

**NextLabs**

**Dynamic Authorization  
Management for SAP  
Advanced Edition**

**User's Guide**

**Version 7.8**

**January 2017**



---

## CONFIDENTIALITY NOTICE

THIS DOCUMENT IS CONFIDENTIAL AND PROPRIETARY TO NEXTLABS, INC. AND MAY NOT BE REPRODUCED, PUBLISHED OR DISCLOSED TO OTHERS WITHOUT COMPANY AUTHORIZATION.

© 2009-2017 NextLabs, Inc. All rights reserved.  
The information in this document is subject to change without notice.

To provide feedback on this document, email NextLabs, Inc. at [techpubs@nextlabs.com](mailto:techpubs@nextlabs.com).

## TRADEMARKS

NextLabs<sup>®</sup>, ACPL<sup>®</sup>, Enterprise Data Protection<sup>™</sup>, and the Enterprise Data Protection logo are registered trademarks of NextLabs, Inc. All other brands or product names used herein are trademarks or registered trademarks of their respective owners.

## LICENSE AGREEMENT

This documentation and the software described in this document are furnished under a license agreement or nondisclosure agreement. The documentation and software may be used or copied only in accordance with the terms of those agreements. No part of this document may be reproduced, stored in a retrieval system or transmitted in any form or any means electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's use, without the prior written permission of NextLabs, Inc.

The content of this document is provided for informational and instructional use only. It is subject to change without notice, and should not be construed as a commitment by NextLabs, Inc.

NextLabs, Inc. assumes no responsibility or liability for any inaccuracies or technical errors that may appear in the content of this document.

Published in San Mateo, CA, by NextLabs, Inc.  
[www.nextlabs.com](http://www.nextlabs.com)  
[info@nextlabs.com](mailto:info@nextlabs.com)  
[support@nextlabs.com](mailto:support@nextlabs.com)  
650.577.9101

Document Revision Number: DAM-SAP7.8

---



# Contents

---

<b>1 Introduction</b>	<b>11</b>
About NextLabs Dynamic Authorization Management for SAP	11
What's New in this Release	12
Key Benefits	12
Classifying objects	12
The SAP Authorization Workflow	13
Classification of Business Objects	13
Access Control	14
Integrated Rights Management	14
Auditing	15
About Extending SAP Authorizations	15
Coordinating Security Classifications and Access Control Contexts	16
Classification and ACC Schemes	16
Extensibility of the Dynamic Authorization Management for SAP	17
Logical Architecture Overview	17
About the Policy Controller	18
Policy Evaluation	18
Monitoring and Auditing	18
Policy Controller Functional Components	18
About the Policy Controller Communication Interface	19
Functional Integration During a Policy Check	19
Transaction/UI Function Integration	20
SAP Agent	21
Security Classification	21
Policy Controller Communication Interface	21
SAP Policy Model	21
Policy Based Security Classification	22
Rights Management Server	22
About SAP EEC and cFolders Integration	22
Contacting Technical Support	24
<b>2 Quick Reference for Set Up</b>	<b>25</b>
Explanation of User Roles	25

Dynamic Authorization Management Installation Procedures (All Cases) . . . . .	26
Example 1: Configuring SAP ECC Only . . . . .	26
Example 2: Configuring SAP ECC, cFolders, and PLM . . . . .	27
Example 3: Configuring SAP ECC and EasyDMS . . . . .	29
Example 4: SAP ECC and BW . . . . .	30
<b>3 Installation and Set Up . . . . .</b>	<b>33</b>
Before You Begin . . . . .	33
Supported Platforms and Products . . . . .	33
Firewall and Port Requirements . . . . .	34
Configuring Firewall Settings . . . . .	34
Gateway Requirements . . . . .	36
Supported Configurations . . . . .	36
1. Policy Controller for Java + Java Connector on Windows . . . . .	36
2. Policy Controller for Java + Java Connector on Red Hat Linux . . . . .	36
3. Server Policy Controller + Java Connector on Windows . . . . .	37
Installation Procedures for Supported Configurations . . . . .	37
Step 1: Installing the Policy Controller . . . . .	38
Step 2: Installing Dynamic Authorization Management for SAP . . . . .	38
Installation Files . . . . .	38
Installation Sequence and Procedure . . . . .	39
Step 3: Making the NextLabs Namespace Modifiable . . . . .	47
Configuring the Policy Controller Communication Interface: Java Connector . . . . .	49
Before You Install . . . . .	50
Install C++ Runtime Environment on the Policy Controller Host . . . . .	50
Define an SAP Gateway Host and Port . . . . .	50
Create an RFC User . . . . .	50
Locate Required Values for Configuration . . . . .	50
Installing the Java Connector . . . . .	51
Script Installation of the Java Connector for the Server Policy Controller . . . . .	52
Manual Installation of the Java Connector for the Server Policy Controller . . . . .	55
Manual Installation of the Java Connector for the Policy Controller for Java . . . . .	55
Configuring the RFC Connection . . . . .	56
Configuring the Properties and Services Files Manually . . . . .	60
Configuring the SAP JavaSDK Properties File (Manual Installation only) . . . . .	60
Configuring the Windows System Services File (Manual and Script Installation) . . . . .	61
Testing the Java Connector Configuration . . . . .	62
Testing the Connection from the Policy Controller Side . . . . .	62
<b>4 Configuring Basic Features . . . . .</b>	<b>65</b>
Configuring NextLabs Control Center for Dynamic Authorization Management for SAP . . . . .	65
Enrolling Users from SAP into Control Center . . . . .	65
Extracting User Data from SAP . . . . .	65



Uploading SAP User Data into Control Center .....	68
Configuring SAP Actions .....	69
Configuring SAP Obligations .....	70
Configuring NextLabs Entitlement Packs and SAP Data .....	71
Using the NextLabs Configuration Tool .....	72
Checking Configuration Status .....	74
Activating Entitlement Packs (EPCONF) .....	77
Adding Composite Keys and Classification Values .....	78
Security Identifiers .....	78
Composite Keys .....	79
Classification Values .....	79
Adding Composite Keys .....	79
Adding New Classification Values .....	81
Linking Composite Keys (SECENH) .....	83
Mapping Security Fields (SECMPG) .....	84
Configuring Security Identifier/Composite Key Value Tables (EPVAL) .....	86
Configuring UI Functions .....	88
Mapping Transaction Codes and UI Functions to Actions (ACTIONS) .....	90
Configuring SAP Data Handling and Connection Settings .....	94
Setting Default Values Automatically .....	95
Changing Connection Configuration Settings .....	96
Recommended Configuration for Implementations with Many Classifications (more than 40,000 rows) .....	102
Configuring View Filtering (EasyDMS Only) .....	102
Configuring Number Range Intervals .....	103
Configuring the NextLabs Number Range .....	104
Configuring Policy Checks Based on Transaction/UI Function .....	109
Configuring Special Fields for the Security Classifications Maintenance Table .....	111
Defining How Security Classifications and Access Control Contexts Should Be Applied .....	112
Defining How Multiple Security Classifications Should Be Applied .....	116
Configuring Access Control Context Settings (PLM Only) .....	121
Standard and Compound Access Control Contexts .....	121
Passing ACCs as Values or Paths .....	121
Configuring the Transactions or Functions to Intercept .....	123
Configuration for Policy-based Security Classification .....	125
Designating Which Create/Edit Functions Should Be Included in the PBSC Queue .....	126
Defining the PBSC Filter for Documents .....	127
Defining the PBSC Filter for Materials .....	128

Configuring the PBSC Conversion Directory .....	129
Defining the Background Job for PBSC .....	131
Configuring the PBSC Custom Obligation .....	136
Configuring Enhancement Implementations .....	137
<b>5 Configuring Optional Features .....</b>	<b>139</b>
Configuring Policy Checks for Value (F4) Help .....	139
General Steps .....	140
Configuring Policy-based Data Segregation .....	141
Defining the BADI Implementation .....	141
BADI Implementation for Filter Policies (ECC) .....	142
BADI Implementation for Filter Policies (EasyDMS) .....	142
BADI Implementation for Access Control Policies (ECC) .....	143
Configuring the Data Segregation Obligations .....	143
Configuring the Check-in and Check-out Actions .....	144
Configuration for Integrated Rights Management (IRM) .....	145
Installing Rights Management Server .....	145
Configuring the IRM Conversion Directory .....	146
Configuring IRM Selection Criteria (Filter) .....	149
Configuring IRM Selection Criteria for SAP ECC .....	149
Configuring IRM Selection Criteria for SAP cFolders .....	151
Defining Background Jobs for IRM .....	153
Defining the IRM_DOWNLOAD Background Job for SAP ECC .....	153
Defining the IRM_CFX Background Job for SAP cFolders .....	159
Configuring the IRM Obligation .....	164
Configuring Encryption Keys .....	165
Configuring the Read Tags Feature .....	166
Configuring the RFC Connection for Read Tags .....	166
Configuring SAP SDK Service and SAPJCo-EDRM Properties Files .....	166
Configuring SAP Data Handling and Connection Settings for Read Tags .....	169
Adding Custom Classification Values .....	169
Mapping Classification Values to the Policy Controller .....	169
Implementation Reference for Read Tags .....	170
Enhancement Implementation for Read Tags .....	170
BADI Implementation to Configure a Temporary File Location for Read Tags .....	170
BADI Implementation for Dynamic User or Resource Attribute .....	171
Reading Tags .....	172
Designing Read Tags Policies .....	174
Example Policy: Temporary File Location .....	174
Example Policy: Read Tags .....	176
<b>6 Using Classifications and Policies .....</b>	<b>179</b>
About Classifications and Policies .....	179

What Can Dynamic Authorization Management Do? .....	180
Entitlement Pack for ECC .....	180
Entitlement Pack for PLM .....	181
Entitlement Pack for EasyDMS .....	183
Entitlement Pack for cFolders .....	184
Entitlement Pack for BW .....	185
About SAP Policies .....	186
Policy Based Security Classification .....	186
About the SAP Resource String .....	187
About SAP Policy Messages .....	187
About Custom Obligations .....	188
Applying Security Classifications .....	188
About Classification Data .....	189
About Compound Classification Keys .....	189
Inheritance of Security Classifications and Access Control Contexts .....	190
Best Practices for Retrieving Security Classification Records .....	190
Viewing Security Classification Records .....	191
Applying Security Classifications Manually .....	194
Applying Security Classifications Based on Policy .....	200
Example Policy: Security Classification Based on Content Analysis .....	200
Updating Classifications for Files Exported to cFolders .....	203
Designing SAP Access Control Policies .....	204
Example Policy: Access Control Based on Classification and ACC .....	205
Example Policy: Access Control Based on Compound Key .....	213
Example Policy: Access Control Based on Resource Attributes .....	215
About SAP Resource Attribute Names .....	218
Example Policy: View Filtering (EasyDMS Only) .....	220
View Filtering in cFolders .....	221
Designing Access Control Policies for SAP BW .....	221
Example Policy: Restrict Access to Classified Data .....	222
Example Policy: Filter Access to Classified Data .....	227
Designing Integrated Rights Management Policies .....	228
Example Policy: IRM Encryption and Tagging .....	228
Designing Data Segregation Policies .....	231
Example Policy: Restrict Storage Locations for Check-in of Classified Data .....	231
Example Policy: Segregate Classified Data and Other Data .....	233
Example Policy: Restrict Download Locations for Check-out of Classified Data .....	235
Verifying the Storage Location of Data .....	235
<b>7 Administration and Maintenance .....</b>	<b>239</b>
Maintenance for Dynamic Authorization Management for SAP ECC .....	239
Viewing NextLabs Log Information in SAP .....	240
Configuration and Management .....	242

Configuration Tools . . . . .	242
Management Activities . . . . .	243
Stopping and Starting Dynamic Authorization Managements . . . . .	243
Monitoring Enforcers . . . . .	243
Uninstalling, Repairing, or Modifying Policy Controllers and Enforcers . . . . .	244
About Service Account Permissions . . . . .	245
About Bundle Encryption . . . . .	246
Authentication Failure . . . . .	246
Decrypting the Bundle: Policy Controller . . . . .	246
Decrypting the Bundle: Policy Controller for Java . . . . .	247
Managing Enforcer Policies . . . . .	248
Managing Event Logging . . . . .	249
Logging Settings . . . . .	249
Changing Logging Levels . . . . .	249
Load Balancing the Policy Controller . . . . .	251
Example: Load Balancing Configuration . . . . .	251
<b>8 Custom Enhancements . . . . .</b>	<b>255</b>
Custom Security Classification Identifiers . . . . .	255
Configuring Custom Security Classification Identifiers . . . . .	255
Adding a Custom Identifier . . . . .	256
Custom Enhancement Activations . . . . .	261
Example: Creating a Custom Enhancement Activation Based on Routing . . . . .	262
Example: Generic Include . . . . .	267
Dynamic User and Resource Attributes . . . . .	268
Configuring Enhancement Activations for Dynamic Attributes . . . . .	268
Referencing Dynamic Attributes in Policies . . . . .	270
Custom Obligations . . . . .	271
Supported Platforms . . . . .	271
Configuring Enhancement Activations for Custom Obligations . . . . .	271
Process Custom Obligation Method . . . . .	274
Import Parameters . . . . .	274
Export Parameters . . . . .	275
Example: Custom Obligation . . . . .	276
Configuring a Custom Obligation . . . . .	276
<b>A Implementation Reference for ECC . . . . .</b>	<b>277</b>
Transactions . . . . .	277
Implementations . . . . .	291
BADI Enhancements . . . . .	291
Explicit Enhancements . . . . .	292
Implicit Enhancements . . . . .	293
User Exit . . . . .	294

---

Implementation Details .....	294
Implementation for Materials (Common Interception) .....	298
<b>B Implementation Reference for EasyDMS .....</b>	<b>301</b>
<b>C Implementation Reference for SAP PLM .....</b>	<b>303</b>
<b>D Implementation Reference for SAP BW .....</b>	<b>305</b>
Types of Access Control .....	305
Implementations .....	306
<b>E Implementation Reference for PBSC .....</b>	<b>309</b>
<b>F Implementation Reference for DFPS .....</b>	<b>311</b>
Implementations for DFPS .....	311
Example DFPS Policies .....	316
Example 1 .....	316
Example 2 .....	317
<b>G Read Tags: External Classifications .....</b>	<b>319</b>



# 1 Introduction

---

This section introduces NextLabs Dynamic Authorization Management for SAP Advanced Edition.

Topics include:

- [About NextLabs Dynamic Authorization Management for SAP](#)
- [What's New in this Release](#)
- [Key Benefits](#)
- [Classifying objects](#)
- [The SAP Authorization Workflow](#)
- [About Extending SAP Authorizations](#)
- [Logical Architecture Overview](#)
- [Functional Integration During a Policy Check](#)
- [About SAP EEC and cFolders Integration](#)
- [Contacting Technical Support](#)

---

## About NextLabs Dynamic Authorization Management for SAP

NextLabs Dynamic Authorization Management for SAP Advanced Edition is an extensible, XACML-based system that enables enterprises to configure, administer, enforce, and audit fine-grained, enterprise-wide, data authorization policies from a central location. It is the only solution that provides enter-prise-class manageability, productivity, and scalability with comprehensive, policy-based, cover-age for SAP data. It supports these SAP components:

- SAP ECC
- SAP PLM
- SAP EasyDMS
- SAP BW
- SAP cFolders
- SAP DFPS

## What's New in this Release

This release introduces a new Defense Forces and Public Security (DFPS) Entitlement Pack that extends fine-grained access control to the Defense Forces and Public Security Add-on. The DFPS Organizational elements, such as Force elements, can be used to determine user access. The solution also allows filtering of data based on user access.

## Key Benefits

The Dynamic Authorization Management for SAP extends role-based SAP authorization to provide policy-driven, fine-grained access control based on user or resource attributes, and supports the interception of SAP transactions for access control. In addition, the Entitlement Manager for SAP can be extended to support other SAP transactions and functions. The Entitlement Manager for SAP is available in a base installation version for SAP, with add-ons, called Entitlement Packs, for SAP ECC, SAP PLM, SAP EasyDMS, SAP BW, SAP cFolders, and SAP DFPS.

*Table 1-1: Entitlement Packs and supported access controls*

Entitlement Pack	Supported access controls
SAP ECC	Objects, such as Materials, Documents, BOMs, Routings, and Engineering Workbench.
SAP PLM	Materials, Documents, BOMs, and Change Masters. The Entitlement Manager for SAP distinguishes between versions of materials and documents, which enables access control based not just on Document, but on Document Type, Part, and Version.
SAP EasyDMS	All actions that users can perform on documents from within EasyDMS.
SAP BW	BW objects including InfoArea, InfoProvider and InfoObjects.
SAP DFPS	DFPS Organizational elements such as Force elements, Storage Locations, and MRP areas.

## Classifying objects

Business objects can be classified manually or automatically when users create or modify business objects. Classifications can be designed to address multiple security or compliance concerns simultaneously, and can be quickly configured, extended, and managed, using batch, interactive, or programmatic interfaces.

In addition, originals stored within SAP can be protected using NextLabs Integrated Rights Management, which automatically applies classifications to originals and protects data so it cannot be accessed by unauthorized users, even after originals are downloaded out of SAP.

Drawing from values in the Security Classification module, as well as defined Access Control Contexts in SAP PLM, policy rules can also take into account an SAP user's identity attributes and



dynamic and contextual factors, such as the computer or location. For example, a policy rule may state, “Allow only US Persons in US locations with Access Control Context Manufacturing to access Materials with Security Classification ITAR.” When users attempt to access the Material from within SAP PLM, this rule is validated in real-time, with no perceptible latency.

## The SAP Authorization Workflow

There are four information-control objectives for managing the creation, access, and distribution of data, also known as the SAP authorization workflow:

- [Classification of Business Objects](#) on page 13
- [Access Control](#) on page 14
- [Integrated Rights Management](#) on page 14
- [Auditing](#) on page 15

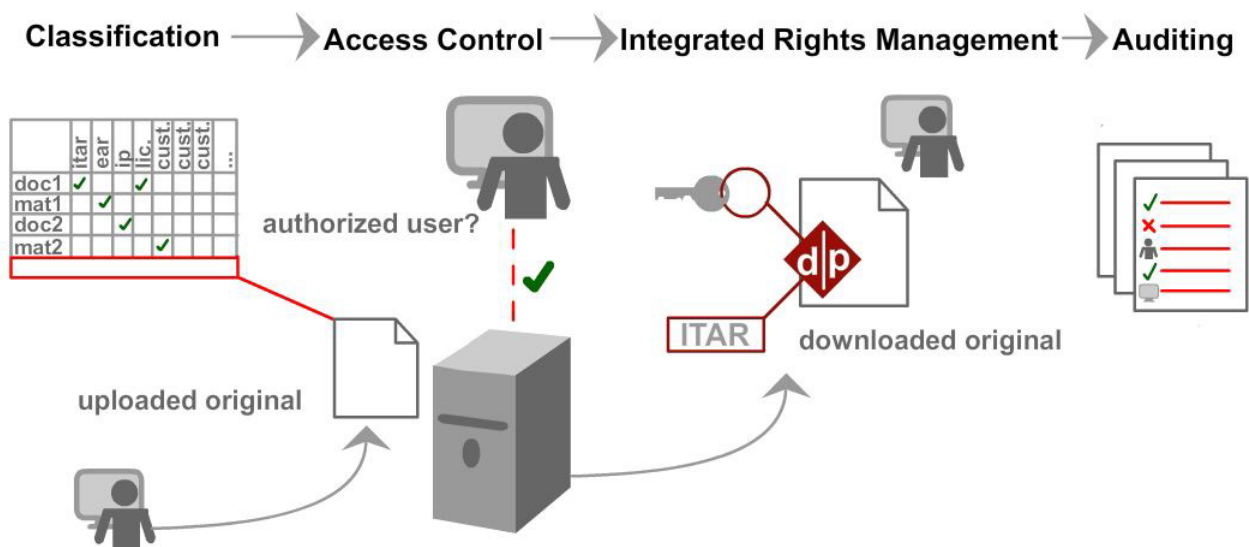


Figure 1-1: The Authorization Workflow

### Classification of Business Objects

The first step to controlling how business objects are accessed and used in SAP is classifying the data. The Security Classification module manages data classifications and simplifies the process of identifying and maintaining classification values. Custom classification values can be created and applied for all transactions, for only selected transactions, and down to the data level, for only certain business objects. The Entitlement Manager for SAP supports both manual, as well as automatic policy-based classification. For Policy Based Security Classification (PBSC), the classification table can be populated automatically based on the originals uploaded into SAP. After classifications are established, they are referenced in [Access Control](#) on page 14 and [Integrated Rights Management](#) on page 14 policies.

Security Classification schemes can include values in the NextLabs Security Classification Maintenance table, as well as Access Control Context (ACC) authorizations maintained in SAP PLM. See:

- [Applying Security Classifications](#) on page 188
- [Designing SAP Access Control Policies](#) on page 204

## Access Control

Access control policies authorize or block user access to specific business data within SAP, based on both the [Classification of Business Objects](#) on page 13 and selected user identity attributes. User alerts can be customized to provide authorized users with guidance and education on compliance restrictions and requirements.

Access control policies can be designed to apply at the transaction level, at the data level (even different versions of data, as described in [About Compound Classification Keys](#) on page 189), and/or at both the transaction and data level. You can control transactions performed in SAP ECC, functions in SAP PLM, and user actions in SAP EasyDMS and SAP cfolders.

For documents accessed through EasyDMS and SAP cfolders, there is an additional layer of access control—view filtering—which determines whether users can view documents they are not authorized to access. If view filtering is enabled, users must be authorized to view documents for files to be displayed.

For SAP BW, user access to data can be controlled in the reporting tool, Business Explorer (BEx) Analyzer. Policies can be written to restrict access to classified InfoArea and InfoProvider objects. More granular access control can be applied to specific classified data in InfoObjects.

- For more information on how to construct access control policies, see [Example Policy: Access Control Based on Classification and ACC](#) on page 205.
- For more information on actions you can control in SAP ECC, SAP PLM, and SAP EasyDMS, see [What Can Dynamic Authorization Management Do?](#) on page 180.
- For more information on view filtering in EasyDMS, see [Example Policy: View Filtering \(EasyDMS Only\)](#) on page 220.
- For more information on View Filtering in cFolders, see [Configuring SAP Data Handling and Connection Settings](#) on page 94.

## Integrated Rights Management

Using Integrated Rights Management (IRM), originals uploaded in SAP can be classified automatically so tags applied to documents persist even when documents are downloaded out of SAP. Using other NextLabs products, you can then design complementary policies that prevent controlled data from being used or distributed in unauthorized ways. A typical example is a policy blocking users from emailing documents that have been downloaded out of SAP to unauthorized users. In addition, IRM enables users to design policies that automatically apply NextLabs encryption to files, which adds an additional layer of persistent protection. To decrypt and access the data, users must have authorized encryption keys

For more information on how to construct Integrated Rights Management policies that tag and apply NextLabs Encryption to files, see [Designing Integrated Rights Management Policies](#) on page 228.

## Auditing

Administrators need insight into the actions users are performing in SAP, including how users are creating, accessing, and downloading data. Audit policies enable administrators to log these details, as well as include notification emails to administrators in appropriately serious cases. In addition, administrators can generate reports to identify the classifications that are being automatically populated through PBSC.

---

## About Extending SAP Authorizations

[Figure 1-2](#) shows how Dynamic Authorization Management for SAP draws from user and document attributes, native to SAP, and applies them in access control policies. When users attempt to access a document through SAP ECC, SAP PLM, SAP EasyDMS, or cFolders, a policy check is triggered. Depending on how policies and classification schemes are designed, the NextLabs Policy Controller can check the request against a Security Classification, PLM ACC setting, or both. In our example, the Security Classification “export security = ITAR” and the Access Control Context “Project A” are applied to a document. A policy limits access to all data that is flagged as “ITAR” and “Project A” to users with the attribute “country = US”.

This is just one example of how you can extend SAP authorizations with Dynamic Authorization Management for SAP. You can also apply more finely grained conditions to access control, for example, to make exceptions for Export Licenses, or to set special conditions for the location of the device (in certain networks or geographic locations, or for remote laptops). [Figure 1-2](#) draws from both Security Classification and Access Control Context, but a policy could also only draw from one or the other, or prioritize one over the other when both are present. For example policies, see [Using Classifications and Policies](#) on page 179.

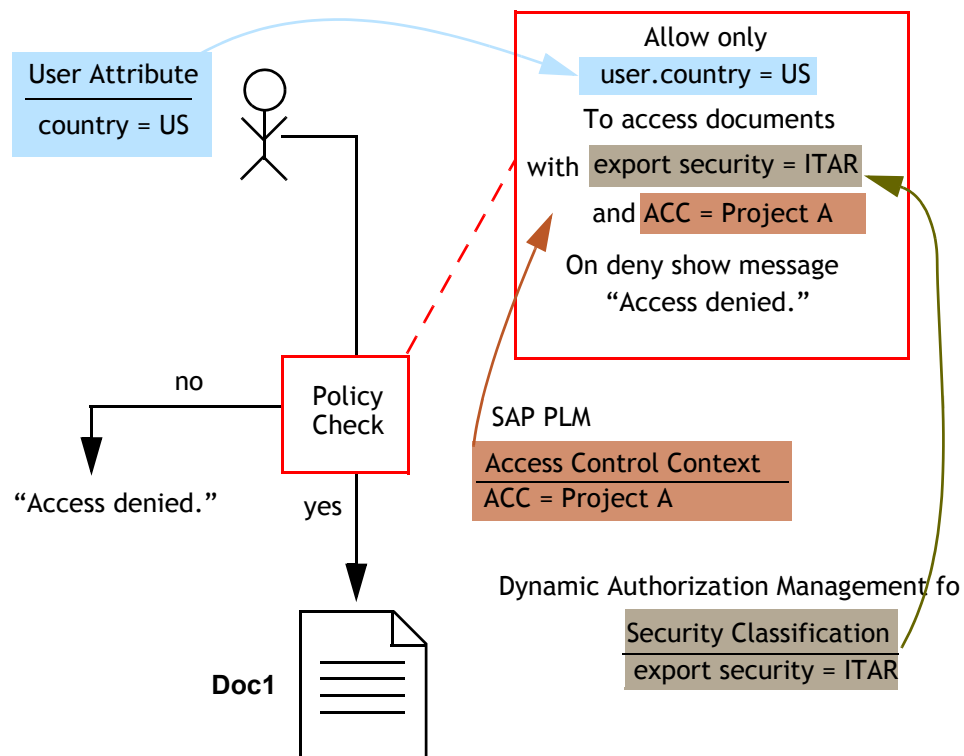


Figure 1-2: Policy-Based Authorization using SAP Attributes

## Coordinating Security Classifications and Access Control Contexts

Policies that control access to SAP business objects can be designed in a variety of ways. However, it is best to coordinate SAP PLM Access Control Contexts (ACC) with Security Classifications to capitalize on the strengths of each authorization scheme.

ACCs are most useful for providing flexible, coarse, role-based authorizations. For instance, you can assign a family of business objects to a department or team. You can also distinguish between long-standing authorizations and temporary ones (for example, granting access to external contractors while a certain project is active). For more information, see [Standard and Compound Access Control Contexts](#) on page 121.

After coarse authorizations are defined, you can design fine-grained, data-level Security Classifications on top of them to classify objects subject to regulatory and internal compliance requirements. Within a team, for instance, you can distinguish members who should be able to view a specific material based on the security classification applied to that material and the identity attributes of the user (i.e., location, citizenship, and so on).

## Classification and ACC Schemes

Depending on how the Entitlement Manager for SAP is configured, and which Entitlement Packs are implemented (ECC, PLM, EasyDMS, or BW), you can design policies that apply any available

classification scheme across all SAP components. In other words, you can combine ACCs defined in SAP PLM and/or classifications from SAP ECC to control how materials and documents are viewed in SAP ECC. You can do the same for SAP PLM, SAP EasyDMS, and SAP BW.

### Extensibility of the Dynamic Authorization Management for SAP

The Dynamic Authorization Management for SAP can be customized to enhance the SAP authorization model:

- Customize which attributes of SAP business objects are checked upon policy evaluation (for example, export license, jurisdiction, and other attributes). For more information, see [Custom Security Classification Identifiers](#) on page 255.
- Intercept SAP transactions. For more information, see [Custom Enhancement Activations](#) on page 261.
- Enable dynamic looks-up of user or resource information; for instance, for user attributes that are not enrolled into the NextLabs policy system, design extensions so this information can be retrieved dynamically at the point of a policy check. For more information, see [Dynamic User and Resource Attributes](#) on page 268.
- Design custom obligations in SAP’s ABAP language that trigger upon a policy enforcement event. For example, upon the event that a user is denied access to a resource, you can automate a workflow or execute an SAP transaction. For more information, see [Custom Obligations](#) on page 271.

## Logical Architecture Overview

The logical architecture of Dynamic Authorization Management for SAP includes a generic *Policy Controller*, and a *Policy Controller Communication Protocol* that provides the connection to *Dynamic Authorization Management for SAP*, as shown in [Figure 1-3](#).

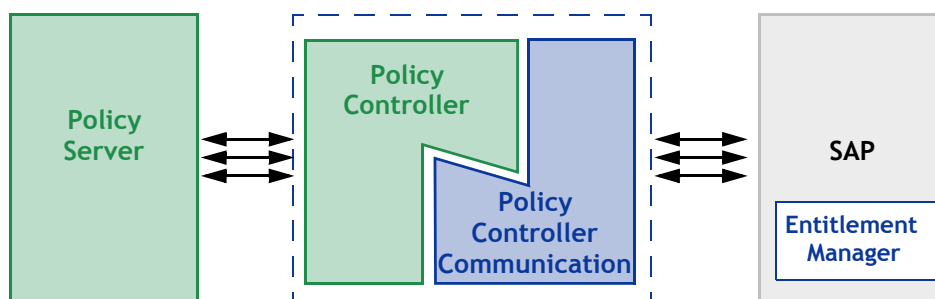


Figure 1-3: Overview of Logical Architecture

The Policy Controller provides the interface to the Policy Server, and functions as the Policy Decision Point, or PDP. It comprises a set of software modules that are delivered ready-to-install and requires no customization.

The Policy Controller Communication Interface provides the connection between the Policy Controller (PDP) and the Dynamic Authorization Management for SAP, which resides in SAP and serves as the Policy Enforcement Point, or PEP. The following sections provide more detail on these components.

## About the Policy Controller

The functions that the Policy Controller provides can be grouped into the categories:

- [Policy Evaluation](#) on page 18
- [Monitoring and Auditing](#) on page 18
- [Policy Controller Functional Components](#) on page 18

### Policy Evaluation

Whenever a policy enforcement point (PEP) detects an event that may be covered by a currently deployed policy, it sends a request to the Policy Controller which acts as the policy decision point (PDP). The Policy Controller applies all context information to the events, and makes decisions on what policy applies and how. It then relays the effects of any relevant policy back to the PEP, which contains system-specific logic to apply the enforcement.

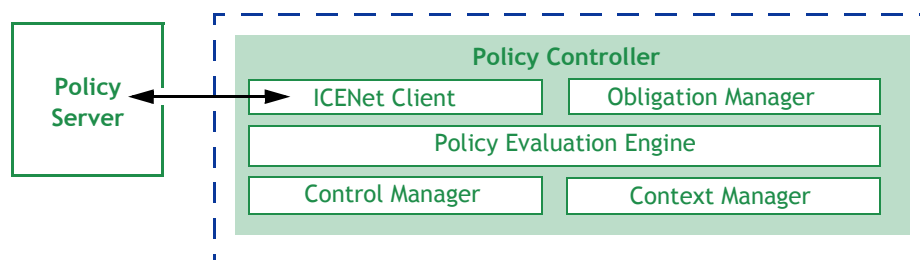
The PEP then instructs the application or file system to respond to the user's action as the policy requires—either allow or deny. If the policy evaluation results in the action being denied, the Policy Controller returns a message indicating that access is denied or that the requested action cannot be performed. These may take the form of standard system errors, or the customized text balloon that was defined with the policy being enforced, or both. Depending on the policy, it may also be accompanied by an obligation being triggered.

### Monitoring and Auditing

Every Policy Controller can monitor how users are accessing documents on that host, so that even if it never blocks any actions by policy subjects, it is still providing a valuable service of capturing who is using what information, how, and when. To do this, create and deploy Allow policies.

### Policy Controller Functional Components

The functional components of the Policy Controller are represented in [Figure 1-4](#).



*Figure 1-4: Policy Controller Detail*

- The **Policy Evaluation Engine** evaluates whether or not every user action is covered by any of the policies currently cached at that enforcement point. It bases this on multiple criteria such as who the user is, what host he is using, how he is connected to the network, what action is being attempted, and on what resource; the date, the time, and so on. It does this in real time, and operates continuously whether the host is connected to the network or not.
- The **Context Manager** keeps constant track of the environmental context of all events, and provides it to the Policy Engine and Policy Adapter. The context includes user identity, computer host name, network connection type, and date and time.
- For any policy that evaluates to True, the **Obligation Manager** initiates an obligation by sending a request to a policy adapter's obligation services or executing built-in obligations. It contains three sub-components:
  - *Policy Logger* - Collects and logs all activity details and policy decision results.
  - *Messaging Services* - Sends message to recipients or targets listed in a policy.
  - *Application Extender* - Launches an application or custom executable that performs some custom obligation.

**Note:** Although it is logically a part of the Policy Controller, the Obligation Manager runs as a separate process, visible in the *Processes* tab in Windows Task Manager.

- The **Control Manager** records non-policy activities, updates the configuration, and secures the controller. Components include:
  - *Activity Recorder* - Records activities tracked by the Policy Adapter in real time.
  - *Configuration Manager* - Applies profile and system configuration changes, in real time.
  - *Policy Authentication* - Authenticates the policy set from the Policy Server and encrypts it on the local file system.
- The **ICENet Client** provides the interface for all communication with the Policy Server. It is used for deploying new or changed policies, periodically sending activity logs from each control point, and providing controller health status.

## About the Policy Controller Communication Interface

The Policy Controller Communication Interface sends policy check event information from the PEP to the PDP, where policy evaluation occurs, and then sends the PDP policy decision and result back to the PEP, where enforcement occurs. This communication interface uses a NextLabs Java Connector (JCo) plug-in that integrates with the SAP Remote Function Call (RFC) interface.

---

## Functional Integration During a Policy Check

This section provides information about the interaction of components during a policy check. The Dynamic Authorization Management for SAP is a Policy Enforcement Point (PEP) that intercepts user actions in SAP ECC, SAP PLM, SAP EasyDMS, and SAP cFolders. When users request access to data or perform actions on an SAP business object, the steps described below occur. The back-end ABAP-based system where the Entitlement Manager for SAP is installed can be SAP ECC, or SAP cFolders, or any combination thereof.

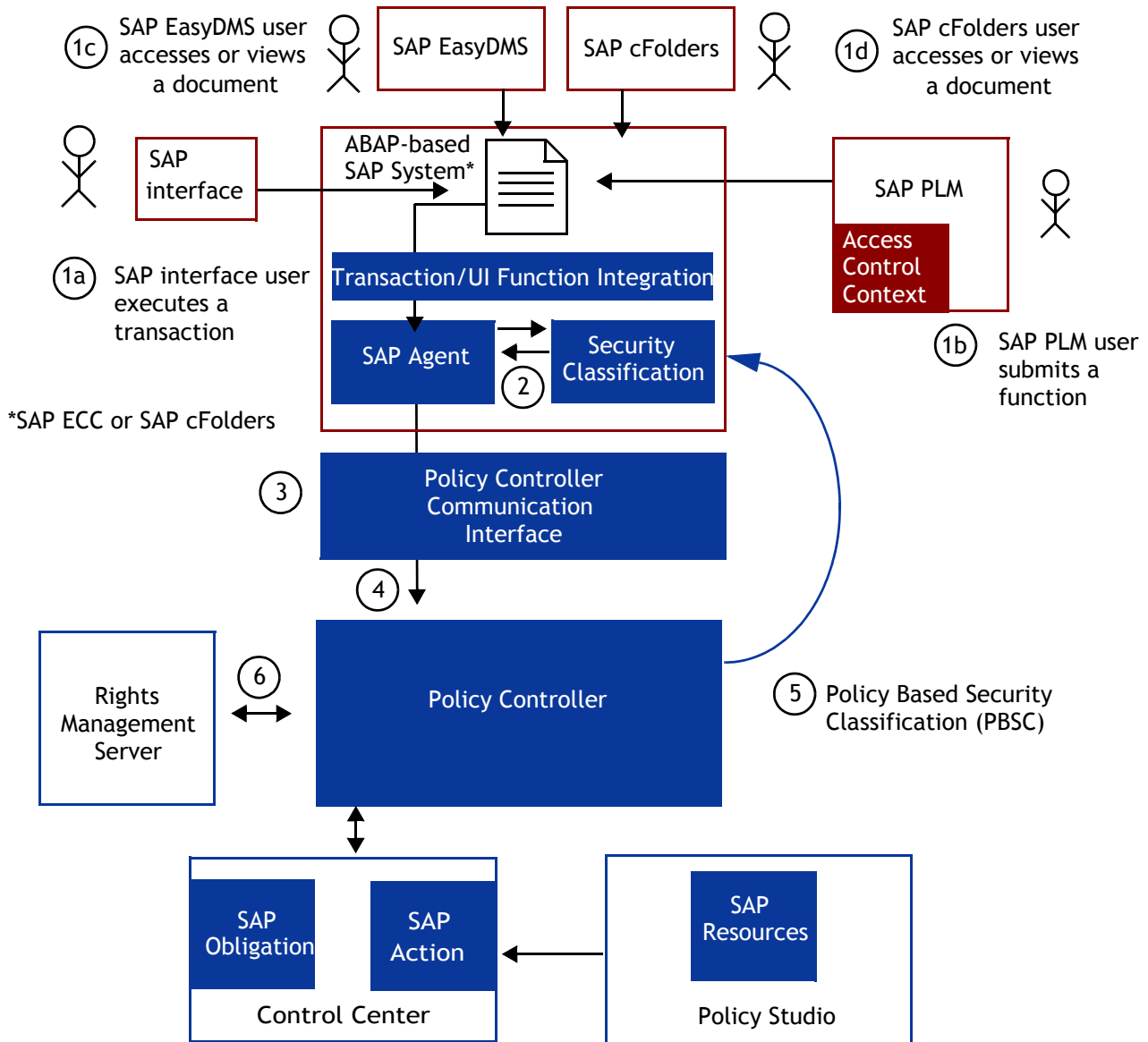


Figure 1-5: Functional Overview

### Transaction/UI Function Integration

The Transaction/UI Function Integration module is one component of Dynamic Authorization Management for SAP that resides within the ABAP-based SAP back-end system (currently, SAP ECC or SAP cFolders). Transaction/UI Function Integration enables NextLabs to “hook” policy checks into specific events, such as when a user executes transaction MM03 in SAP ECC, submits a Display Material UI function in SAP PLM, or attempts to access a document through EasyDMS or cFolders. Information about each event is routed through the Transaction/UI Function Integration module. (Figure 1-5, Steps 1a, 1b, 1c, and 1d.)



## SAP Agent

The **SAP Agent** is another one of the three components of Dynamic Authorization Management for SAP that reside within the ABAP-based back-end system. The SAP Agent collects information from the Security Classification module and Transaction Integration and routes it to enterprise services created using SOA Manager (Figure 1-5, Step 2.)

## Security Classification

**Security Classification** is another one of the three components of Dynamic Authorization Management for SAP that reside within the ABAP-based SAP back-end system (SAP ECC or SAP cFolders). The Security Classification module stores classification settings for SAP business objects (Materials, Documents, and Transactions). Classification settings, which are customized upon implementation, can be applied manually in the *Security Classification Maintenance* screen.

The steps necessary to configure Security Classifications are discussed in [Quick Reference for Set Up](#) on page 25. This includes defining the security classification identifiers and establishing the classification priority to use when multiple identifiers are present for a single transaction. For more information, see [Applying Security Classifications](#) on page 188 or [Example Policy: Access Control Based on Classification and ACC](#) on page 205.

## Policy Controller Communication Interface

The Entitlement Manager for SAP implementation must include an interface that enables communication between the ABAP-based SAP back-end system (either SAP ECC or cFolders, or both) and the NextLabs Policy Controller (Figure 1-5, Step 3).

When a transaction occurs, transaction information is routed from the SAP Agent to the Policy Controller Communication Interface. The Interface routes that information to the Policy Controller, which checks the transaction and classification information against any pertinent policies. The Policy Controller sends the allow or deny result back through the communication interface to the SAP Agent. For PLM UI functions, the response is then routed through the Java Application Server to the SAP PLM web application interface.

This communication interface is configured using the NextLabs Java Connector (JCo) plug-in. The NextLabs Java Connector is a middleware component that enables development of SAP-compatible components and applications in Java. It is configured with SAP ECC using the Remote Function Call (RFC) interface.

For more information about configuring the Java Connector, see [Configuring the Policy Controller Communication Interface: Java Connector](#) on page 49.

## SAP Policy Model

A few SAP-specific components must be configured in NextLabs Control Center and Policy Studio. These components enable the integration of SAP Security Classification settings with policies created in NextLabs Policy Studio.

## SAP Obligations

Included in a Policy Controller response might be a custom SAP Obligation. There are several options for custom obligations. One obligation displays an SAP message informing users that

access is denied or allowed (with educational warnings on how sensitive data should be handled). Another customer obligation applies file tagging to an SAP document so that Security Classification settings persist even outside SAP ECC and SAP PLM. In addition, custom obligations can be designed in ABAP code, so that any program created in ABAP can be triggered upon a policy evaluation.

SAP Obligations must be registered with the Control Center, as is discussed in [Configuring SAP Obligations](#) on page 70 and [Configuring the IRM Obligation](#) on page 164. The example policies in [Using Classifications and Policies](#) on page 179 describe how to associate the obligations with policies.

### SAP Action

As is discussed in [Using Classifications and Policies](#) on page 179, when creating SAP policies in Policy Studio, you might need to select SAP-specific actions, such as “Copy from SAP.” SAP actions must be registered with the Control Center, as discussed in [Configuring SAP Actions](#) on page 69.

