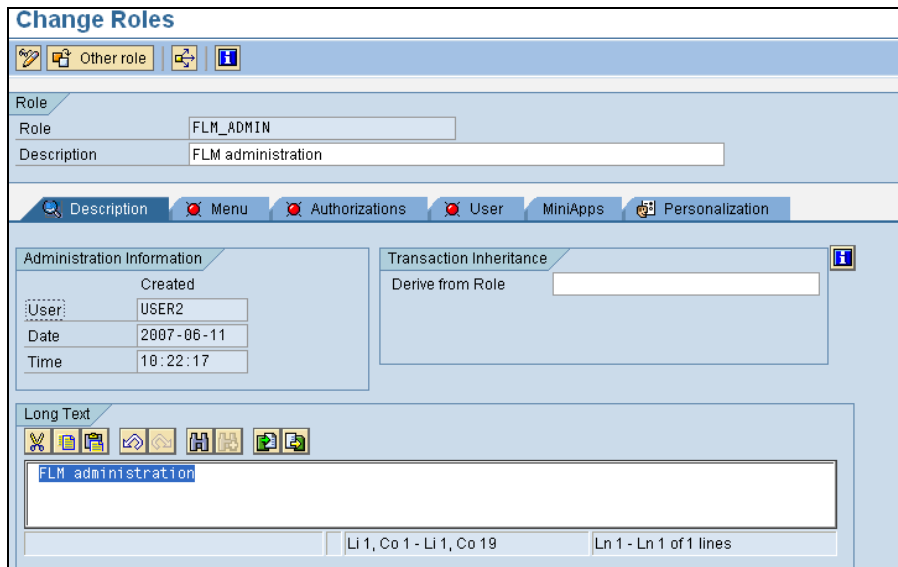
4.4 Roles

There are various ways of linking the authorization object to user ids using roles and profiles. In this section we describe one method, using a 'single role' and a 'generated profile'.

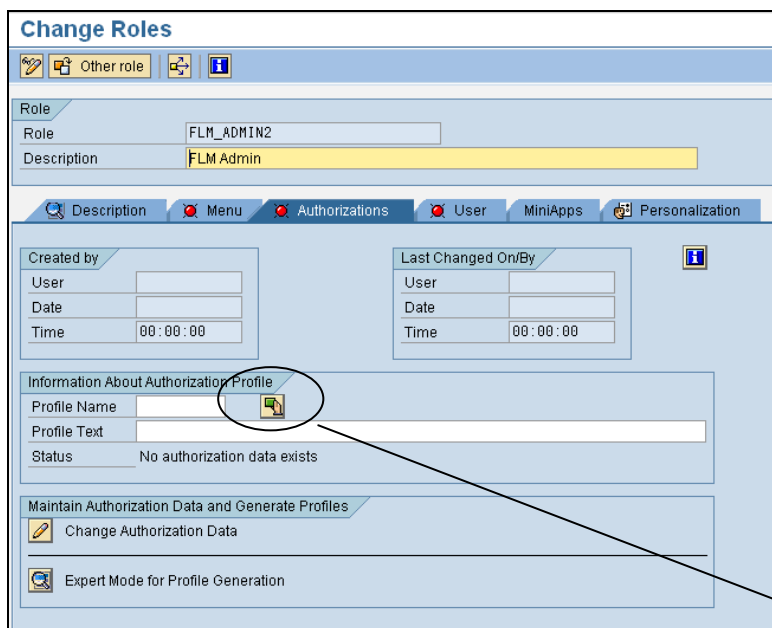
4.4.1 Creating an FLM Admin Role

We need one role for FLM administration

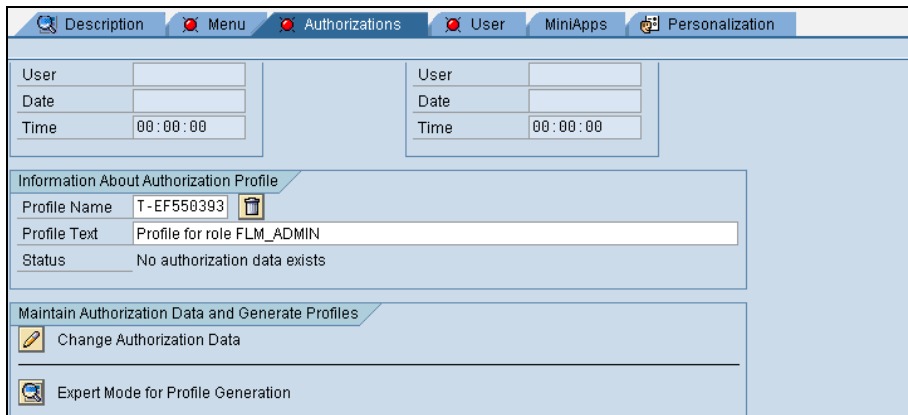
- Or transaction code PFCG or select the menu path 'Tools->Administration->User Maintenance->Role Administration->Roles.'
- Enter role name 'FLM_ADMIN' and choose the 'Single Role' pushbutton.
- Enter the role description then Save



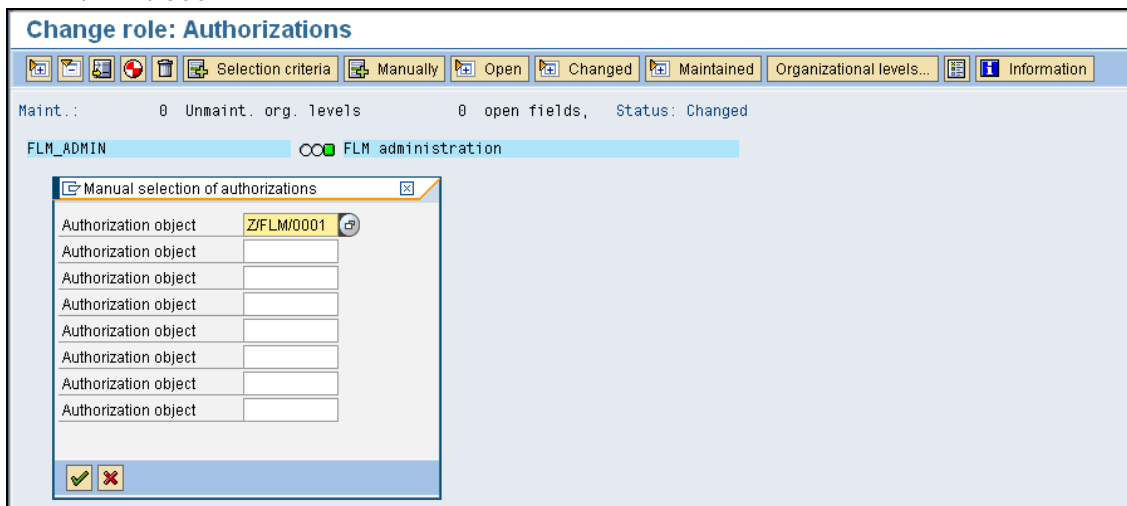
- Select the Authorizations tab



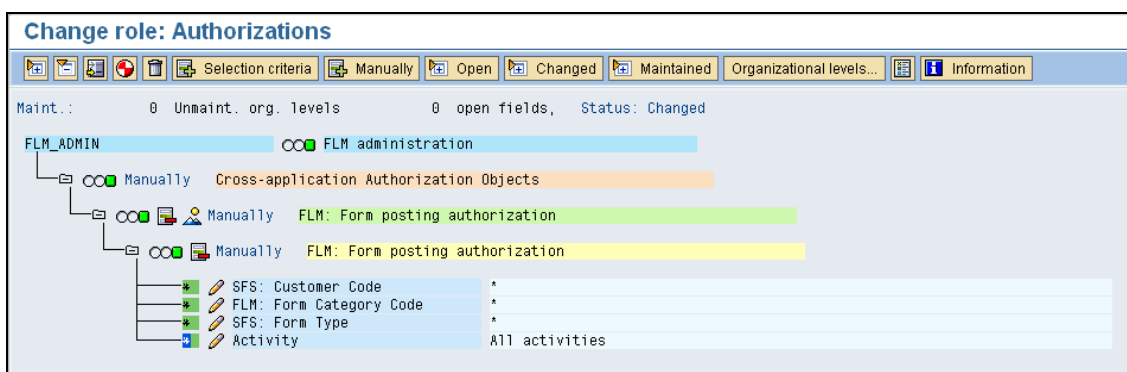
- In the 'Information About Authorisation Profile' box, select the 'Propose Profile Names' pushbutton, and a profile name is generated automatically. Save.



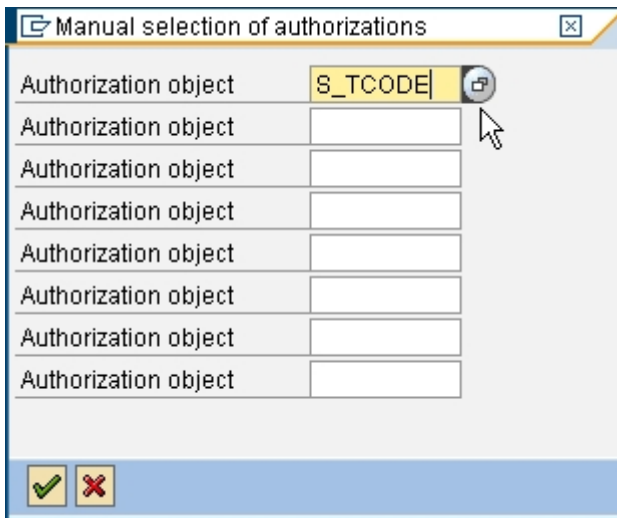
- Select the Change Authorization Data pushbutton. A pop-up window is displayed. Select the 'Do not select templates' pushbutton.
- Now select the 'Manually' pushbutton, and enter the authorization object 'Z/FLM/0001'.



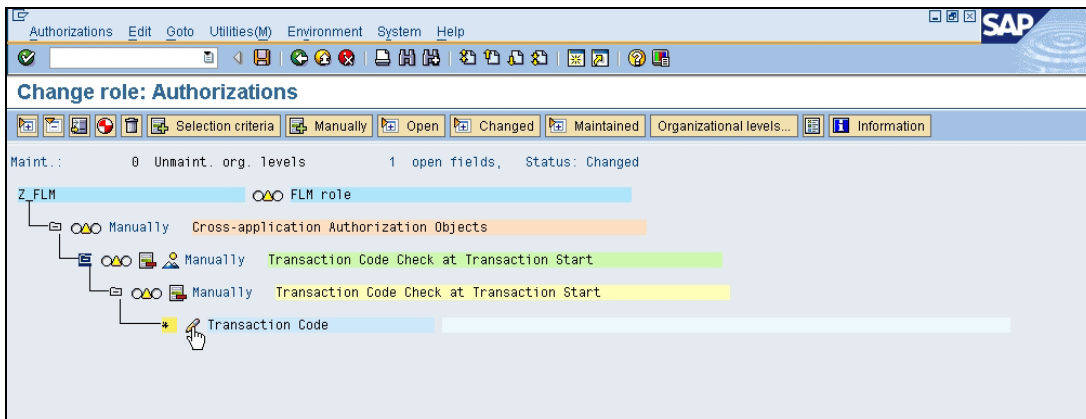
- Expand the hierarchy and then enter values for each field: select the '*' symbol in order for this role to include all form types, categories and activities.