### Policy Based Security Classification

Depending on system configuration, classification policies can be designed to automatically add classification values to the *Security Classification Maintenance* screen upon user actions, such as the creation or editing of a material or document ([Figure 1-5](#), Step 5). For instance, a policy could monitor for the creation of documents with certain keywords. When users save documents, a policy check detects the keyword, and based on policy, automatically populates classification information that can be referenced by access control and IRM policies. For more information on required configuration, see [Configuration for Policy-based Security Classification](#). For an example, see [Example Policy: Security Classification Based on Content Analysis](#).

### Rights Management Server

The NextLabs Dynamic Authorization Management for SAP can be integrated with the NextLabs Rights Management Server. Integrated Rights Management (IRM) enables you to embed classifications into originals as they are uploaded into SAP. That way, classifications are present when documents are downloaded out of SAP. The Rights Management Server can also apply NextLabs encryption to files. For more information on configuring the Rights Management Server, see [Configuration for Integrated Rights Management \(IRM\)](#). For example policies, see [Example Policy: IRM Encryption and Tagging](#).

To access NextLabs encrypted files, Rights Management Client must be installed on all endpoints. For more information, see the *NextLabs Rights Management Client User's Guide*.

---

## About SAP ECC and cFolders Integration

SAP cFolders (Collaboration folders) is the SAP web-based application for sharing information. It can be integrated with SAP ECC and SAP PLM. Users can select business objects within SAP ECC and export them to SAP cFolders, where other users may access and modify documents (including files associated with Documents, Materials, and BOMs).

As with SAP ECC, if the Entitlement Pack for cFolders is implemented, you can apply security classifications to documents that are created or edited in cFolders. If your implementation includes Entitlement Packs for both SAP ECC and SAP cFolders, classification values applied within SAP ECC also persist when data is exported from SAP ECC to cFolders. If your implementation includes SAP PLM, you can also pass ACC values to data that is exported to cFolders. You can also design policies that apply Policy Based Security Classification (PBSC) and Integrated Rights Management (IRM) to files within cFolders.

Figure 1-6 and Figure 1-7 show an example integration scenario for an implementation that includes both SAP ECC and SAP cFolders.

In the example, a document is attached to a business object in SAP ECC. (1) PBSC runs to assess how the file should be classified, and based on the presence of key words, sets the file's Export Classification value to ITAR in the Security Classification Maintenance table within SAP ECC. (2) IRM also runs to insert this classification value into the file's metadata (tag it), and apply NextLabs encryption to the file (protect it). (3) A user exports the file to the cFolders system, and automatically, the system updates the classification in the cFolder's Security Classification Maintenance table. An authorized user attempts to access the file, and a policy check references the cFolders classification to ensure access should be allowed. (4) This authorized user modifies the file in cFolders.

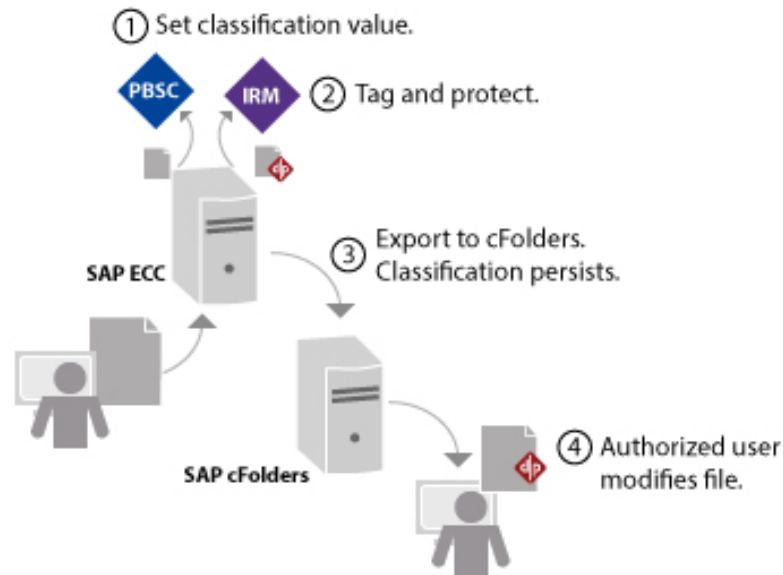


Figure 1-6: Exporting Files from SAP ECC to cFolders

(5) After the user finishes modifying the file, PBSC runs again, this time within SAP cFolders. Based on the presence of new keywords in the file's content, PBSC resets the classification value of the file (for example, to Export Security = EAR). (6) PBSC can also be configured to append a new classification (rather than overwrite it). IRM also runs in cFolders, and overwrites the old tag on the file with the new tag. IRM can also append a new tag, rather than overwrite.

When the file is imported back into SAP ECC from cFolders (7), it is managed by SAP ECC as a new version (different from the original that was exported). This new version is initially unclassified in SAP ECC, meaning there is no classification associated with the document in the *Security Classification Maintenance* screen. The file is reclassified when PBSC runs again (8) to analyze the contents of the file and determine the proper classification.

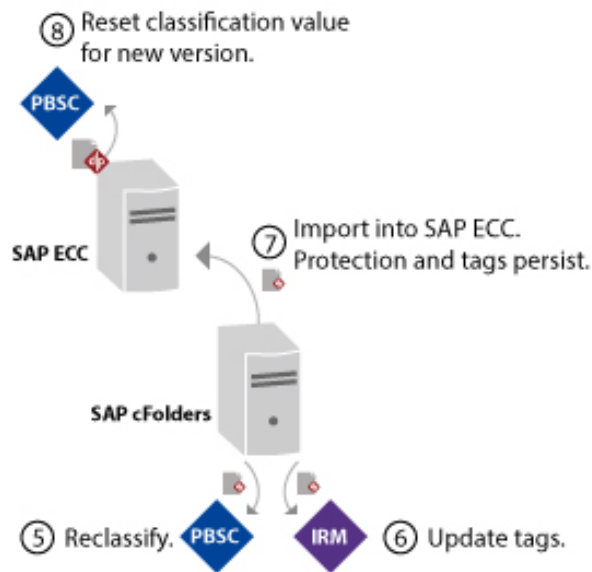


Figure 1-7: Importing Files Back into SAP ECC

---

## Contacting Technical Support

For help with NextLabs products, email Technical Support at [support@nextlabs.com](mailto:support@nextlabs.com).

## 2 Quick Reference for Set Up

This section provides links to the installation and configuration procedures for Dynamic Authorization Management for SAP and all the Entitlement Packs. In addition, this section lists the organizational roles typically responsible for performing each procedure.

Dynamic Authorization Management for SAP must be installed and configured on all implementations. Installing Entitlement Packs is optional.

Topics:

- [Explanation of User Roles](#)
- [Dynamic Authorization Management Installation Procedures \(All Cases\)](#)
- [Example 1: Configuring SAP ECC Only](#)
- [Example 2: Configuring SAP ECC, cFolders, and PLM](#)
- [Example 3: Configuring SAP ECC and EasyDMS](#)
- [Example 4: SAP ECC and BW](#)

If your implementation is not represented in one of the examples, go through the installation and configuration sections to identify the required procedures.

---

### Explanation of User Roles

[Table 2-1](#) lists users and associated permissions that are required to install and configure Dynamic Authorization Management for SAP and all Entitlement Packs. These roles are referenced in the tables that follow. The roles at your organization may vary. For example, an SAP Consultant might be responsible for the SAP ABAP Developer functions. However, the permission requirements are still relevant.

*Table 2-1: Installation and Configuration Roles*

Role	Description/Requirements
NextLabs Administrator	Must have Administrator privileges on the host where the NextLabs Policy Server and Policy Controller are installed; must have Administrator Profile password and NextLabs Administrator password
Basis Administrator	Must have permissions to perform Basis administrator functions, including installing .car files and configuring enterprise services within SAP
SAP Functional Consultant	Must have access to all tables, transactions, and programs within the /NEXTLABS/ namespace
SAP ABAP Developer	Must have a developer key

## Dynamic Authorization Management Installation Procedures (All Cases)

[Table 2-2](#) provides links to the procedures for installing and configuring Dynamic Authorization Management for SAP, including Entitlement Packs, and then performing basic configuration. These procedures must be performed for ALL implementations, regardless of the Entitlement Packs being deployed.

Begin with the procedures in [Table 2-2](#), then proceed to the procedures required for each Entitlement Pack.

**Note:** These procedures do not include NextLabs Control Center installation, which is covered in the *NextLabs Control Center Installation Guide*.

*Table 2-2: Procedures for Installing and Configuring Dynamic Authorization Management for SAP*

Procedure(s)	System Access	Role
<a href="#">Step 1: Installing the Policy Controller</a> on page 38	Policy Controller Administrator Password	NextLabs Administrator
<a href="#">Configuring the Policy Controller Communication Interface: Java Connector</a> on page 49	IIS configuration	NextLabs Administrator and SAP Basis Administrator
<a href="#">Enrolling Users from SAP into Control Center</a> on page 65	In SAP: /NEXTLABS/USER_EXTRACT and On Control Center: NextLabs Enrollment Manager	SAP Functional Consultant and NextLabs Administrator
<a href="#">Configuring SAP Actions</a> on page 69	Control Center Policy Server	NextLabs Administrator
<a href="#">Configuring SAP Obligations</a> on page 70	Control Center Policy Server	NextLabs Administrator

## Example 1: Configuring SAP ECC Only

[Table 2-3](#) lists the procedures for configuring an implementation that includes the Entitlement Pack for SAP ECC only. This is the most basic configuration for Entitlement Packs.

*Table 2-3: Configuring the Entitlement Pack for SAP ECC*

Procedure(s)	Transaction/System Access	Role
<a href="#">Activating Entitlement Packs (EPCONF)</a> on page 77	SM30 > /NEXTLABS/EPCONF	SAP Functional Consultant
<a href="#">Adding Composite Keys and Classification Values</a> on page 78	SE11 > /NEXTLABS/SECENH_CLS or > /NEXTLABS/CLS_APPEND	SAP Functional Consultant
<a href="#">Linking Composite Keys (SECENH)</a> on page 83	SM30 > /NEXTLABS/SECENH	SAP Functional Consultant
<a href="#">Mapping Security Fields (SECMPG)</a> on page 84	SM30 > /NEXTLABS/SECMPG	SAP Functional Consultant
<a href="#">Configuring Security Identifier/Composite Key Value Tables (EPVAL)</a> on page 86	SM30 > /NEXTLABS/EPVAL	SAP Functional Consultant

Table 2-3: Configuring the Entitlement Pack for SAP ECC (Continued)

Procedure(s)	Transaction/System Access	Role
<a href="#">Mapping Transaction Codes and UI Functions to Actions (ACTIONS) on page 90</a>	SM30 > /NEXTLABS/ACTION	SAP Functional Consultant
<a href="#">Configuring SAP Data Handling and Connection Settings on page 94</a>	SA38 > /NEXTLABS/CONCFG_MAINTAIN and SM30 > /NEXTLABS/CONCFG	SAP Functional Consultant
<a href="#">Configuring Number Range Intervals on page 103</a>	SM30 > /NEXTLABS/NRCONF	SAP Functional Consultant
<a href="#">Configuring Policy Checks Based on Transaction/UI Function on page 109</a>	SM30 > /NEXTLABS/CHKCLS	SAP Functional Consultant
<a href="#">Configuring Special Fields for the Security Classifications Maintenance Table on page 111</a>	SM30 > /NEXTLABS/EPCLS	SAP Functional Consultant
<a href="#">Defining How Multiple Security Classifications Should Be Applied on page 116</a>	SM30 > /NEXTLABS/OPTCFG	SAP Functional Consultant
<a href="#">Configuring the Transactions or Functions to Intercept on page 123</a>	/NEXTLABS/TXFLTR	SAP Functional Consultant
<a href="#">Configuring the NextLabs Number Range on page 104</a>	SNRO	SAP Functional Consultant
<a href="#">Configuration for Policy-based Security Classification on page 125</a>	SM30 > /NEXTLABS/PBSCFG SM30 > /NEXTLABS/PBSDIR SM30 > /NEXTLABS/PBSMAT /nFile SM36 and Control Center Policy Server	SAP Functional Consultant  and NextLabs Administrator
<a href="#">Configuration for Integrated Rights Management (IRM) on page 145</a>	/nFile SE38 > /NEXTLABS_IRM_DOWNLOAD SE38 > /NEXTLABS_IRM_CFX SM36 Control Center Policy Server Rights Management Server	SAP Functional Consultant  and NextLabs Administrator
<a href="#">Configuring Enhancement Implementations on page 137</a>	CMOD, SE18, SE24, SE37, SE38	SAP ABAP Developers

## Example 2: Configuring SAP ECC, cFolders, and PLM

Table 2-4 lists the procedure for configuring an Implementation that includes Entitlement Packs for SAP ECC, cFolders, and PLM. Additional procedures (which are not required with the most basic configuration, [Example 1: Configuring SAP ECC Only on page 26](#)), are indicated with a plus (+) symbol.

Table 2-4: Configuring the Entitlement Packs for SAP ECC, cFolders, and PLM

Add-on Steps	Procedure(s)	Transaction/System Access	Role
	<a href="#">Activating Entitlement Packs (EPCONF) on page 77</a>	SM30 > /NEXTLABS/EPCONF	SAP Functional Consultant

Table 2-4: Configuring the Entitlement Packs for SAP ECC, cFolders, and PLM (Continued)

Add-on Steps	Procedure(s)	Transaction/System Access	Role
	<a href="#">Adding Composite Keys and Classification Values</a> on page 78	se11 > /NEXTLABS/SECENH_CLS or > /NEXTLABS/CLS_APPEND	SAP Functional Consultant
	<a href="#">Linking Composite Keys (SECENH)</a> on page 83	SM30 > /NEXTLABS/SECENH	SAP Functional Consultant
	<a href="#">Mapping Security Fields (SECMPG)</a> on page 84	SM30 > /NEXTLABS/SECMPG	SAP Functional Consultant
	<a href="#">Configuring Security Identifier/ Composite Key Value Tables (EPVAL)</a> on page 86	SM30 > /NEXTLABS/EPVAL	SAP Functional Consultant
+	<a href="#">Configuring UI Functions</a> on page 88	SM30 > UIFUNC	SAP Functional Consultant
	<a href="#">Mapping Transaction Codes and UI Functions to Actions (ACTIONS)</a> on page 90	SM30 > /NEXTLABS/ACTION	SAP Functional Consultant
	<a href="#">Configuring SAP Data Handling and Connection Settings</a> on page 94	sa 38 > /NEXTLABS/CONCFG_MAINTAIN and SM30 > /NEXTLABS/CONCFG	SAP Functional Consultant
	<a href="#">Configuring Number Range Intervals</a> on page 103	SM30 > /NEXTLABS/NRCONF	SAP Functional Consultant
	<a href="#">Configuring Policy Checks Based on Transaction/UI Function</a> on page 109	SM30 > /NEXTLABS/CHKCLS	SAP Functional Consultant
	<a href="#">Configuring Special Fields for the Security Classifications Maintenance Table</a> on page 111	SM30 > /NEXTLABS/EPCLS	SAP Functional Consultant
+	<a href="#">Defining How Security Classifications and Access Control Contexts Should Be Applied</a> on page 112	SM30 > /NEXTLABS/EVLIDT	SAP Functional Consultant
	<a href="#">Defining How Multiple Security Classifications Should Be Applied</a> on page 116	SM30 > /NEXTLABS/OPTCFG	SAP Functional Consultant
+	<a href="#">Configuring Access Control Context Settings (PLM Only)</a> on page 121	SM30 > /NEXTLABS/ACCMNT	SAP Functional Consultant
	<a href="#">Configuring the Transactions or Functions to Intercept</a> on page 123	/NEXTLABS/TXFLTR	SAP Functional Consultant
	<a href="#">Configuring the NextLabs Number Range</a> on page 104	SNRO	SAP Functional Consultant
	<a href="#">Configuration for Policy-based Security Classification</a> on page 125	SM30 > /NEXTLABS/PBSCFG SM30 > /NEXTLABS/PBSDIR SM30 > /NEXTLABS/PBSMAT /nFile SM36 Control Center Policy Server	SAP Functional Consultant  and NextLabs Administrator
	<a href="#">Configuration for Integrated Rights Management (IRM)</a> on page 145	/nFile SE38 > /NEXTLABS_IRM_DOWNLOAD SE38 > /NEXTLABS_IRM_CFX SM36 Control Center Policy Server Rights Management Server	SAP Functional Consultant  and NextLabs Administrator
	<a href="#">Configuring Enhancement Implementations</a> on page 137	CMOD, SE18, SE24, SE37, SE38	SAP ABAP Developers



## Example 3: Configuring SAP ECC and EasyDMS

Table 2-5 lists the procedure for configuring the Entitlement Packs for SAP ECC and EasyDMS. Additional procedures (which are not required with the most basic configuration, [Example 1: Configuring SAP ECC Only](#) on page 26), are indicated with a plus (+) symbol.

Table 2-5: Configuring the Entitlement Packs for SAP ECC and EasyDMS

Add on Step	Procedure(s)	Transaction/System Access	Role
	<a href="#">Activating Entitlement Packs (EPCONF)</a> on page 77	SM30 > /NEXTLABS/EPCONF	SAP Functional Consultant
	<a href="#">Adding Composite Keys and Classification Values</a> on page 78	SE11 > /NEXTLABS/SECENH_CLS or > /NEXTLABS/CLS_APPEND	SAP Functional Consultant
	<a href="#">Linking Composite Keys (SECENH)</a> on page 83	SM30 > /NEXTLABS/SECENH	SAP Functional Consultant
	<a href="#">Mapping Security Fields (SECMPG)</a> on page 84	SM30 > /NEXTLABS/SECMPG	SAP Functional Consultant
	<a href="#">Configuring Security Identifier/Composite Key Value Tables (EPVAL)</a> on page 86	SM30 > /NEXTLABS/EPVAL	SAP Functional Consultant
+	<a href="#">Configuring UI Functions</a> on page 88	SM30 > UIFUNC	SAP Functional Consultant
	<a href="#">Mapping Transaction Codes and UI Functions to Actions (ACTIONS)</a> on page 90	SM30 > /NEXTLABS/ACTION	SAP Functional Consultant
	<a href="#">Configuring SAP Data Handling and Connection Settings</a> on page 94	SA38 > /NEXTLABS/CONCFG_MAINTAIN and SM30 > /NEXTLABS/CONCFG	SAP Functional Consultant
+	<a href="#">Configuring View Filtering (EasyDMS Only)</a> on page 102	SM30 > /NEXTLABS/EDMS	SAP Functional Consultant
	<a href="#">Configuring Number Range Intervals</a> on page 103	SM30 > /NEXTLABS/NRCONF	SAP Functional Consultant
	<a href="#">Configuring Policy Checks Based on Transaction/UI Function</a> on page 109	SM30 > /NEXTLABS/CHKCLS	SAP Functional Consultant
	<a href="#">Configuring Special Fields for the Security Classifications Maintenance Table</a> on page 111	SM30 > /NEXTLABS/EPCLS	SAP Functional Consultant
	<a href="#">Defining How Multiple Security Classifications Should Be Applied</a> on page 116	SM30 > /NEXTLABS/OPTCFG	SAP Functional Consultant
	<a href="#">Configuring the Transactions or Functions to Intercept</a> on page 123	/NEXTLABS/TXFLTR	SAP Functional Consultant
	<a href="#">Configuring the NextLabs Number Range</a> on page 104	SNRO	SAP Functional Consultant
	<a href="#">Configuration for Policy-based Security Classification</a> on page 125	SM30 > /NEXTLABS/PBSCFG SM30 > /NEXTLABS/PBSDIR SM30 > /NEXTLABS/PBSMAT /nFile SM36 Control Center Policy Server	SAP Functional Consultant  and NextLabs Administrator

Table 2-5: Configuring the Entitlement Packs for SAP ECC and EasyDMS (Continued)

Add on Step	Procedure(s)	Transaction/System Access	Role
	<a href="#">Configuration for Integrated Rights Management (IRM) on page 145</a>	/nFile SE38 > /NEXTLABS_IRM_DOWNLOAD SE38 > /NEXTLABS_IRM_CFX SM36 Control Center Policy Server Rights Management Server	SAP Functional Consultant  and NextLabs Administrator
	<a href="#">Configuring Enhancement Implementations on page 137</a>	CMOD, SE18, SE24, SE37, SE38	SAP ABAP Developers

## Example 4: SAP ECC and BW

Table 2-6 lists the procedure for configuring the Entitlement Packs for SAP ECC and SAP BW. Additional procedures (which are not required with the most basic configuration, [Example 1: Configuring SAP ECC Only on page 26](#)), are indicated with a plus (+) symbol.

Table 2-6: Configuring the Entitlement Packs for SAP ECC and BW

Add on Step	Procedure(s)	Transaction/System Access	Role
	<a href="#">Activating Entitlement Packs (EPCONF) on page 77</a>	SM30 > /NEXTLABS/EPCONF	SAP Functional Consultant
	<a href="#">Adding Composite Keys and Classification Values on page 78</a>	SE11 > /NEXTLABS/SECENH_CLS or > /NEXTLABS/CLS_APPEND	SAP Functional Consultant
	<a href="#">Linking Composite Keys (SECENH) on page 83</a>	SM30 > /NEXTLABS/SECENH	SAP Functional Consultant
	<a href="#">Mapping Security Fields (SECMPG) on page 84</a>	SM30 > /NEXTLABS/SECMPG	SAP Functional Consultant
	<a href="#">Configuring Security Identifier/Composite Key Value Tables (EPVAL) on page 86</a>	SM30 > /NEXTLABS/EPVAL	SAP Functional Consultant
+	<a href="#">Configuring UI Functions on page 88</a>	SM30 > UIFUNC	SAP Functional Consultant
	<a href="#">Mapping Transaction Codes and UI Functions to Actions (ACTIONS) on page 90</a>	SM30 > /NEXTLABS/ACTION	SAP Functional Consultant
	<a href="#">Configuring SAP Data Handling and Connection Settings on page 94</a>	SA38 > /NEXTLABS/CONCFG_MAINTAIN and SM30 > /NEXTLABS/CONCFG	SAP Functional Consultant
	<a href="#">Configuring Number Range Intervals on page 103</a>	SM30 > /NEXTLABS/NRCONF	SAP Functional Consultant
	<a href="#">Configuring Policy Checks Based on Transaction/UI Function on page 109</a>	SM30 > /NEXTLABS/CHKCLS	SAP Functional Consultant
	<a href="#">Configuring Special Fields for the Security Classifications Maintenance Table on page 111</a>	SM30 > /NEXTLABS/EPCLS	SAP Functional Consultant
+	<a href="#">Defining How Security Classifications and Access Control Contexts Should Be Applied on page 112</a>	SM30 > /NEXTLABS/EVLIDT	SAP Functional Consultant

Table 2-6: Configuring the Entitlement Packs for SAP ECC and BW (Continued)

Add on Step	Procedure(s)	Transaction/System Access	Role
	<a href="#">Defining How Multiple Security Classifications Should Be Applied</a> on page 116	SM30 > /NEXTLABS/OPTCFG	SAP Functional Consultant
	<a href="#">Configuring the Transactions or Functions to Intercept</a> on page 123	/NEXTLABS/TXFLTR	SAP Functional Consultant
	<a href="#">Configuring the NextLabs Number Range</a> on page 104	SNRO	SAP Functional Consultant
	<a href="#">Configuration for Policy-based Security Classification</a> on page 125	SM30 > /NEXTLABS/PBSCFG SM30 > /NEXTLABS/PBSDIR SM30 > /NEXTLABS/PBSMAT /nFile SM36 Control Center Policy Server	SAP Functional Consultant  and NextLabs Administrator
	<a href="#">Configuration for Integrated Rights Management (IRM)</a> on page 145	/nFile SE38 > /NEXTLABS_IRM_DOWNLOAD SE38 > /NEXTLABS_IRM_CFX SM36 Control Center Policy Server Rights Management Server	SAP Functional Consultant  and NextLabs Administrator
	<a href="#">Configuring Enhancement Implementations</a> on page 137	CMOD, SE18, SE24, SE37, SE38	SAP ABAP Developers



# 3 Installation and Set Up

---

This section describes the basic installation and set up procedures required for the Entitlement Manager for SAP.

Topics:

- [Before You Begin](#)
- [Step 1: Installing the Policy Controller](#)
- [Step 2: Installing Dynamic Authorization Management for SAP](#)
- [Step 3: Making the NextLabs Namespace Modifiable](#)
- [Configuring the Policy Controller Communication Interface: Java Connector](#)

**Note:** After installation and setup is complete, additional configuration steps are required for basic and optional features. See:

- [Configuring Basic Features](#)
- [Configuring Optional Features](#)

---

## Before You Begin

Before you begin, ensure that your environment meets the following requirements:

- [Supported Platforms and Products](#)
- [Firewall and Port Requirements](#)
- [Gateway Requirements](#)
- [Supported Configurations](#)

## Supported Platforms and Products

The following platforms are supported:

- For the Policy Controller for Microsoft File Servers: Windows Server 2008, 64-bit; with Application Server Role, with ASP.net enabled
- For the Policy Controller for Java: Windows Server 2008, or Red Hat Enterprise Linux 6.6 and 7.0
- SAP ECC 6.0 and above
- SAP PLM 6.0 and above
- SAP EasyDMS 7.0 and above

- SAP BW 7.3 and 7.4
- SAP cFolders 4.5
- NextLabs Control Center 7.7
- NextLabs Policy Controller 7.7
- NextLabs Policy Studio 7.7
- NextLabs Rights Management Server 8.2
- NextLabs Java Connector 7.6

### Firewall and Port Requirements

Each Windows PC where the NextLabs Enforcer is being installed must be configured to enable communication between the Control Center over the required ports. These port settings are required to enable the Enforcer to send heartbeat messages to the Control Center and receive policy bundles in return. The ports are also used by the administrative applications that connect to the Enforcers for status updates and other information.

The following are the default port assignments:

- Push Deployment port: 2000
- Policy Controller port: 8443
- RMI Registry port: 1099. If your implementation uses the Policy Controller for Java, an exception for the RMI Registry port. To change this port assignment, you need to modify the `JavaSDKService.properties` file.

**Note:** The port numbers listed here are defaults. If your Control Center uses different port assignments, the exception ports on each Windows PC must match the Control Center port assignments. The Push Deployment port in particular is special in that it can be set differently for each Enforcer profile, in the *Enforcer Profile Configuration* tab in the Control Center Administrator interface. For details, see the *Control Center Administrator's Guide*.

### Configuring Firewall Settings

Configure firewall settings on each Windows PC where the enforcer is being installed to allow connections to the appropriate ports.

#### Procedure

- 1 In the Windows Control Panel, select **System and Security**, then select **Windows Firewall**.
- 2 Select **Advanced settings**.
- 3 Select **Inbound Rules**, then select **New Rule**, as shown in [Figure 3-1](#).

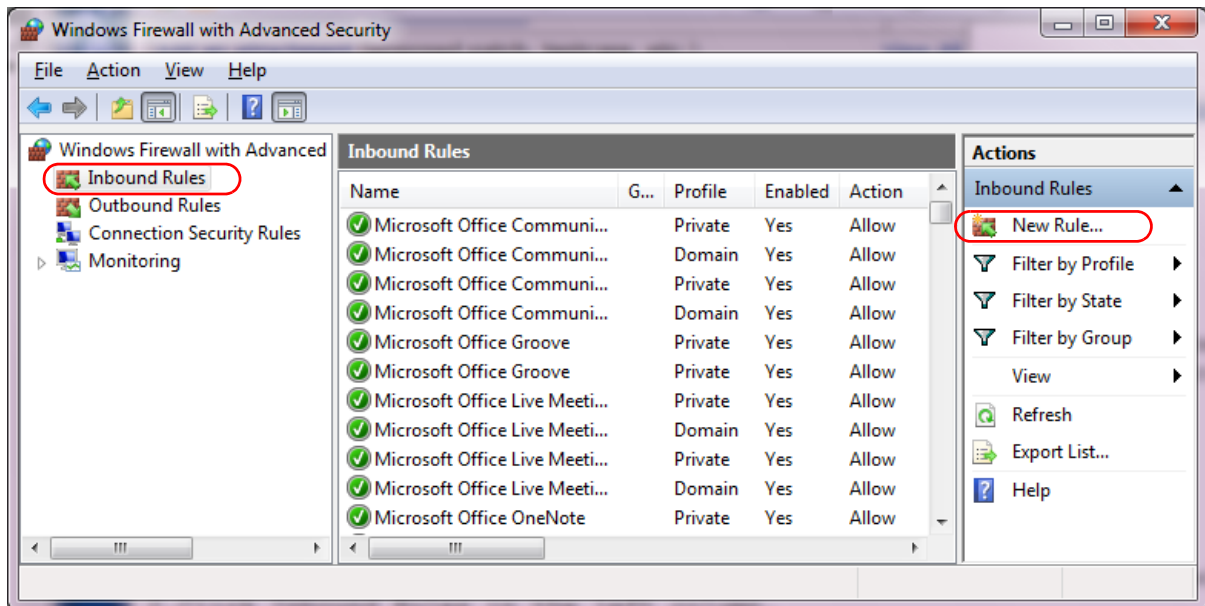


Figure 3-1: Creating a new inbound rule

- 4 Select **Port** as the type of rule to create, then click **Next**.
- 5 Select **TCP**, then select **Specific local ports** and type the port number or numbers, as shown in Figure 3-2, then click **Next**.

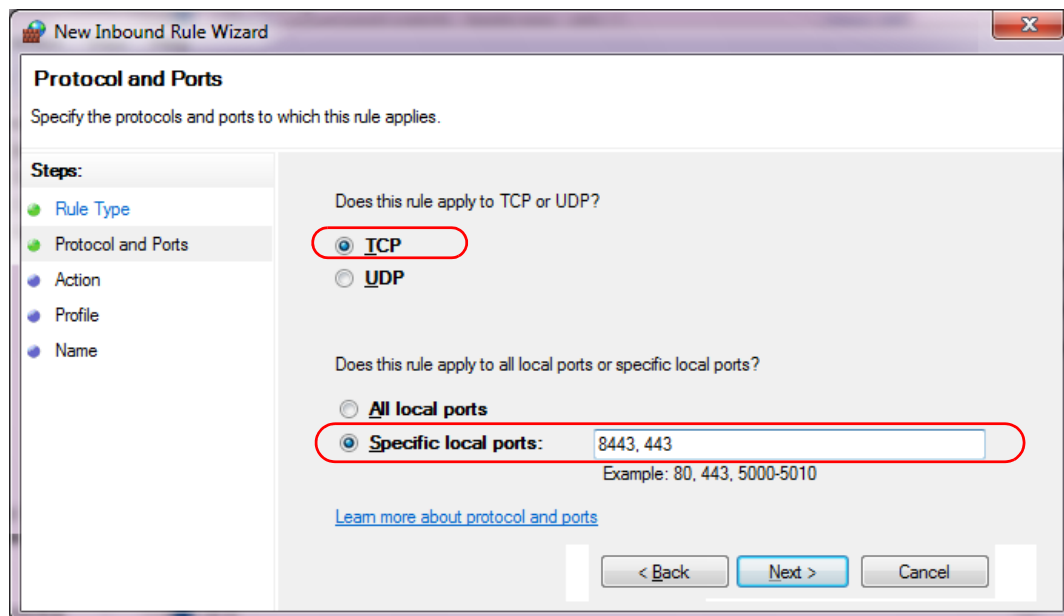


Figure 3-2: Specifying the ports

- 6 Select **Allow the Connection**, then click **Next**.
- 7 Select **Domain**, **Private**, and **Public**, then click **Next**.
- 8 Type a name and description for the port rule, then click **Finish**.

## Gateway Requirements

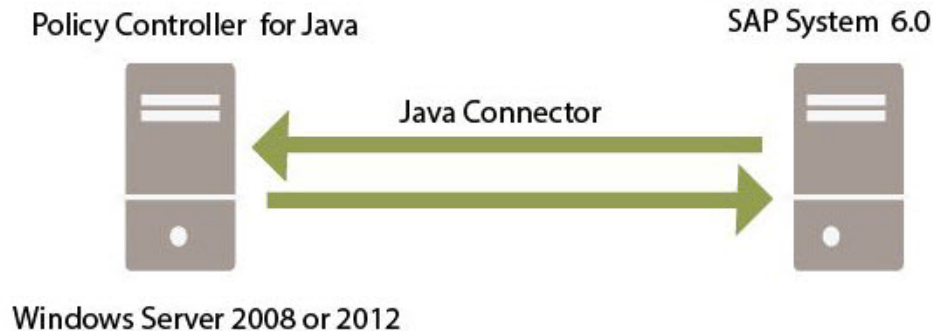
If you are installing Dynamic Authorization for SAP on a server with SAP Basis release 750 or later, see the [Configuring Gateway Settings](#) section for the steps necessary to ensure that the SAP instance is able to communicate with the Policy Controller.

## Supported Configurations

Dynamic Authorization Management for SAP offers several configuration options to support different platforms and communication interfaces. The SAP system can be installed on any platform certified by SAP. The following sections describe the configurations that are supported.

### 1. Policy Controller for Java + Java Connector on Windows

The NextLabs Policy Controller for Java can be installed on Microsoft Server 2008 or 2012. When you install the Java Connector with the Policy Controller for Java, you must manually perform all the installation steps for installing the Java Connector.



*Figure 3-3: Java Policy Controller + Java Connector on Windows*

### 2. Policy Controller for Java + Java Connector on Red Hat Linux

The NextLabs Policy Controller for Java can also run on Red Hat Linux. All the steps of the Java Connector installation on Red Hat Linux must be performed manually; no installation script is available.



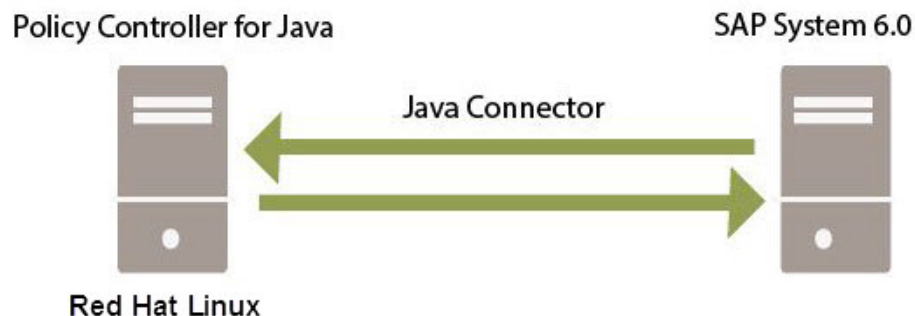


Figure 3-4: Policy Controller for Java + Java Connector on RHEL

### 3. Server Policy Controller + Java Connector on Windows

The NextLabs Policy Controller for Windows File Servers only runs in Windows Server 2008 or 2012 environments. If you install Java Connector and the Server Policy Controller on Windows, NextLabs supplies an installation script that installs files in the proper locations and configures a properties file.

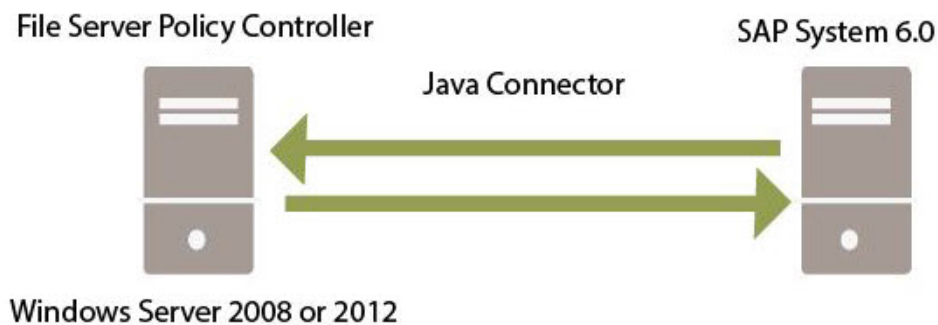


Figure 3-5: Server Policy Controller + Java Connector on Windows

## Installation Procedures for Supported Configurations

This section provides the order of steps that should be followed:

- [Step 1: Installing the Policy Controller](#)
- [Step 2: Installing Dynamic Authorization Management for SAP](#)
- [Step 3: Making the NextLabs Namespace Modifiable](#)
- [Configuring the Policy Controller Communication Interface: Java Connector](#)

---

## Step 1: Installing the Policy Controller

You must install one of the following Policy Controllers:

- The Server Policy Controller.
- The Policy Controller for Java.

For more information about installing Policy Controllers, see the Installing Policy Controllers section in the *NextLabs Control Center 7.7 Installation Guide*.

### **Next Steps**

The next step is [Step 2: Installing Dynamic Authorization Management for SAP](#) on page 38.

---

## Step 2: Installing Dynamic Authorization Management for SAP

This section provides information on a new installation of Dynamic Authorization Management for SAP. NextLabs Professional Services supplies you with the appropriate `.car` files for Dynamic Authorization Management products you are installing. This section includes information on the following topics:

- [Installation Files](#) on page 38
- [Installation Sequence and Procedure](#) on page 39
- [Install the `.car` files for additional Entitlement Packs as needed.](#) on page 39

### **Installation Files**

The following table lists the installation files for Dynamic Authorization Management for SAP and for different Entitlement Packs.

*Table 3-1: Installation Files for Dynamic Authorization Management for SAP*

Product	Installation File
Dynamic Authorization Management for SAP	NextLabs_Base_770_001.CAR
Entitlement Pack for SAP ECC	NXLECC_770_004.CAR
Entitlement Pack for SAP DFPS	NXLDFPS_780.CAR

## Installation Sequence and Procedure

As described in [Table 3-1](#), Dynamic Authorization Management for SAP and each applicable Entitlement Pack has its own `.car` files (Compressed Archive files). The procedure for installing the `.car` files is the same, no matter which products you are installing.

Install files in this sequence:

- 1 Install Dynamic Authorization Management for SAP, either on the SAP ECC system, on the SAP cFolders system, or on both, using the `NextLabs_Base_770_001.CAR` file.
- 2 If your base ABAP data system is SAP ECC, install the appropriate `.car` file for the Entitlement Pack for SAP ECC:
  - If your implementation includes support for ACCs, use the `NextLabs_ECC_AC-C_DMS_770.CAR` file.
  - If you do not require support for ACCs, use the `NextLabs_ECC_770_004.CAR` file.
- 3 Install the `.car` files for additional Entitlement Packs as needed.

### **Before You Begin**

Contact NextLabs Support at [support@nextlabs.com](mailto:support@nextlabs.com) to obtain the necessary installation passwords.

### **Procedure**

- 1 In the SAP interface, log in to client 000.
- 2 Enter transaction `SAINT`. The *Add-on Installation Tool* screen shows a list of all pre-configured add-ons and systems.
- 3 Click **Start** to begin a new Add-on Installation.

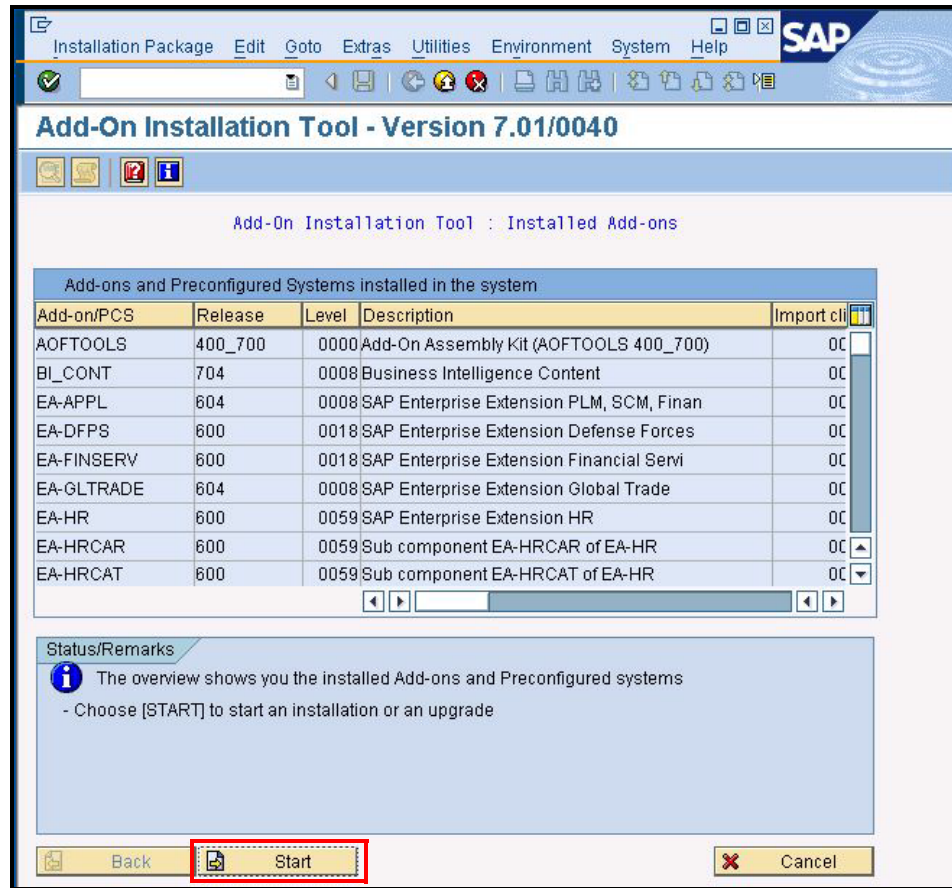


Figure 3-6: Starting a New Add-On Installation

- 4 Locate an Installation Package from Installation Package > Load packages > From Front End.

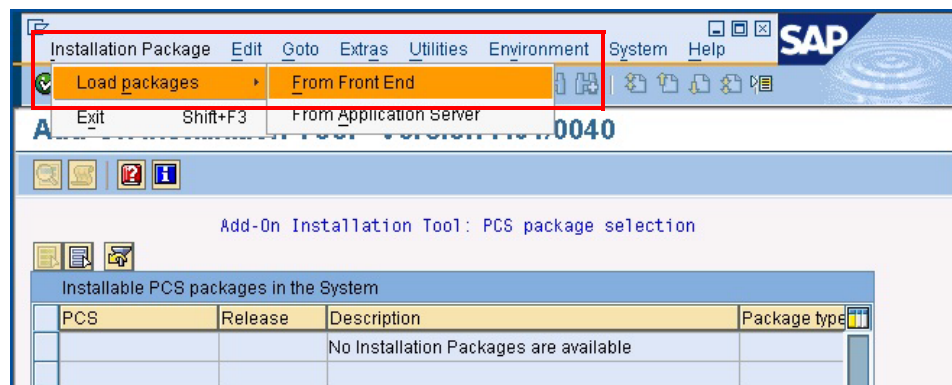


Figure 3-7: Loading an Installation Package

- 5 Navigate to the appropriate .car file for your installation, then click **Open**. To identify the appropriate file, see the section [Installation Sequence and Procedure](#) on page 39.
- 6 When prompted, click **Decompress** to extract the installer files.

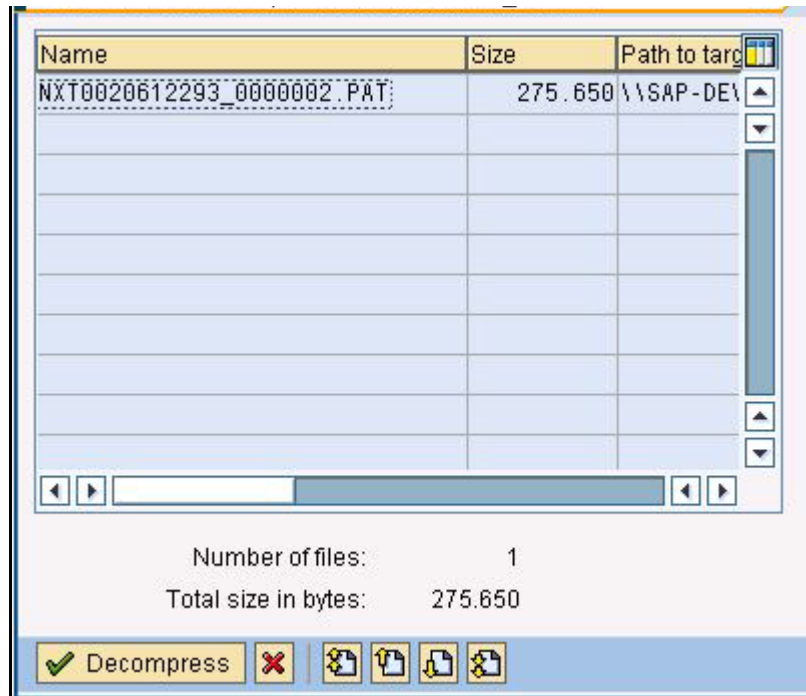


Figure 3-8: Decompressing the Installation Files

- 7 Click **Start**.

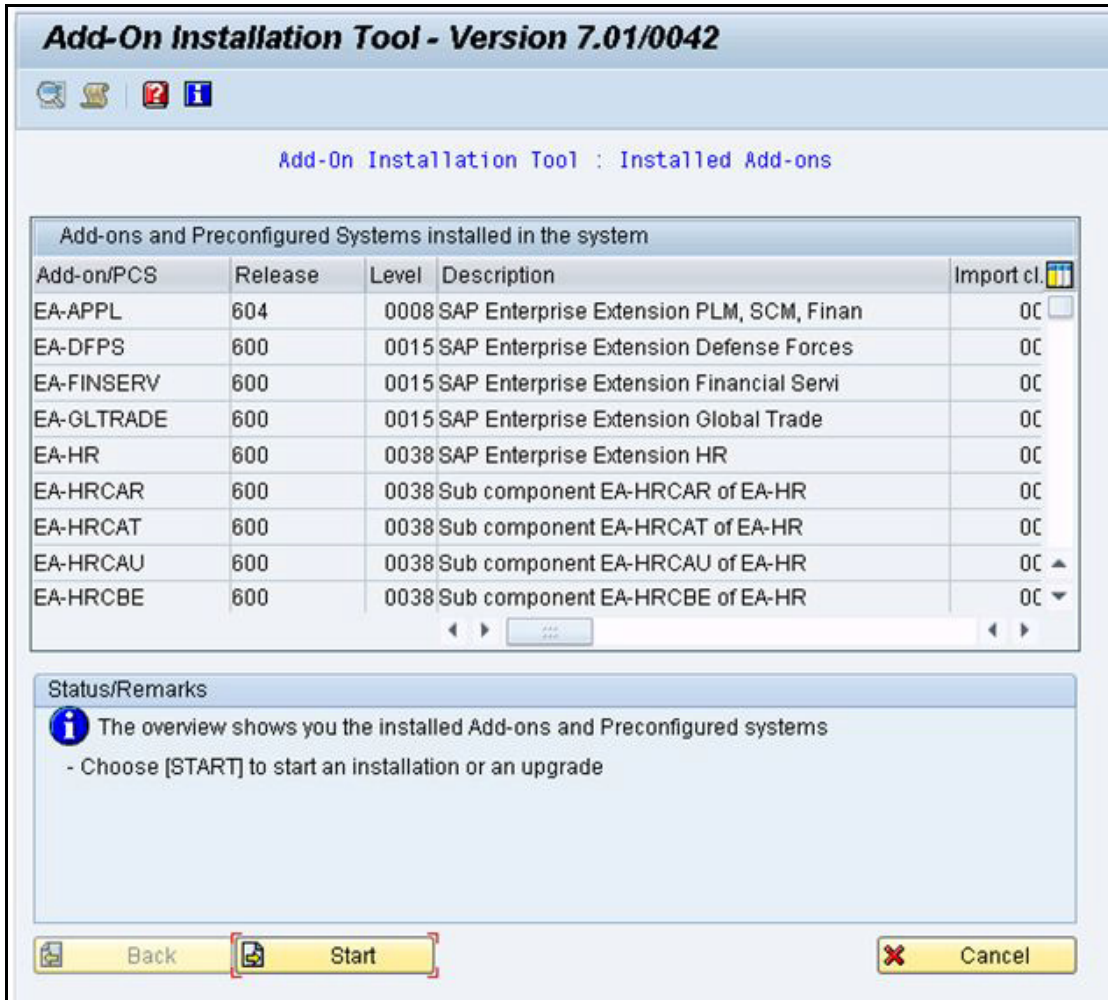


Figure 3-9: Starting the Installation

- 8 Select the Installation Package from the Installation queue and click **Continue** to launch the installation.

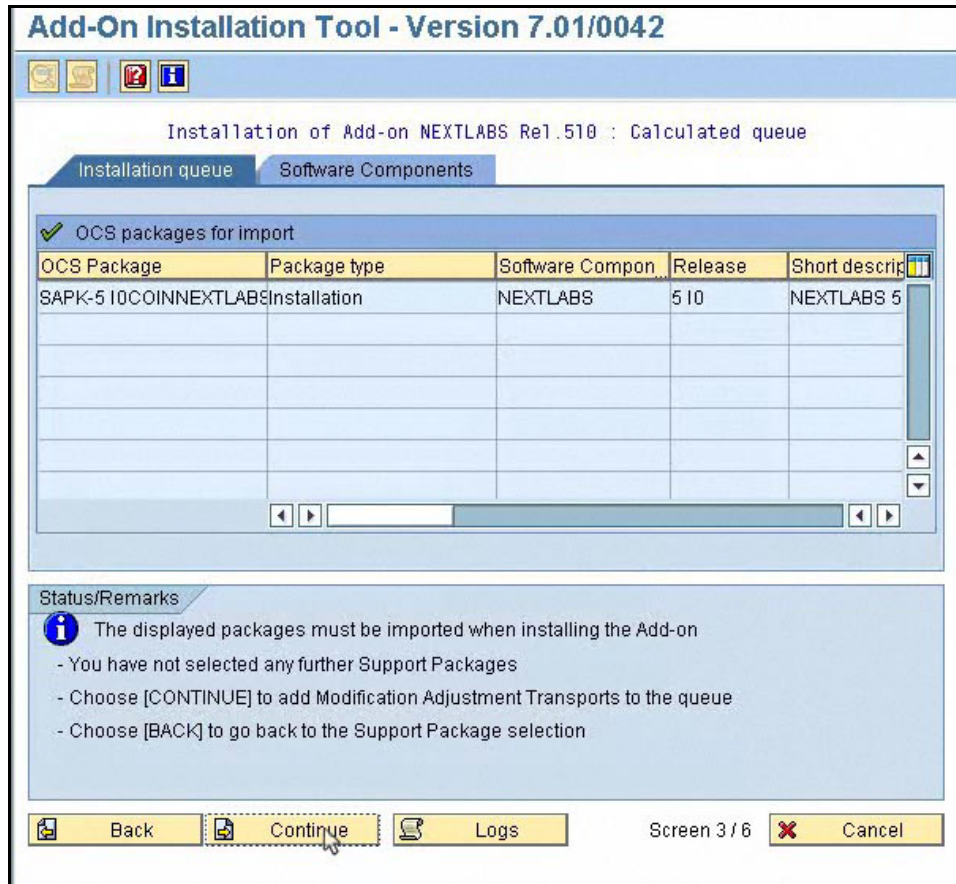


Figure 3-10: Selecting the Installation Package from the Queue

The screen shown in Figure 3-11 appears. You do not need to enter any information.



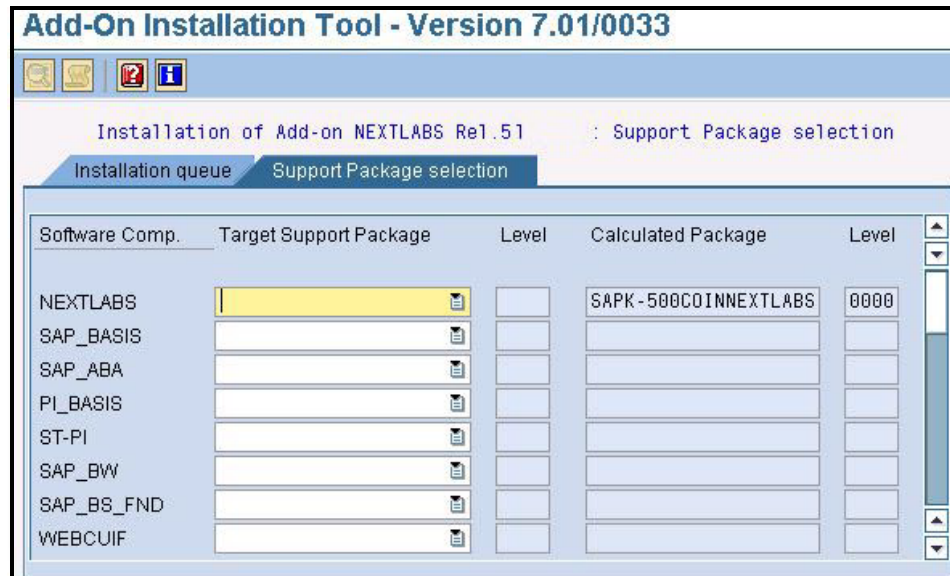


Figure 3-11: Add-On Installation Tool continued

- 9 Click **Continue** three times through the screens that display.
- 10 When prompted to add Modification Adjustment Transports to the queue, click **No**.

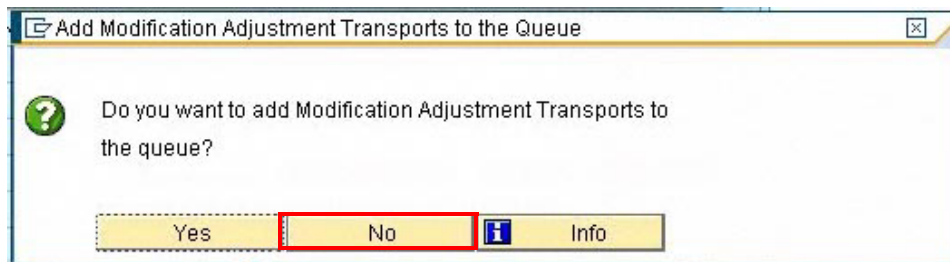
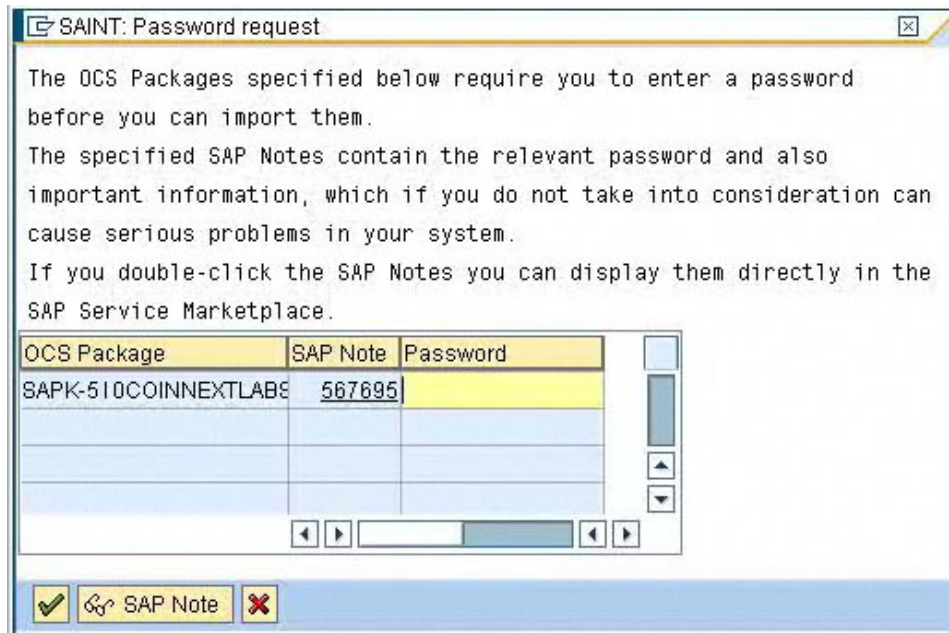


Figure 3-12: Modification Adjustment Transports

- 11 When prompted, enter the appropriate password for the OCS package then click **OK**. You must obtain this password from NextLabs.





*Figure 3-13: Enter the Appropriate Password*

After the installation process runs, a message appears stating that the installation succeeded.

- 12 Click **Logs** to view the installation Logs and confirm the installation was successful, then click **Finish**.

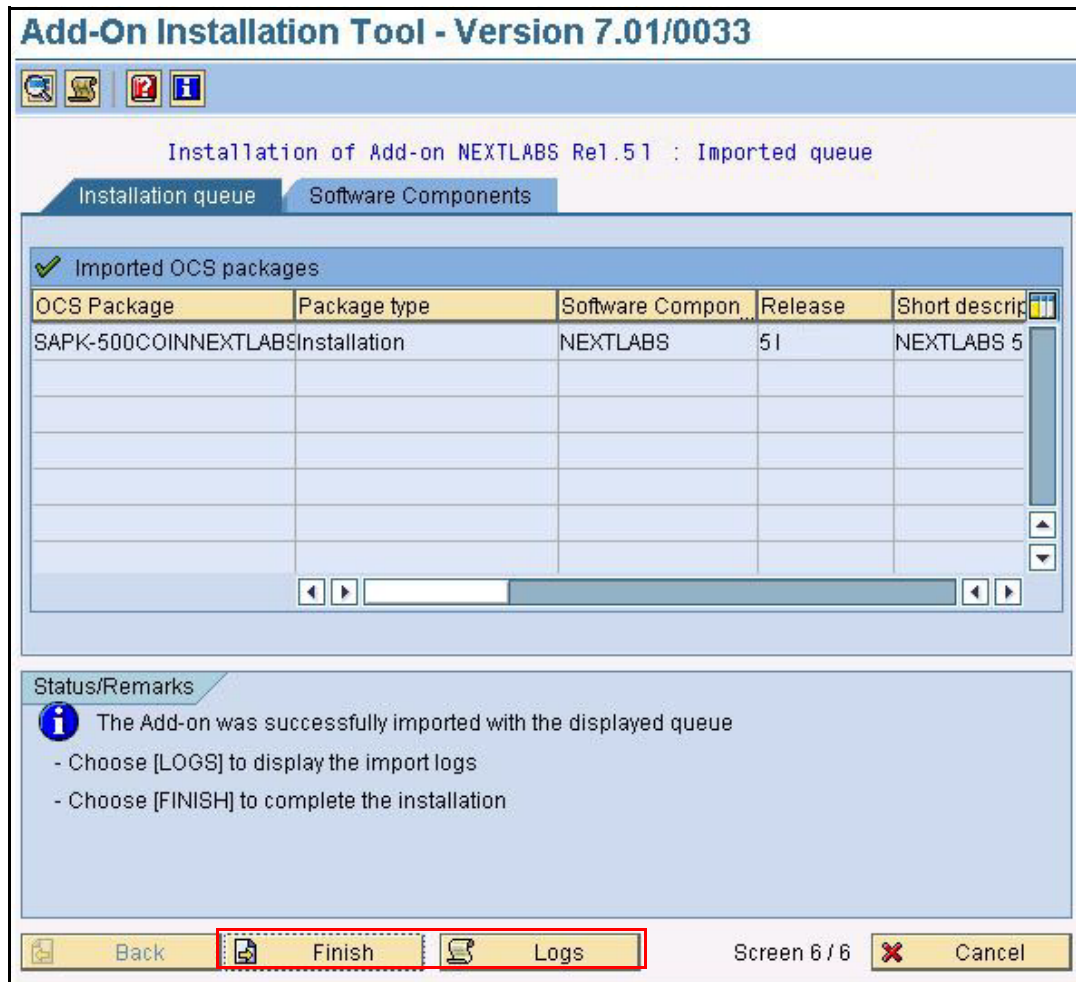


Figure 3-14: Viewing Logs and Completing the Installation

When you click **Finish**, the final screen indicates that the installation succeeded.

**Note:** You can verify in the Status/Remarks that the installation was successful. You can also scroll through the components to locate the newly installed NextLabs Add-on.

- 13 Repeat the preceding steps, depending on the type of installation you are performing. See [Installation Sequence and Procedure](#) on page 39.
- 14 After you ave installed all the necessary NextLabs components, log off client 000 and log in to client 100. This step is required to continue with the configuration.

### Next Steps

The next step is [Step 3: Making the NextLabs Namespace Modifiable](#) on page 47.

---

## Step 3: Making the NextLabs Namespace Modifiable

Upon initial system setup, you must configure the NextLabs Namespace to be modifiable. This step must be performed prior to configuring the NextLabs Number Range. The NextLabs Namespace should be made unmodifiable after configuration is complete.

**Note:** The only modifications supported for the NextLabs namespace are officially released NextLabs product code. Customers should not store other modifications to code in the NextLabs namespace because it can result in installation and upgrade issues.

### Procedure

- 1 In the SAP interface, enter transaction `SE03`. The *Transport Organizer Tools* screen appears.
- 2 Select **Set System Change Option**.

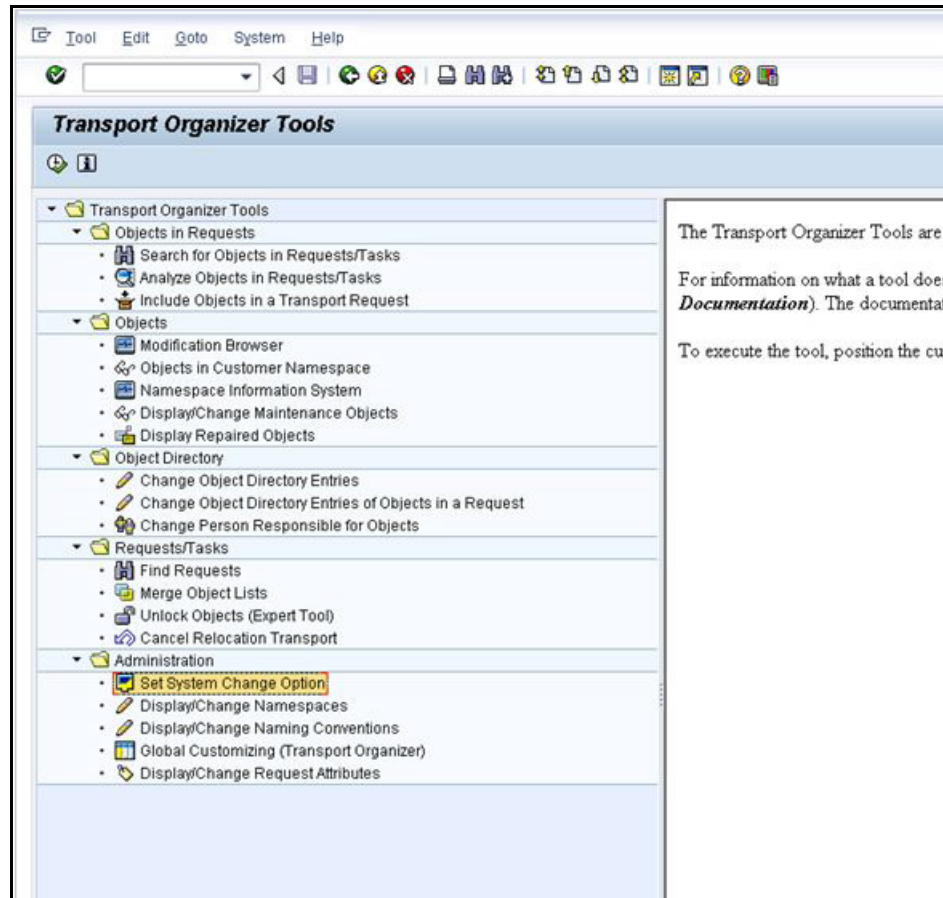


Figure 3-15: Transport Organizer Tools

- 3 In the *System Change Option* screen, scroll down to the NextLabs NameSpace.
- 4 In the **Modifiable** field, change Not Modifiable to Modifiable.

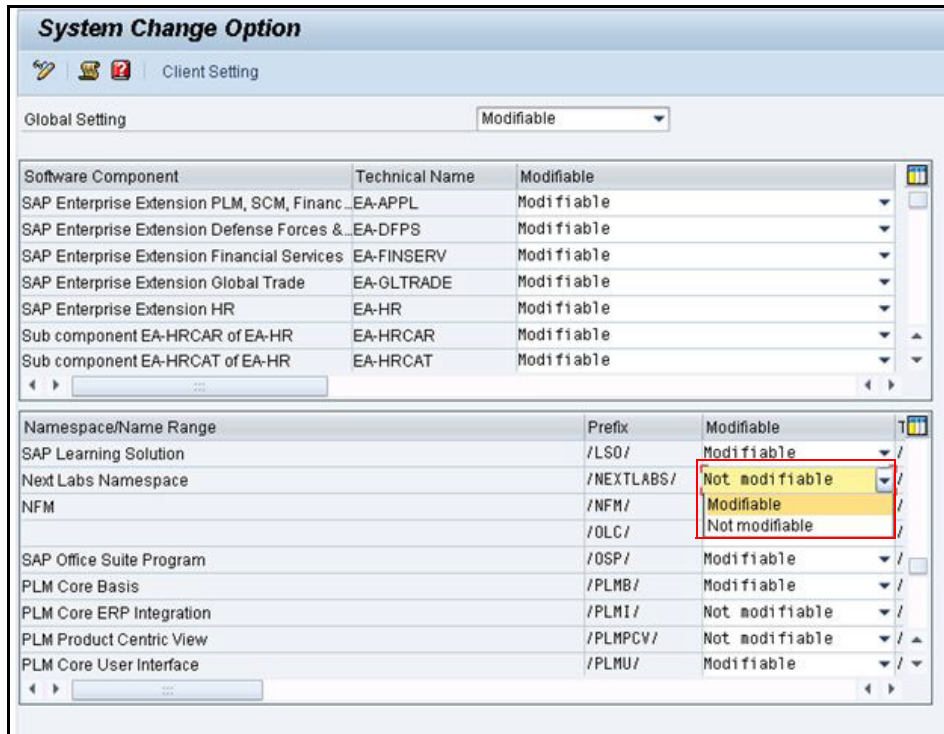


Figure 3-16: Making the NextLabs Namespace Modifiable

5 Click the **Save** button.

### Next steps

The next step is to configure the communication interface between the Policy Controller and SAP. See [Configuring the Policy Controller Communication Interface: Java Connector](#) on page 49.

## Configuring the Policy Controller Communication Interface: Java Connector

Setting up the communication interface between the Policy Controller and SAP Agent involves the following procedures:

- Verify that all prerequisites are met (see [Before You Install](#) on page 50)
- [Installing the Java Connector](#) on page 51
- [Configuring the RFC Connection](#) on page 56
- [Configuring the SAP JavaSDK Properties File \(Manual Installation only\)](#) on page 60
- [Configuring the Windows System Services File \(Manual and Script Installation\)](#) on page 61
- [Testing the Java Connector Configuration](#) on page 62

## Before You Install

The procedures discussed in this section must be performed before you set up the NextLabs Java Connector.

### Install C++ Runtime Environment on the Policy Controller Host

Before installing the Java Connector on Windows, you must install runtime components of Visual C++ libraries.

For 64 bit machines:

- Install the 64-bit version of Microsoft Visual C++ 2005 SP1 Redistributable Package. This download is available for free from the Microsoft Download Center.
- Install the patch for Microsoft Visual C++ 2005 SP1. This download is available for free from the Microsoft Download Center.

For 32 bit machines:

- Install the 32-bit version Microsoft Visual C++ 2005 SP1 Redistributable Package. This download is available for free from the Microsoft Download Center.

### Define an SAP Gateway Host and Port

The SAP Gateway Host name and port must be defined within SAP ECC. In the steps below, the host name and port must be inserted into a properties file used by the Java Policy Connector.

### Create an RFC User

An RFC user must already be configured within the SAP system (of type Communication, with the assigned Communication role), before you configure the Java Policy Connector. In addition, the RFC User must have the following authorizations defined as RFC Name Values:

- /NEXTLABS/\*
- RFC1
- SDIFRUNTIME
- SYST

In the procedures that follow, the RFC user must be inserted into a properties file used by the Java Policy Connector.

### Locate Required Values for Configuration

Before you install and configure the Java Connector, make sure you have all the values that you need to enter as part of set up. Required parameters are listed in [Table 3-2](#), along with

procedures for how to locate the values (where applicable). You need these values whether you are installing manually or using the install script.

*Table 3-2: Required Parameters for Configuring the JavaSDK.properties File (for both Script Install and Manual Procedure)*

Parameter	Explanation	How to Retrieve
pc_hostname	FQDN for host where the Policy Controller is installed	Contact IT
ashost	Host where SAP ECC is installed	Contact IT
user	RFC user	This user should be created by the Basis Administrator. This user should be Communication type and have the appropriate role for a RFC user, including the RFC name values: <ul style="list-style-type: none"> <li>• /NEXTLABS/*</li> <li>• RFC1</li> <li>• SDIFRUNTIME</li> <li>• SYST</li> </ul>
passwd	Password for the RFC User	If you are performing a manual install, you must encrypt the RFC User password (the script installation does this automatically, so you can just enter the plain text version of the password in the installation menu. For a manual installation, this password should be encrypted using the mkpasswd.bat utility supplied with the installation of the Control Center: <ul style="list-style-type: none"> <li>• On the host where the Policy Server is installed, open a command prompt and browse to &lt;install dir&gt;Policy Server/tools/crypt.</li> <li>• Run mkpasswd.bat -password &lt;plain text password&gt;</li> <li>• The encrypted version of the plain text password appears. Copy it for use when you configure the SAP JavaSDK Properties File manually.</li> </ul>
gwhost	FQDN of host where SAP ECC is installed	Contact IT
gwserv and port	The Gateway Service Name and port defined within SAP ECC	This gateway service name is configured in the following location: <ul style="list-style-type: none"> <li>• Enter transaction SMGW.</li> <li>• Navigate to Goto &gt; Expert functions &gt; Host Name Buffer.</li> <li>• At the bottom of the screen, Gateway Service Ports and Names are defined.</li> </ul>
progid	The program ID for the Remote Function Call defined for the Java Connector communication interface	This program ID was defined in the procedures discussed in the section <a href="#">Configuring the RFC Connection</a> on page 56.
<system>	The SAP system name	Contact the Basis Administrator

## Installing the Java Connector

If your implementation includes the Server Policy Controller, the Java Connector can be installed using a script or manually. If your implementation includes the Policy Controller for Java, the Java Connector must be installed manually.

- [Script Installation of the Java Connector for the Server Policy Controller](#) on page 52
- [Manual Installation of the Java Connector for the Server Policy Controller](#) on page 55
- [Manual Installation of the Java Connector for the Policy Controller for Java](#) on page 55

### Script Installation of the Java Connector for the Server Policy Controller

If you are installing the Java Connector with the Server Policy Controller, a script is available that automatically installs and un-installs the Java Connector files and adds the required values to the Properties file. This script is only available for Server Policy Controllers running on Windows Server 2008 and 2012, and it must be run from the command line.

**Note:** Although the script automatically installs and uninstalls files, you still need to enter values in the Properties file when running the script.

#### **Before You Begin**

- Obtain the Java Connector installation file from NextLabs. For more information, contact NextLabs Technical Support at [support@nextlabs.com](mailto:support@nextlabs.com).
- To stop the Policy Controller, you must have the profile password defined in the Control Center *Administrator* interface.

#### **Procedure**

- 1 Stop the Policy Controller, if it is running.
- 2 Locate the Java Connector installation file from NextLabs support (SAPJCo-Entitlement-Manager-<version number>.zip).
- 3 Open a command prompt as Administrator and navigate to the folder where SAP Java Connector zip file is extracted.
- 4 Run `deployManager.bat`. The install menu appears in the console.





Figure 3-17: JCO Deployment Manager First Screen

- 5 In the first screen, enter 1 and specify the root location of the Policy Controller. The default is C:\Program Files\Nextlabs\Policy Controller.

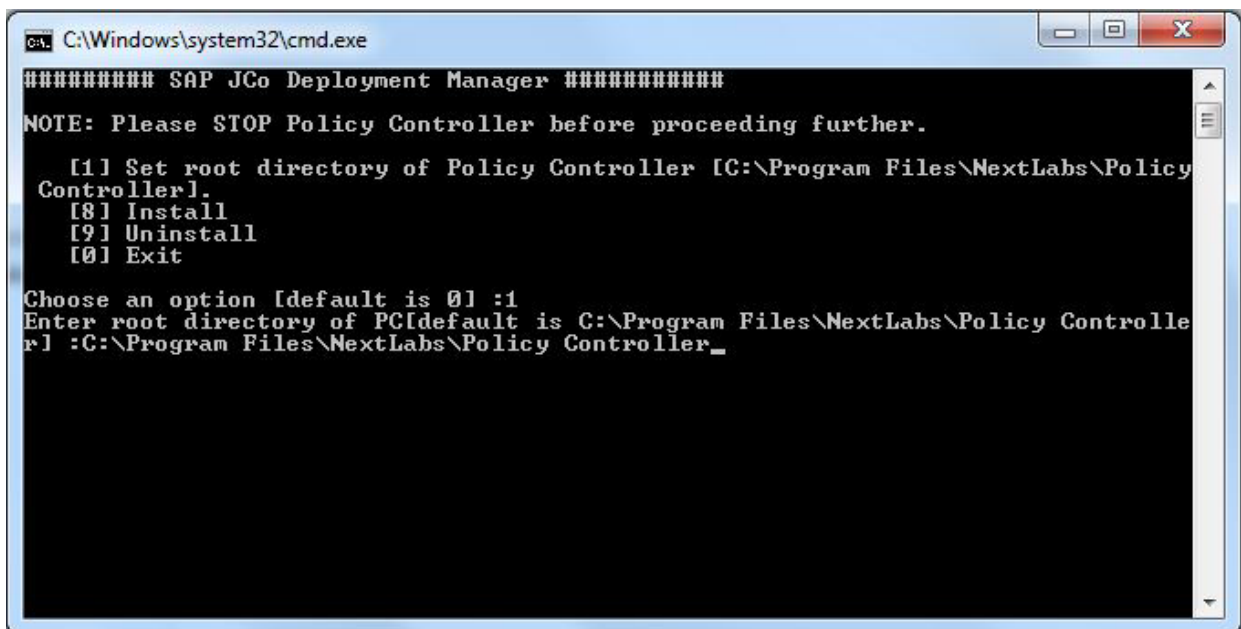
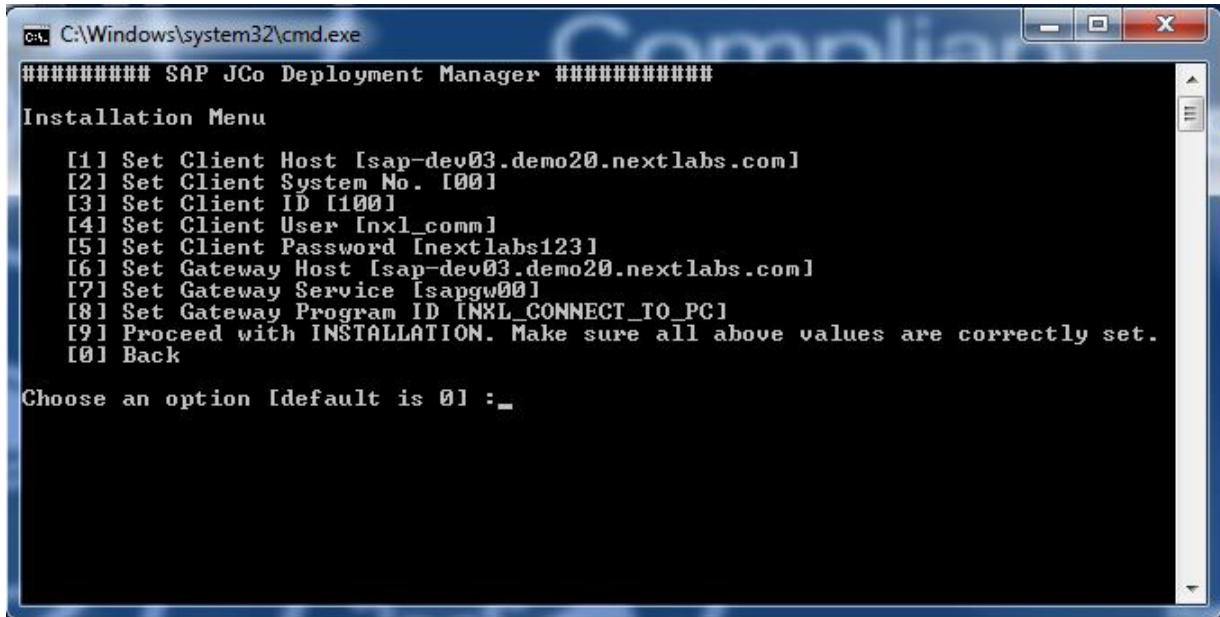


Figure 3-18: Setting Location of the Policy Controller

- 6 Enter 8 to access the install menu. A screen appears for each of the configuration values you need to define.



```

C:\Windows\system32\cmd.exe
##### SAP JCo Deployment Manager #####
Installation Menu

[1] Set Client Host [sap-dev03.demo20.nextlabs.com]
[2] Set Client System No. [00]
[3] Set Client ID [100]
[4] Set Client User [inx1_comm]
[5] Set Client Password [nextlabs123]
[6] Set Gateway Host [sap-dev03.demo20.nextlabs.com]
[7] Set Gateway Service [sapgw00]
[8] Set Gateway Program ID [INXL_CONNECT_TO_PC]
[9] Proceed with INSTALLATION. Make sure all above values are correctly set.
[0] Back

Choose an option [default is 0] :_

```

Figure 3-19: Installation Menu Options

- 7 Using the numbered options 1 through 8 in the Installation Menu, enter the values for the Properties file. The instructions for retrieving these values are provided in [Locate Required Values for Configuration](#) on page 50.

**Note:** When you install the Java Connector using the Deployment Manager script, the RFC user password can be supplied in plain text form. You do not need to use the `mkpassword.bat` utility to encrypt the password (as you do for the manual process).

Uninstall deletes all the jars, dlls, and configured values in the `SAPJavaSDK-service.properties` file.

It is recommended that you make a backup of the `SAPJavaSDKService.properties` file.

- 8 When you have entered all the Properties values, select 9 to proceed with installation.
- 9 There is one installation file for the Java Connector that the script cannot install because it is specific to the host where the Java Connector is being installed. From the extracted contents of the .zip file, open the `xlib` folder. Open the folder that corresponds with the operating system of the host on which you are installing the Java Connector:
- "Windows 64-bit, Intel-Arch: Open the `NTia64` folder
  - "Windows 64-bit, AMD: open the `NTamd64` folder

- 10 In the appropriate folder, select `sapjco3.jar` and `sapjco3.dll`. Copy the files to `<install dir>/Policy Controller/jre/lib/ext`.

## Manual Installation of the Java Connector for the Server Policy Controller

If your implementation uses the Server Policy Controller, you can manually install the Java Connector as described in this section.

### Before You Begin

- The Policy Controller must be installed.
- Obtain the Java Connector installation file from NextLabs. For more information, contact NextLabs Technical Support at [support@nextlabs.com](mailto:support@nextlabs.com).
- To stop the Policy Controller, you must have the profile password defined in the Control Center *Administrator* interface.

### Procedure

- 1 Stop the Policy Controller, if it is running.
- 2 Create the following folders in the Policy Controller install directory:
 

```
<install dir>/Policy Controller/jservice/config
<install dir>/Policy Controller/jservice/jar/sap
<install dir>/Policy Controller/jre/lib/ext
```
- 3 Locate the Java Connector installation file from NextLabs support (`SAPJCo-Entitlement-Manager-<version number>.zip`).
- 4 In the zip file, locate the `SAPJavaSDKService.Properties` file at `SAPJCo-Entitlement-Manager-<version number>.zip/SAPJCo-EntitlementManager/config`.
- 5 Copy the `SAPJavaSDKService.Properties` file to `<install dir>/Policy Controller/jservice/config`.
- 6 In the zip file, locate the `SAPJco-EntitlementManager.jar` file at `SAPJCo-Entitlement-Manager-<version number>.zip/SAPJCo-EntitlementManager`.
- 7 Copy the `SAPJco-EntitlementManager.jar` file to `<install dir>/Policy Controller/jservice/jar/sap`.
- 8 From the extracted contents of the .zip file, open the `xlib` folder that matches the host on which you are installing the Java Connector:
  - Windows 64-bit, Intel-Arch: open the `NTia64` folder
  - Windows 64-bit, AMD: open the `NTamd64` folder
- 9 In the appropriate folder, select `sapjco3.jar` and `sapjco3.dll`. Copy the files to `<install dir>/Policy Controller/jre/lib/ext`.

## Manual Installation of the Java Connector for the Policy Controller for Java

If your implementation includes the Policy Controller for Java, the Java Connector must be installed manually. The procedure is the same for both Windows and Red Hat Linux systems.

### **Before You Begin**

- Obtain the Java Connector installation file from NextLabs. For more information, contact NextLabs Technical Support at [support@nextlabs.com](mailto:support@nextlabs.com).
- To stop the Policy Controller, you must have the profile password defined in the Control Center *Administrator* interface.

### **Procedure**

- 1 Stop Policy Controller, if it is running.
- 2 Create the following folders, if these do not exist (where `<tomcat-home>` is the installation location of the Tomcat server).  

```
<tomcat-home>/nextlabs/dpc/jservice/config  
<tomcat-home>/nextlabs/dpc/jservice/jar/sap  
<tomcat-home>/nextlabs/dpc/jservice/jar/javasdk  
<tomcat-home>/nextlabs/shared_lib/
```
- 3 Locate the Java Connector installation file from NextLabs support: `SAPJCo-Entitlement-Manager-<version number>.zip`.
- 4 From the extracted contents of the `.zip` file, locate `config/SAPJavaSDKService.properties`.
- 5 Copy the `SAPJavaSDKService.properties` file to `<tomcat-home>/nextlabs/dpc/jservice/config`.
- 6 From the extracted contents of the `.zip` file, locate `jars/SAPJCo-EntitlementManager.jar`.
- 7 Copy the `SAPJCO-EntitlementManager.jar` file to `<tomcat-home>/nextlabs/dpc/jservice/jar/sap`.
- 8 From the extracted contents of the `.zip` file, open the `xlib` folder appropriate for the host on which you are installing the Java Connector:
  - "Windows 64-bit, Intel-Arch: Open the `NTia64` folder
  - "Windows 64-bit, AMD: open the `NTamd64` folder
- 9 In the appropriate folder, select `sapjco3.jar` and `sapjco3.dll`. Copy the files to `[tomcat-home]/nextlabs/shared_lib/`.

### **Next Steps**

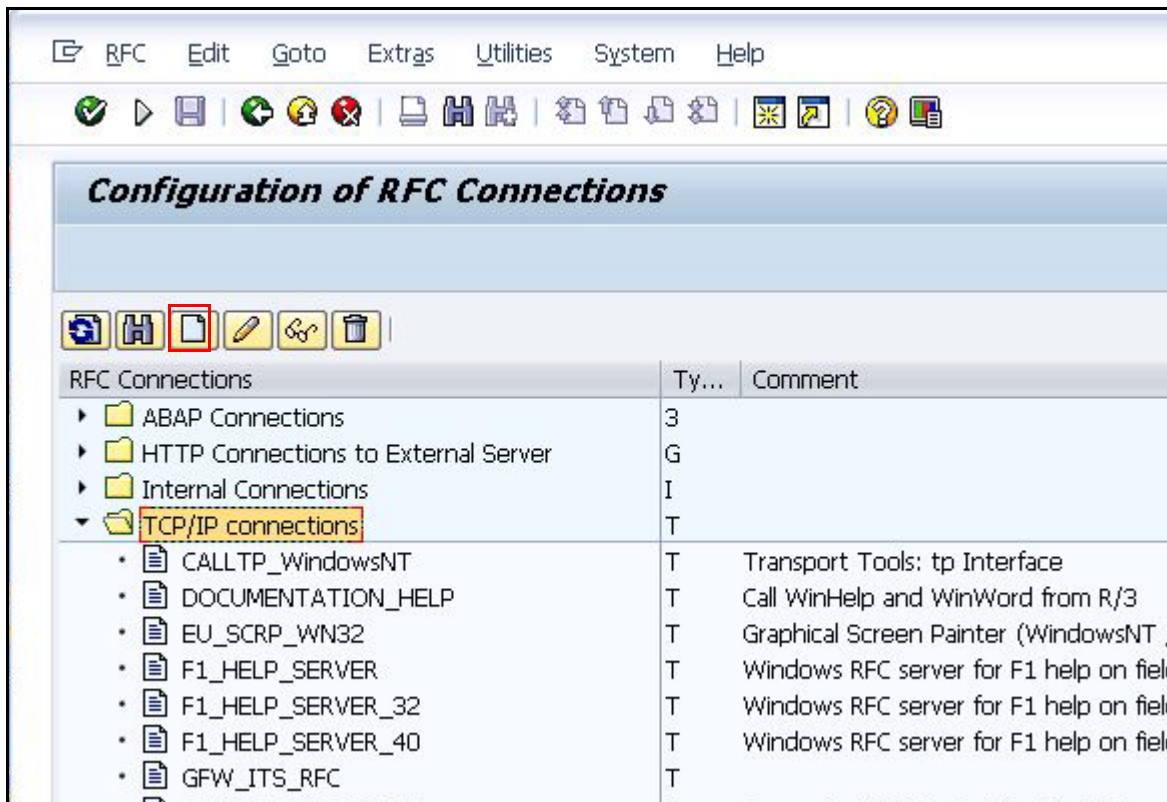
After installation of the Java Connector is complete, the next step is to configure the Remote Function Call (RFC) connection. See [Configuring the RFC Connection](#) on page 56.