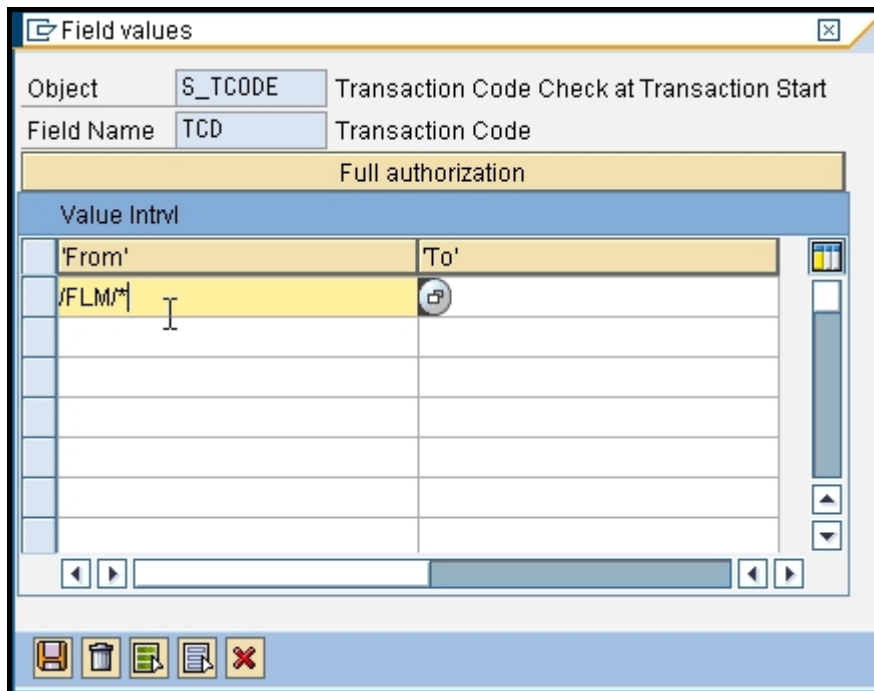
- Select the 'Manually' pushbutton again, and enter the authorization object S_TCODE



- Expand the Tree out and select the Change Transaction code button.



- Enter '/FLM/*' into the 'Form' field as shown. This is granting authority for ALL /FLM transactions to the FLM_ADMIN role.



- Generate. (Save)
- Role FLM_ADMIN is now ready to be linked to user records.

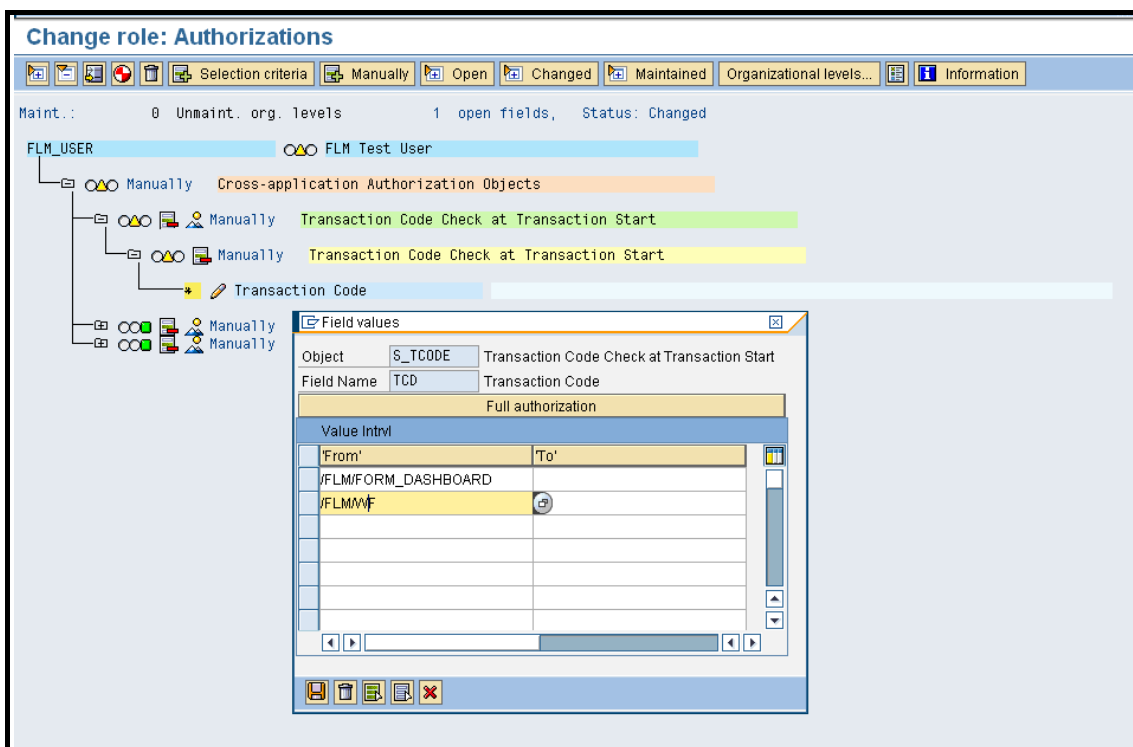
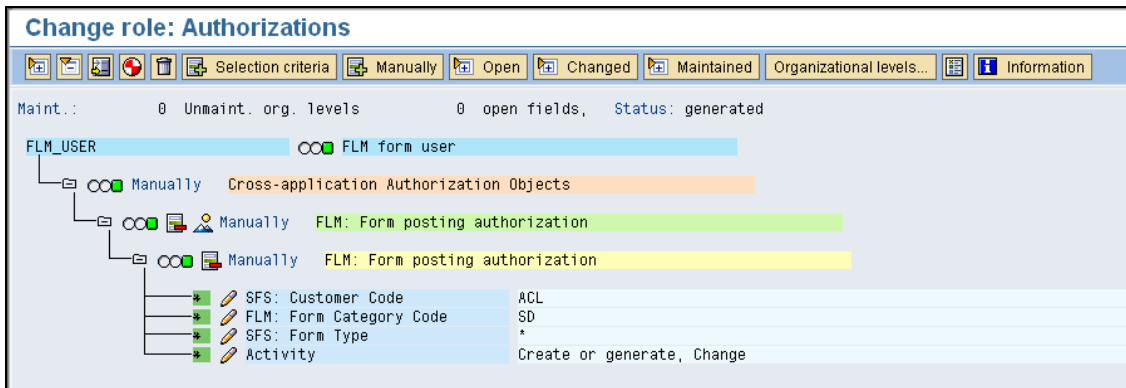
4.4.2 Creating Form User Roles

Follow the same steps as for setting up the FLM_ADMIN role, but in the final step, limit the role by form category, form type and /or /FLM transaction code.

For example:

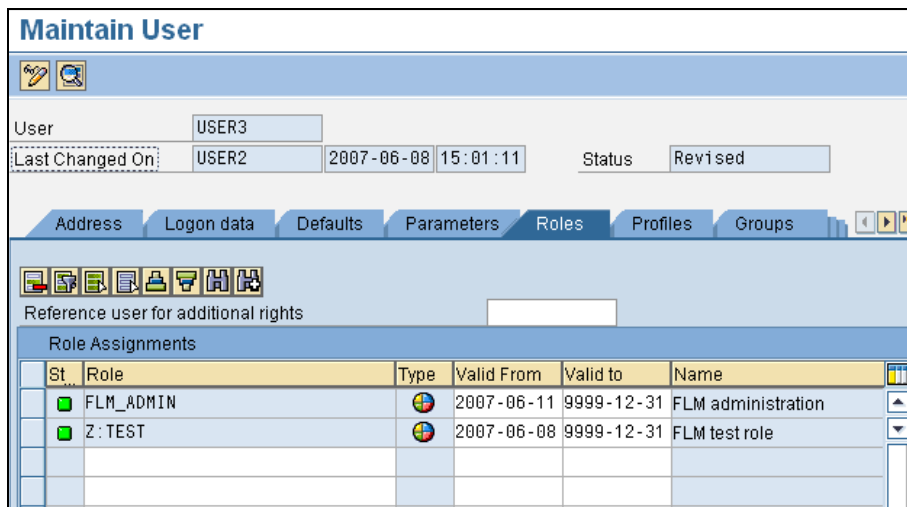
If the user only submits forms but cannot change submitted forms, then select activity 01 but not 02 for authorisation object Z/FLM/0001.

If the user can display forms in the Dashboard (transaction code /FLM/FORM_DASHBOARD), but not post forms in FPE (transaction /FLM/FPE) or trigger form routing escalations (transaction /FLM/WF) for example ,select only the relevant transaction code(s) in place of the /FLM/* assigned to the admin role against authorisation object S_TCODE

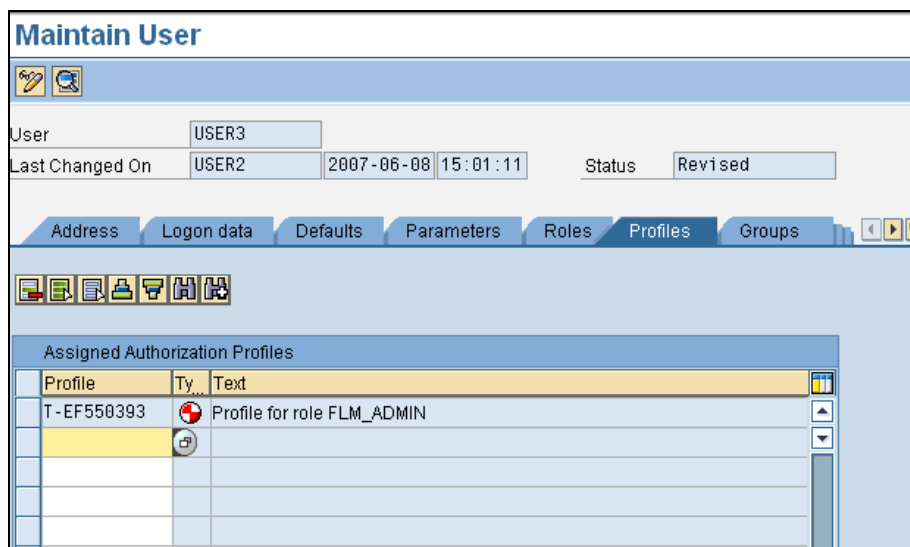


4.5 User Masters Roles

Once the roles and profiles have been created, it is necessary to link the roles to the user master records as normal. This can be achieved on a user-by-user basis, or by adding the FLM roles to existing composite roles. Users are maintained in transaction SU01. The role is added as shown in the 'Roles' tab.