## **Configuring the RFC Connection**

After installing the Java Connector, the next step is to configure a Remote Function Call. This step must be performed regardless of the installation method used.

**Procedure**

- 1 In the SAP interface, enter transaction `SM59`. The RFC connection configuration screen appears.
- 2 Select the TCP/IP connections folder, then click **Create**.



*Figure 3-20: Creating a TCP/IP RFC Connection*

- 3 Set the Connection type to `T`.
- 4 Enter an RFC Destination name, for example, `NEXTLABS_PC`.
 

**Note:** Make note of the RFC Destination name. You need to enter this name in the `NEXTLABS/CONCFG` table in a subsequent step.
- 5 In the Technical Settings tab, select `Registered Server Program`.
- 6 Enter a Program ID, for example, `NXL_CONNECT_TO_PC`.
 

**Note:** Make note of the Program ID. You need to enter this ID in a properties file in a subsequent procedure.

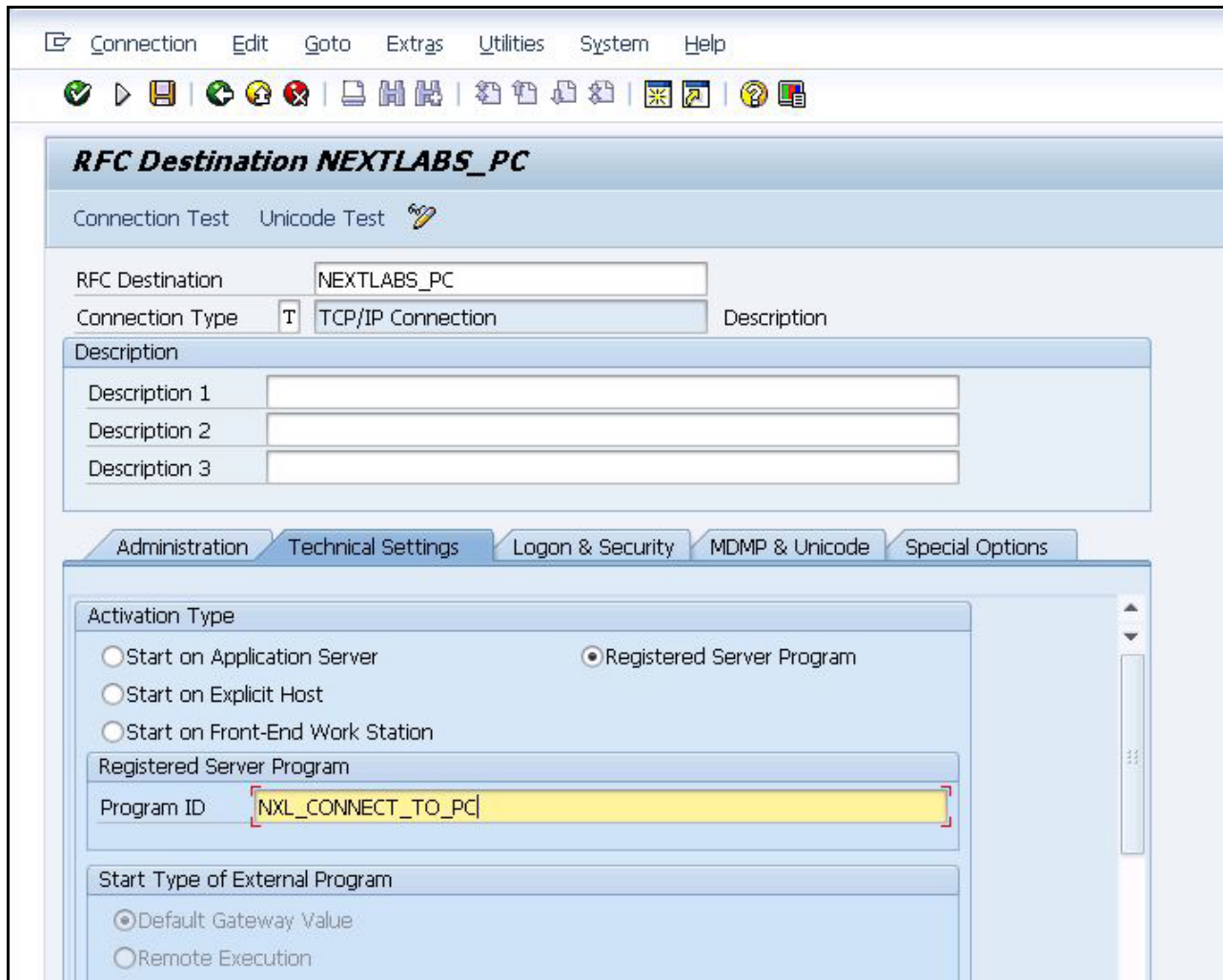


Figure 3-21: Defining the RFC

- 7 Click the **MDMP & Unicode** tab.
- 8 Select `Unicode`.

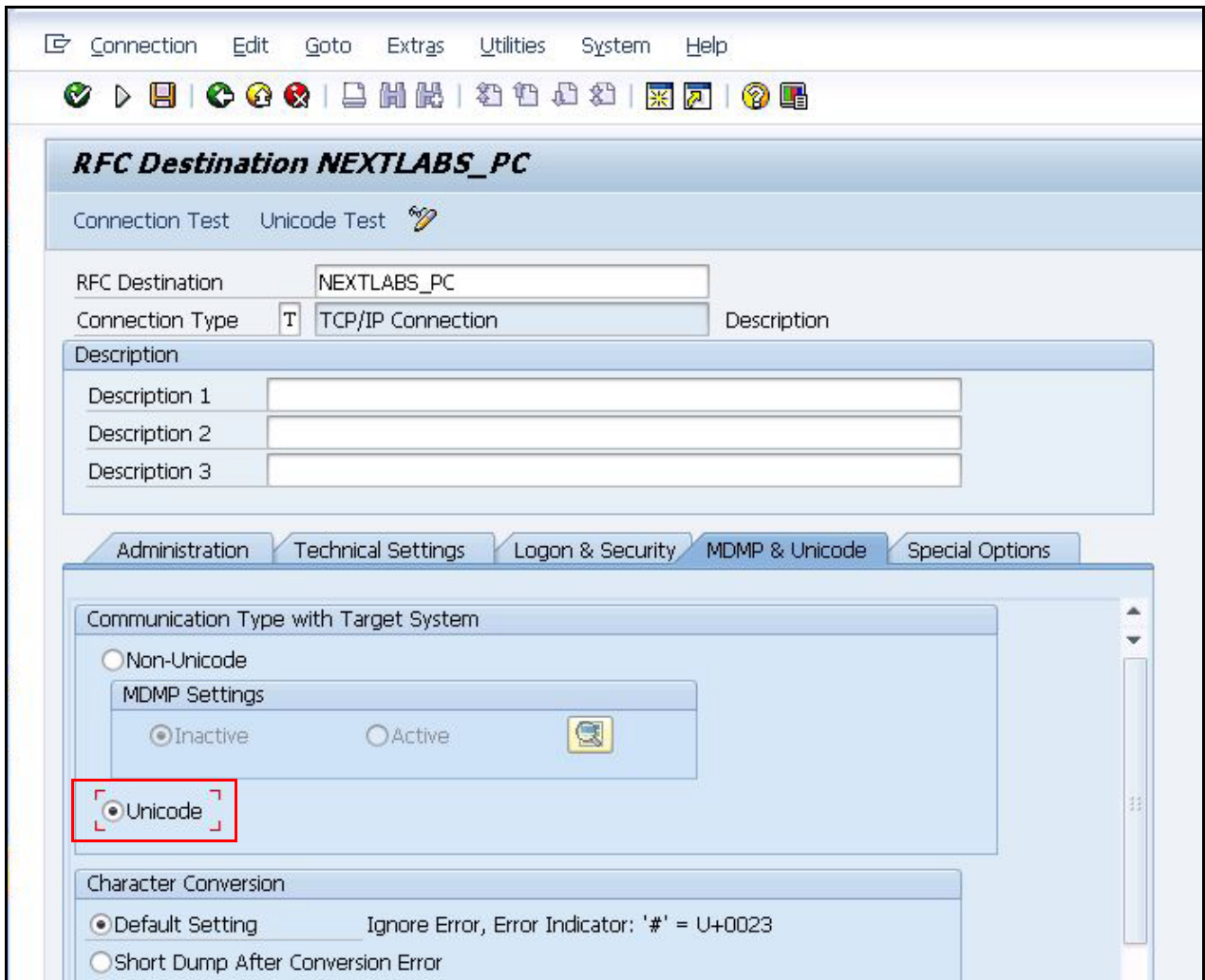


Figure 3-22: Selecting Unicode

9 Save the configuration.

**Next Steps**

After RFC connection is configured, the next step is to configure the properties file used by the Java Connector. See [Configuring the SAP JavaSDK Properties File \(Manual Installation only\)](#) on page 60.



## Configuring the Properties and Services Files Manually

Two additional files must be configured for the Java Connector: the `SAPJavaSDKService.properties` file and the Windows services file.

If you used the script method of installation, then the `SAPJavaSDKService.properties` file is already configured (the script provides a menu where you can enter the required values, which are written to the properties file automatically). However, you must still configure the Windows Services file manually.

### Configuring the SAP JavaSDK Properties File (Manual Installation only)

If the Java Connector files are installed manually, the `SAPJavaSDK` properties file must be configured to communicate with the SAP Server and the Policy Controller.

**Note:** If you use the installation script, the properties file is configured automatically. Configuring the SAP JavaSDK properties file when using the script is necessary only if you want to change information after the initial installation, or add additional SAP Servers or Policy Controllers to your implementation. For more information on how to configure multiple Policy Controllers for load balancing, see [Load Balancing the Policy Controller](#) on page 251.

### Procedure

- 1 Stop the Policy Controller.
- 2 Navigate to and open the `SAPJavaSDKService.properties` file in the following location:

```
<tomcat-home>/nextlabs/dpc/jservice/config
```

- 3 Verify that `jar-path` is set to the following:

```
<install dir>/Policy Controller/jservice/jar/sap/SAPJCo-EntitlementManager.jar
```

- 4 Locate the section for `server destination details prefix`.
- 5 If necessary, add a new set of server and destination parameters to reflect the number of servers in your implementation.

For each server instance, you must append prefix values to the `dest_name` and `server_name` property in the following format: `DEST<number>;` and `SERV<number>;` where `<number>` refers to the server instance.

For example, below is the configuration for three server instances.

```
#server destination details prefix
dest_prefix=DEST1_;DEST2_;DEST3_;
server_prefix=SERV1_;SERV2_;SERV3_;
```

If our implementation included four server instances, we would add `DEST4;` to `dest_prefix` and `SERV4;` to `server_prefix`.

- 6 Locate the `destination data provide connection details` section. Enter the following information:



- `ashost` = FQDN for the host where SAP ECC is installed
- `sysnr` = System number
- `client` = client
- `user` = RFC user name
- `passwd` = RFC user password. This password must be encrypted using the `mkpasswd.bat` utility installed with the Policy Server. These instructions are provided in [Locate Required Values for Configuration](#) on page 50.

**Note:** For more information on how to use the `mkpasswd.bat` utility, see *Control Center Installation Guide*.

- `lang` = language code (`en` for English)

7 **Locate the** `server data provider connection details` section. Enter the following information:

- `gwhost` = Gateway host name
- `gwserver` = Gateway service name
- `progid` = Program ID that was defined in the RFC configuration step (in our example, `NEXTLABS_CONNECT_TO_PC`)

8 If you are upgrading from a previous release of Dynamic Authorization Management for SAP, enter the following line in the `SAP module names` section:

```
jpc_query_mval_handler=/NEXTLABS/JPC_MAIN_MVAL
```

9 After you have made all the changes, save and close the properties file.

## Configuring the Windows System Services File (Manual and Script Installation)

You must enter the SAP Gateway service and port number in the Windows Services file on the host where the Policy Controller is installed.

### Procedure

- 1 Browse to `<Windows Home>System 32\drivers\etc`.
- 2 Open the services file using Wordpad or Notepad.
- 3 Add a new entry in the file for the Gateway Service name (the `gwserver` parameter value) and port. For example:

```
sapgw00          3300/tcp    #sap connection for NextLabs
```

- 4 Save the file.

### Next Steps

After configuring the Properties and Services files, the next step is to test the connection. See [Testing the Java Connector Configuration](#) on page 62.

There is an additional step to configure the Java Connector: you must define the `AGENT_COMMUNICATION_OPTION` and `AGENT_RFC_NAME` in `/NEXTLABS/CONCFG`. This step is discussed in [Configuring SAP Data Handling and Connection Settings](#) on page 94.

## Testing the Java Connector Configuration

After the configuration is complete, the final step is to test the Java Connector configuration.

### Procedure

- 1 In SAP ECC, enter transaction `SMGW`. The Gateway Monitor appears.
- 2 In the *Go to* menu, select **Logged in Clients**.
- 3 Check for the Program ID that was defined for the RFC in `SM59`.

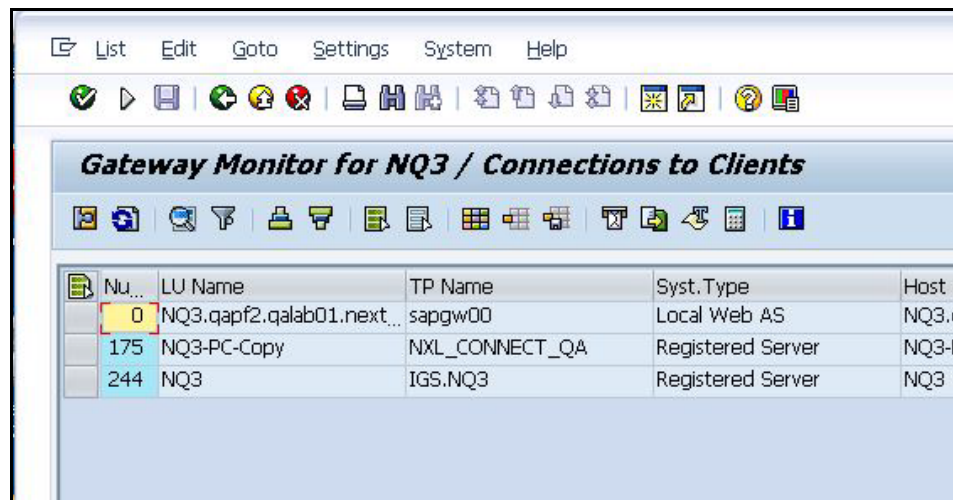


Figure 3-23: Testing the Java Connector Connection

## Testing the Connection from the Policy Controller Side

You can test the connection from the Policy Controller side provided that you have performed the following tasks:

- Installed Dynamic Authorization Management for SAP.
- In `/NEXTLABS/CONCFG` table (accessed through `SM30`), made the following configuration changes:
  - `AGENT_COMMUNICATION_OPTION` should be set to `RFC`.
  - `AGENT_RFC_NAME` should be set the RFC Destination Name that was defined in `SM59`.

After you have performed these tasks and started the Policy Controller, you can test the connection from the Policy Controller side.

### Procedure

- 1 Browse to the location of the Agent log at `<InstallDir>\Policy Controller\Agentlog`.

- 2 Search the log for the following message, which indicates the starting of the SAPJavaSDK, used in the Communication interface.

```
INFO: SAPJAVASDK init() started.
```

The installation is complete.

### Next steps

The next step is to configure Dynamic Authorization Management for SAP. See [Configuring Next-Labs Control Center for Dynamic Authorization Management for SAP](#) on page 65.



# 4 Configuring Basic Features

---

This section describes the basic configuration steps for the Entitlement Manager for SAP. It covers configuration of the base product and all the Entitlement Packs.

Topics:

- [Configuring NextLabs Control Center for Dynamic Authorization Management for SAP](#)
- [Configuring NextLabs Entitlement Packs and SAP Data](#)
- [Using the NextLabs Configuration Tool](#)
- [Configuration for Policy-based Security Classification](#)
- [Configuring Enhancement Implementations](#)

---

## Configuring NextLabs Control Center for Dynamic Authorization Management for SAP

This section explains how to configure NextLabs Control Center for Dynamic Authorization Management for SAP. The configuration procedures in this section are required regardless of the Entitlement Packs and features installed on your implementation.

- [Enrolling Users from SAP into Control Center](#)
- [Configuring SAP Actions](#)
- [Configuring SAP Obligations](#)

---

## Enrolling Users from SAP into Control Center

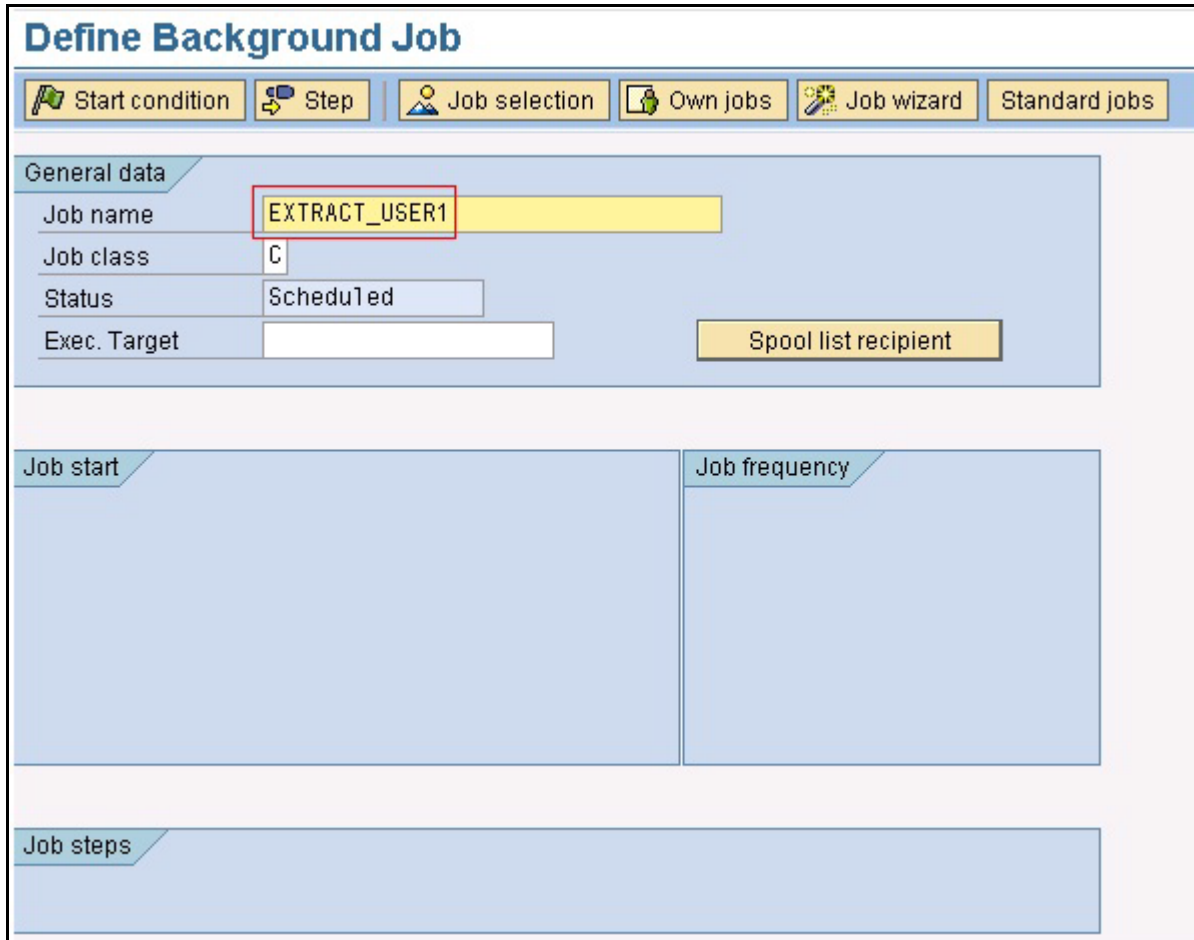
You can enroll SAP users into NextLabs Control Center so that they can be added to User Components in Policy Studio. This is a two-stage process: first extract user data from SAP; then upload user data into Control Center.

### Extracting User Data from SAP

You can download user data from SAP by scheduling a batch job to run the extract program: /NEXTLABS/USER\_EXTRACT.

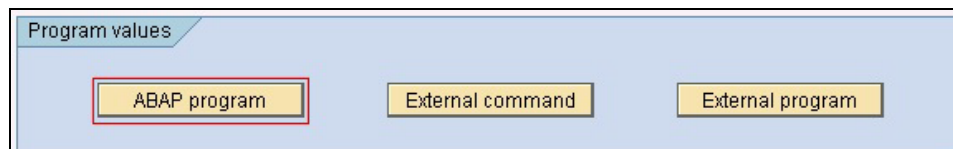
**Procedure**

- 1 In the SAP interface, enter transaction SM36. The *Define Background Jobs* screen appears. See [Figure 4-1](#).



*Figure 4-1: Define Background Job Screen*

- 2 Enter the Job name and click **Step** on the application tool bar. A pop-up window appears.



*Figure 4-2: Program Values Pop-up Window*

- 3 Click **ABAP program**, and enter the name of the ABAP program: /NEXTLABS/USER\_EXTRACT.

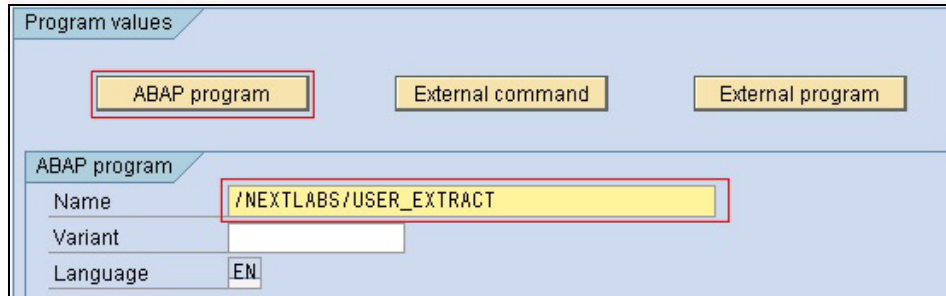


Figure 4-3: Enter the ABAP program name

- 4 Click **Check and Save** at the bottom of the window, then, return to the *Define Background Job* screen by clicking **Back**.
- 5 Click **Start Condition** in the tool bar. A pop-up window appears, prompting you to schedule the job.

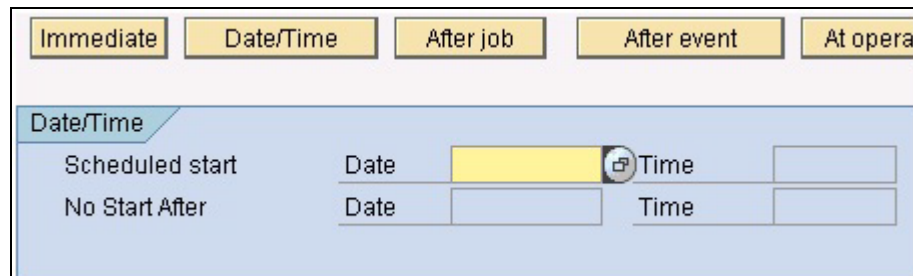


Figure 4-4: Scheduling the background job

- 6 Schedule the start Date and Time conditions, or click **Immediate** to start the job at once. You can also schedule the job to run repeatedly by clicking on the **Date/Time** button and specifying the frequency. Click **Check and Save**.
- 7 After the pop-up disappears, be sure to click **Save** to save the job definition in the original screen. This is very important. If the job definition is not saved in the *Define Background Job* screen, the job is not scheduled. After you save the job, a message appears at the bottom of the screen in the status bar displaying the job saved or released message.
- 8 To verify the job definition, enter transaction `SM37` and click **Execute**. Make sure the User name in the selection screen is the same as the one used to define the job. The jobs scheduled or released by this user are displayed as in [Figure 4-5](#).

**Note:** The download path for the user extract file can be configured using the `FILE` transaction.

Job	Ln	Job CreatedB	Status	Start date	Start time	Duration(sec.)	Delay (sec.)
CODE_INSPECTOR_DELETION		DEVELOPERS	Finished	30.08.2010	01:00:12	4	12
EU_PUT		DEVELOPERS	Finished	30.08.2010	00:10:11	1	11
EU_REORG		DEVELOPERS	Finished	30.08.2010	01:40:12	210	12
EXTRACT_USER1		DEVELOPERS	Finished	30.08.2010	17:17:35	1	0

Figure 4-5: Verifying the job definition

## Uploading SAP User Data into Control Center

After you have extracted user information from SAP, you need to upload it into NextLabs Control Center.

### Before You Begin

Obtain the Enrollment Adapter file from NextLabs. For more information, contact NextLabs Technical Support at [support@nextlabs.com](mailto:support@nextlabs.com).

### Procedure

- 1 Locate the Enrollment Adapter file from NextLabs Technical Support.
- 2 Extract the Enrollment Adapter zip file to `C:\Program Files\NextLabs\PolicyServer\tools\enrollment`. This creates a folder named `SAPEnrollmentAdapter`, which contains the following files:
  - `Mapping.properties` (maps SAP export columns to the ldif template)
  - `Sap_nextlabs_com.def` (ldif enrollment definition file that maps ldif to the dictionary)
  - `Sap2ldif.bat` (script that calls the ldif converter)
  - `SAPDataLdifConverter.jar` (converter program)
- 3 Change the mapping properties enrollment domain in two places to reflect the SAP system and company, for example, “`EC2.nextlabs.com`.”
  - `dn: cn=<<UserID>>,dc=sap,dc=domain_name,dc=com`
  - `userPrincipalName: <<UserID>>@sap.domain_name.com`
- 4 Set two file paths in `Sap2ldif.bat`. The first points to the folder on the SAP ECC server where the user extract is being placed. You should share this folder on the ECC system so that the enrollment process can reach it. The second is the output location of the ldif file.
  - `"C:\Program Files\NextLabs\PolicyServer\java\bin\java" -jar SAPDataLdifConverter.jar mapping.properties "\\DEMO20-SAP02\src\userdata.txt"`
  - `"C:\Users\user.name\Documents\sapldif\sap_nextlabs_com.ldif"`
- 5 In `Sap_nextlabs_com.def`, change the location of the ldif filename to `"C:\Users\user.name\Documents\sap ldif\sap_nextlabs_com.ldif"` (This is the same path as the output file.)



- `ldif.filename C:/Users/user.name/Documents/sapldif  
sap_nextlabs_com.ldif`

6 To run the conversion, run `sap2ldif`.

7 Set up enrollment for the ldif using the `enroll` and `sync` commands of the `enrollmgr` utility.

```
enrollmgr -u Administrator -enroll -t LDIF -n sap.domain_name.com
-d "C:\Program Files\NextLabs\PolicyServer\tools\enrollment\SAPEn-
rollmentAdapter\sap_nextlabs_com.def"
```

```
enrollmgr -u Administrator -sync -t LDIF -n sap.domain_name.com
-d "C:\Program Files\NextLabs\PolicyServer\tools\enrollment\SAPEn-
rollmentAdapter\sap_nextlabs_com.def"
```

### Next steps

The next step is [Configuring SAP Actions](#) on page 69 in Policy Studio.

---

## Configuring SAP Actions

Before you can write SAP policies, you must add SAP actions to Policy Studio. Actions include SAP Copy-From, Execute Transaction, Check-in, and others.

### Procedure

- 1 Use Notepad or WordPad to open one of the following XML files supplied by NextLabs Technical Support:
  - Open `SAP Actions_FullBuild.xml` if you are installing Dynamic Authorization Management for SAP for the first time. This file contains all the SAP actions.
  - Open `SAP Actions_Upgrade.xml` if you are upgrading Dynamic Authorization Management for SAP. This file contains only the new SAP actions added in the current release.
- 2 Open the main configuration file, `configuration.xml`, on the Control Center host. By default, this file is located in

```
<installDirectory>\PolicyServer\server\configuration
```

- 3 Copy the actions from `SAP Actions_FullBuild.xml` or `SAP Actions_Upgrade.xml`.
- 4 Paste the actions into `configuration.xml` in the `<ActionList> </ActionList>` section.

The following are examples of actions in the `<ActionList>` section of `configuration.xml`.

```
<ActionList>
  <Action>
    <Name>SAP_COPY_FROM</Name>
    <DisplayName>SAP Copy-From</DisplayName>
    <ShortName>CF</ShortName>
    <Category>Transform</Category>
  </Action>
```

```
<Action>
  <Name>CHECK_IN</Name>
  <DisplayName>Check_in</DisplayName>
  <ShortName>CN</ShortName>
  <Category>Transform</Category>
</Action>
.
.
.
</ActionList>
```

5 Save the changes and restart Control Center.

6 After the system restarts, test the configuration by opening Policy Studio and searching for the new actions in the Actions component pane.

### Next steps

The next step is [Configuring SAP Obligations](#) on page 70 for Policy Studio.

---

## Configuring SAP Obligations

You also must insert obligations for Entitlement Manager for SAP in the Control Center configuration.xml file. The following obligations may be required, depending on your configuration:

- **SAP User Alert:** Display messages defined in Policy Studio to users in SAP
- **SAP Message Class Display:** Display messages from SAP Message classes as part of a NextLabs policy notification
- **Set Classification Value:** Trigger Policy Based Security Classification (PBSC)
- **Data Segregation - Blacklist:** Specify the content servers where classified data cannot be stored
- **Data Segregation - Whitelist:** Specify the content servers where classified data can be stored

**Note:** Another obligation pertinent to Rights Management Server may be required to implement Integrated Rights Management policies. See [Configuration for Integrated Rights Management \(IRM\)](#) on page 145.

### Procedure

- 1 Use WordPad or NotePad to open one of the following XML files, supplied by NextLabs Professional Services:
  - Open SAP Obligations\_FullBuild.xml if you are installing Dynamic Authorization Management for SAP for the first time. This file contains all the SAP obligations.
  - Open SAP Obligations\_Upgrade.xml if you are upgrading Dynamic Authorization Management for SAP. This file contains only the new SAP obligations added in the current release.
- 2 Open the main configuration file, configuration.xml, on the Control Center host. This file is located at:

```
<installDirectory>\NextLabs\PolicyServer\server\configuration
```

- 3 Copy the obligations you require from SAP Obligations\_FullBuild.xml or SAP Obligations\_Upgrade.xml.
- 4 Paste the obligations into configuration.xml file in the <Obligations> </Obligations> section.
- 5 Save your changes and restart Control Center.
- 6 After the system restarts, test the configuration by opening Policy Studio and creating a new policy. The new obligations should appear in the drop-down list of Custom Obligations.

**Note:** For example policies that use these obligations, see [Designing SAP Access Control Policies](#) on page 204, [Designing Integrated Rights Management Policies](#) on page 228 and [Designing Data Segregation Policies](#) on page 231.

This completes the configuration of NextLabs Control Center to support Dynamic Authorization Management for SAP.

### Next steps

The next steps are [Configuring NextLabs Entitlement Packs and SAP Data](#).

---

## Configuring NextLabs Entitlement Packs and SAP Data

After NextLabs Control Center has been configured for Dynamic Authorization Management for SAP, Entitlement Packs and SAP data can be configured. The required procedures vary depending on which Entitlement Packs are being used. Many of the procedures described in this section, however, apply to all the Entitlement Packs.

- [Activating Entitlement Packs \(EPCONF\)](#) on page 77
- Configuring Data in the Security Classification Maintenance table:
  - [Adding Composite Keys and Classification Values](#) on page 78
  - [Linking Composite Keys \(SECENH\)](#) on page 83
  - [Mapping Security Fields \(SECM PG\)](#) on page 84
  - [Configuring Security Identifier/Composite Key Value Tables \(EPVAL\)](#) on page 86
- [Configuring UI Functions](#) on page 88
- [Mapping Transaction Codes and UI Functions to Actions \(ACTIONS\)](#) on page 90
- [Configuring SAP Data Handling and Connection Settings](#) on page 94
- [Configuring View Filtering \(EasyDMS Only\)](#) on page 102
- [Configuring Number Range Intervals](#) on page 103
- [Configuring the NextLabs Number Range](#) on page 104
- [Configuring Policy Checks Based on Transaction/UI Function](#) on page 109
- [Configuring Special Fields for the Security Classifications Maintenance Table](#) on page 111
- [Defining How Security Classifications and Access Control Contexts Should Be Applied](#) on page 112

- [Defining How Multiple Security Classifications Should Be Applied](#) on page 116
- [Configuring Access Control Context Settings \(PLM Only\)](#) on page 121
- [Configuring the Transactions or Functions to Intercept](#) on page 123

---

## Using the NextLabs Configuration Tool

Use the NextLabs configuration tool to configure Dynamic Authorization Management and Entitlement Packs using SPRO (SAP Project Reference Object).

**Note:** The configuration tool covers only the configuration procedures performed in SAP. You must already have installed the software components as described in [Installation and Set Up](#) on page 33, and performed the configuration procedures in the NextLabs Control Center as described in [Configuring NextLabs Control Center for Dynamic Authorization Management for SAP](#) on page 65.

### Procedure

- 1 In the SAP interface, do one of the following:
  - Enter transaction/NEXTLABS/CONFIG. This transaction shows the installed NextLabs components only, as shown in [Figure 4-6](#).

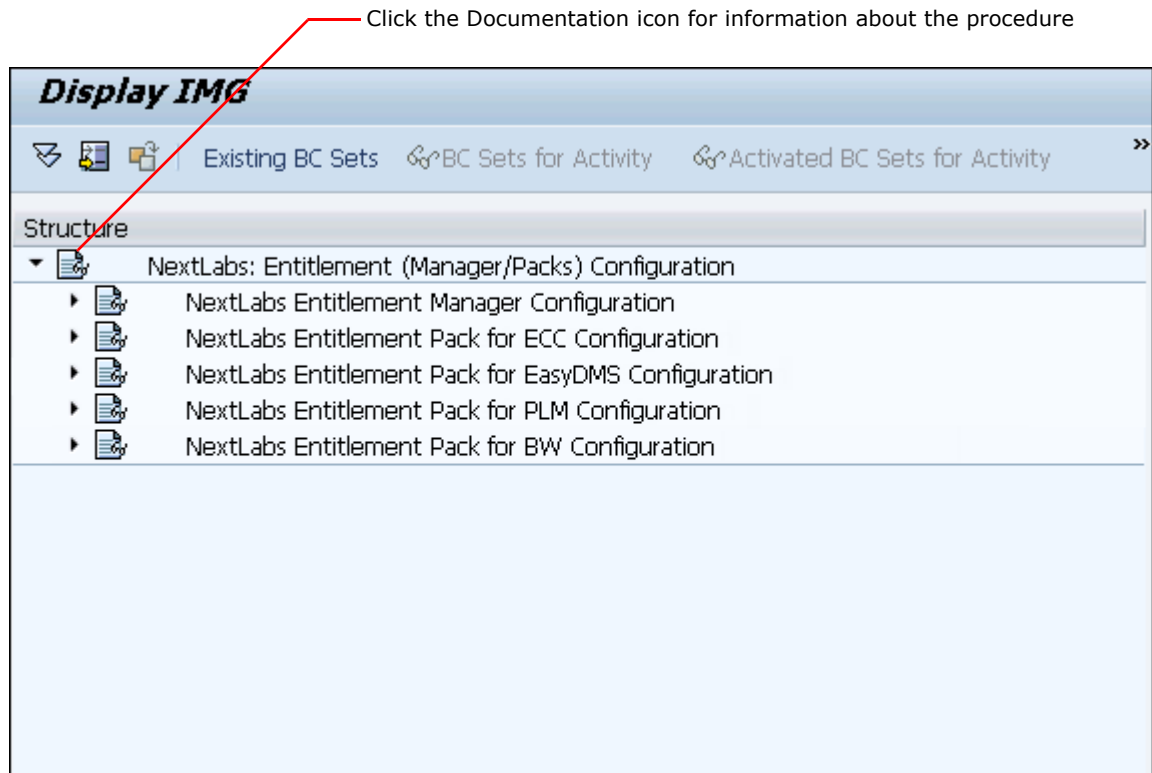


Figure 4-6: Using `/NEXTLABS/CONFIG` to access the configuration tool

- Enter transaction `SPRO`, then do the following:
  - a Click **SAP Reference IMG**, then expand the *Cross-Application Components* node.
  - b Expand *NextLabs: Entitlement (Manager/Packs) Configuration*. As shown in [Figure 4-7](#), the configuration tool organizes procedures by Dynamic Authorization Management and Entitlement Packs. You see only the Entitlement Packs that you have installed.

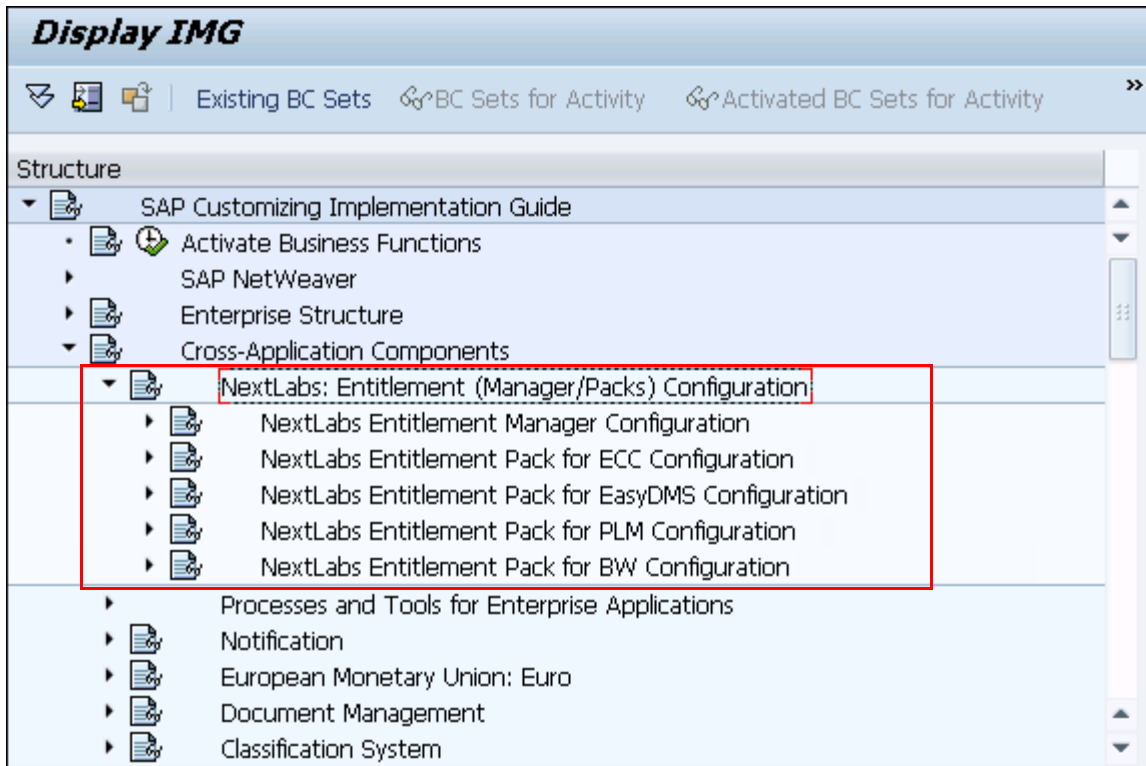
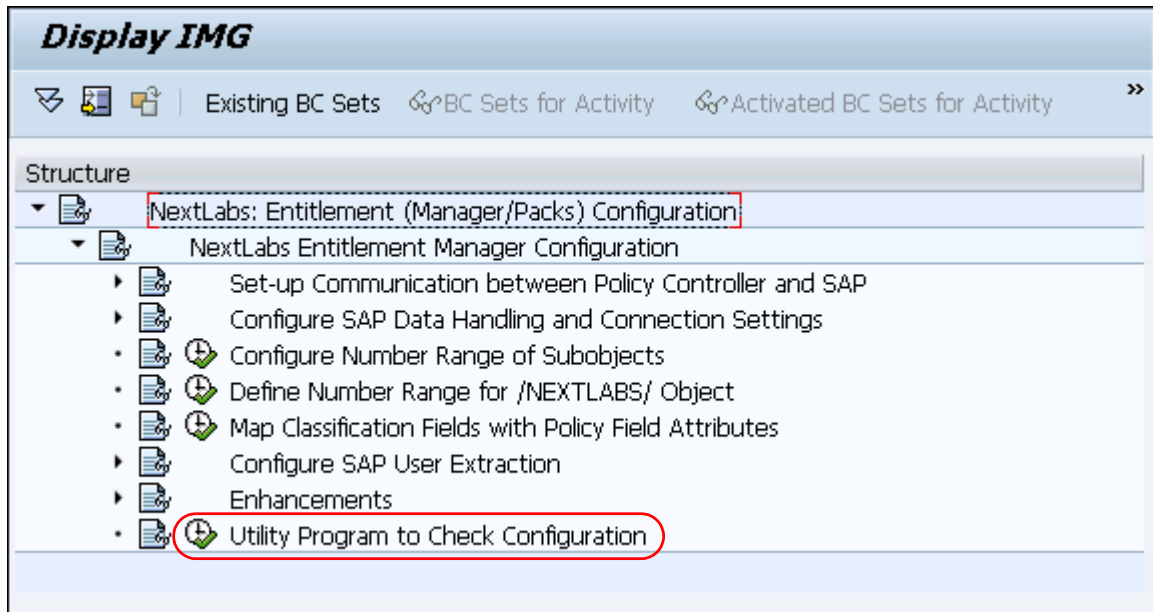


Figure 4-7: Using SPRO to access the configuration tool

- 3 Configure the NextLabs Dynamic Authorization Management, which is the base product.
- 4 Configure each Entitlement Pack that you want to implement. See [Configuring NextLabs Entitlement Packs and SAP Data](#) on page 71.
- 5 For more information about procedures, click the Documentation icon shown in [Table 4-1](#).