When the role is added to the user, the profile appears in the 'Profiles' tab.



4.6 Offline User

The FLM system requires one SAP user as the so-called 'Offline User'. The FLM framework uses this user for processing offline forms [ie form sent to user not via the FLM Portal], since at this time no SAP dialog user is available for authorisation checking.

It is recommended that an SAP user is given the auth object Z_FLM... with open authorisations to the whole FLM system as follows:

/FLM/CUST	*
/FLM/FCAT	*
/FLM/FTYPE	*
ACTVT	*

This user master record is entered into the FLM configuration under the 'Set Customer Code' IMG activity against the 'Offline User' field. = user name attributed to offline email user activity

4.7 FLM Developer User Role and Authorisations

1. [SE16] Insert entry in table TBRG as below to define a class for FLM customising tables.

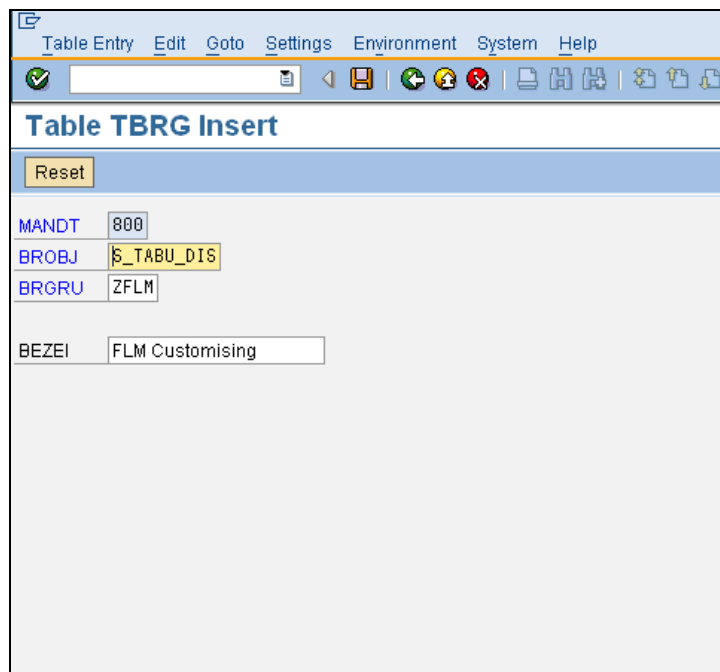


Table TBRG Insert	
MANDT	800
BROBJ	\$_TABU_DIS
BRGRU	ZFLM
BEZEI	FLM Customising

2. [SE16] Change entries in table TDDAT for each /FLM/* table - change the authorisation group from &NC& to ZFLM

Before	After

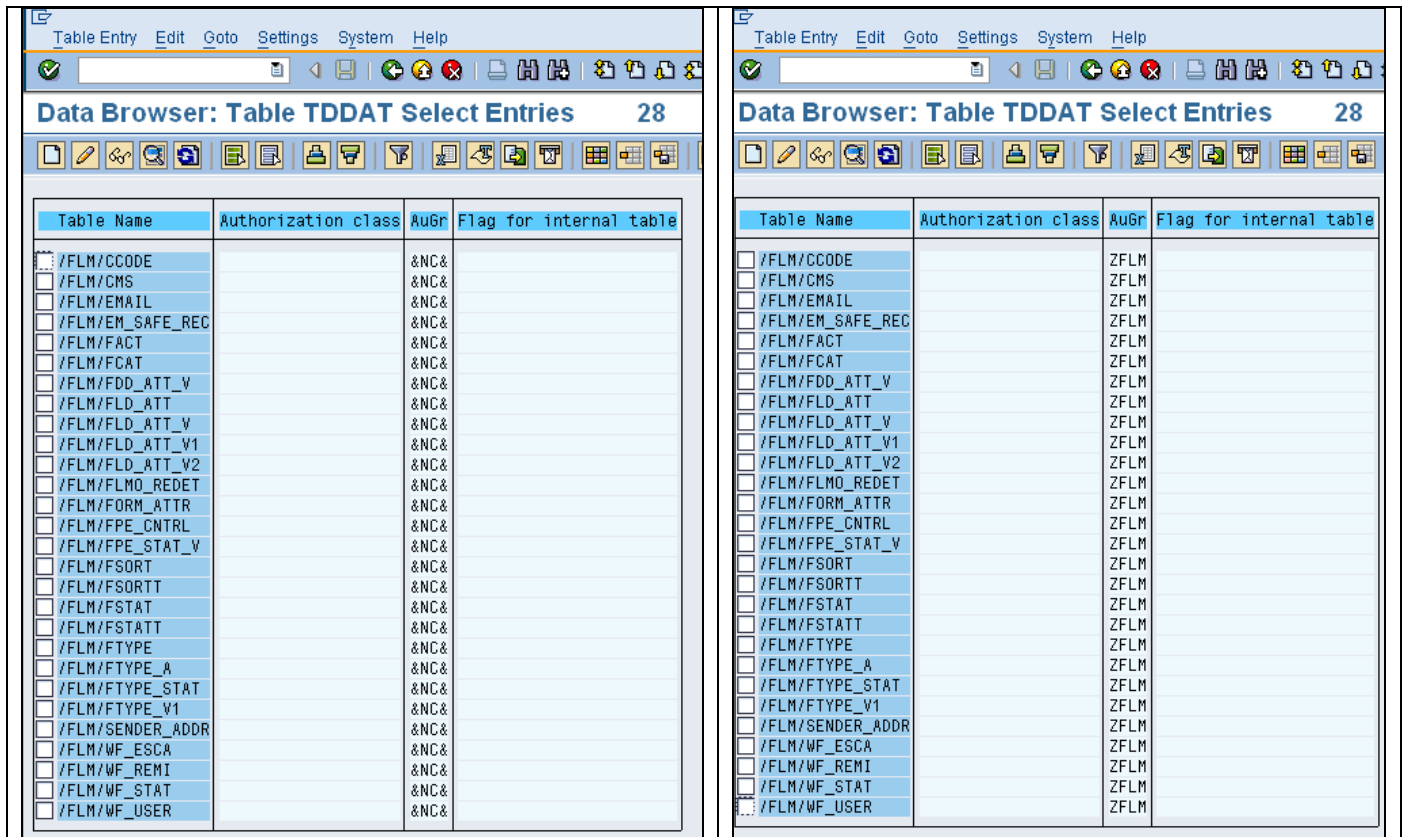
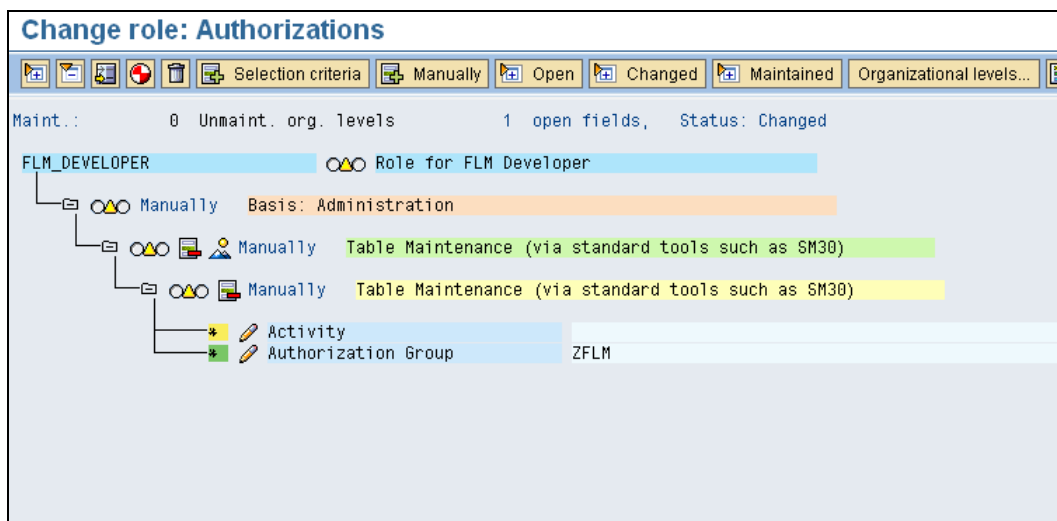


Table Name	Authorization class	AuGr	Flag for internal table
<input type="checkbox"/> /FLM/CCODE		&NC&	
<input type="checkbox"/> /FLM/CMS		&NC&	
<input type="checkbox"/> /FLM/EMAIL		&NC&	
<input type="checkbox"/> /FLM/EM_SAFE_REC		&NC&	
<input type="checkbox"/> /FLM/FACT		&NC&	
<input type="checkbox"/> /FLM/FCAT		&NC&	
<input type="checkbox"/> /FLM/FDD_ATT_V		&NC&	
<input type="checkbox"/> /FLM/FLD_ATT		&NC&	
<input type="checkbox"/> /FLM/FLD_ATT_V		&NC&	
<input type="checkbox"/> /FLM/FLD_ATT_V1		&NC&	
<input type="checkbox"/> /FLM/FLD_ATT_V2		&NC&	
<input type="checkbox"/> /FLM/FLMO_REDET		&NC&	
<input type="checkbox"/> /FLM/FORM_ATTR		&NC&	
<input type="checkbox"/> /FLM/FPE_CNTRL		&NC&	
<input type="checkbox"/> /FLM/FPE_STAT_V		&NC&	
<input type="checkbox"/> /FLM/FSORT		&NC&	
<input type="checkbox"/> /FLM/FSORTT		&NC&	
<input type="checkbox"/> /FLM/FSTAT		&NC&	
<input type="checkbox"/> /FLM/FSTATT		&NC&	
<input type="checkbox"/> /FLM/FTYPE		&NC&	
<input type="checkbox"/> /FLM/FTYPE_A		&NC&	
<input type="checkbox"/> /FLM/FTYPE_STAT		&NC&	
<input type="checkbox"/> /FLM/FTYPE_V1		&NC&	
<input type="checkbox"/> /FLM/SENDER_ADDR		&NC&	
<input type="checkbox"/> /FLM/WF_ESCA		&NC&	
<input type="checkbox"/> /FLM/WF_REMI		&NC&	
<input type="checkbox"/> /FLM/WF_STAT		&NC&	
<input type="checkbox"/> /FLM/WF_USER		&NC&	