## Checking Configuration Status

As you finish configuring Dynamic Authorization Management and each Entitlement Pack, you can run a utility to check if each component has been configured correctly. To run this utility, either run transaction `/NEXTLABS/UTIL_CNFG`, or execute the **Utility Program to Check Configuration** activity, as shown in [Figure 4-8](#).



*Figure 4-8: Executing the utility to check configuration status*

The utility displays a report showing the status of each configuration activity, as shown in [Figure 4-9](#). The **Activity Message** column provides a description of the status. The **Activity Result** column provides a graphical status. A green symbol indicates that the configuration meets the minimum requirements, a yellow symbol is advisory (for example, an optional item is inactive), and a red symbol indicates an error that needs to be fixed.

The utility's primary function is to report which configuration activities have been completed and which have not. Even if all activities are complete and have the green or yellow status, this does not necessarily mean that the configuration is appropriate for your system or that policies will run as intended. The utility is comparable to a programming tool that can detect syntax errors, but not errors in logic.

**NextLabs: Entitlement Manager/Packs Configuration Check**

To Check Config, Doubleclick the Node!

NextLabs Entitlement Config Check	Activity Result	CO...	Activity Message
▶ INSTALLER_CHECK	■		
▼ CONFIG_CHECK	▲		
▼ EM_CONFIG	▲		
▶ CONCFG_TABLE	▲		
• SECMPG_TABLE	■		SECMPG Table is configured correctly
• NRCONF_TABLE	■		NRCONF Table is configured correctly
• NUMBER_RANGE	■		NEXTLABS Number Range is configured Correctly
▼ EP_CONFIG	■		
• EPCONF_TABLE	■		EPCONF Table is configured correctly
• EPCLS_TABLE	■		EPCLS Table is configured correctly
• SECENH_TABLE	■		Compound Key is Configured in the system
• EVLIDT_TABLE	■		EVLIDT Table is configured correctly
• CHKCLS_TABLE	■		CHKCLS Table is configured correctly
• EPVAL_TABLE	■		Security Identifier Value table is configured correctly
• ACTION_TABLE	■		ACTION Table is configured correctly
• OPTCFG_TABLE	■		Security Identifier Default configuration with Transaction Code is
• PBSCFG_TABLE	■		PBSC Configuration for,Security Identifier with UI Func/Function
▼ ECC_CONFIG	■		
• EPCONF_TABLE	■		EPCONF Table is configured correctly
• PBSMAT_TABLE	■		PBSC Configuration for,Material Identifier is configured correctly
• PBSDIR_TABLE	■		PBSC Configuration for,Document Identifier is configured correct
▼ EDMS_CONFIG	■		
• EPCONF_TABLE	■		EPCONF Table is configured correctly
• EDMS_FUNC	■		EasyDMS: UI Function Table is configured
• EDMS_FILTER	■		EasyDMS: View Filtering is active
▼ PLM_CONFIG	■		
• EPCONF_TABLE	■		EPCONF Table is configured correctly
• PLM_FUNC	■		PLM: UI Function Table is configured
• PLM_ACC	■		PLM: Access Control Context Table is configured
▶ ENHANCEMENTS CHECK	▲		

Figure 4-9: Configuration status report



---

## Activating Entitlement Packs (EPCONF)

The next configuration step is to activate the Entitlement Pack for the SAP back-end systems in the `/NEXTLABS/EPCONF` table.

### Procedure

- 1 In the SAP interface, enter transaction `SM30`. The *Table Maintenance View* screen appears.
- 2 In **Table/View**, enter `/NEXTLABS/EPCONF` to access the *NextLabs Entitlement Pack Configuration* screen.
- 3 Click **Maintain**.
- 4 Enter the NextLabs Entitlement Pack you are installing:
  - For SAP BW, enter `NXLBW`
  - For SAP cFolders, enter `NXLCFX`
  - For SAP EasyDMS, Enter `NXLEDMS`
  - For SAP ECC, enter `NXLECC`
  - For SAP PLM, enter `NXLPLM`
  - For SAP S/4HANA, enter `NXLS4H`
- 5 Select **Flag for Activation of EP**.
- 6 Enter the Identifier Append Structure used for the Entitlement Pack.
  - For SAP BW, enter `/NEXTLABS/SECIDT_BW`
  - For SAP cFolders, enter `/NEXTLABS/SECIDT_CF`
  - For SAP EasyDMS, enter `/NEXTLABS/SECIDT_ECC`
  - For SAP ECC, enter `/NEXTLABS/SECIDT_ECC`
  - For SAP PLM, enter `/NEXTLABS/SECIDT_ECC`
  - For SAP S/4HANA, enter `/NXLS4H/SECIDT_S4HANA`
- 7 Click **Save** to save the configuration. [Figure 4-10](#) shows a configuration example to activate SAP cFolders and SAP ECC.

NextLabs Entitlement Pack	Flag for Activation of EP	Identifier Append Structure
MXLCFX	<input checked="" type="checkbox"/>	/NEXTLABS/SECIDT_CF
MXLECC	<input checked="" type="checkbox"/>	/NEXTLABS/SECIDT_ECC

Figure 4-10: Example of NextLabs Entitlement Pack Configuration

### Next Steps

The next step is [Linking Composite Keys \(SECENH\)](#) on page 83 for Entitlement Pack for ECC.

## Adding Composite Keys and Classification Values

The first step of configuring data for the Security Classification Maintenance table is identifying the data to be displayed. The following are the types of data that appear in the Security Classification Maintenance table:

- [Security Identifiers](#) on page 78
- [Composite Keys](#) on page 79
- [Classification Values](#) on page 79

### Security Identifiers

A Security Identifier is the primary business object that classifications are applied to. Several Security Identifiers are configured in the *Security Classification Maintenance* screen upon installation, including Material and Document. Custom Security Identifiers can also be configured. However, this addition is only relevant if you are adding custom enhancements for new transactions, and this procedure is discussed in [This section explains the custom enhancements available for the NextLabs Dynamic Authorization Management for SAP](#). on page 255.

## Composite Keys

The Security Classification Maintenance table also contains Composite Keys. Composite keys enable you to create a unique version of a Security Identifier that can be classified independently. For example, if the Identifier is Document and a Composite Key is Document Version, you can maintain a unique classification for the combination of Document *and* Document Version. A Security Identifier can have any number of Composite Keys associated with it.

## Classification Values

The Security Classification Maintenance table also contains Classification Values. These are classifications customers want to associate with business objects. Some common examples are export control, license, and IP control category. Classification values should be configured to reflect requirements addressed in a customer implementation.

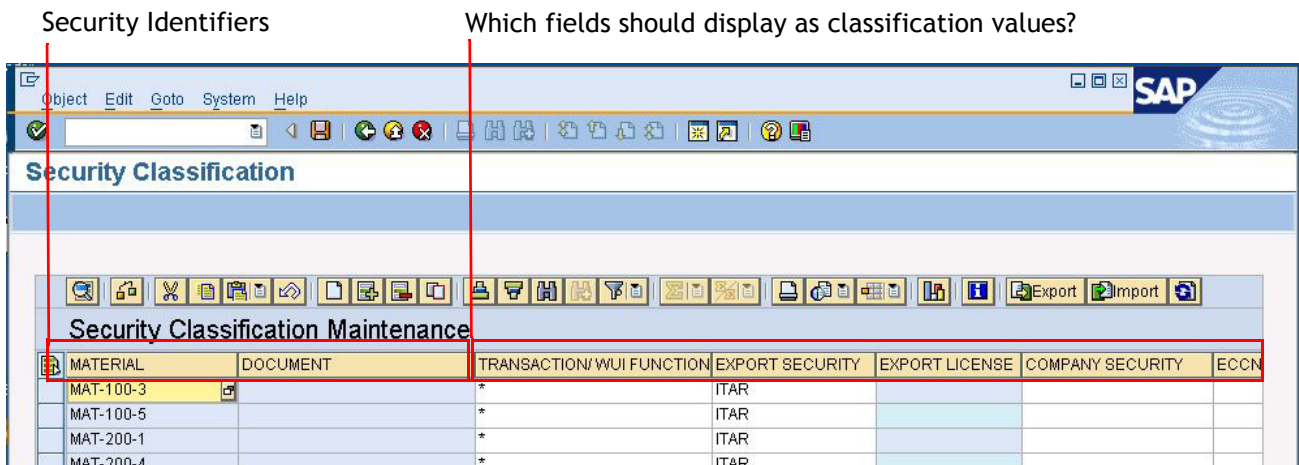


Figure 4-11: Customizing the SAP Security Classification Screen

## Adding Composite Keys

Composite Keys are not required, but are commonly used in SAP systems to track different states of business objects. For example, the Composite Keys for Document are included in the NextLabs installation because they are so common; these include DOKTL, DOKAR, and DOKVR. You can add composite keys as needed.

**Note:** To prevent custom enhancements from being overwritten by NextLabs upgrades, create Append Structures in your customer namespace. Do not you add Append Structures to the NextLabs namespace.

### Procedure

- 1 In the SAP interface, enter transaction `SE11`.
- 2 Select **Data Type**, and in the field, enter the structure name `/NEXTLABS/SECENH_CLS`, then click **Display**. The pre-configured composite keys display.

3 Click **Append Structure** on the toolbar.

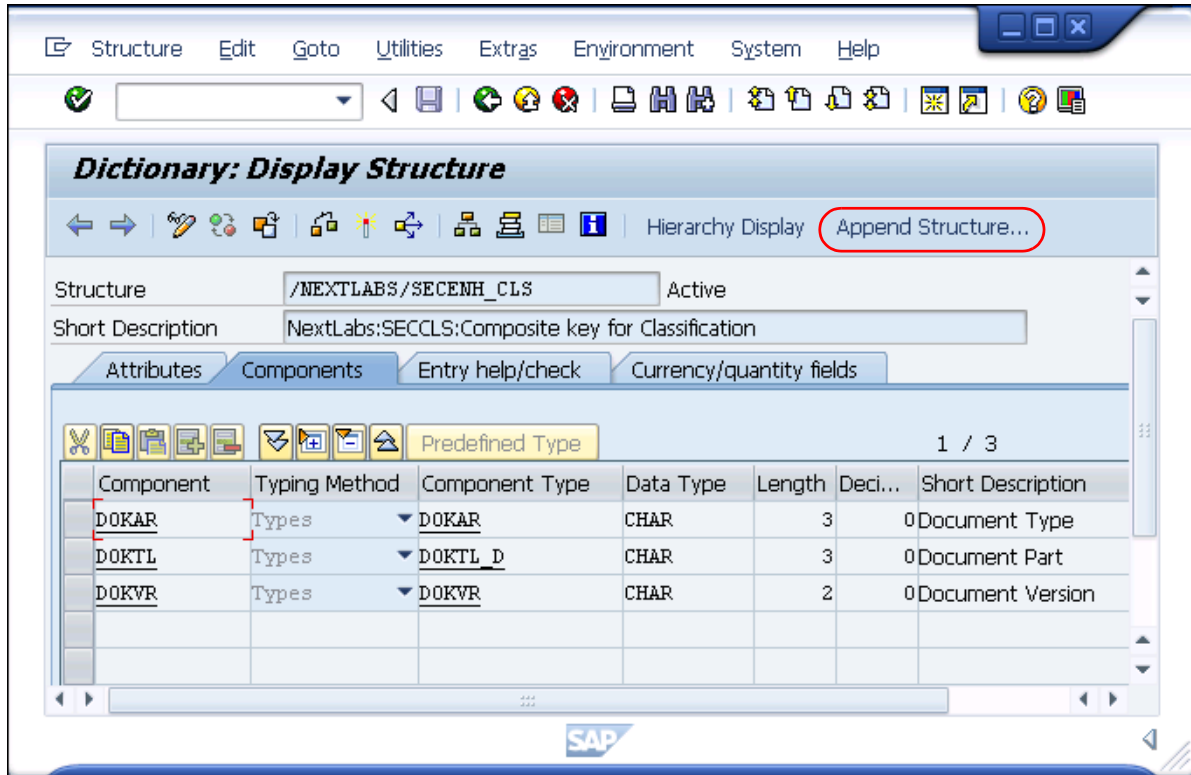


Figure 4-12: Append Structure

- 4 If the message No append defined appears, click **OK**.
- 5 Click **Create Append**, enter an Append Structure name, then click **Continue**.
- 6 In the *Append Structure* screen, enter information for the Component you want to add. In our example, we add a `WERKS` Component for Plant.

**Note:** To add additional Composite Keys, you can add them to the same Append Structure.

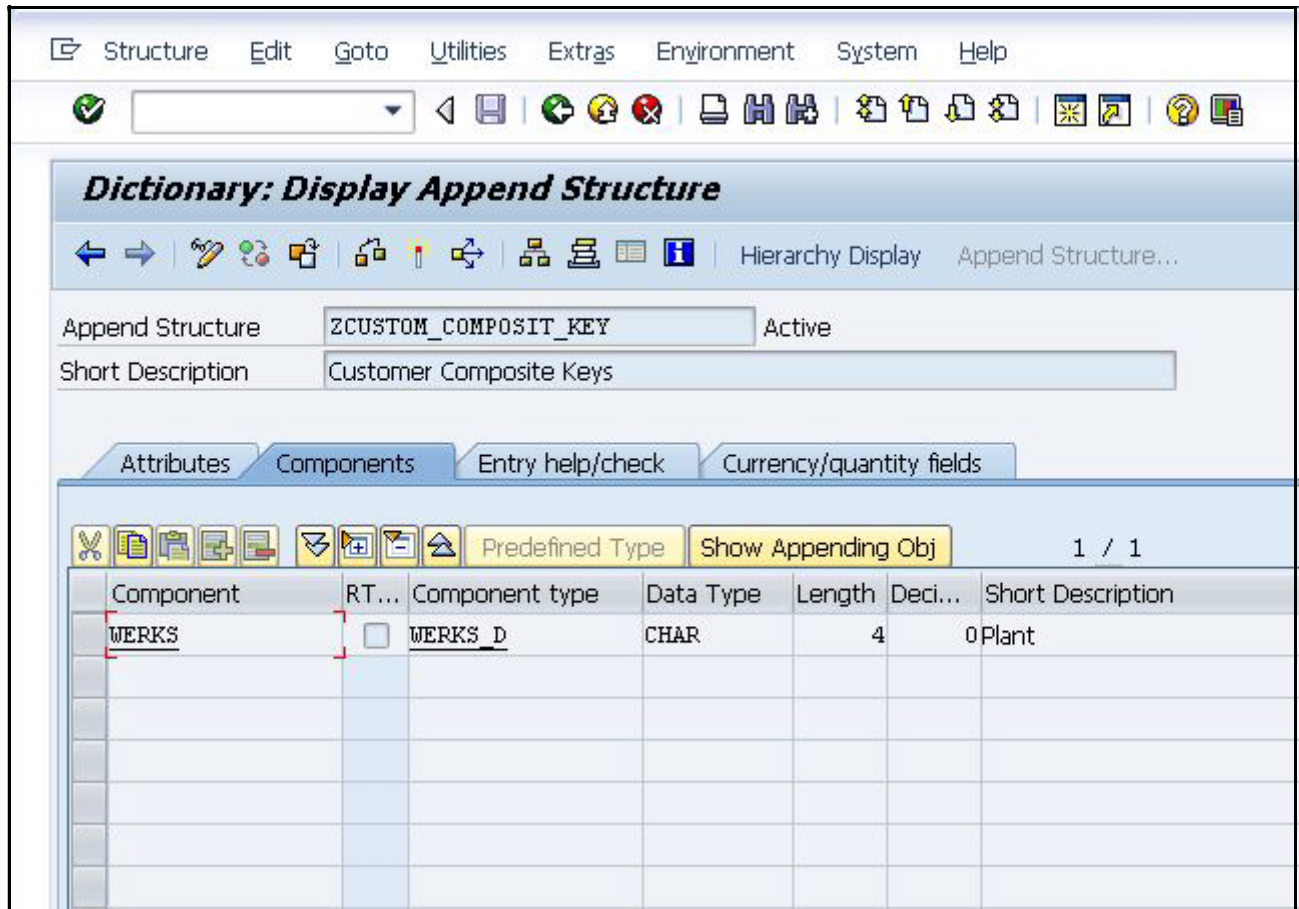


Figure 4-13: Adding a Composite Key as a New Component

- 7 Save and Activate the new component.
- 8 Save and Activate the new Append Structure.

## Adding New Classification Values

Custom Classification values can be added to the Security Classification Maintenance table.

**Note:** To prevent custom enhancements from being overwritten by NextLabs upgrades, create Append Structures in your customer namespace. Do not add them to the NextLabs namespace.

### Procedure

- 1 In the SAP interface, enter transaction SE11.
- 2 Select **Data Type**, and in the field, enter the structure name `/NEXTLABS/CLS_APPEND`. The default Classification values that default on installation display.

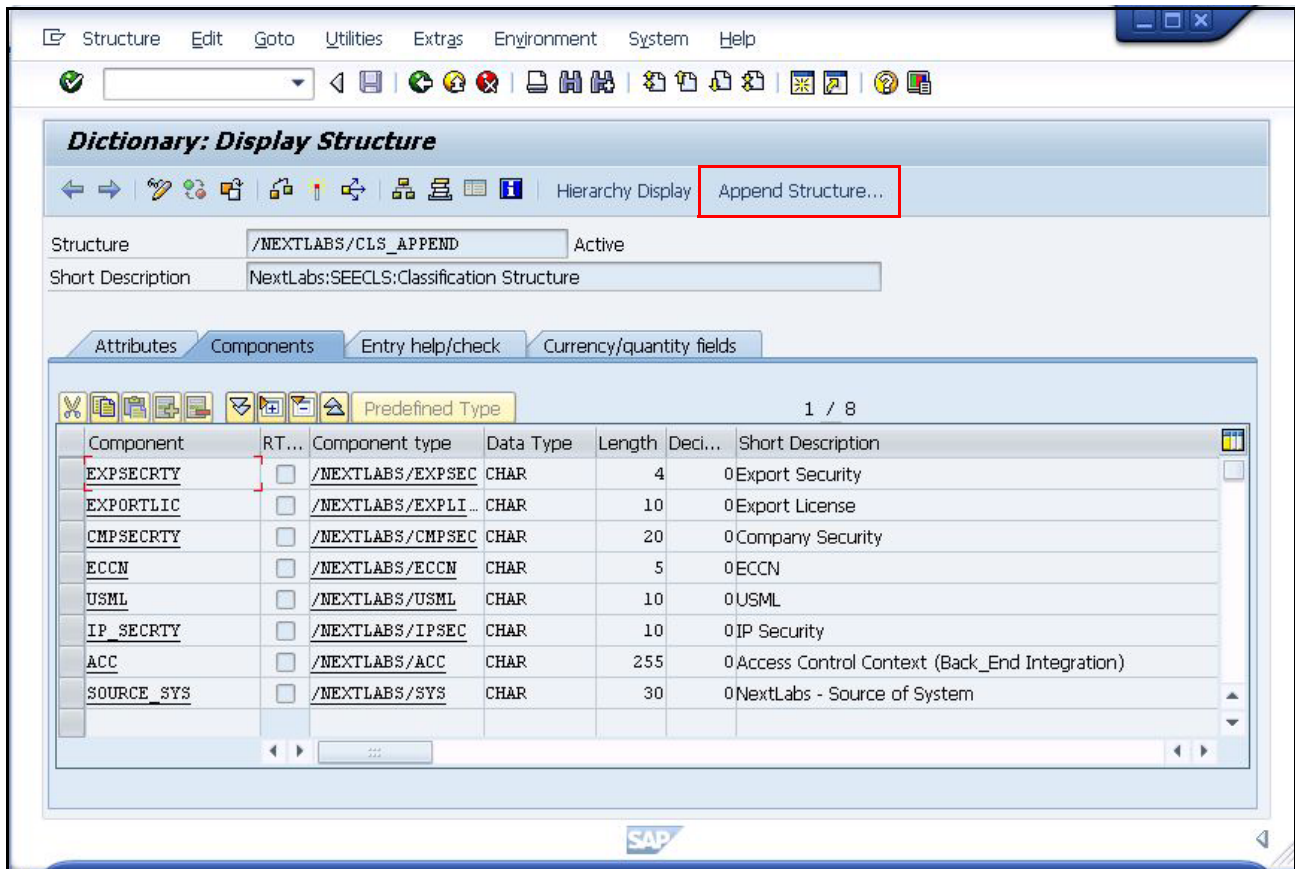
3 Click **Append Structure** on the toolbar.

Figure 4-14: Append Structure

- 4 If the message `No append defined` appears, click **OK**.
- 5 Click **Create Append**, enter an Append Structure name, then click **OK**.
- 6 In the *Append Structure* screen, enter information for a new Component(s).
  - a Enter the Component name.
  - b If you want to select a predefined Component type, enter it in the *Component type* screen (the characteristics of the Component type default in the Data Type, Length, and other fields).
  - c If you do not want to select a pre-defined Component type, define the component characteristics:
    - Enter the component Data Type (for example, CHAR).
    - Enter the Length of the field.
    - Enter a Short Description of the component.

- 4 **Save and Activate** New Component.
- 5 **Save and Activate** the new Append Structure.

### Next steps

The next step for Composite Keys is to associate them with a Security Identifier (see [Linking Composite Keys \(SECENH\)](#) on page 83). This step is not necessary for Classification Values. If you are configuring Classification Values only, the next step is [Mapping Security Fields \(SECM PG\)](#) on page 84.

---

## Linking Composite Keys (SECENH)

After adding Composite Keys, you must link them to a Security Identifier. This is done in SECENH.

### Procedure

- 1 In the SAP interface, enter transaction `SM30`. The *Table Maintenance View* screen appears.
- 2 In **Table/View**, enter `/NEXTLABS/SECENH`. Click **Display**. The *Maintain Security Identifier Composite Key* screen appears.
- 3 Change from Display to Change view.
- 4 Click **New Entries**.
- 5 For the existing Security Identifier `DOCNUM`, enter the following three Key Fields (as shown in [Figure 4-15](#)):
  - `DOKAR`: For Document Type
  - `DOKTL`: For Document Part
  - `DOKVR`: For Document Version
- 6 Associate each new Composite Key configured in the previous step with a Security Identifier. In [Figure 4-15](#), the Security Identifier for Material (`MATNR`) is linked with a Composite Key for Plant (`WERKS`).

**Note:** You can use the Search Help to see the list of valid Security Identifiers and Key Field (Composite Key) Names.



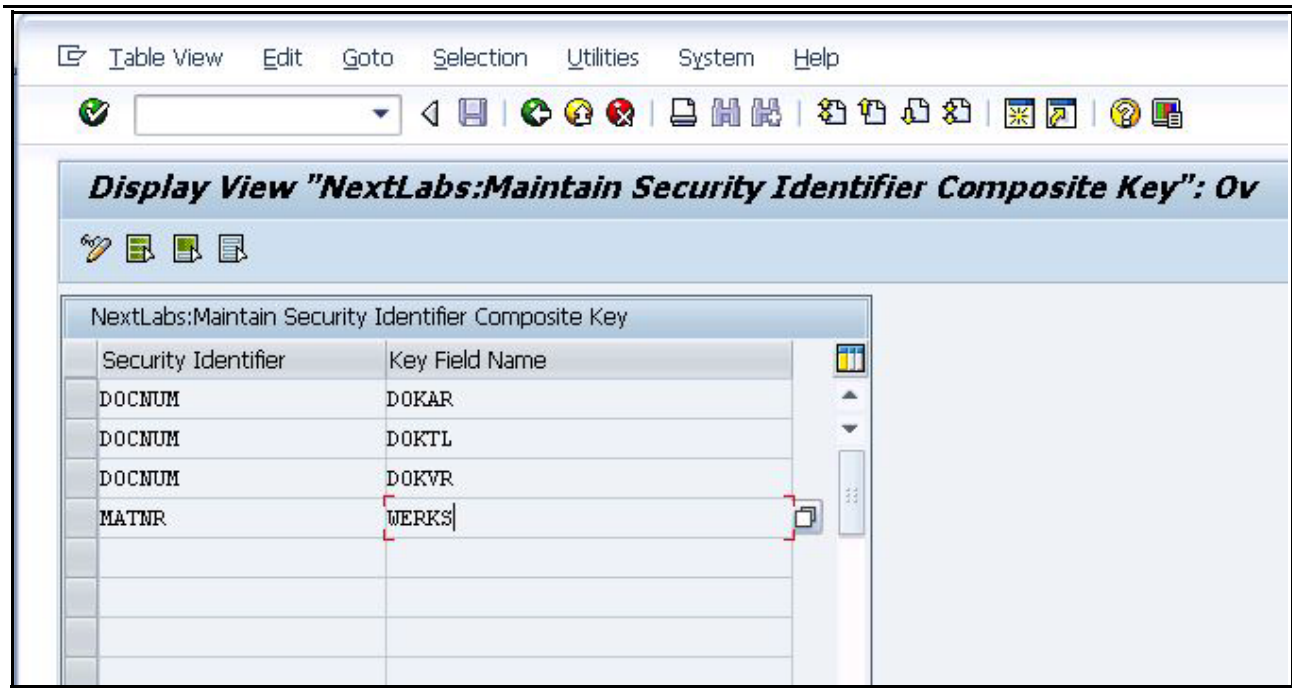


Figure 4-15: Linking Composite Keys to Security Identifiers

7 Save the changes.

### Next steps

The next step is [Mapping Security Fields \(SECMPG\)](#) on page 84 to the Policy Controller.

## Mapping Security Fields (SECMPG)

For classification fields configured in the Security Classification Maintenance table, you must perform the step of mapping values to the Policy Controller. This procedure is required for fields to be passed to the Policy Controller.

### Procedure

- 1 In the SAP interface, enter transaction `SM30`. The *Table Maintenance View* screen appears.
- 2 In **Table/View**, enter `/NEXTLABS/SECMPG`.



**Display View "Nextlabs: Security Fields Mapping": Overview**

Field name	Property name	Cardinality	Source
ECCN	ECCN	Single	Classification Data
EXPORTLIC	EXPORT LICENSE	Single	Classification Data
EXPSECRY	EXPORT SECURITY	Single	Classification Data
IP_SECRY	IP SECURITY	Multiple	Classification Data
TCODE	TRANSACTION	Single	Transaction Data
USML	MUNITIONS	Single	Classification Data

Figure 4-16: NextLabs Security Fields Mapping

- 3 For each classification field to configure:
  - a Enter the **Field Name** that was added to the Append Structure for Classification Values.
  - b Enter the **Property Name**. This is the name of the value that should be entered in Policy Studio when the classification value is added to policy in a Resource component.
 

**Note:** For more information on referencing classification values in policies, see [Designing SAP Access Control Policies](#) on page 204.
  - c Select **Multiple** or **Single** Cardinality.
  - d Enter the source of the classification data in the Source column:
    - If the value should be passed to the Policy Controller from the Security Classification table, select **Classification Data**.
    - If the value should be passed from a transaction, select **Transaction Data**.
- 5 Save the changes.

### Next steps

The next step is [Configuring Security Identifier/Composite Key Value Tables \(EPVAL\)](#) on page 86.

## Configuring Security Identifier/Composite Key Value Tables (EPVAL)

For each composite key, default security identifier, and custom security identifier in your implementation, designate the master data table and field from which data should be drawn, and select options for restricting the data and records to be validated.

This procedure must be performed for each SAP back-end system.

### Procedure

- 1 In the SAP interface, enter transaction `SM30`. The *Table Maintenance View* screen appears.
- 2 In **Table/View**, enter `/NEXTLABS/EPVAL` to access the *Security Identifier/Composite Key Value* screen.
- 3 Click **New Entries**. For each entry, specify the following information:
  - In **Security Identifier/Composite Key**, enter the security identifier or composite key added as part of your implementation.
  - In **Value Table**, enter the master data table to use to supply values for the security identifier or composite key.
  - In **Field Name**, enter the name of the field in the master data table that supplies the values for the security identifier (skip this step if you do not require a value table and select **Inactive for Validation**).
  - In **Validation Attributes**, select one of the following options:
    - **Active for Validation**: Select this option to validate field entries against a designated value table. If this option is selected, the field can come from any value table maintained in SAP. During validation, the table is checked to validate whether the user is entering a valid value in the Security Classification Maintenance table.
    - **Combine Validation with other fields**: Select this option to require a Security Identifier and/or its Composite key(s) to come from the same value table. If this option is selected, the field must come from the same record in the value table as other fields that both (1) come from the same table and (2) have **Combined Validation with other fields** selected. In other words, you can have one Security Identifier (DOCNUM) and two composite keys (DOKAR and DOKVR) mapped to the same table (DRAW). If **Combined Validation with other fields** is selected for DOCNUM, DOKAR, and DOKVR, a validation check ensures that the values come from the same record in the same table. In this case, the Security Identifier and Composite Keys are always bound as a data set.
    - **Inactive for Validation**: Select this option if you do not want to require validation against a value table in SAP. In this case, the values do not need to come from a table maintained within the SAP system. Any data can be entered in the *Security Classification Maintenance* table by a user.

Figure 4-17 shows an example of security identifiers configured in the EPVAL table.

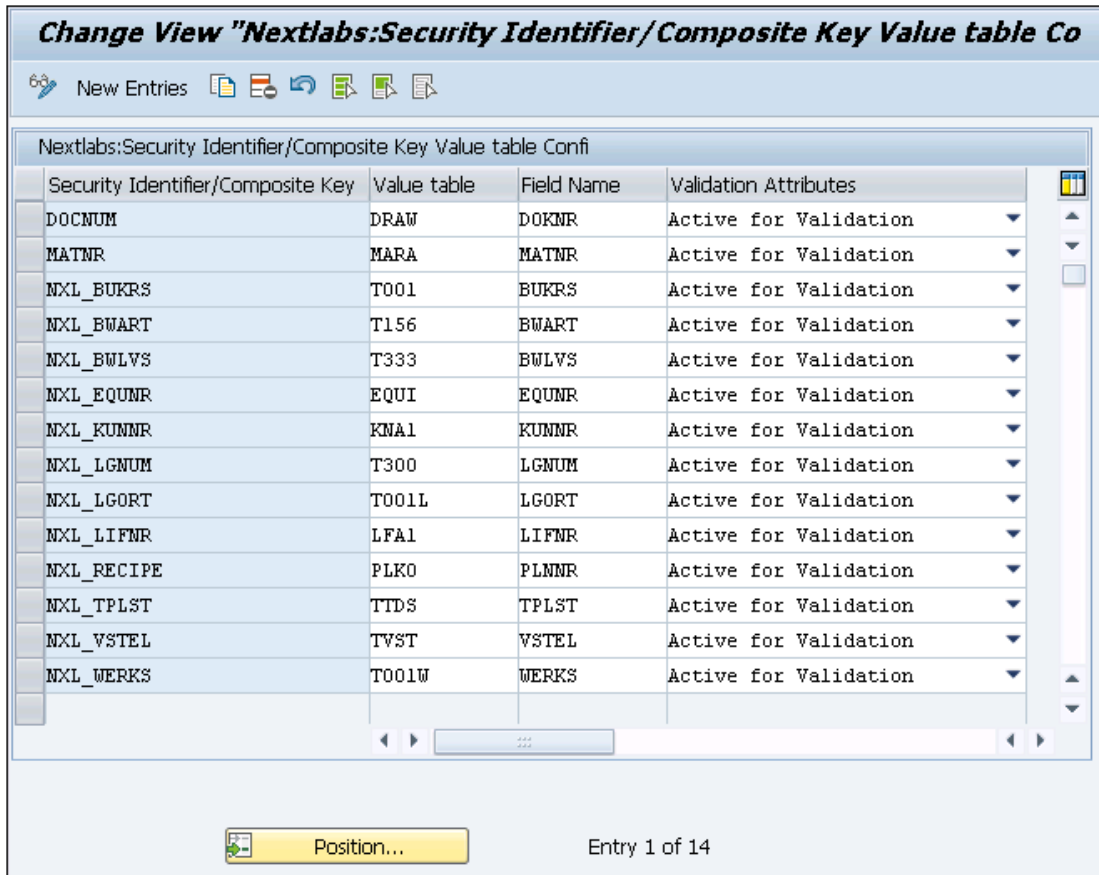


Figure 4-17: Configuring Security Identifiers and Value Tables

4 Click **Save**.

Table 4-1 lists the security identifiers and composite keys for all the Entitlement Packs, and their associated value tables and fields. The required identifiers must be configured for each installed Entitlement Pack.

Table 4-1: Security identifiers, value tables and fields

Entitlement Pack	Identifier/Composite Key	Value Table	Field
ECC/EasyDMS/PLM	DOCNUM	DRAW	DOKNR
ECC/EasyDMS/PLM	DOKAR	DRAW	DOKAR
ECC/EasyDMS/PLM	DOKTL	DRAW	DOKTL
ECC/EasyDMS/PLM	DOKVR	DRAW	DOKVR
ECC	MATNR	MARA	MATNR
ECC	NXL_BUKRS	T001	BUKRS
ECC	NXL_BWART	T156	BWART
ECC	NXL_BWLVS	T333	BWLVS

Table 4-1: Security identifiers, value tables and fields (Continued)

Entitlement Pack	Identifier/Composite Key	Value Table	Field
ECC	NXL_EQUNR	EQUI	EQUNR
ECC	NXL_KUNNR	KNA1	KUNNR
ECC	NXL_LGNUM	T300	LGNUM
ECC	NXL_LGORT	T001L	LGORT
ECC	NXL_LIFNR	LFA1	LIFNR
ECC	NXL_RECIP	PLKO	PLNNR
ECC	NXL_TPLST	TTDS	TPLST
ECC	NXL_VSTEL	TVST	VSTEL
ECC	NXL_WERKS	T001W	WERKS
cFolders	MAT_ID	CFF_LOIO	PROP09
cFolders	DOC_ID	CFF_LOIO	PROP09
BW	NXL_INFOAREA	RSDAREA	INFOAREA
BW	NXL_INFOPROV	RSDDTALOC	INFOPROV
S/4HANA	NXL_KOSTL	CSKS	KOSTL
S/4HANA	NXL_PARTNER	BUT000	PARTNER
S/4HANA	NXL_PRCTR	CEPC	PRCTR
S/4HANA	NXL_SAKNR	SKA1	SAKNR

### Next steps

If your implementation includes Entitlement Packs for PLM, EasyDMS, BW, SAP S/4HANA, or cFolders, the next step is [Configuring UI Functions](#) on page 88. Otherwise, skip to [Mapping Transaction Codes and UI Functions to Actions \(ACTIONS\)](#) on page 90.

## Configuring UI Functions

Before policies can be designed for SAP PLM, EasyDMS, SAP cFolders, SAP S/4HANA, or SAP BW, the UI functions related to the installed Entitlement Packs must be configured in the `/NEXTLABS/UIFUNC` table.

### Procedure

- 1 In the SAP interface, enter transaction `SM30`. The *Table Maintenance View* screen appears.
- 2 In **Table/View**, enter `/NEXTLABS/UIFUNC`, then click **Display**.
- 3 Click **Edit** to toggle from Display to Edit mode.

New Entries: Overview of Added Entries			
Maintain NextLabs Function			
NextLabs EM	UI Mode	Web Dynpro Component	Function

Figure 4-18: Entering New Functions in the UIFUNC table

- 4 Click **New Entries**.
- 5 Based on which Entitlement Packs you are configuring, enter values from [Table 4-2](#).

Table 4-2: Required Settings in the NextLabs Function Maintenance Screen

NextLabs EM	UI Mode	Web Dynpro Component	Function
NextLabs BW Addon	Display	ANALYZER	ANALYZER_DISPLAY
NextLabs BW Addon	Filter	ANALYZER	ANALYZER_FILTER
NextLabs cFolder Addon	Change	BOM_CFOLDER	CHANGE BOM
NextLabs cFolder Addon	Change	DOC_CFOLDER	CHANGE DOCUMENT
NextLabs cFolder Addon	Change	MAT_CFOLDER	CHANGE MATERIAL
NextLabs cFolder Addon	Display	BOM_CFOLDER	DISPLAY BOM
NextLabs cFolder Addon	Display	DOC_CFOLDER	DISPLAY DOCUMENT
NextLabs cFolder Addon	Display	MAT_CFOLDER	DISPLAY MATERIALS
NextLabs cFolder Addon	Deletion	BOM_CFOLDER	DELETE BOM
NextLabs cFolder Addon	Deletion	DOC_CFOLDER	DELETE DOCUMENT
NextLabs cFolder Addon	Deletion	MAT_CFOLDER	DELETE MATERIAL
NextLabs cFolder Addon	Filter	BOM_CFOLDER	FILTER BOM
NextLabs cFolder Addon	Filter	DOC_CFOLDER	FILTER DOCUMENT
NextLabs cFolder Addon	Filter	MAT_CFOLDER	FILTER MATERIAL
NextLabs cFolder Addon	Insertion	BOM_CFOLDER	CREATE BOM
NextLabs cFolder Addon	Insertion	DOC_CFOLDER	CREATE DOCUMENT
NextLabs cFolder Addon	Insertion	MAT_CFOLDER	CREATE MATERIAL
NextLabs cFolder Addon	Copy	BOM_CFOLDER	COPY BOM
NextLabs cFolder Addon	Copy	DOC_CFOLDER	COPY DOCUMENT
NextLabs cFolder Addon	Copy	MAT_CFOLDER	COPY MATERIAL
NextLabs EasyDMS Addon	Change	(Leave this field blank)	CHANGE DOCUMENT

Table 4-2: Required Settings in the NextLabs Function Maintenance Screen (Continued)

NextLabs EM	UI Mode	Web Dynpro Component	Function
NextLabs EasyDMS Addon	Display	(Leave this field blank)	DISPLAY DOCUMENT
NextLabs EasyDMS Addon	Filter	(Leave this field blank)	FILTER DOCUMENT
NextLabs EasyDMS Addon	Print	(Leave this field blank)	PRINT DOCUMENT
NextLabs EasyDMS Addon	Copy	(Leave this field blank)	COPY DOCUMENT
NextLabs PLM Addon	Change	/PLMU/WDA_DIR_OIF	CHANGE DOCUMENT
NextLabs PLM Addon	Change	/PLMU/WDA_ECN_OIF	CHANGE CHANGE MASTER
NextLabs PLM Addon	Change	/PLMU/WDA_MAT_OIF	CHANGE MATERIAL
NextLabs PLM Addon	Change	/PLMU/WDA_MBOM_OIF	CHANGE MATERIAL BOM
NextLabs PLM Addon	Display	/PLMU/WDA_DIR_OIF	DISPLAY DOCUMENT
NextLabs PLM Addon	Display	/PLMU/WDA_ECN_OIF	DISPLAY CHANGE MASTER
NextLabs PLM Addon	Display	/PLMU/WDA_MAT_OIF	DISPLAY MATERIAL
NextLabs PLM Addon	Display	/PLMU/WDA_MBOM_OIF	DISPLAY MATERIAL BOM
NextLabs PLM Addon	Deletion	/PLMU/WDA_ECN_OIF	DELETE CHANGE MASTER
NextLabs PLM Addon	Deletion	/PLMU/WDA_MBOM_OIF	DELETE MATERIAL BOM
NextLabs PLM Addon	Insertion	/PLMU/WDA_DIR_OIF	CREATE DOCUMENT
NextLabs PLM Addon	Insertion	/PLMU/WDA_MAT_OIF	CREATE MATERIAL
NextLabs PLM Addon	Insertion	/PLMU/WDA_MBOM_OIF	CREATE MATERIAL BOM
NextLabs PLM Addon	Copy	/PLMU/WDA_DIR_OIF	COPY DOCUMENT
NextLabs PLM Addon	Copy	/PLMU/WDA_ECN_OIF	COPY CHANGE MASTER
NextLabs PLM Addon	Copy	/PLMU/WDA_MAT_OIF	COPY MATERIAL
NextLabs PLM Addon	Copy	/PLMU/WDA_MBOM_OIF	COPY MATERIAL BOM

6 Click **Save**.

### Next steps

The next step is [Mapping Transaction Codes and UI Functions to Actions \(ACTIONS\)](#) on page 90.

## Mapping Transaction Codes and UI Functions to Actions (ACTIONS)

For each Entitlement Pack you are installing, you must also map transactions or UI functions (or both) to Policy Studio actions. By default, all SAP actions, whether transactions executed in SAP ECC or UI functions in SAP PLM, EasyDMS, or cFolders, are mapped to the Run action in Policy Studio. But this can be changed if a different mapping is required.