Table Name	Authorization class	AuGr	Flag for internal table
<input type="checkbox"/> /FLM/CCODE		ZFLM	
<input type="checkbox"/> /FLM/CMS		ZFLM	
<input type="checkbox"/> /FLM/EMAIL		ZFLM	
<input type="checkbox"/> /FLM/EM_SAFE_REC		ZFLM	
<input type="checkbox"/> /FLM/FACT		ZFLM	
<input type="checkbox"/> /FLM/FCAT		ZFLM	
<input type="checkbox"/> /FLM/FDD_ATT_V		ZFLM	
<input type="checkbox"/> /FLM/FLD_ATT		ZFLM	
<input type="checkbox"/> /FLM/FLD_ATT_V		ZFLM	
<input type="checkbox"/> /FLM/FLD_ATT_V1		ZFLM	
<input type="checkbox"/> /FLM/FLD_ATT_V2		ZFLM	
<input type="checkbox"/> /FLM/FLMO_REDET		ZFLM	
<input type="checkbox"/> /FLM/FORM_ATTR		ZFLM	
<input type="checkbox"/> /FLM/FPE_CNTRL		ZFLM	
<input type="checkbox"/> /FLM/FPE_STAT_V		ZFLM	
<input type="checkbox"/> /FLM/FSORT		ZFLM	
<input type="checkbox"/> /FLM/FSORTT		ZFLM	
<input type="checkbox"/> /FLM/FSTAT		ZFLM	
<input type="checkbox"/> /FLM/FSTATT		ZFLM	
<input type="checkbox"/> /FLM/FTYPE		ZFLM	
<input type="checkbox"/> /FLM/FTYPE_A		ZFLM	
<input type="checkbox"/> /FLM/FTYPE_STAT		ZFLM	
<input type="checkbox"/> /FLM/FTYPE_V1		ZFLM	
<input type="checkbox"/> /FLM/SENDER_ADDR		ZFLM	
<input type="checkbox"/> /FLM/WF_ESCA		ZFLM	
<input type="checkbox"/> /FLM/WF_REMI		ZFLM	
<input type="checkbox"/> /FLM/WF_STAT		ZFLM	
<input type="checkbox"/> /FLM/WF_USER		ZFLM	

- [PFCG] Add the authorisation object S_TABU_DIS with activity '*' and Authorisation Group 'ZFLM' to an FLM_DEVELOPER role as normal



Change role: Authorizations

Maint.: 0 Unmaint. org. levels 1 open fields, Status: Changed

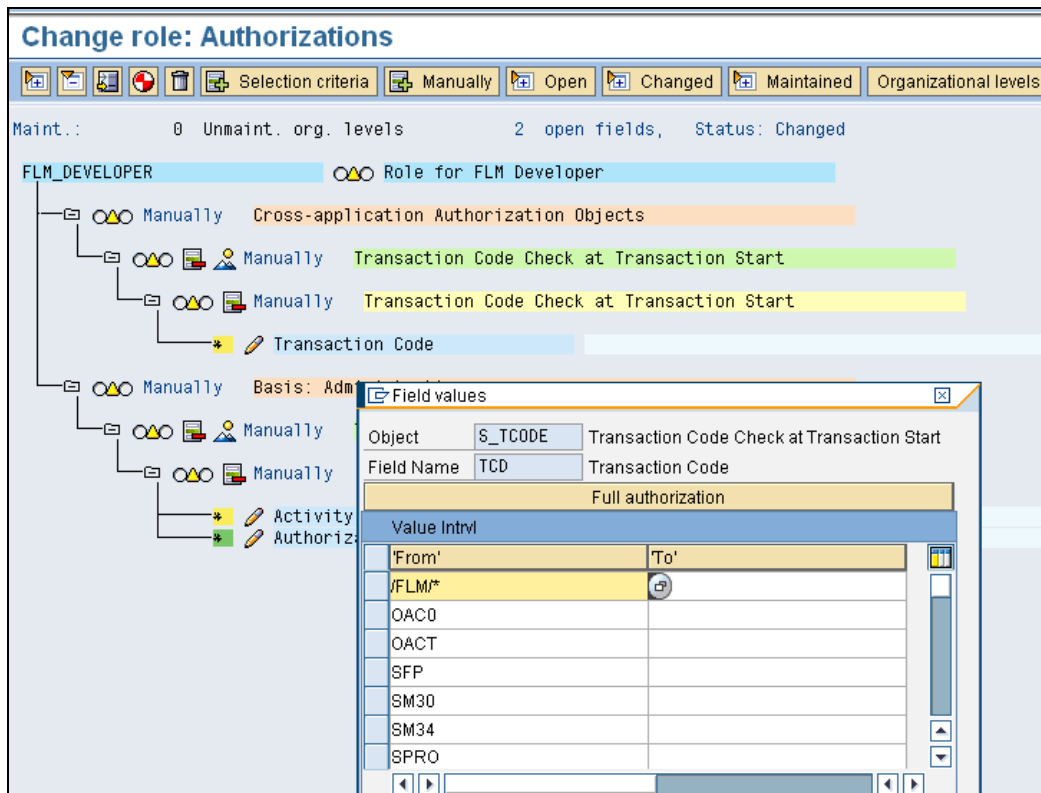
FLM_DEVELOPER Role for FLM Developer

- Manually Basis: Administration
 - Manually Table Maintenance (via standard tools such as SM30)
 - Manually Table Maintenance (via standard tools such as SM30)
 - Activity *
 - Authorization Group ZFLM

- Also add the authorisation object S_TCODE with the following transactions:

- /FLM/*
- OACO

- OACT
- SFP
- SM30
- SM34
- SPRO
- SOST
- SCOT



5. Add authorisation objects S_CTS_ADMI and S_TRANSPRT to enable the role to create and administer transport requests.
6. Add role FLM_DEVELOPER to the developer user, along with a normal ABAP developer role such as SAP_BC_DWB_ABAPDEVELOPER

5 Deploying the FLM Portal

5.1 Introduction

In order make use of the FLM Portal for on-line forms, the FLM Portal must be deployed to the Java Server using the Software Deployment Manager [SDM].

5.2 Enterprise Archive

The FLM portal is shipped as an Enterprise Archive on the installation disks. This file is called "flm.com~flmgui.ear" and is located in the DATA directory.

5.2.1 SDM Procedure

The high-level procedure for deploying this archive is as follows. For step-by-step guidance on any particular step, please consult the appropriate SAP documentation:

1. Copy the archive from the installation media to a local file available to the Java application server.
2. Start the SDM and login to the Java server.
3. Select 'Deploy New Archive' and navigate to the file from step 1)
4. Follow the on-screen instructions to deploy the file

5.2.2 Java Connectors

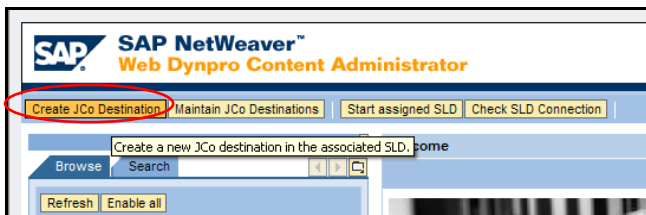
Once the archive has been successfully deployed, the new java application must be connected to the ABAP backend system by configuring necessary "Java Connectors" [JCo's].

There are two java connectors that must be configured:

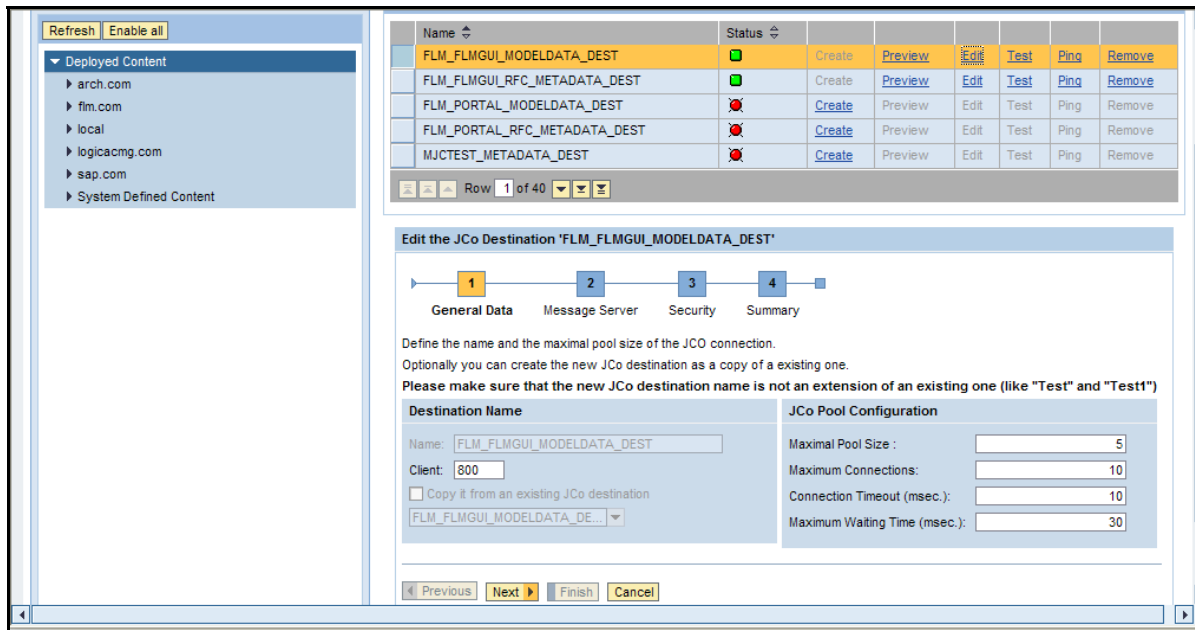
- FLM_FLMGUI_RFC_METADATA_DEST
- FLM_FLMGUI_MODELDATA_DEST

These two connectors will have been registered in the System Landscape Directory [SLD] during the deployment of the archive in the previous step. Hence each JCo can now be configured using the Content Manager application of the local Java stack.

Login to the User Management portal and click 'create JCo destination'.



Here, create the two Java connectors as detailed above and configure their settings and security using the wizard:



Please consult SAP standard documentation for further guidance on the step-by-step process of configuring Java Connectors.

Note:

It is highly recommended to configure the JCo's to make use of 'Logon Tickets', rather than by hardcoding usernames and passwords, as this ensures that user management is centralised into the ABAP stack and that all SAP-standard authorisation checks are enforced.

For the purposes of prototyping and development, FLM can be used with hardcoded JCO userreferences as the FLM authorisations due not rely on the user maintained in these connections, but rather the user who has logged into the FLM Portal.