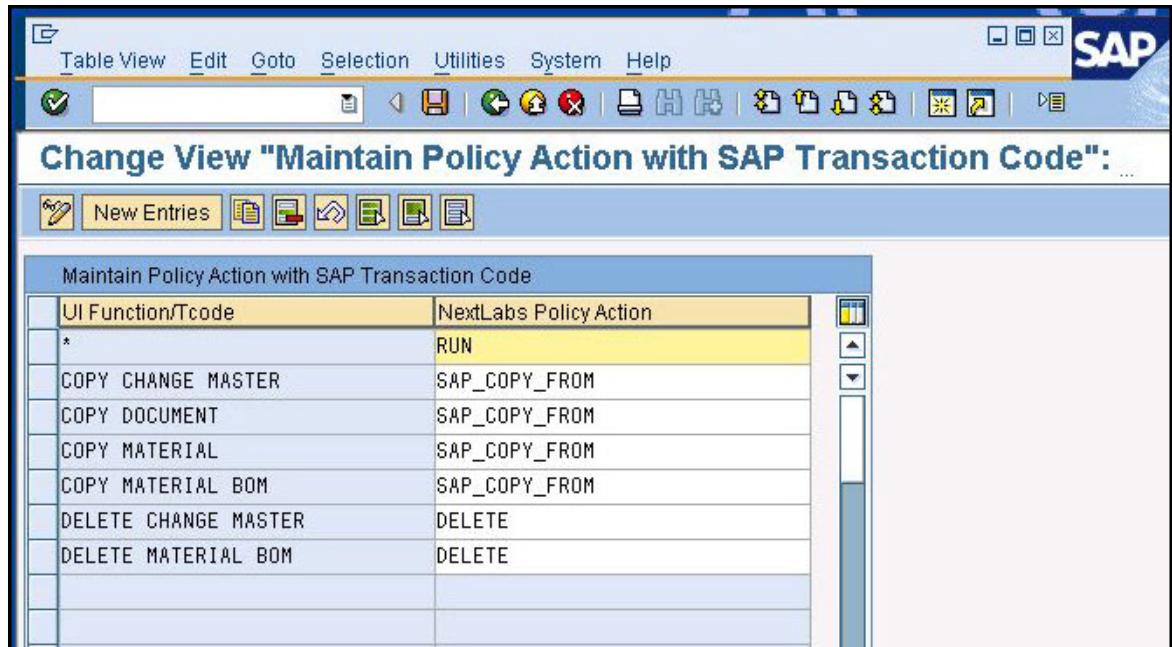
If your installation includes SAP PLM, it is recommended that you map two PLM UI functions (Copy From and Delete) to separate policy actions, to make the policies more fine-grained.

**Note:** For more information on defining the Copy from SAP action, see [Configuring SAP Actions](#) on page 69. For more information about the user events that trigger policy checks, see [What Can Dynamic Authorization Management Do?](#) on page 180

## Procedure

- 1 In the SAP interface, enter transaction SM30. The *Table Maintenance View* screen appears.
- 2 In *Table/View*, enter /NEXTLABS/ACTION, then Click **Display**.

All the entered action mappings display. In [Figure 4-19](#), a setting that defaults on installation maps all SAP Transactions and UI functions to Run. Additional mappings are entered for Copy From and Delete.



UI Function/Tcode	NextLabs Policy Action
*	RUN
COPY CHANGE MASTER	SAP_COPY_FROM
COPY DOCUMENT	SAP_COPY_FROM
COPY MATERIAL	SAP_COPY_FROM
COPY MATERIAL BOM	SAP_COPY_FROM
DELETE CHANGE MASTER	DELETE
DELETE MATERIAL BOM	DELETE

*Figure 4-19: Policy Action Mapping*

- 3 Click the toggle button to change from Display to Change view.
- 4 Click **New Entries** to add a new Action mapping.

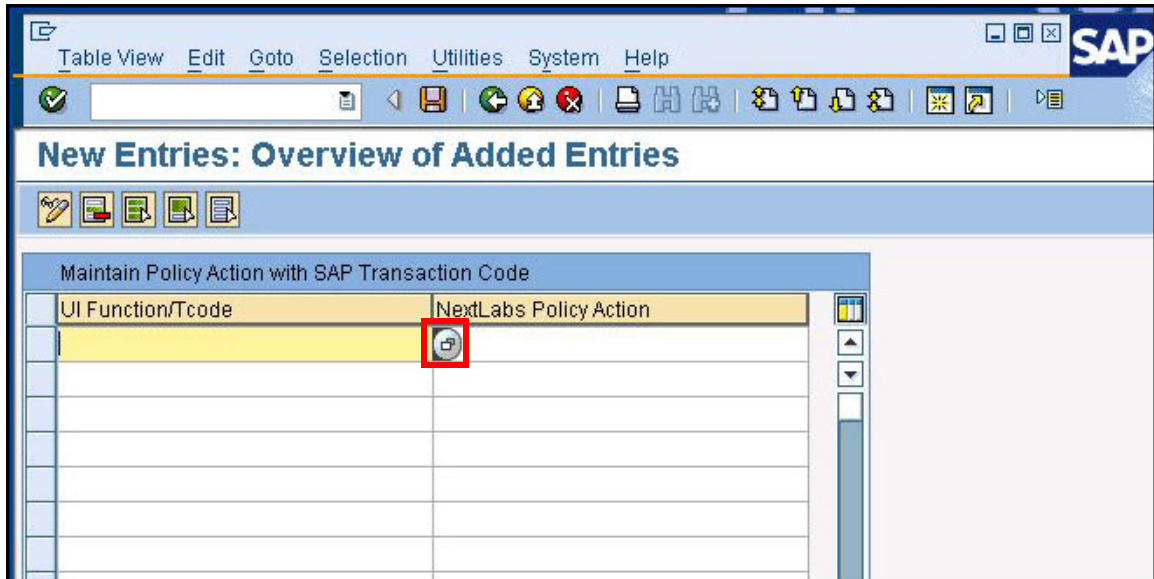


Figure 4-20: Adding a New Entry for Action Mapping

- 5 Enter a valid UI Function or Transaction Code, or click the Search Help icon on the right side of the field for the Search Help, where you can search for valid UI Functions or Tcodes, and select one from the list.

**Note:** If you are mapping the same action for multiple Add-on components (for example, “Display Document” for both EasyDMS and PLM), you only need to add it once. You cannot map the same action from different add-on components to different actions.



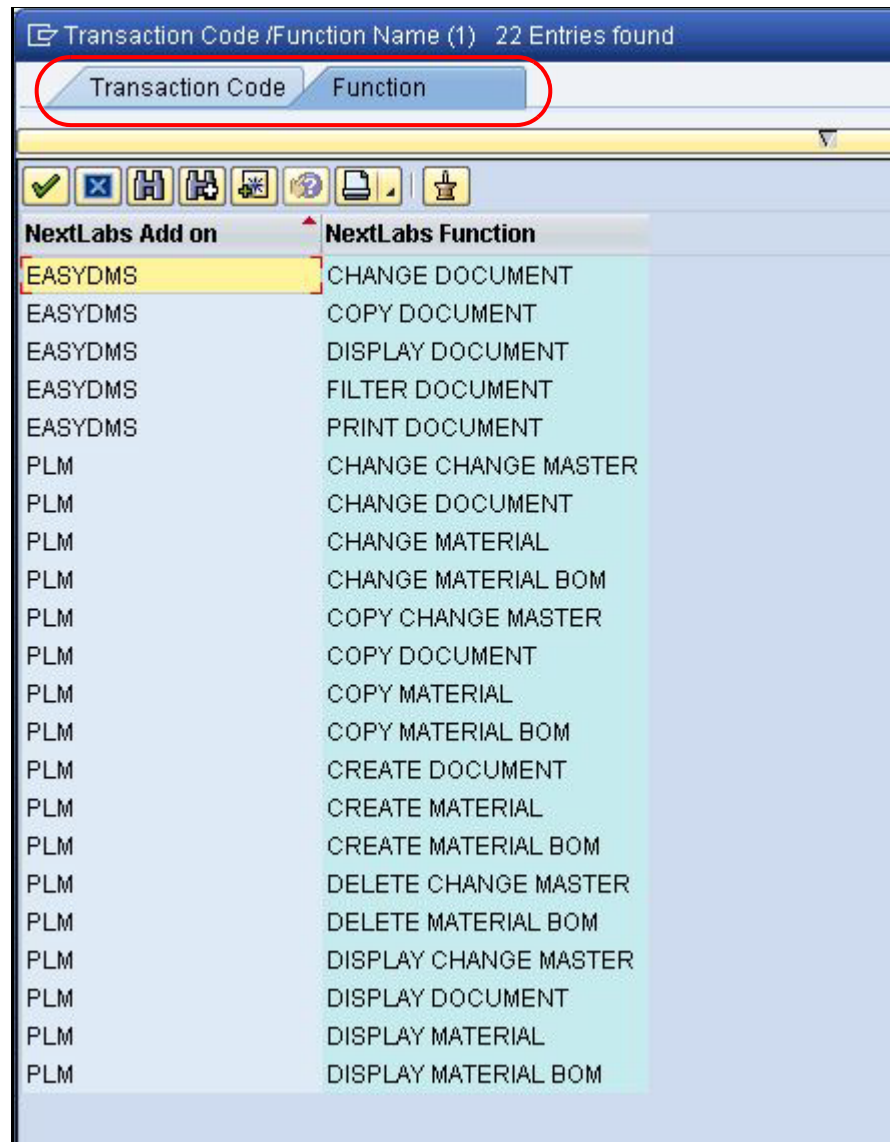


Figure 4-21: Search Help tabs for Transaction Codes and UI Functions

- 6 Enter the exact name of the action to which you are mapping the Transaction Code or UI Function. See [Table 4-3](#).

**Note:** There is no Search Help option for this field. Be sure to enter the Action name exactly as it appears in Policy Studio.

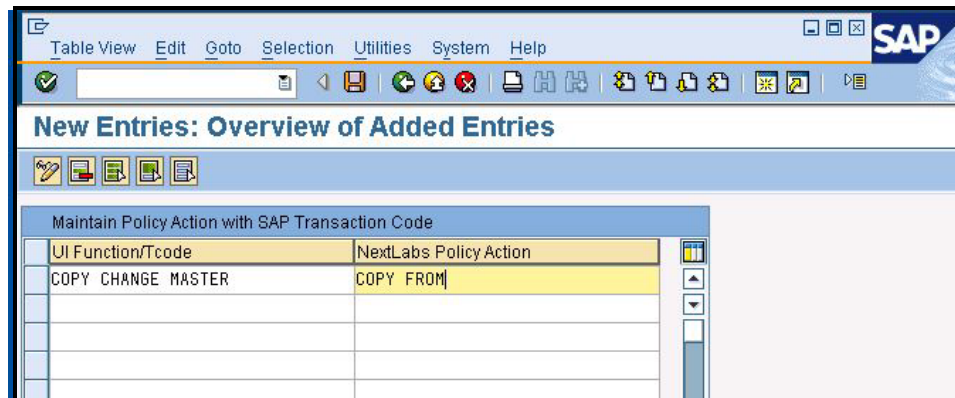


Figure 4-22: Mapping UI Function/Transaction Codes and Policy Actions

You can map any valid Transaction and UI Functions to any valid policy action to match your business processes and needs. However, the recommended mappings are shown in Table 4-3.

**Note:** All entries except for the last entry are for implementations that include SAP PLM only.

Table 4-3: Recommended Policy Action Mapping

Tcode/UI Transaction	Policy Action
*	RUN
COPY CHANGE MASTER	SAP_COPY_FROM
COPY DOCUMENT	SAP_COPY_FROM
COPY MATERIAL	SAP_COPY_FROM
COPY MATERIAL BOM	SAP_COPY_FROM
DELETE CHANGE MASTER	DELETE
DELETE MATERIAL BOM	DELETE

7 When you have finished entering new Policy Action mappings, click **Save**.

**Next steps**

The next step is [Configuring SAP Data Handling and Connection Settings](#) on page 94.

## Configuring SAP Data Handling and Connection Settings

The Connection Configuration table (/NEXTLABS/CONCFG) enables you to define numerous data handling and connection settings. This section describes each of these settings, including what

values are available. This section also describes how to use a configuration program to make this process more efficient.

## Setting Default Values Automatically

Rather than entering values for the settings in the Connection Configuration table manually, you can use the `/NEXTLABS/CONCFG_MAINTAIN` program to automatically populate the table with default settings. After these default settings are entered, you can change them as needed.

### Procedure

- 1 In the SAP interface, enter transaction `SE38` or `SA38`. The *Program Execution* screen appears.
- 2 Enter `/NEXTLABS/CONCFG_MAINTAIN` then click **Execute**.

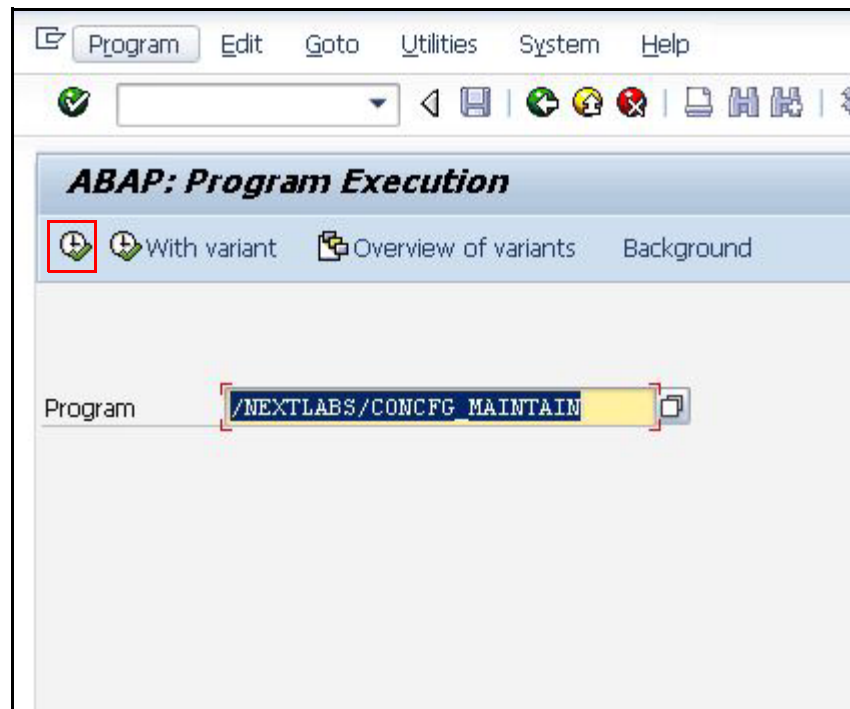


Figure 4-23: CONCFG Maintain Program

- 3 When prompted to select whether you want to both create the activity entries and **Update Entries with Default** settings (you can change the settings manually afterward), select or deselect this option, and click **Execute**.

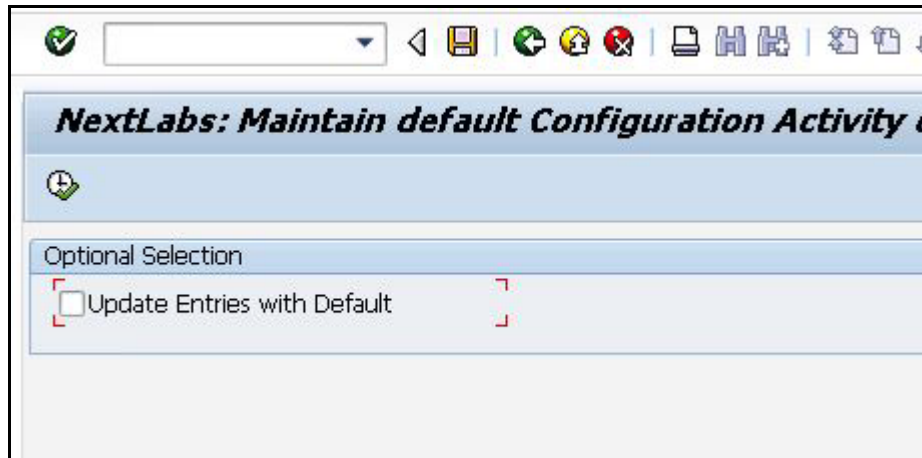


Figure 4-24: Update Entries with Default Settings Option

- 4 Optional: To change the settings later, go to the /NEXTLABS/CONCFG table. See [Changing Connection Configuration Settings](#) on page 96.

### Changing Connection Configuration Settings

The options that are available in the CONCFG table depend on which NextLabs products you have installed. For example, you only see the configuration settings related to cFolders if you installed and are configuring the NextLabs Entitlement Pack for cFolders. You can change the default settings in the /NEXTLABS/CONCFG table.

#### Procedure

- 1 In the SAP interface, enter transaction SM30. The *Table Maintenance View* screen appears.
- 2 In **Table/View**, enter /NEXTLABS/CONCFG. Click **Display**. The *Activity Maintenance Table* appears.
- 3 Toggle the table from Display to Change view.
- 4 Change the default settings, based on your business requirements. [Table 4-4](#) describes each setting and lists the valid options.

Table 4-4: Connection Configuration Settings

Activity Name	Description	Valid Settings
AGENT_COMMUNICATION_OPTION	Method for Policy Controller Communication Interface	RFC. In earlier versions, Webservice was an option. However, the web service interface is no longer supported in this release.

Table 4-4: Connection Configuration Settings (Continued)

Activity Name	Description	Valid Settings
AGENT_DEFAULT_DENY_OBLIGATION	If a policy definition does not include a default Deny message, you can define whether the default message that appears should come from a default NextLabs user alert (SAPMSG) or a default SAP Message Class (SAPMCL). The default SAPMSG is hard-coded and cannot be changed. The default SAPMCL message class (/NEXTLABS/CA000) can be modified.	SAPMSG or SAPMCL
AGENT_DEFAULT_IP_INCASE_OF_ERR	If your SAP implementation includes users for which there may not be an available IP addresses (for example, a service account user for running a background job), you can define a default IP address here that will be supplied for the user accounts.	IP address
AGENT_LOGICAL_PORT_NAME	Deprecated in release 7.5. In previous releases, this was the logical port name used to establish web services communication between SAP and the Policy Controller	Deprecated in release 7.5.
AGENT_RFC_ERR_WAITTIME_SECS	The number of seconds that should pass before an RFC call is considered an error	Integer
AGENT_RFC_NAME	Remote Function Call Destination name	The RFC Destination name created using the SM59 transaction (see <a href="#">Configuring the RFC Connection</a> )
BYPASS_TCODE_WHITELIST_CHECK	For internal configuration only. Do not change the default value without consulting NextLabs Support.	Blank: Disabled (Default)
CFOLDER_COPY_SECCLS_ON_CHANGE	cFolders classification information is stored in the Description. If the description is changed, the file can become unclassified. This setting determines whether the classification should be copied to a new Description. The default setting is Enabled.	<ul style="list-style-type: none"> <li>• X: Enabled (Default)</li> <li>• Blank: Disabled</li> </ul>
CFOLDER_VIEW_FILTER_ACTIVE	For cFolders, you can enable View Filtering, which means that users who are not authorized to access a file cannot view it in cFolders. If this setting is disabled, the file appears in the user's cFolders, but the user is denied access.	<ul style="list-style-type: none"> <li>• X: Enabled</li> <li>• Blank: Disabled (Default)</li> </ul>
ECC_SYNC_POLICY_CHECK	This setting enables a policy check to be performed whenever users run the /NEXTLABS/UPDCLS transaction. This transaction is used to update documents that have been exported to cFolders when their classification value changed in SAP ECC. The benefit of such a policy check would be to prevent unauthorized reclassification of documents in cFolders. (This setting does not automatically create the policy. The policy would need to be defined in Policy Studio.)	Yes: Enabled (Default) Space: Disabled
IRM_POLICY CHECK	This setting enables a policy check to be performed whenever users execute /NEXTLABS/IRM programs for cFolders and ECC. The benefit of the policy check would be to prevent unauthorized classification and/or encryption of originals. (This setting does not automatically create the policy. The policy would need to be defined in Policy Studio.)	Yes: Enabled (Default) Space: Disabled

Table 4-4: Connection Configuration Settings (Continued)

Activity Name	Description	Valid Settings
IRM_UPLOAD_NO_IF_RECORDS_WARNG	Determines how many upload records may exist before a warning appears. Users can configure IRM batch processing to ignore this warning.	Numeric Values (1 to 9999) Default is 999
IRM_UPLOAD_WAIT_TIME	This setting determines how long the interval should be between IRM upload processes (which process the OUT folder on the RMS and uploads originals back into SAP ECC or SAP cFolders).	Numeric Values in seconds (1 to 180 seconds) Default is 60
MULTI_QUERY_TIMEOUT_IN_MILLISC	The SAP Agent collects transaction information from multiple policy queries and routes it to the Policy Controller. In the event that communication fails, because the Policy Controller is down or some other reason, SAP needs to be configured for when and how it should respond. Each multi-query request sent to the Policy Controller contains groups of objects. A timeout occurs when any one group exceeds the timeout value you set.	Activity Type: value in milliseconds, must be greater than 0. The default is 40000.  Activity Handler: in the event of a timeout, SAP takes the action that is specified in the Activity Handler for TIME_PERIOD_IN_MILLISECONDS.
NO_SEC_IDT_SELECTED	Determines what should happen if a user enters a transaction that has not been assigned a default Identifier. For example, if a user attempts to display a BOM, which contains multiple Materials with different classification settings, but there is no setting for how to prioritize these Identifiers, this setting determines whether SAP should allow or prevent the execution of the transaction.	<ul style="list-style-type: none"> <li>• Error: SAP displays an Error message to the user and prevents the transaction.</li> <li>• Information: SAP displays an Informational message to the user and allows the transaction.</li> <li>• Blank: if the field is left blank, SAP allows the transaction to proceed with displaying a message to the user.</li> </ul>
PBSC_CREATE_QUEUE_FOR_CFOLDER	Before a PBSC queue is created on the cFolders system, the system checks this setting to learn whether or not PBSC is enabled.	X: Enabled Space: Disabled
PBSC_POLICY_CHECK	This setting enables a policy check to be performed whenever users execute /NEXTLABS/PBSC programs for cFolders and ECC. The benefit of the policy check would be to prevent unauthorized classification of business objects. (This setting does not automatically create the policy. The policy would need to be defined in Policy Studio.)	Yes: Enabled (Default) Space: Disabled
PBSC_UPLOAD_WAIT_TIME	This setting determines how long the interval should be between PBSC upload jobs (which take any new files in the OUT folder of upload originals back into SAP ECC or SAP cFolders).	Numeric Values in seconds (1 to 180 seconds) Default is 60
READTAG_AGENT_RFC_NAME	Remote Function Call Destination name for Read Tags	The RFC Destination name created using the SM59 transaction (see <a href="#">Configuring the RFC Connection for Read Tags</a> )
SEC_CLS_POLICY_CHECK	This setting enables a policy check to be performed whenever users execute /NEXTLABS/SEC_CLS programs for cFolders and ECC. The benefit of the policy check would be to prevent unauthorized classification of business objects. (This setting does not automatically create the policy. The policy would need to be defined in Policy Studio.)	Yes: Enabled (Default) Space: Disabled

Table 4-4: Connection Configuration Settings (Continued)

Activity Name	Description	Valid Settings
SEC_CLS_SCREEN_CHANGE_HISTORY	This setting determines whether or not Change History appears by default in the <i>NextLabs Security Classification Selection</i> screen. This setting is the default. Users can change it if needed.	X: Display Change History as a selection option Blank: Do not display change history as a selection option
SEC_CLS_SCREEN_CLASSIFICATION	This setting determines whether or not classification fields display by default in the <i>NextLabs Security Classification Selection</i> screen. This setting is the default. Users can change it if needed.	X: Display classification fields as a selection option Blank: Do not display classification fields as a selection option
SEC_CLS_SCREEN_COMPOSITE_KEY	This setting determines whether or not Composite keys display by default in the <i>NextLabs Security Classification Selection</i> screen. This setting is the default. Users can change it if needed.	X: Display Composite keys as a selection option Blank: Do not display Composite keys as a selection option
SEC_CLS_SCREEN_IDENTIFIER	This setting determines whether or not Identifiers display by default in the <i>NextLabs Security Classification Selection</i> screen. This setting is the default. Users can change it if needed.	X: Display Identifier as a selection option Blank: Do not display Identifier as a selection option
SEC_CLS_SCREEN_WARNING_MESSAGE	When more than a configured number of records are fetched, based on criteria entered in the <i>NextLabs Security Classification Selection</i> screen, a warning appears. Users are asked whether or not they wish to continue. This setting determines the number of records that triggers this warning.  If a large warning number is supplied here, it can impact the amount of time users have to wait for fetched records.  Also, this warning message does not prevent users from entering large numbers, but it does enable them to cancel the request.	Integer for the number of records before a warning is displayed (default 200)
SEC_CLS_UPLOAD	This setting determines the behavior of importing security classification information (from an Excel or .csv file) into the Security Classification Maintenance table. The options are whether you want to update new values only, or overwrite (meaning, delete all values and write new ones to the table).	Overwrite: overwrite values (default) Blank: update only
SEC_CLS_UPLOAD_HAS_HEADER	This setting determines behavior for importing security classification data (from an Excel or .csv file) into the Security Classification Maintenance table. If your source files have a header you can configure the update process to ignore the header when the values are imported.	X: Ignore first row (it is a header) Blank: Do not ignore first row (it is not a header)
SEC_CLS_UPLOAD_NO_OF_COLS	This setting determines behavior for importing security classification data (from an Excel or .csv file) into the Security Classification Maintenance table. You can set the number of columns to be imported from the source file. If the number you set is less than the number of columns in a source file, the excess columns are not imported.	Numeric Values (1 to 30) Default is 20



Table 4-4: Connection Configuration Settings (Continued)

Activity Name	Description	Valid Settings
SEC_CLS_UPLOAD_NO_OF_ROWS	This setting determines behavior for importing security classification data (from an Excel or .csv file) into the Security Classification Maintenance table. You can set the number of rows to be imported from the source file.	Numeric values (1 to 64000) Default is 50000
SEC_CLS_WILDCARD_COMPOSITE_KEY	This setting determines whether to allow wildcard entries for Composite key fields in the Security Classification Maintenance table  There are important best practice recommendations associated with this setting. For more information, see <a href="#">Recommended Configuration for Implementations with Many Classifications (more than 40,000 rows)</a> on page 102.	X: Allow wildcards for Composite keys Blank: Do not allow wildcards for Composite keys
SEC_CLS_WILDCARD_IDENTIFIER	Allow wildcard entries to Identifier value fields in the Security Classification Maintenance table  There are important best practice recommendations associated with this setting. For more information, see <a href="#">Recommended Configuration for Implementations with Many Classifications (more than 40,000 rows)</a> on page 102.	X: Allow wildcards for Identifier values Blank: Do not allow wildcards for identifier values
TIME_PERIOD_IN_MILLISECONDS	The SAP Agent collects transaction information from a single policy query and routes it to the Policy Controller. In the event that communication fails, because the Policy Controller is down or some other reason, SAP needs to be configured for how it should respond.	Activity Type: value in milliseconds, must be greater than 0. The default is 25000.  Activity Handler: enter what should happen in the event of a timeout. The option you select here also applies to MULTI_QUERY_TIMEOUT_IN_MILLISC. <ul style="list-style-type: none"> <li>• Error: SAP displays an error message to the user and prevents the transaction.</li> <li>• Information: SAP displays an Informational message to the user and allows the transaction.</li> <li>• Blank: If the field is left blank, SAP allows the transaction to proceed without displaying a message to the user.</li> </ul>

- 5 Verify that your changes are associated with the appropriate transport request by following these steps:
  - a Select **Menu > Table View > Transport**. A pop-up requests that you specify whether you want to create a new request or use an existing configuration request.
  - b Click the green check mark.





Figure 4-25: Customizing Request Screen

- c Click the **Select all** icon (or F7).
- d Click **Include in Request**.

**Display View "Nextlabs: Default Configuration Maintenance": Overview**

Activity Name	Activity Type	Activity Handler
AGENT_COMMUNICATION_OPTION	RFC	
AGENT_DEFAULT_DENY_OBLIGATION	SAPMSG	
AGENT_DEFAULT_IP_INCASE_OF_ERR	255.255.255.0	
AGENT_LOGICAL_PORT_NAME	/NEXTLABS/SAPAGENT_TO_WSDL	
AGENT_RFC_NAME	NEXTLABS_PC	
BYPASS_TCODE_WHITELIST_CHECK		
CONFIG_POLICY_CHECK	YES	
MULTI_QUERY_TIMEOUT_IN_MILLISC	40000	
PBSC_POLICY_CHECK	YES	
PBSC_UPLOAD_WAIT_TIME	60	
READTAG_AGENT_RFC_NAME	NEXTLABS_READTAG	
SEC_CLS_POLICY_CHECK	YES	
SEC_CLS_SCREEN_CHANGE_HISTORY	X	
SEC_CLS_SCREEN_CLASSIFICATION		
SEC_CLS_SCREEN_COMPOSITE_KEY		
SEC_CLS_SCREEN_IDENTIFIER	X	
SEC_CLS_SCREEN_WARNING_MESSAGE	200	
SEC_CLS_UPLOAD	OVERWRITE	
SEC_CLS_UPLOAD_HAS_HEADER	X	
SEC_CLS_UPLOAD_NO_OF_COLS	20	
SEC_CLS_UPLOAD_NO_OF_ROWS	50000	
SEC_CLS_WILDCARD_COMPOSITE_KEY	X	
SEC_CLS_WILDCARD_IDENTIFIER		
TIME_PERIOD_IN_MILLISECONDS	25000	Error

Figure 4-26: Select All and Include in Request

- 5 Click **Save** to save the changes you made to /NEXTLABS/CONCFG.

## Recommended Configuration for Implementations with Many Classifications (more than 40,000 rows)

If you anticipate having a large number of classification records (more than 40,000 rows) in the Security Classification Maintenance table, NextLabs recommends that you do one of the following to improve system performance:

- Do not use wildcards for Identifiers and Composite Keys. These settings are configured in the `SEC_CLS_WILDCARD_COMPOSITE_KEY` and `SEC_CLS_WILDCARD_IDENTIFIER` fields (see [Connection Configuration Settings](#) on page 96)
- If you must use wildcards for Identifiers and Composite keys, your BASIS Administrator should create an index in the `/NEXTLABS/SECIDT` table for the `MANDT` field and the relevant Security Identifiers and/or Composite keys for which wildcards are being used.

### Next steps

If your configuration includes the Entitlement Pack for EasyDMS, the next step is [Configuring View Filtering \(EasyDMS Only\)](#) on page 102. Otherwise, skip to [Configuring Number Range Intervals](#) on page 103.

---

## Configuring View Filtering (EasyDMS Only)

You can configure view filtering for EasyDMS using the *EasyDMS Configuration* screen. When this configuration is enabled, you can design a filter policy that restricts which documents display when users log in to EasyDMS.

If view filtering is not enabled, users can view the presence of documents that they do not have authority to access. Users cannot, however, access (display, change, edit) the documents, as determined by policies.

### Procedure

- 1 In the SAP interface, enter transaction `SM30`. The *Table Maintenance View* screen appears.
- 2 In **Table/View**, enter `/NEXTLABS/EDMS`, then click **Display**.
- 3 Toggle the mode from Display to Change.
- 4 Click **New Entries**, and select **Allow View Filter Policy**.

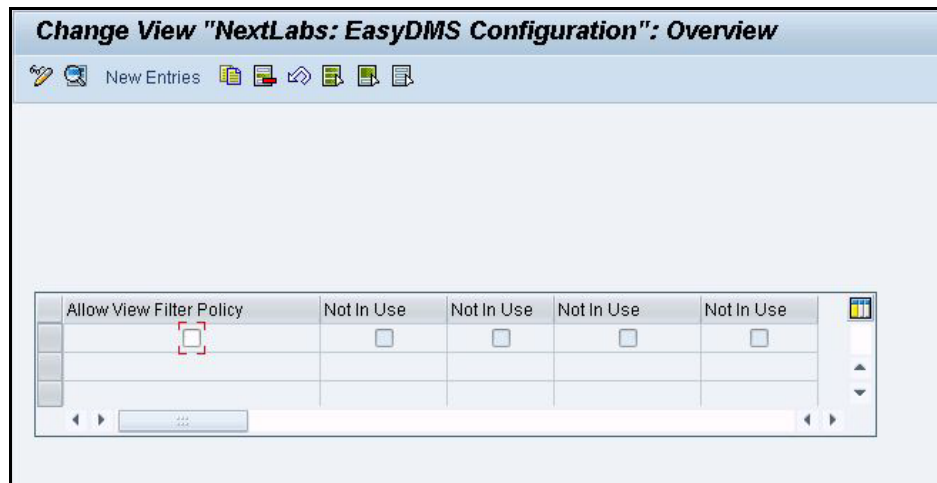


Figure 4-27: Selecting Allow View Filter Policy

5 Click **Save**.

### Next steps

The next step is [Configuring Number Range Intervals](#) on page 103.

## Configuring Number Range Intervals

Number range intervals are used for IDs for NextLabs internal operations, for instance, for security classification records and for PBSC logs and queues. Number Range Intervals are configured in the `/NEXTLABS/NRCONF` table.

After you have configured Number Ranges here, you can embed them as Sub objects of the NextLabs Number Range (which is done as a later configuration step; see [Configuring the NextLabs Number Range on page 104](#)).

### Procedure

- 1 In the SAP interface, enter transaction `SM30`. The *Table Maintenance View* screen appears.
- 2 In **Table/View**, enter `/NEXTLABS/NRCONF` and click **Display**. The Activity Maintenance Table appears.
- 3 Toggle from View to Change mode.
- 4 Define the following Sub Objects:
  - Security Classification Number Range

- PBSC Log Number Range
- PBSC Queue Number Range

**Note:** For implementations that do not include PBSC, no number ranges are required.

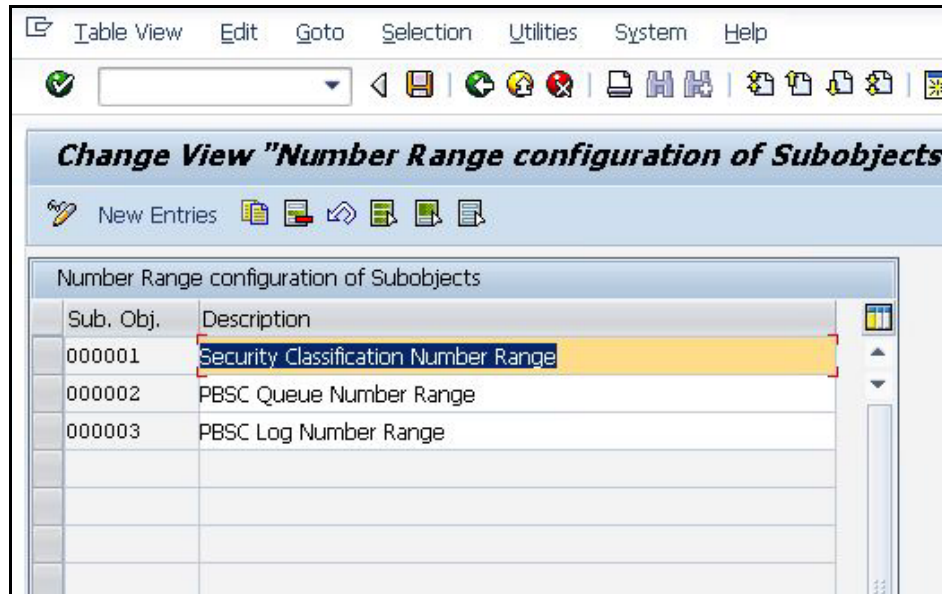


Figure 4-28: Reference ID Number Range Interval

5 Click **Save** to save your changes.

### Next steps

The next step is [Configuring the NextLabs Number Range](#) on page 104.

## Configuring the NextLabs Number Range

After you have assigned Number Ranges for NextLabs processes (see [Configuring Number Range Intervals on page 103](#)), you are ready to associate them as sub objects of the NextLabs object. If SAP ECC, SAP cFolders, or SAP BW are installed on separate systems, this procedure must be performed on each system.

### Procedure

- 1 In the SAP interface, enter transaction `SNRO`. The *Number Range Object Maintenance* screen appears.

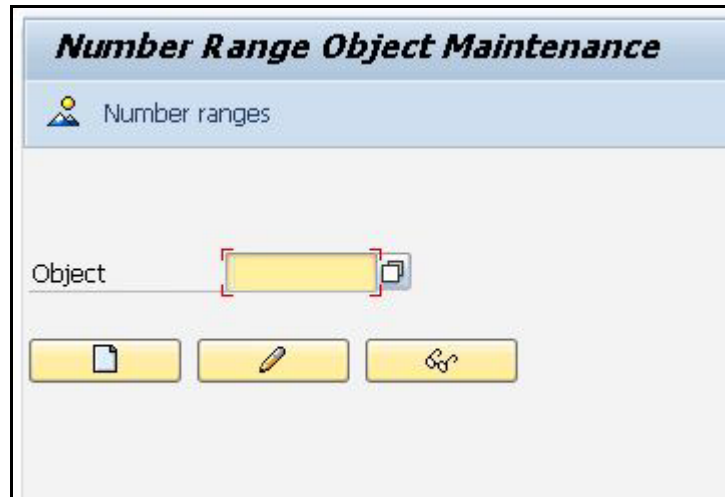


Figure 4-29: Number Range Object Maintenance

- 2 Enter or select the `/NEXTLABS/` object and click **Change** to edit the object.

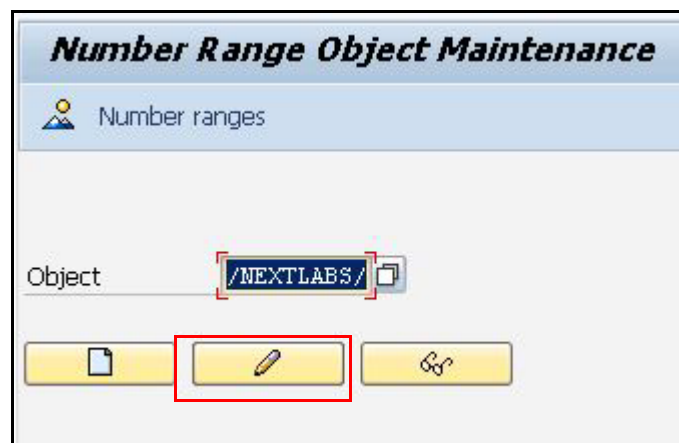


Figure 4-30: Modifying the `/NEXTLABS/` Object

- 3 In the *Interval Change* screen, select the **Number Range** option.

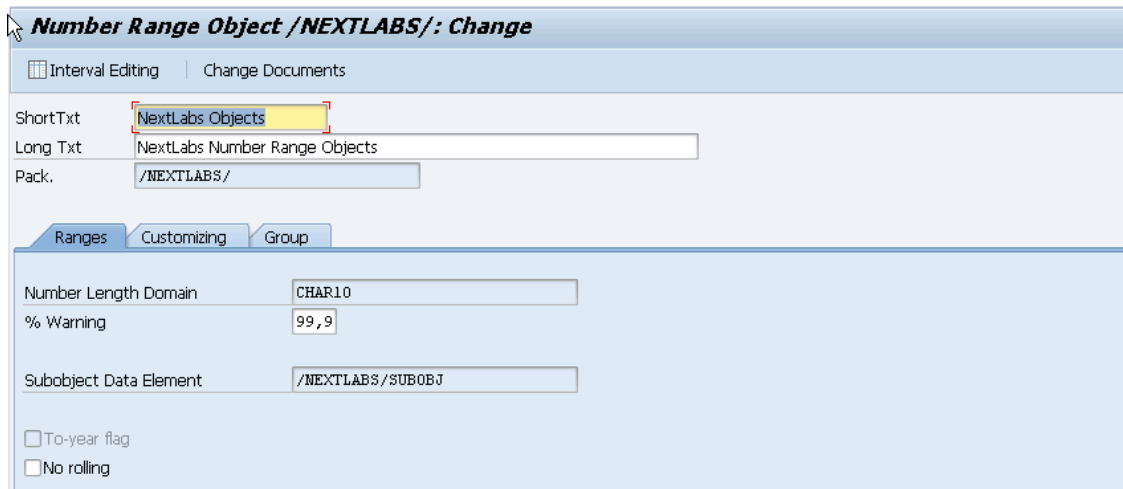


Figure 4-31: Number Ranges

- 4 In the Sub Object field, you can click the Search Help button to retrieve a list of sub objects that have been defined for /NEXTLABS/ (as part of a prior configuration step).

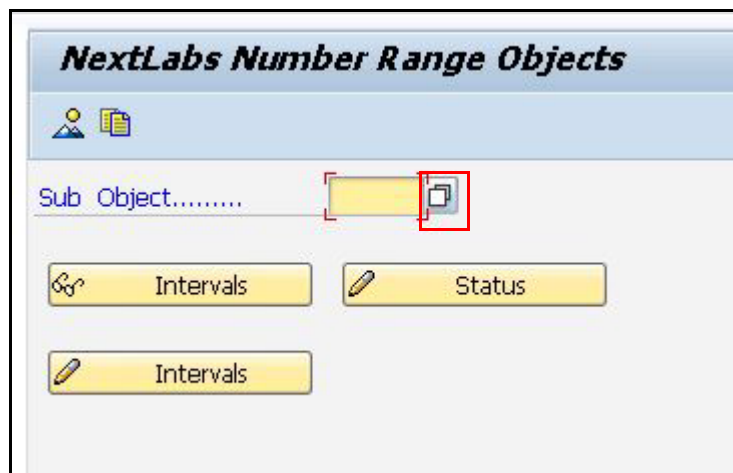


Figure 4-32: Searching for Pre-Defined Sub Objects

- 5 Select a Sub Object by double-clicking it. (You need to perform this procedure for all Sub Objects.)

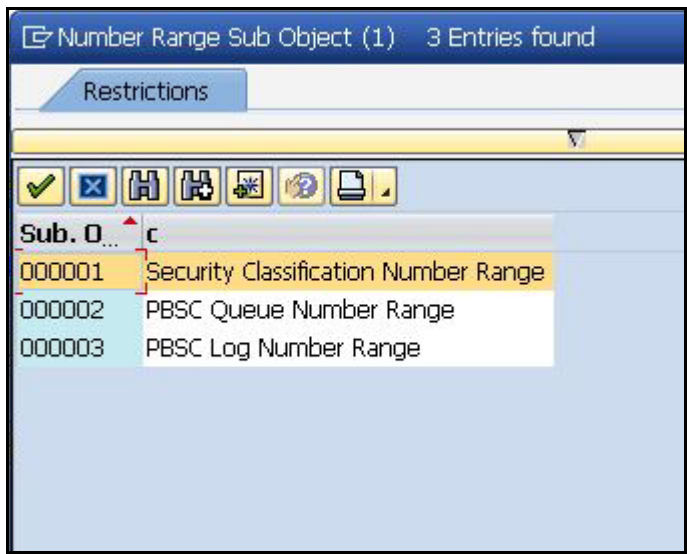


Figure 4-33: Selecting a Pre-Defined Sub Object

6 With the sub object selected, click **Change Intervals**.

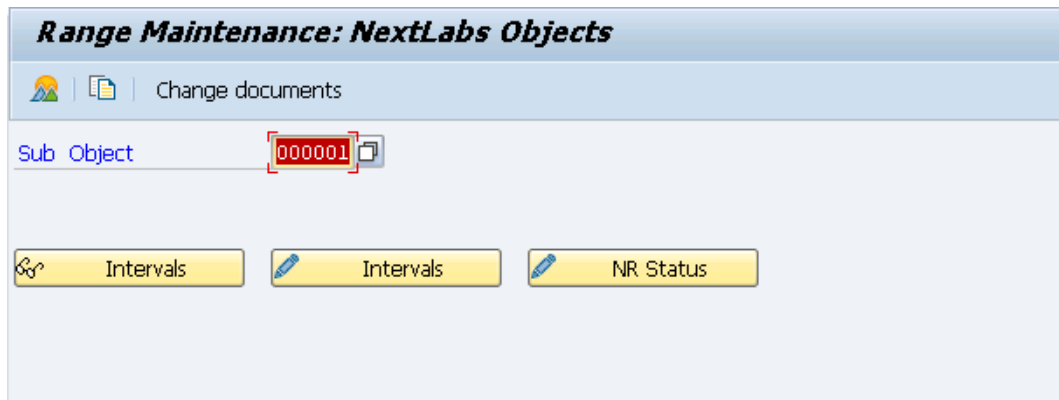


Figure 4-34: Changing Intervals

7 In the *Maintain Number Range Intervals* screen, click **Insert Interval**.

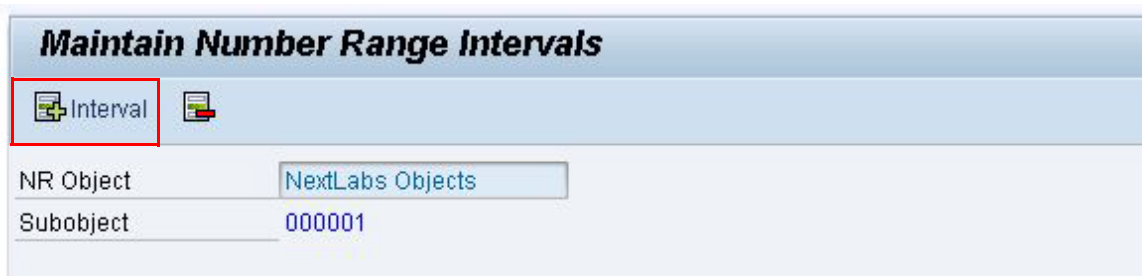


Figure 4-35: Inserting an Interval

- 8 Enter the number as From 1 and To 999999999 and click Insert Interval.



Figure 4-36: Inserting an Interval

- 9 Click **Save** to save your configuration changes.
- 10 Repeat these steps for any other Sub Objects in the /NEXTLABS/ object. Or, you can copy the defined intervals from one Sub Object to another using the Copy button.

**Note:** If your implementation does not include PBSC, you do not need to include the number ranges for PBSC.



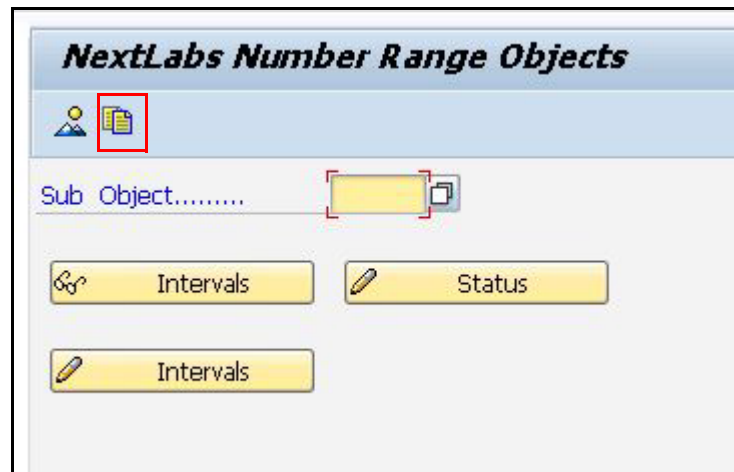


Figure 4-37: Copying Sub Object Definition to Another Sub Object

### Make the NextLabs Namespace Unmodifiable

After configuring the NextLabs Number Range, you should make the NextLabs namespace unmodifiable. This prevents users from making changes in the NextLabs namespace, which run the risk of being overwritten on a subsequent installation or upgrade of a NextLabs product.

**Note:** The only modifications supported for the NextLabs namespace are officially released NextLabs product code. Customers should not store other modifications to code in the NextLabs namespace because it can result in installation and upgrade issues.

### Next steps

The next step is [Configuring Policy Checks Based on Transaction/UI Function](#) on page 109.

## Configuring Policy Checks Based on Transaction/UI Function

You can configure one of two options for how policy checks should occur for business objects: to occur only when there is a classification applied to the object, or to occur for all business objects regardless of whether they are classified. You can set this configuration for each transaction and/or UI function. In other words, you may want policy checks for every business object for certain transactions/UI functions, but policy checks should only occur for classified business objects for other transactions/UI functions.

### Procedure

- 1 In the SAP interface, enter transaction `SM30`. The *Table Maintenance View* screen appears.

2 In **Table/View**, enter `/NEXTLABS/CHKCLS` and click **Display**. The Activity Maintenance Table appears. The default setting (\*) specifies that policy checks are enabled for all transactions/UI functions.

You can make exceptions to the default setting by adding different settings for specific transaction codes or functions.

3 Toggle from **View** to **Change** mode.

4 Click **New Entities** to add a new record to the table.

5 Enter a transaction and/or UI function and select the desired security classification check behavior. The options are:

- **Check Classified Only:** enable policy checks only when there is classification data for a business object.
- **Check All:** enables policy checks for classified and unclassified business objects.

In [Figure 4-38](#), transaction MM01 is configured to have policy checks occur only when classifications are present for the business object. For all other transactions and UI functions, policy checks occur whether or not classification values are present.

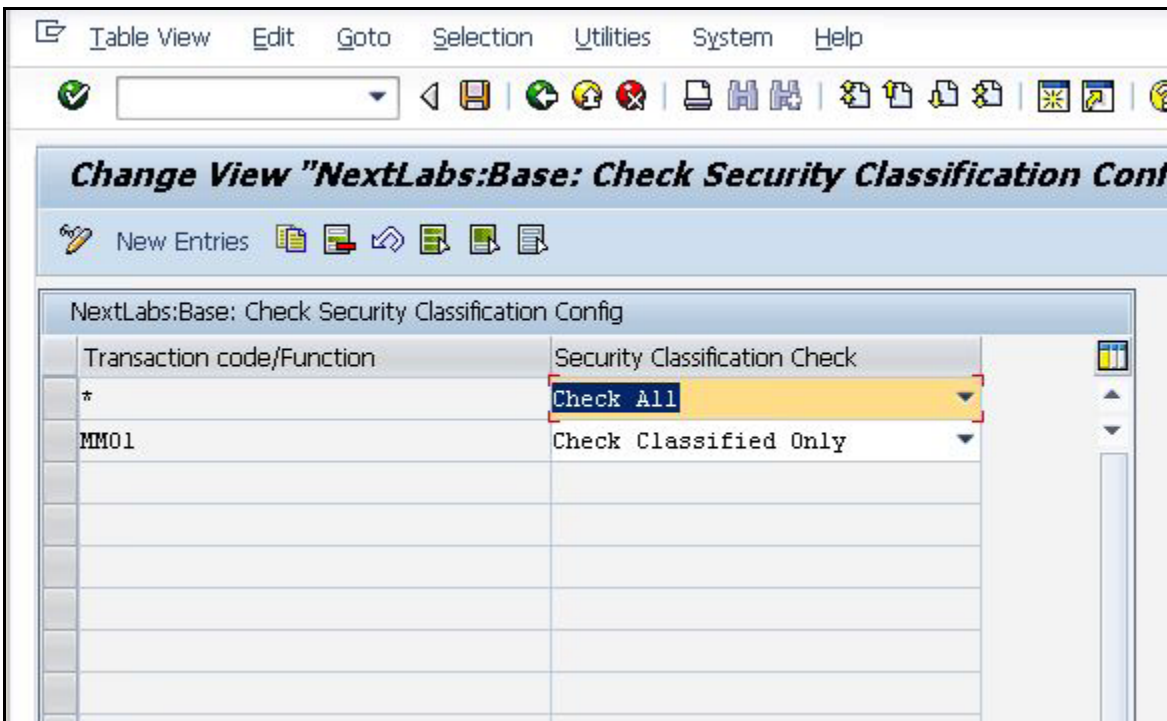


Figure 4-38: Configuring Policy Checks based on Transaction/UI Function

6 When you have entered all the required information, click **Save**.

## Next steps

The next step is [Configuring Special Fields for the Security Classifications Maintenance Table](#) on page 111.

## Configuring Special Fields for the Security Classifications Maintenance Table

In the `/NEXTLABS/EPCLS` table, you can change the properties of fields that exist in the Security Classification Maintenance table. In particular, you can mark fields as Read Only or Do Not Display.

The fields that need to be configured in this table depend on which Entitlement Packs are included in your implementation. One field, `ACC`, should be configured to support the exporting of ACCs to cFolders. Another field, `SOURCE_SYS`, should be configured to track the source system where a business object originated.

### Procedure

- 1 In the SAP interface, enter transaction `SM30`. The *Table Maintenance View* screen appears.
- 2 In **Table/View**, enter `/NEXTLABS/EPCLS` and click **Display**. The *Activity Maintenance Table* appears.
- 3 Toggle from **Display** to **Change** mode.
- 4 Click **New Entities** to add new values to the table.
- 5 Enter the values for the Entitlement Packs in your implementation as shown in [Figure 4-1](#).

*Table 4-5: Values for NextLabs Entitlement Packs*

NextLabs Entitlement Pack	Classification Field	Field Option in SECCLS UI
NXLECC	ACC	Read Only
NXLECC	SOURCE_SYS	Do Not Display
NXLBW	EXPSECRTY	Read Only
NXLBW	SOURCE_SYS	Do Not Display
NXLEDMS	EXPSECRTY	Read Only
NXLEDMS	SOURCE_SYS	Do Not Display
NXLPLM	EXPSECRTY	Read Only
NXLPLM	SOURCE_SYS	Do Not Display

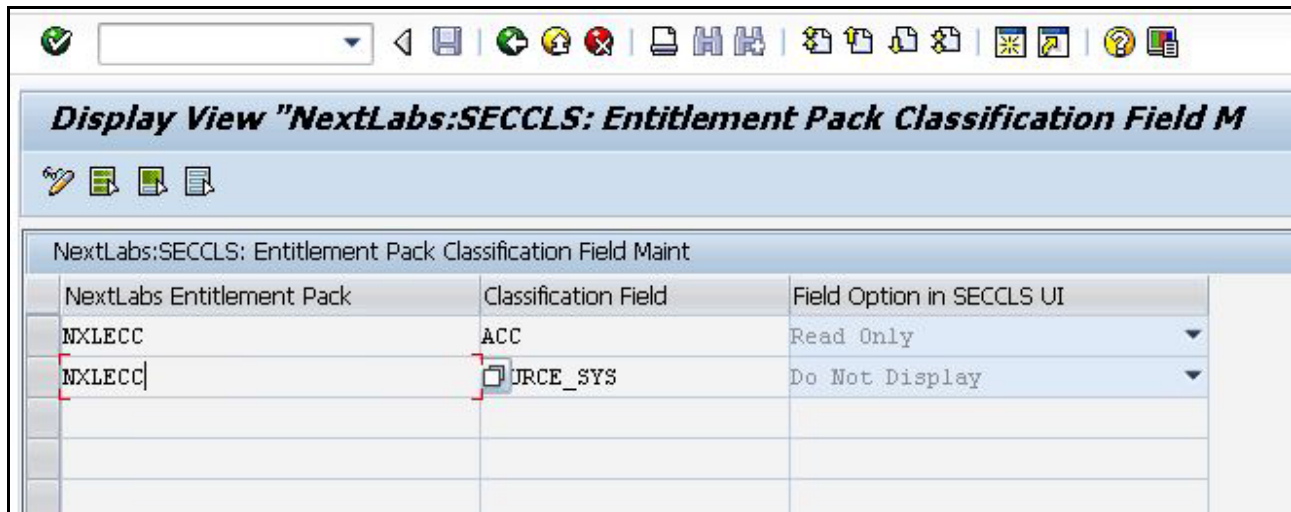


Figure 4-39: Configuring Special Fields for Security Classification Maintenance

6 Click Save.

### Next steps

If your implementation includes the Entitlement Pack for PLM, The next step is [Configuring Access Control Context Settings \(PLM Only\)](#) on page 121. Otherwise, skip to [Defining How Multiple Security Classifications Should Be Applied](#) on page 116.

## Defining How Security Classifications and Access Control Contexts Should Be Applied

You can configure whether a Security Classification, Access Control Context, or both, should be applied to a transaction or UI function. At the very least, you must set a default configuration for all transactions and UI functions that are applicable to your implementation. After this default is set, you can specify exceptions for transactions or UI functions that are exceptions to the default rule.

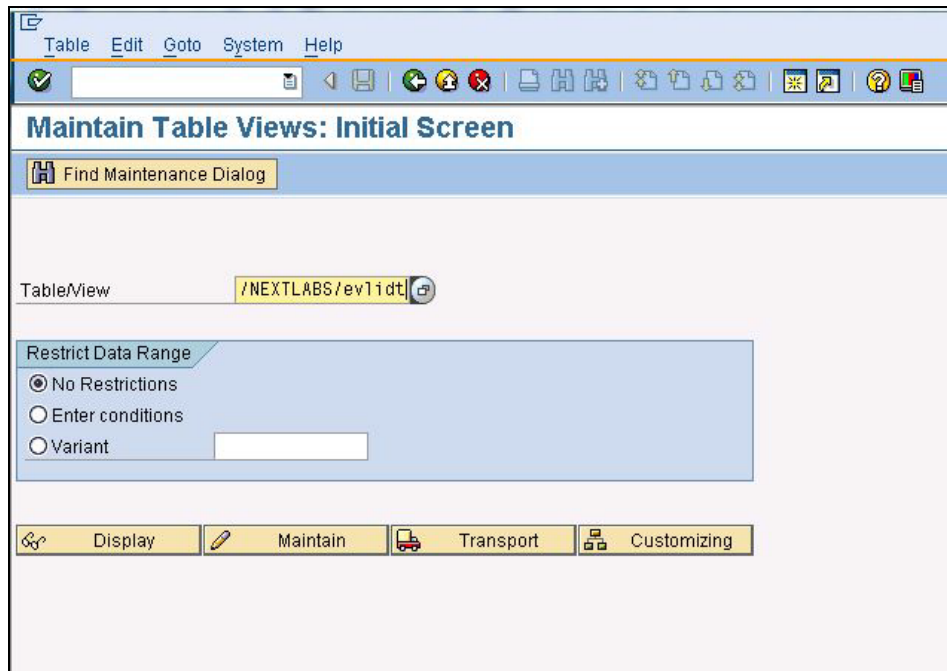
**Note:** For information about the transactions and UI functions that can be entered in this table, see [What Can Dynamic Authorization Management Do?](#) on page 180

For cases where a business object can also contain other business objects (as with Materials, Documents, BOMs, Routings, Engineer Workbench, and Change Masters), you can configure Dynamic Authorization Management so that a Security Classification, Access Control Context, or both, are applied for the parent business object. However, child business objects are evaluated by their Security Classification only. In other words, if you have a Document with multiple Materials associated with it, a policy check includes both the Security Classification and the Access Control Context for the parent Document, but only the applicable Security Classifications for the child Materials linked to that Document.

You can configure when Security Classification or Access Control Context should be prioritized.

### Procedure

- 1 In the SAP interface, enter transaction SM30. The *Table Maintenance View* screen appears.
- 2 In *Table/View*, enter `/NEXTLABS/EVLIDT`, then click **Display**.



*Figure 4-40: Accessing the Evaluation Identifier Table*

The *NextLabs Evaluation Identifier* screen displays all records that have been defined. In the example in [Figure 4-41](#), a wildcard (\*) specifies the default that both Access Control Context and Security Classification be used for evaluation when both identifiers are present on a single transaction or UI function. The example also shows settings that specify that Access Control Context is prioritized when users Copy, Change, or Display Change Masters in SAP PLM.



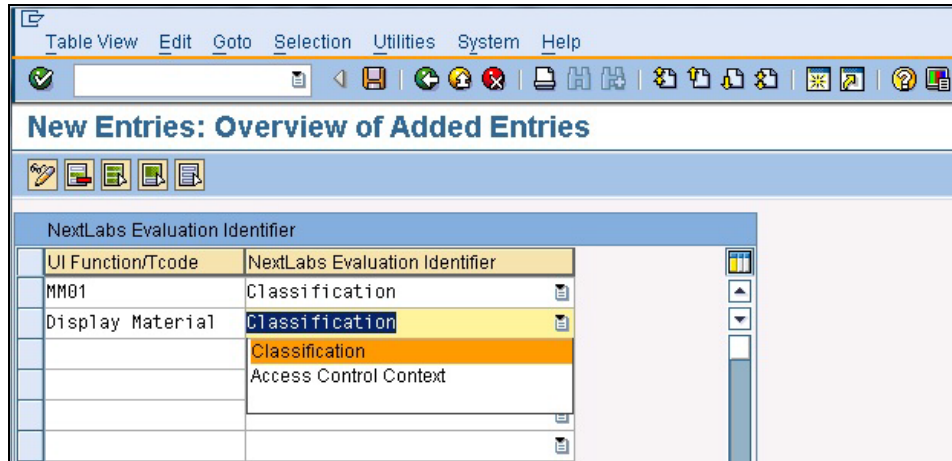


Figure 4-42: Adding New Evaluation Settings

- 7 When you have finished entering activation settings, click **Enter** and **Back** to return to the *NextLabs Security Classification Selection* screen. Each new activation record you created appears as a new row.
- 8 When you have entered all the necessary settings, click **Save**.

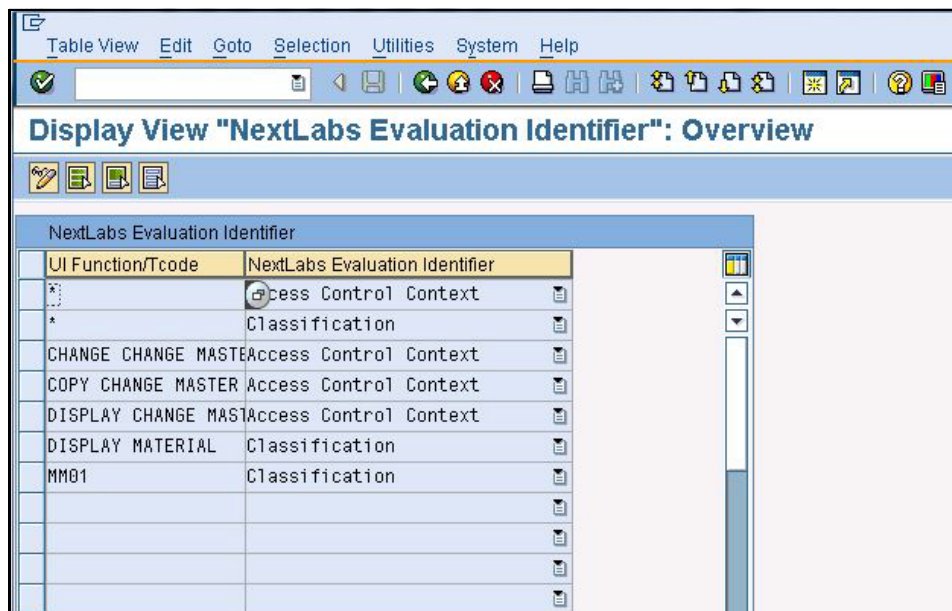


Figure 4-43: Added Lines in the Evaluation Identifier Screen

### Next steps

The next step is [Defining How Multiple Security Classifications Should Be Applied](#) on page 116.



---

## Defining How Multiple Security Classifications Should Be Applied

After setting up all the identifiers (columns in the Security Classification table), you must specify the transactions for which you want each identifier to be checked. You should also decide how classifications should be applied when multiple values are present on a single transaction, UI function, or document access event in EasyDMS. This can be the case for Materials, Documents, versions of Materials or Documents, BOMs, Routings, Engineer Workbench, and Change Masters (which are all SAP objects that may contain one or more other SAP objects within them).

You can configure how Security Classifications should be applied in the NextLabs Security Identifier Selection table. This table configures how Security Classifications and Access Control Contexts are applied for given user events.

**Note:** For more information on how to determine whether a Security Classification or Access Control Context should be applied when both are present for a parent business object, see [Defining How Security Classifications and Access Control Contexts Should Be Applied](#) on page 112.

For instance, let's say a Document has multiple Materials linked on it, and that the parent Document and children Materials all have different Security Classifications. You can configure how these classifications should be applied when a user executes a transaction or UI function to access the parent Document.

For certain transactions or UI functions, you may want to prioritize the classification status of the parent Document over the children Materials. For other transactions or UI functions, you may want to ensure that the Security Classification of both parents and children are enforced (in other words, that users must be authorized to access *both* to access the parent business object). Aside from the default Identifiers Document and Material, you may also need to prioritize the classification settings for the other custom Identifiers you have defined.

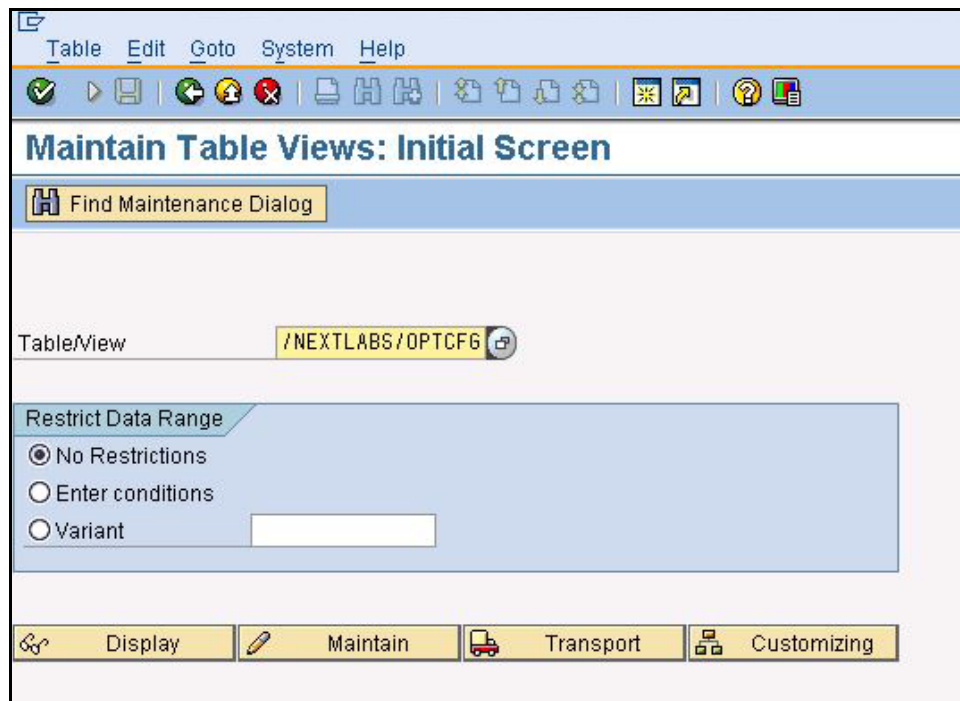
You configure how multiple classifications should be applied in the *Security Classification Selection Configuration* screen by selecting identifiers for specific transactions, UI functions, and/or EasyDMS actions. It is good practice to create a default activation that applies to all transactions, UI functions, and actions for an identifier (for example, a default that all transactions, UI functions, and actions executed that involve a Document should apply the classification status of the Document over any materials linked to the document), and then make determinations for how classifications should be activated for individual transactions as exceptions to the default rule. All transactions that are not explicitly activated receive the default activation setting.

**Note:** For information about the transactions and UI functions that you can include in this table, see [What Can Dynamic Authorization Management Do?](#) on page 180.

### Procedure

- 1 In the SAP interface, enter transaction SM30. The *Table Maintenance View* screen appears.
- 2 In *Table/View*, enter /NEXTLABS/OPTCFG, then click **Display**.





*Figure 4-44: Accessing the Option Configuration Table*

The *NextLabs Security Classification Selection* screen displays all records that have been defined. As shown in [Figure 4-45](#), a wildcard is used to set the default behavior that, for all transactions, both Materials and Documents are activated by default.

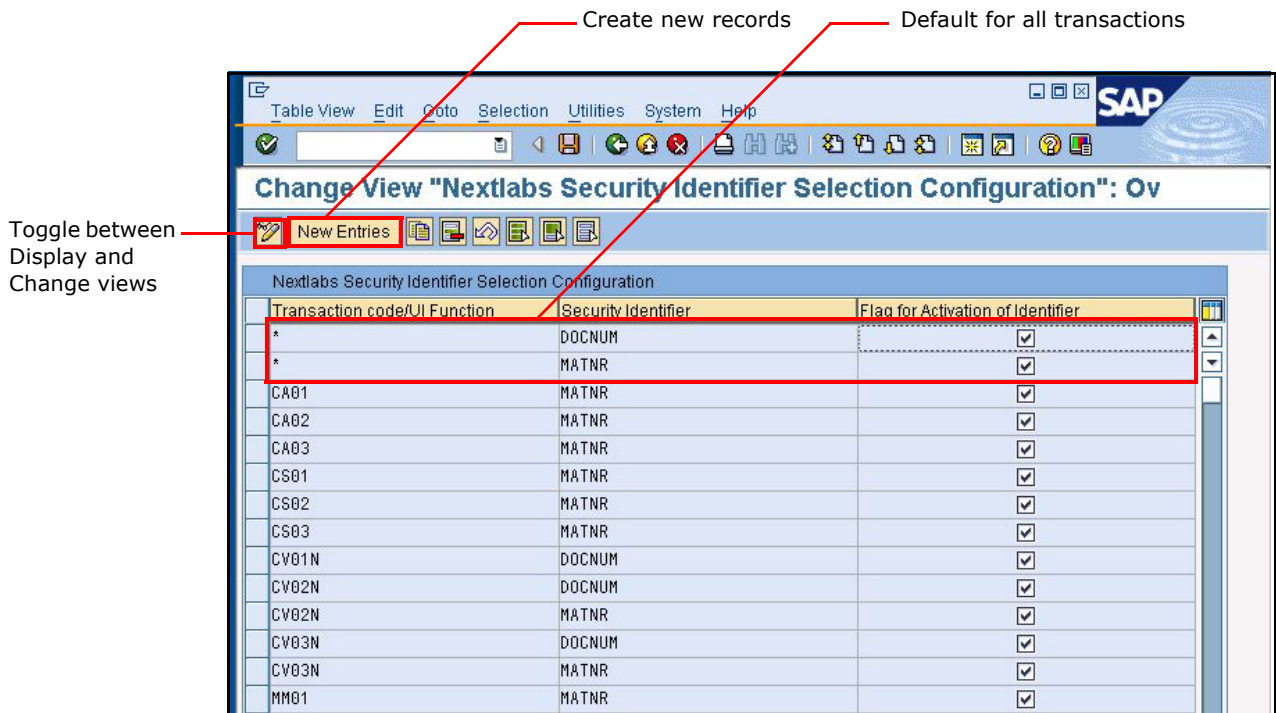


Figure 4-45: Defining the Priority of Security Identifiers

- 3 Toggle the view from Display to Change.
- 4 Click **New Entries**. A blank screen appears where you can enter new activation settings for Transaction code/UI Function and Security Identifier combinations.
- 5 Enter a valid Transaction Code or UI Function by entering the name or by clicking the Search Help icon on the right side of the field.
- 6 Enter a valid Security Identifier by entering the name or by clicking the Search Help icon on the right side of the field.

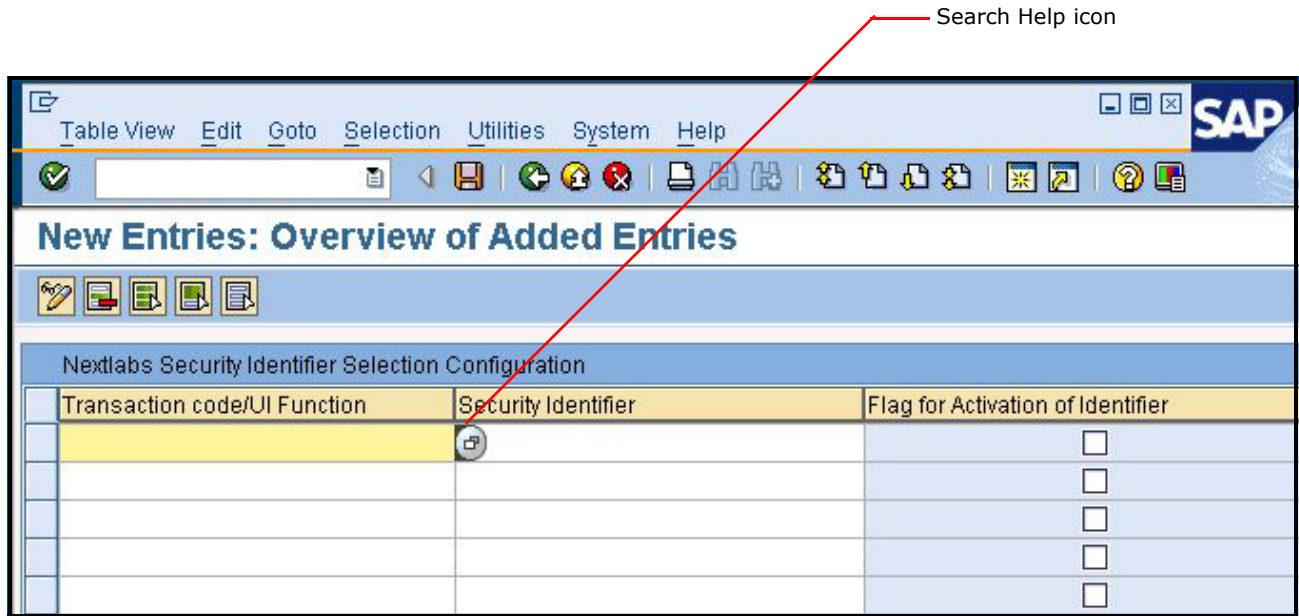


Figure 4-46: Adding a New Activation

**Note:** The Search Help option includes a separate tab for searching SAP ECC transaction codes and PLM UI Functions.

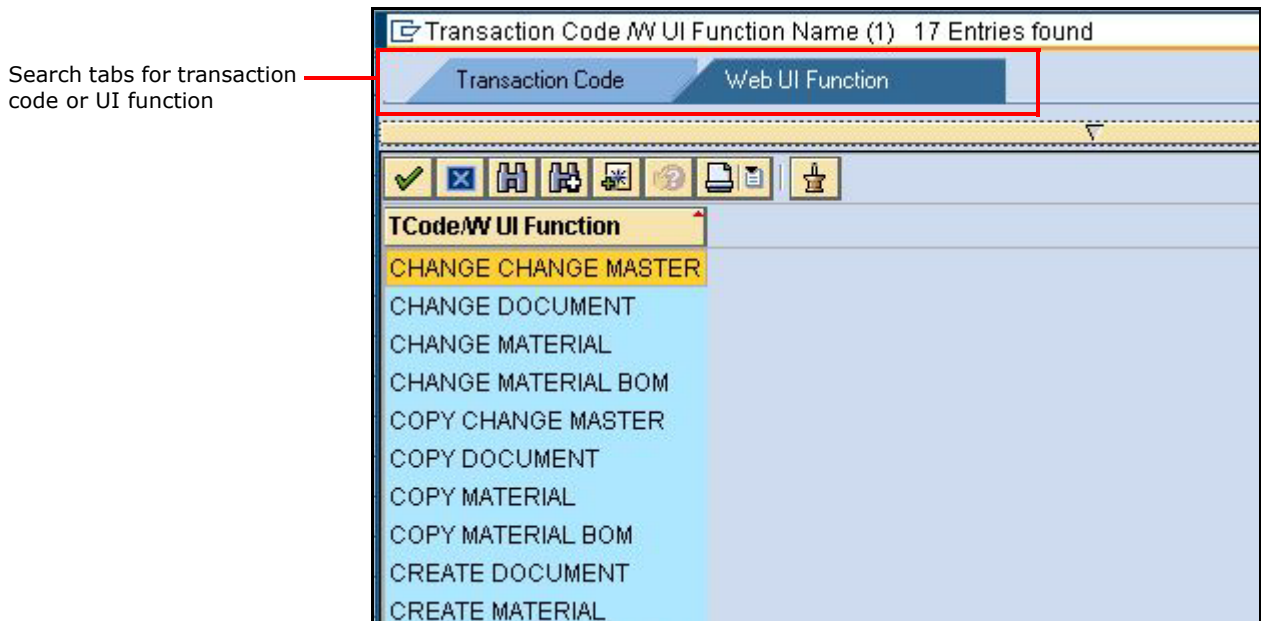


Figure 4-47: Search Help Screen

7 To activate the Security Identifier for the Transaction code, select the **Flag for Activation of Identifier** check box.

- When you finish entering activation settings, click **Save** and return to the *NextLabs Security Classification Selection* screen. Each new activation record you created appears as a new row, as shown in [Figure 4-48](#).

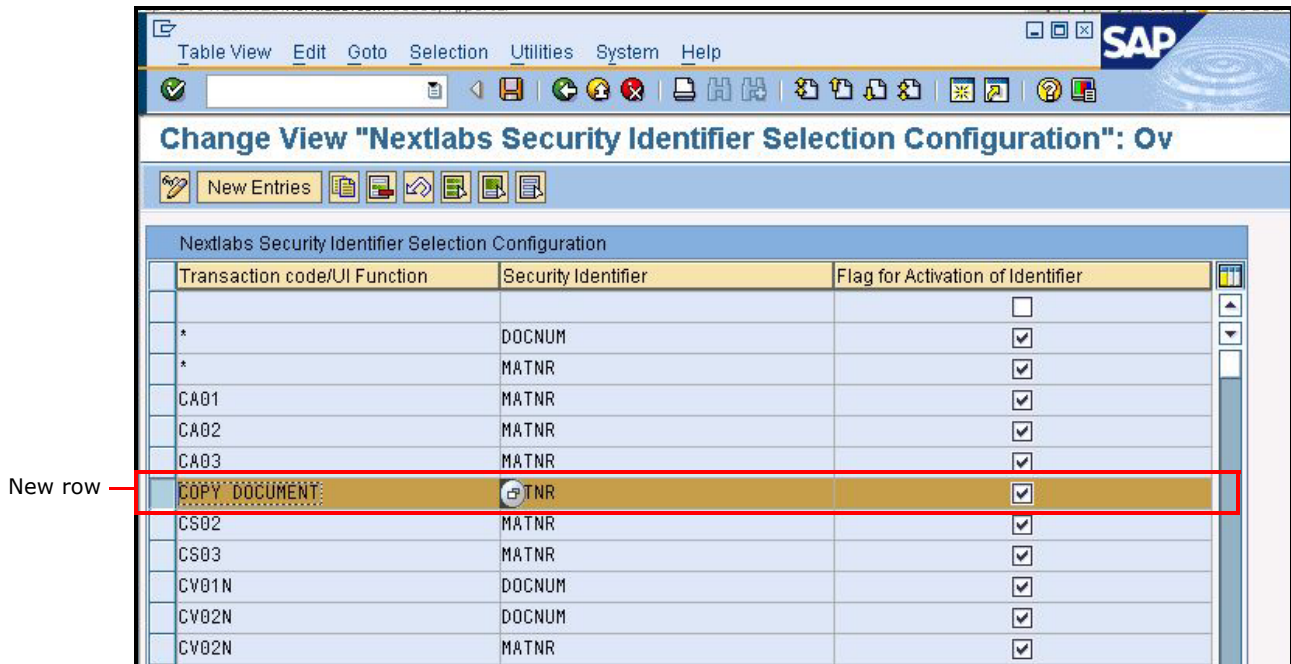


Figure 4-48: New entries in the Security Classification Selection Configuration screen

**Note:** After a row has been added for a Transaction Code/Security Identifier pair, you can change the Activation setting anytime by selecting or deselecting the **Flag for Activation of Identifier** check box.

- When you finish configuring these settings, click **Save**.

After these activation settings are defined, they determine how Security Classifications are prioritized. For example, in the settings defined in [Figure 4-4](#), a new configuration is added so that when a user attempts to create a copy of a document in SAP PLM, the evaluation views the Security Classification of any Materials associated with document when determining whether to grant access.

**Next steps**

If your implementation includes the Entitlement Pack for PLM, The next step is [Configuring Access Control Context Settings \(PLM Only\)](#) on page 121. Otherwise, skip to [Configuring the Transactions or Functions to Intercept](#) on page 123.

## Configuring Access Control Context Settings (PLM Only)

Access Control Context values are passed to Dynamic Authorization Management for SAP ECC and SAP PLM SOA Agent, where they are evaluated against existing policies (see the section [Functional Integration During a Policy Check on page 19](#)). In the *Access Control Context Attributes Maintenance* screen, you can configure how Access Control Context values are passed during a policy check.

### Standard and Compound Access Control Contexts

Access Control Contexts are constructed hierarchically in SAP PLM. Every context setting must have a Parent context. Every context can be either a “Standard” or a “Compound” value. A “Standard” context is the child of only one parent, whereas a “Compound” context can be the child of more than one Parent.

In [Figure 4-49](#), “100\_Mat” and “200\_Mat” are the Standard children of the context “Engineering.” “Project A” is a Compound context that is the child of both “200\_Mat” and “Contractors.” This diagram reflects a common business use for Compound contexts: as temporary or external access granted on a project-by-project basis. For example, the context “Project A” grants Contractors access to a family of materials for the duration of the project.

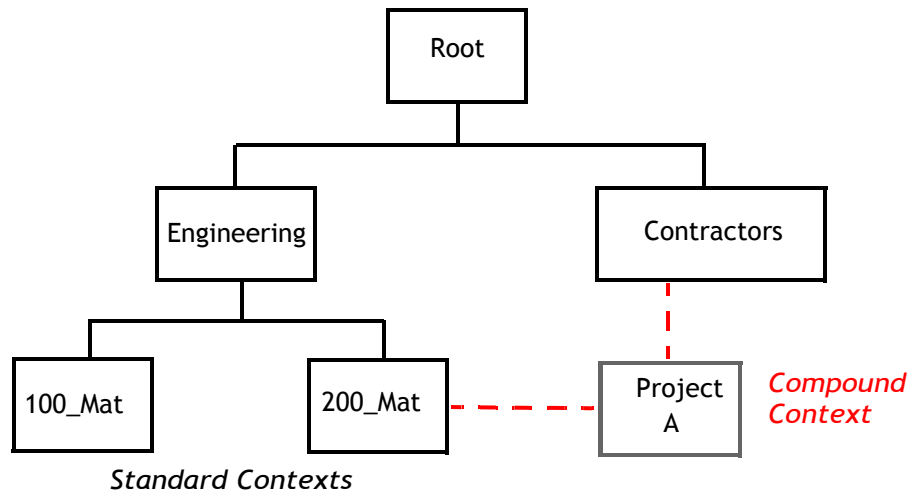


Figure 4-49: Standard and Compound Contexts

### Passing ACCs as Values or Paths

One part of setting up Dynamic Authorization Management is defining whether you want both Standard and Compound contexts to be passed to the SOA Agent for evaluation. You can also configure whether you want a context passed as a value, or as a complete path (which would reference the location of the value in relation to the entire context hierarchy).

This difference is important for policy design. For example, if you passed values only, when the value “Assembly Plans” might be passed, it could be as the child of “Manufacturing” or “Project B.” In contrast, if you configure the system to pass the full ACC hierarchy, you can write policies that target the ACC value “Assembly Plans” only when it is associated with one parent or the other (/Root/Engineering/Project B/Assembly Plans or /Root/Manufacturing/Assembly Plans).

You configure these settings in the *PLM Access Control Context Attributes* screen.

**Procedure**

- 1 In the SAP interface, enter transaction SM30 . The *Table Maintenance View* screen appears.
- 2 In *Table/View*, enter /NEXTLABS/ACCMNT, then click **Display**.

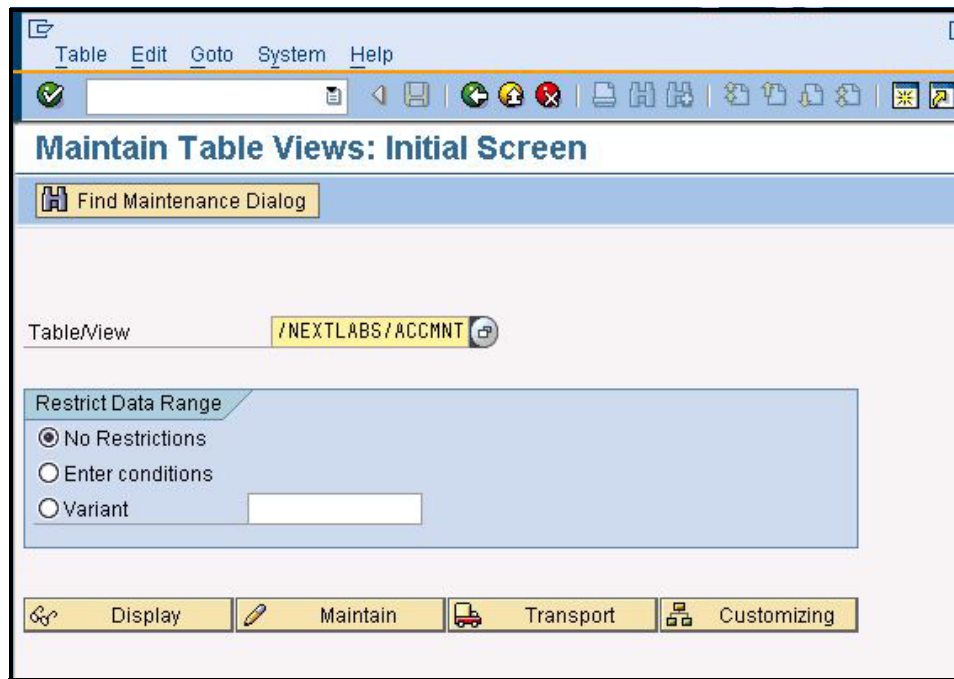


Figure 4-50: Accessing the Access Control Context Attributes Maintenance Screen

- 3 The Access Control Context settings default in Display mode. Click the toggle button to change the view from Display to Change.

Select one, both, or none                      Select one only

NextLabs EM	Owner Context	Compound Context	ACC Policy attributes pass as Hierarchy	ACC Policy attributes pass as value
NextLabs cFolder Addon	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
NextLabs EasyDMS Addon	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
NextLabs ECC Addon	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
NextLabs PLM Addon	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Figure 4-51: Maintain PLM Access Control Context Attributes Screen

- 4 In the NextLabs EM column, select the SAP component you want to configure.
- 5 In the Access Control Context Type column:
  - Select the **Owner Context** check box if you want Standard Context values to be passed to the SOA Agent.
  - Select the **Compound Context** check box if you want Compound Context values to be passed to the SOA Agent.

**Note:** It is recommended that you enable both Owner and Compound context to be able to reference all the Owner and Compound context values you have defined for policy evaluation.
- 6 In the ACC Policy attributes Pass as... areas:
  - Select the **pass as Value** check box if you want ACC attributes passed as values.
  - Select the **pass as Hierarchy** check box if you want the entire path passed.
- 7 Click **Save**.

### Next steps

The next step is [Configuring the NextLabs Number Range](#) on page 104.

## Configuring the Transactions or Functions to Intercept

For the ECC, BW, and S/4HANA Entitlement Packs, you must specify explicitly the transactions and functions that you want Dynamic Authorization Management to intercept for policy checks.



For a list of the ECC transactions supported for interception, see [Implementation Reference for ECC](#) on page 277. For a list of the BW functions supported for interception, see [Implementation Reference for SAP BW](#) on page 305. For a list of the S/4HANA functions supported for interception, see [Implementation Reference for SAP S/4HANA](#) on page 311.

The transactions to intercept must be configured through the /NEXTLABS/TXFLTR transaction and also in the /NEXTLABS/OPTCFG table. Transactions specified through /NEXTLABS/TXFLTR are either enabled (Active status) or disabled for policy checks. Transactions specified in the OPTCFG table are assigned to specific security identifiers; when activated, policy checks are enabled for the transaction and its associated security identifier. For information about this configuration, see [Defining How Multiple Security Classifications Should Be Applied](#) on page 116.

### Procedure

- 1 In the SAP interface, enter transaction /NEXTLABS/TXFLTR. The *Maintain Executable Objects for Common Interception* screen appears, as shown in [Figure 4-52](#).

*Figure 4-52: Maintain Executable Objects for Common Interception Screen*

- 2 To specify new transactions to intercept, click **Create**. To view or modify transactions that are already in the table, click **Modify/Display**. In **Existing Objects Selection**, you can specify criteria for filtering the list of transactions.
- 3 If you clicked **Create**, add new transactions. For each transaction to add, do the following:
  - Click **Insert Row**.
  - In **Executable Object**, enter a transaction code.
  - In **Active**, enter **x** to enable interception of the transaction. Use the Active setting to disable or enable interception for each transaction as your authorization requirements change.

[Figure 4-53](#) shows an example of one transaction and its status set to Active.

As an alternative to entering each transaction code here, you can click **Import** to import the values from an Excel file, if you choose to type the codes in Excel.



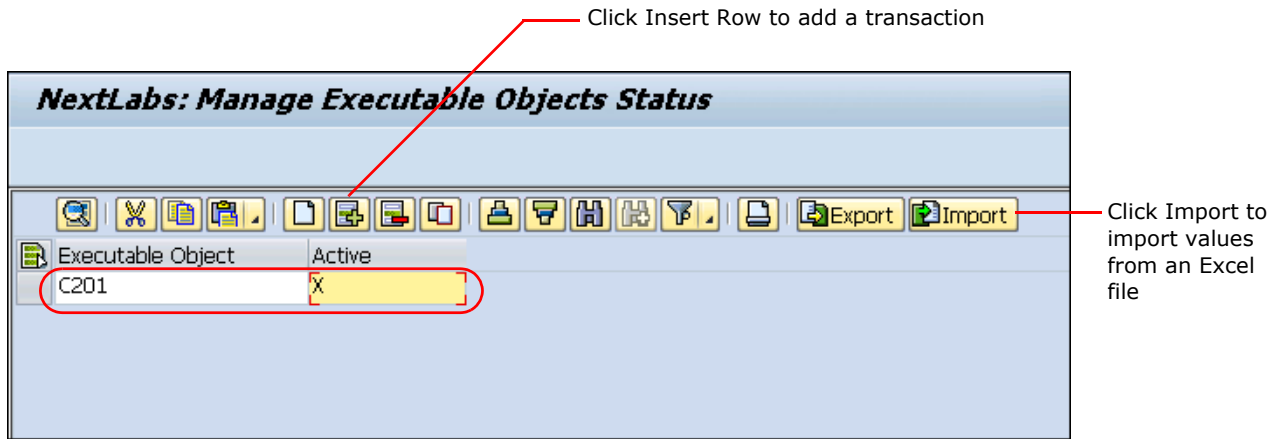


Figure 4-53: Adding a transaction to the /NEXTLABS/TXFLTR table

- 4 Click **Save** when you finish entering all the transactions. If you want to save all the values to an Excel file, click **Export**.

**Next steps**

The next step is [Configuration for Policy-based Security Classification](#) on page 125.

## Configuration for Policy-based Security Classification

Classifications can be defined manually, by entering rows in the Security Classification Maintenance table, or automatically based on policy. Policy Based Security Classification (PBSC) can run as a batch process to automatically classify data in SAP ECC and SAP cFolders. In SAP ECC, documents can be added to a PBSC queue upon the creation or editing of originals on a parent business object. In SAP cFolders, documents can be added to the PBSC queue when they become Exported to cFolders, or are created and/or modified there.

Documents in the queue are evaluated based on policy when the batch process runs. Referencing defined policies, the batch process determines whether or not new classification records (rows) should be added to the Security Classification Maintenance table for the parent business objects (in the case of SAP ECC) or for the documents themselves (in the case of SAP cFolders).

**Note:** For an example of how these policies are defined, see [Example Policy: Security Classification Based on Content Analysis](#) on page 200.

Configuring Policy Based Security Classification includes the following procedures:

- [Designating Which Create/Edit Functions Should Be Included in the PBSC Queue](#) on page 126
- [Defining the PBSC Filter for Documents](#) on page 127
- [Defining the PBSC Filter for Materials](#) on page 128

- [Configuring the PBSC Conversion Directory](#) on page 129
- [Defining the Background Job for PBSC](#) on page 131
- [Configuring the PBSC Custom Obligation](#) on page 136

## Designating Which Create/Edit Functions Should Be Included in the PBSC Queue

Documents and materials intended for the PBSC queue can be filtered based on UI function or data. This section describes how to configure UI functions for Documents and Materials being sent to the queue. This must be done for all Entitlement Packs where you are deploying PBSC policies.

To designate which create and edit functions should trigger a PBSC queuing of materials and documents, configure these functions in the `/NEXTLABS/PBSCFG` table. Before this step can be performed, transaction codes and UI functions must have already been mapped in the `/NEXTLABS/UIFUNC` table.

### Procedure

- 1 In the SAP interface, enter transaction `SM30`. The *Table Maintenance View* screen appears.
  - 2 In **Table/View**, enter `/NEXTLABS/PBSCFG`, then click **Display**.
  - 3 Click the toggle button to change the view from Display to Change.
  - 4 Click **New Entries**. For all Entitlement Packs included in your implementation where you plan to deploy PBSC policies, enter the following information (see [Table 4-6](#)):
    - **UI function:** transaction or function used to create and edit materials and documents
    - **Field name:** the field name for Documents and Material records (`DOCNUM` or `MATNR` for SAP ECC, SAP PLM, and EASYDMS, `DOC_ID` or `MAT_ID` for cFolders)
    - **UI Mode:** select `Insertion` for the create action and `Change` for the edit action.
    - **Indicator:** activate the UI function for PBSC
    - **Text:** enter a description. This description appears in the log for PBSC.
- Note:** [Table 4-6](#) provides a sample of valid entries. Your entries will vary, depending on which Entitlement Packs you are implementing and how you have mapped policies in the `UIFUNC` table.

*Table 4-6: Example entries in the PBSCFG table*

UI Function	Field Name	UI Mode	Text
/PLMi/EASYDMS_DOCUMENT_CREATE2	DOCNUM	Insertion	EASYDMS Create Document
BAPI_DOCUMENT_CREATE2	DOCNUM	Insertion	EASYDMS Create Document
BAPI_DOCUMENT_CREATEFROMSRC2	DOCNUM	Insertion	EASYDMS Create Document
CVAPI_DOC_COPY_REF_FILES	DOCNUM	Insertion	EASYDMS Copy/Paste
EASYDMS_DOCS_CREATE_NEWVERSION	DOCNUM	Insertion	EASYDMS New Version
EASYDMS_DOCUMENT_CHANGE2	DOCNUM	Change	EASYDMS Change Document
EASYDMS_MASS_DELETE_DOCUMENTS	DOCNUM	Change	EASYDMS Delete Document

Table 4-6: Example entries in the PBSCFG table (Continued)

UI Function	Field Name	UI Mode	Text
EASYDMS_RENAME	DOCNUM	Change	EASYDMS Rename Document
/PLMU/WDA_DIR_OIF	DOCNUM	Change	PLM Modify Document
/PLMU/WDA_DIR_OIF	DOCNUM	Insertion	PLM Create Document
/PLMU/WDA_MAT_OIF	MATNR	Change	PLM Modify Material
/PLMU/WDA_MAT_OIF	MATNR	Insertion	PLM Create Material
CFI01	DOCNUM	Insertion	cFolders Import Document
CFI01	MATNR	Insertion	cFolders Import Material
CFI02	DOCNUM	Insertion	cFolders Import Document
CFI02	MATNR	Insertion	cFolders Import Material
CHANGE BOM	MAT_ID	Change	cFolders Change BOM
CHANGE DOCUMENT	DOC_ID	Change	cFolders Change Document
CHANGE MATERIAL	MAT_ID	Change	cFolders Change Material
CREATE BOM	MAT_ID	Insertion	cFolders Create BOM
CREATE DOCUMENT	DOC_ID	Insertion	cFolders Create Document
CREATE MATERIAL	MAT_ID	Insertion	cFolders Create Material
CV01N	DOCNUM	Insertion	SAP ECC Create Document
CV02N	DOCNUM	Change	SAP ECC Modify Document
MM01	MATNR	Insertion	SAP ECC Create Material
MM02	MATNR	Change	SAP ECC Modify Material

5 Click **Save** to save your configuration changes.

## Defining the PBSC Filter for Documents

Documents and materials intended for the PBSC queue can be filtered based on data characteristics. This section describes how to configure which Documents to send to the queue. You do this in the *PBSC filter for Documents* screen. If you do not set a filter, or if you apply a wildcard (\*), originals associated with all Documents (for SAP ECC), and all kinds of documents (for SAP cFolders) are sent to the queue for evaluation. In many cases, organizations may wish to limit which business objects are being queued, because only certain objects are considered sensitive or controlled, and/or to avoid an impact on system performance.

### Procedure

- 1 In the SAP interface, enter transaction **SM30**. The *Table Maintenance View* screen appears.
- 2 In **Table/View**, enter **/NEXTLABS/PBSDIR**, then click **Display**.
- 3 Change the view from **Display** to **Edit**.

- 4 Click **New Entries**. Enter values to filter what originals are sent to the queue. Notice that, for SAP ECC, you can also filter objects by Compound Key. In [Figure 4-54](#), the filter is set so that only documents of type DRW are added to the PBSC queue.

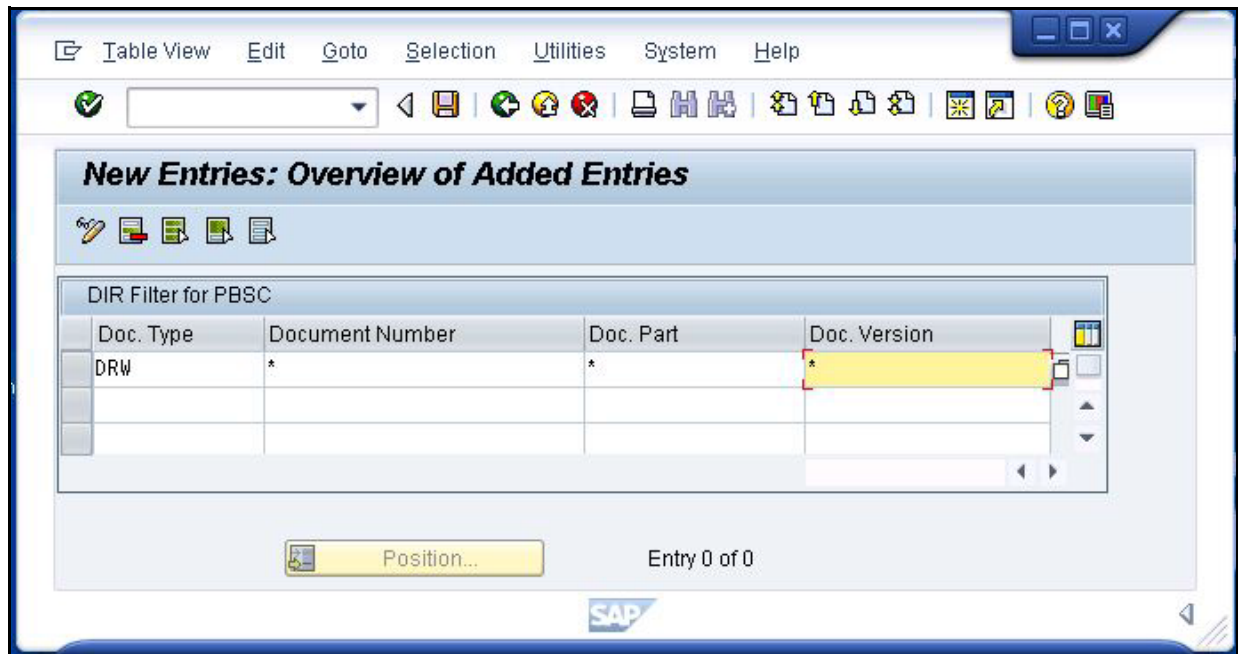


Figure 4-54: Filtering by Doc. Type DRW

- 5 When you have entered all the filter criteria, click **Save**.

## Defining the PBSC Filter for Materials

Documents and materials intended for the PBSC queue can be filtered based on data characteristics. This section describes how to configure which Materials are sent to the queue. You do this in the *PBSC Filter for Materials* screen. If you do not set a filter, or if you apply a wildcard (\*), originals associated with all Materials are sent to the queue for evaluation. In many cases, organizations may wish to limit which business objects are being queued, because only certain objects are considered sensitive or controlled, and/or to avoid an impact on system performance.

### Procedure

- 1 In the SAP interface, enter transaction `SM30`. The *Table Maintenance View* screen appears.
- 2 In **Table/View**, enter `/NEXTLABS/PBSMAT`, then click **Display**.
- 3 Change the view from **Display** to **Edit**.
- 4 Click **New Entries**. Enter values to filter what materials should be sent to the queue. In [Figure 4-55](#), a wildcard is used so that all materials are added to the PBSC queue.

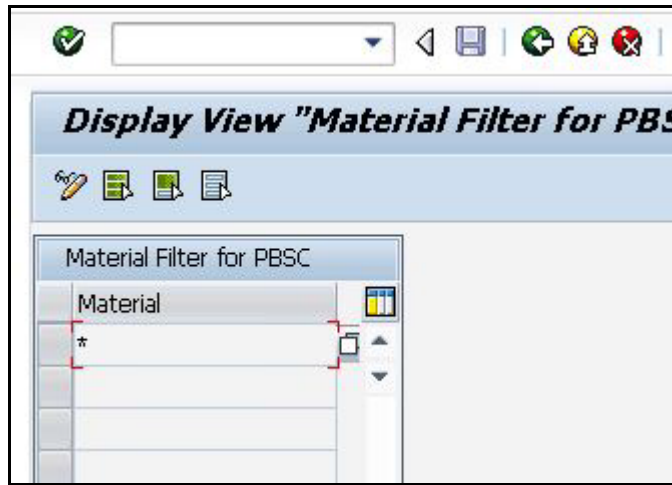


Figure 4-55: Setting Material Filters

- 5 When you have entered all the filter criteria, click **Save** to save the record.

## Configuring the PBSC Conversion Directory

Policy Based Security Classification (PBSC) requires a location to temporarily store files while batch processes are being performed. This location must be on the host where the Policy Controller is installed. A logical path pointing to this location must also be configured within SAP.

**Note:** One of the landscape requirements for your implementation is that the SAP user under which the PBSC process runs has access to write and read files within the conversion directory that is located on the Policy Controller server.

### Procedure

- 1 On the same host as the Policy Controller, create a folder in the `c:` drive to serve as the conversion directory (in our example, this is simply `c:\Conversion`). This is the root folder, and may include additional subfolders.
- 2 In the SAP interface, enter transaction `/nFile`. The *Logical File Path Definition* screen appears.
- 3 In the list of logical paths, locate `/NEXTLABS/PBSC_DOWNLOAD`. Select the path and click **Assignment of Physical Path to Logical Path**.

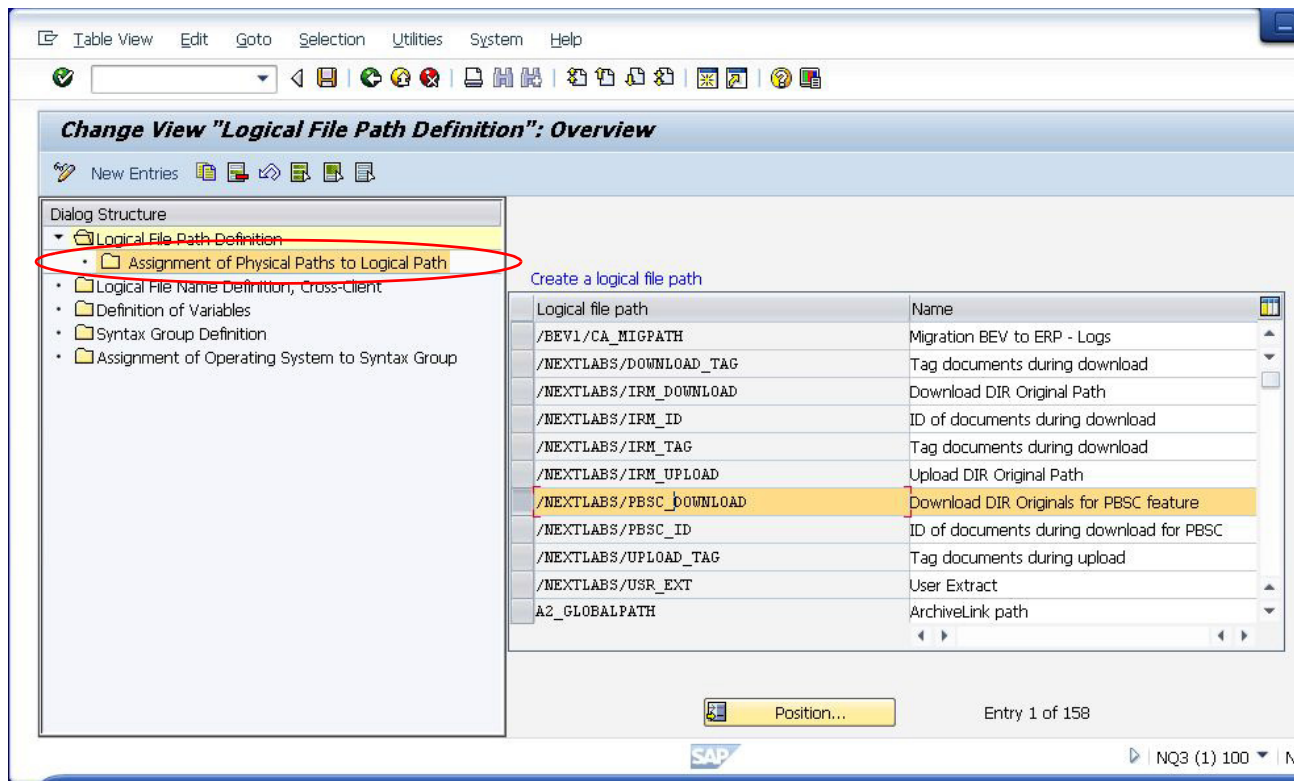


Figure 4-56: Assignment of Physical Path to Logical Path

- 4 In the Physical path field, specify the path to the conversion directory, using the following format:

```
\\<hostname>\Conversion\<PARAM_2>\IN\<FILENAME>
```

where <hostname> implicitly references the host's C:\ drive. In other words, the path above points to C:\Conversion. If additional folders appear under the root conversion folder, you must also include them in the path. <PARAM\_2>\IN\<FILENAME> must be written exactly as is.

- 5 In the list of logical paths, locate /NEXTLABS/DOWNLOAD\_TAG. Select the path and click **Assignment of Physical Path to Logical Path**.
- 6 In the Physical path field, change the <host name> to point to the location of the conversion directory. <hostname> implicitly references the host's C:\ drive. In other words, the path above points to C:\Conversion. If there are additional folders under the root conversion folder, you must also include them in the path.
- 7 In the list of logical paths, locate /NEXTLABS/PBSC\_ID. Select the path and click **Assignment of Physical Path to Logical Path**.
- 8 In the Physical path field, change the <host name> to point to the location of the conversion directory. <hostname> implicitly references the host's C:\ drive. In other words, the

path above points to `C:\Conversion`. If there are additional folders under the root conversion folder, you must also include them in the path.

- 9 Save the change, or click **Logical Path Definition** to return to the list of Logical paths and configure another one.

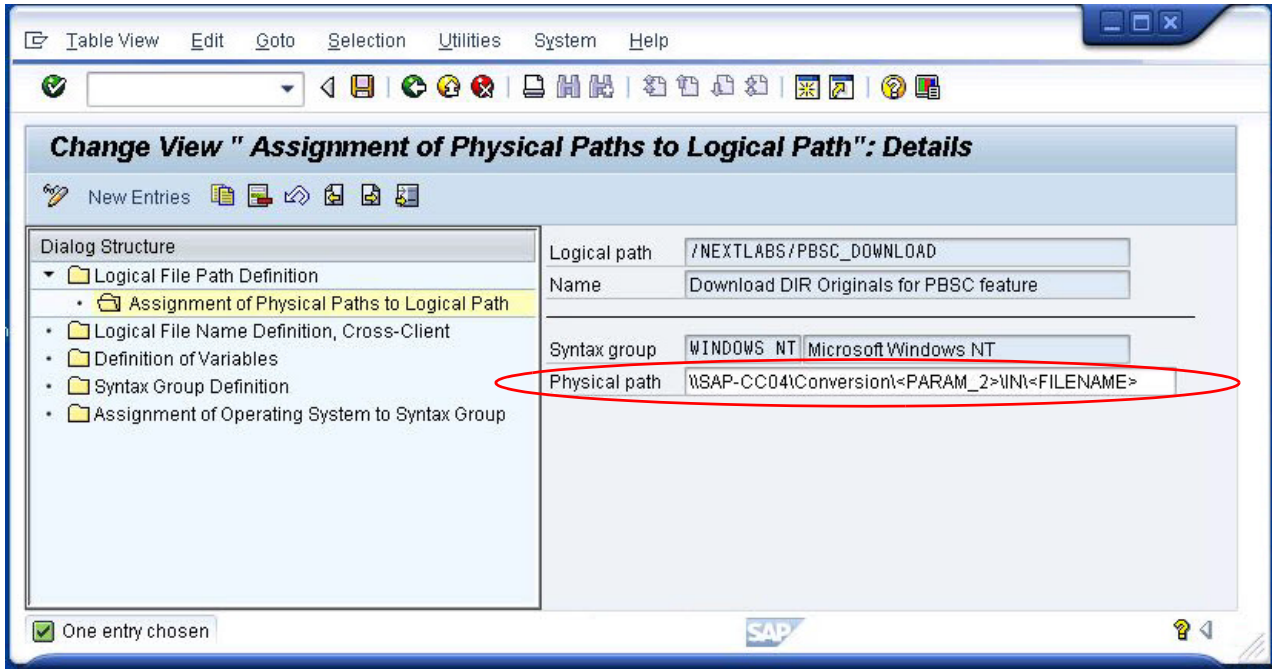


Figure 4-57: Configuring the Logical Path of the Conversion Directory

## Defining the Background Job for PBSC

PBSC runs as a batch process that must be defined as a background job in SAP. This ABAP program to be configured in the job is `/NEXTLABS/PBSC`. The program is the same for SAP ECC and SAP cFolders. The background job should be scheduled so that PBSC updates occur at a frequency appropriate for addressing your business and authorization requirements.

### Procedure

- 1 In the SAP interface, enter transaction `SM36`. The *Define Background Job* screen appears.
- 2 Click **Job Wizard** to launch the job wizard.



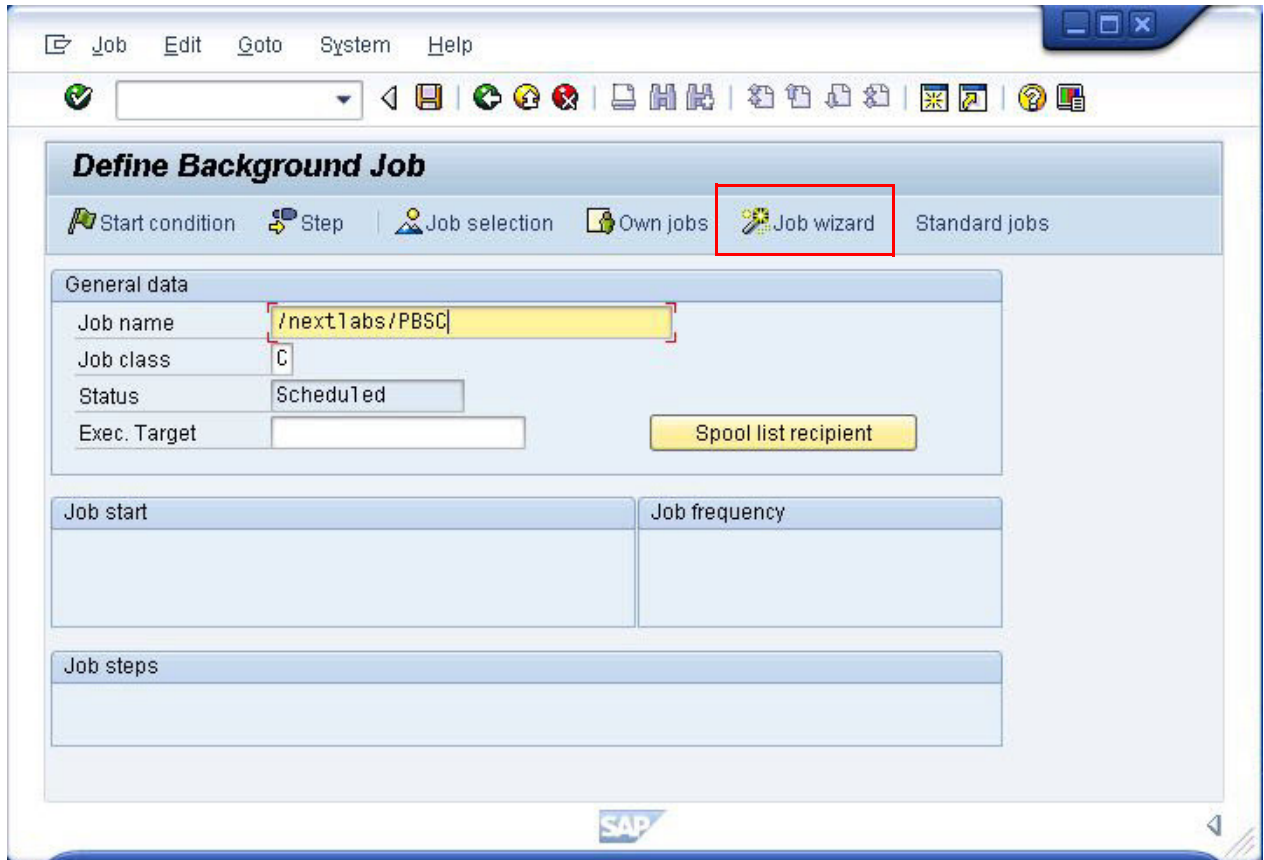


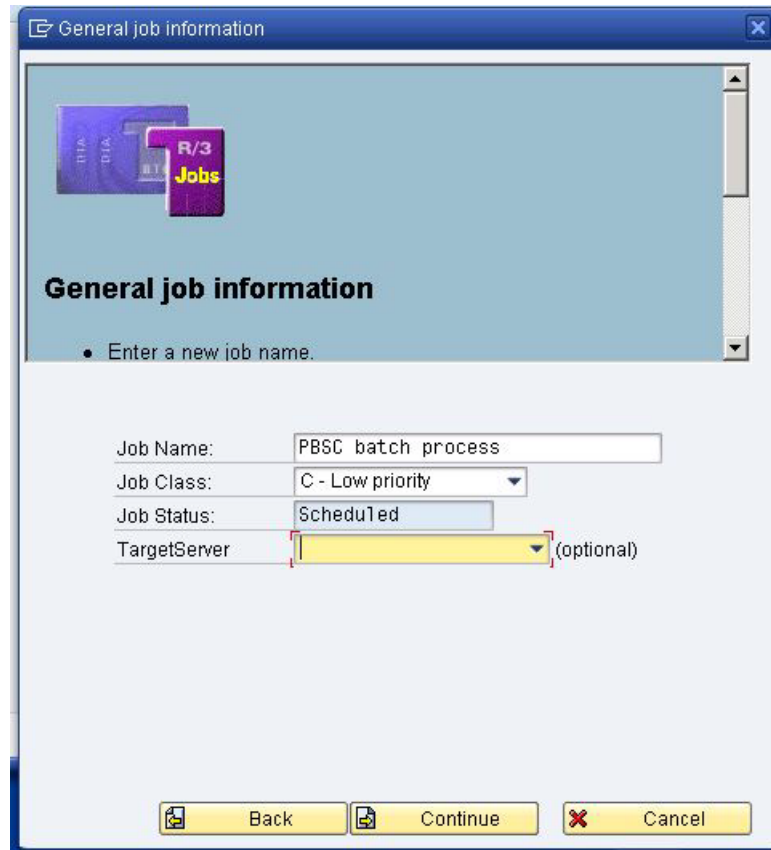
Figure 4-58: Job Wizard

3 Enter background information about the job:

- The Job Name can be anything you specify.
- Select the priority (Job Class) for the batch process to reflect your business requirements. Because this is a frequently running process, we select *Low Priority*. However, if you want the job to take priority over other batch processes, because of the high risk of exposing sensitive data, select a higher priority for the job.
- TargetServer is not required.

4 Click **Continue**.





*Figure 4-59: Defining a Background Job for PBSC*

5 In ABAP program name, enter `/NEXTLABS/PBSC`, and click **Continue**.

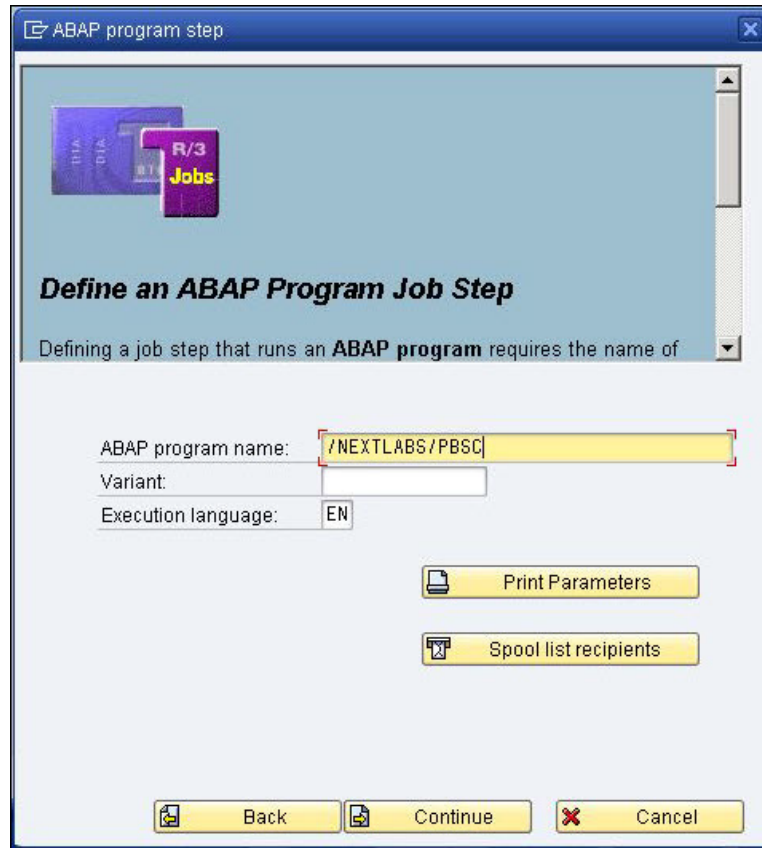
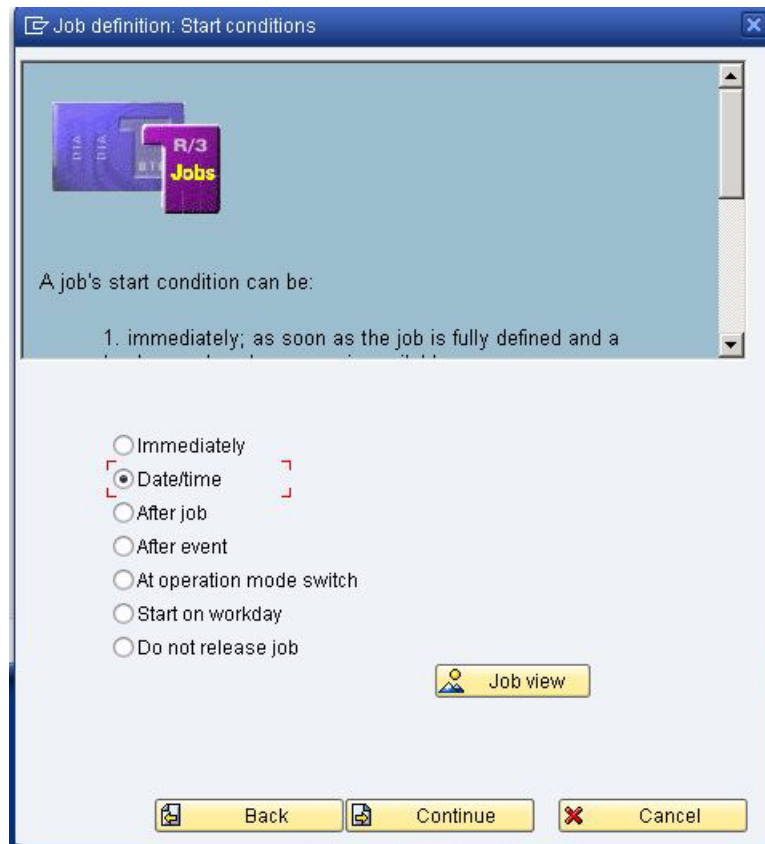


Figure 4-60: /NEXTLABS/PBSC

- 6 In the *Multi-step job* screen, do not click **Add additional steps**, and click **Continue**.
- 7 In the scheduling screen, select the frequency for running the batch process, based on your business requirements, then click **Continue**.



*Figure 4-61: Scheduling the Job*

A message appears stating that the job has been defined successfully.

**8 Click Complete.**

**9** You can use the transaction `SM37` to verify the job has been created and scheduled.

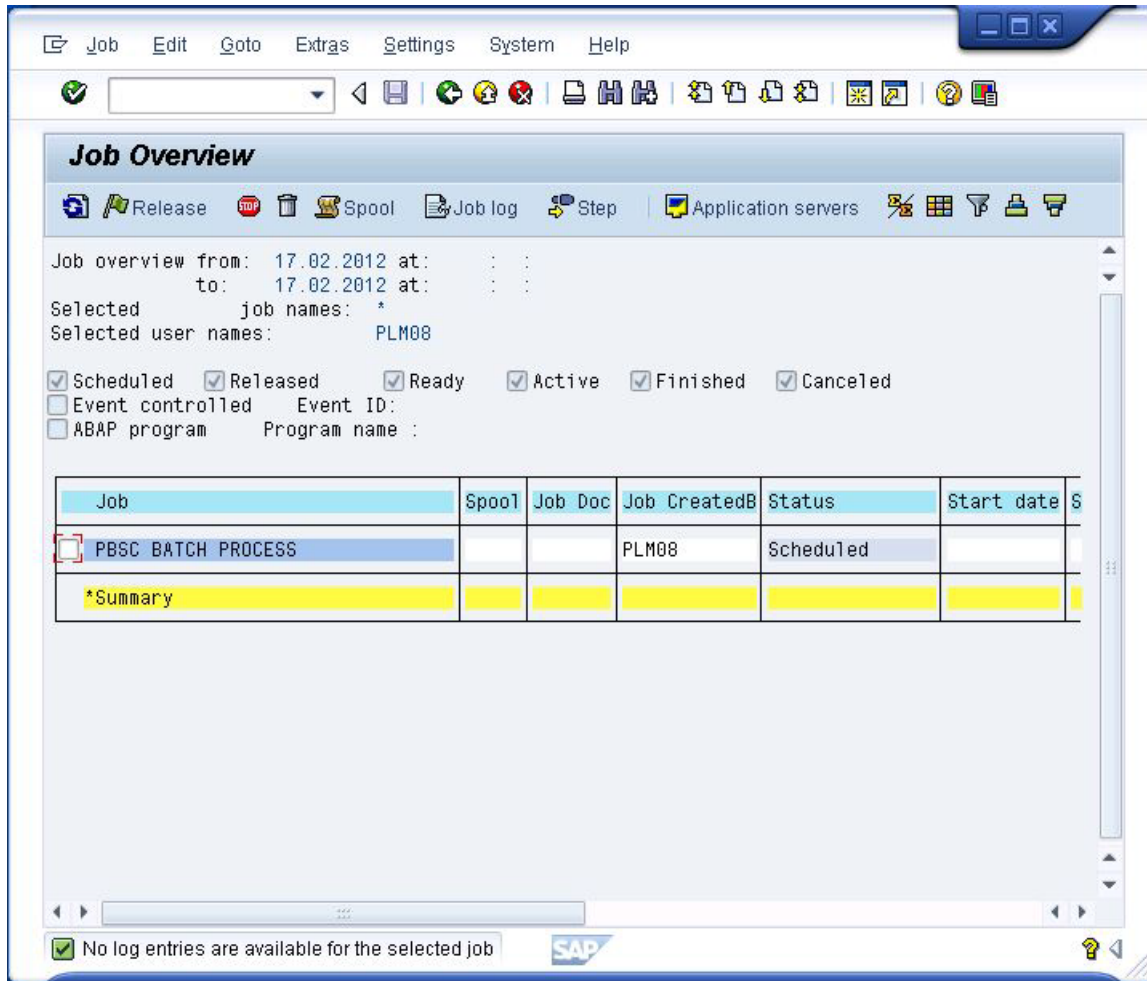


Figure 4-62: Verifying the Scheduled Job

### Configuring the PBSC Custom Obligation

PBSC policies require the configuration of the PBSC custom obligation. This procedure is discussed as part of configuring NextLabs Control Center. The same custom obligation is used for SAP ECC and SAP cFolders. See [Configuring SAP Obligations](#) on page 70.

### Next steps

The next step is [Configuring Enhancement Implementations](#) on page 137.

---

## Configuring Enhancement Implementations

The last configuration step is to configure enhancement implementations for all Entitlement Packs in your solution. The enhancements implement the interception points for policy checks for transactions and business objects. The Entitlement Pack for ECC alone supports the interception of over 300 transactions and policy checks for more than a dozen business objects, including material, document, vendor, and customer.

Enhancement implementations require a developer license and knowledge of your SAP environment. The latter is important because the instructions provided in this guide cannot anticipate or address certain variables in your landscape that might impact how you should configure enhancements. For instance, one unknown variable is what enhancement implementations are already in use in your system, and how NextLabs code should be integrated with existing implementations.

For information about the transactions and functions supported for interception, and instructions for configuring enhancement implementations, see the Implementation Reference for the Entitlement Packs installed on your system.

- [Implementation Reference for ECC](#) on page 277
- [Implementation Reference for EasyDMS](#) on page 301
- [Implementation Reference for SAP PLM](#) on page 303
- [Implementation Reference for SAP BW](#) on page 305
- [Implementation Reference for PBSC](#) on page 309
- [Implementation Reference for DFPS](#) on page 311
- [Implementation Reference for SAP S/4HANA](#) on page 311



# 5 Configuring Optional Features

---

This section describes the optional configuration procedures for the Entitlement Manager for SAP. Your configuration is specific to your product implementation, classification scheme, and access-control goals.

Topics:

- [Configuring Policy Checks for Value \(F4\) Help](#)
- [Configuring Policy-based Data Segregation](#)
- [Configuration for Integrated Rights Management \(IRM\)](#)
- [Configuring the Read Tags Feature](#)
- [Implementation Reference for Read Tags](#)
- [Reading Tags](#)
- [Designing Read Tags Policies](#)

---

## Configuring Policy Checks for Value (F4) Help

Practically all the enhancement implementations you configure for the Entitlement Pack for ECC are to intercept transactions and trigger policy checks for business objects, such as Materials, Customers, Vendors, and Warehouses. For example, if you configured Warehouse Numbers for policy checks, then a policy check is triggered whenever a user enters a warehouse number and executes the transaction. The user is then either denied or permitted access to that information.

However, rather than typing a value for a particular business object, users can press F4 for help providing the value. You can filter the list of values that is displayed to the user to include only the values that she is permitted to access. [Figure 5-1](#) shows an example of when a policy check for F4 help is triggered. In this example, the user sees only the warehouse numbers to which she is granted access.

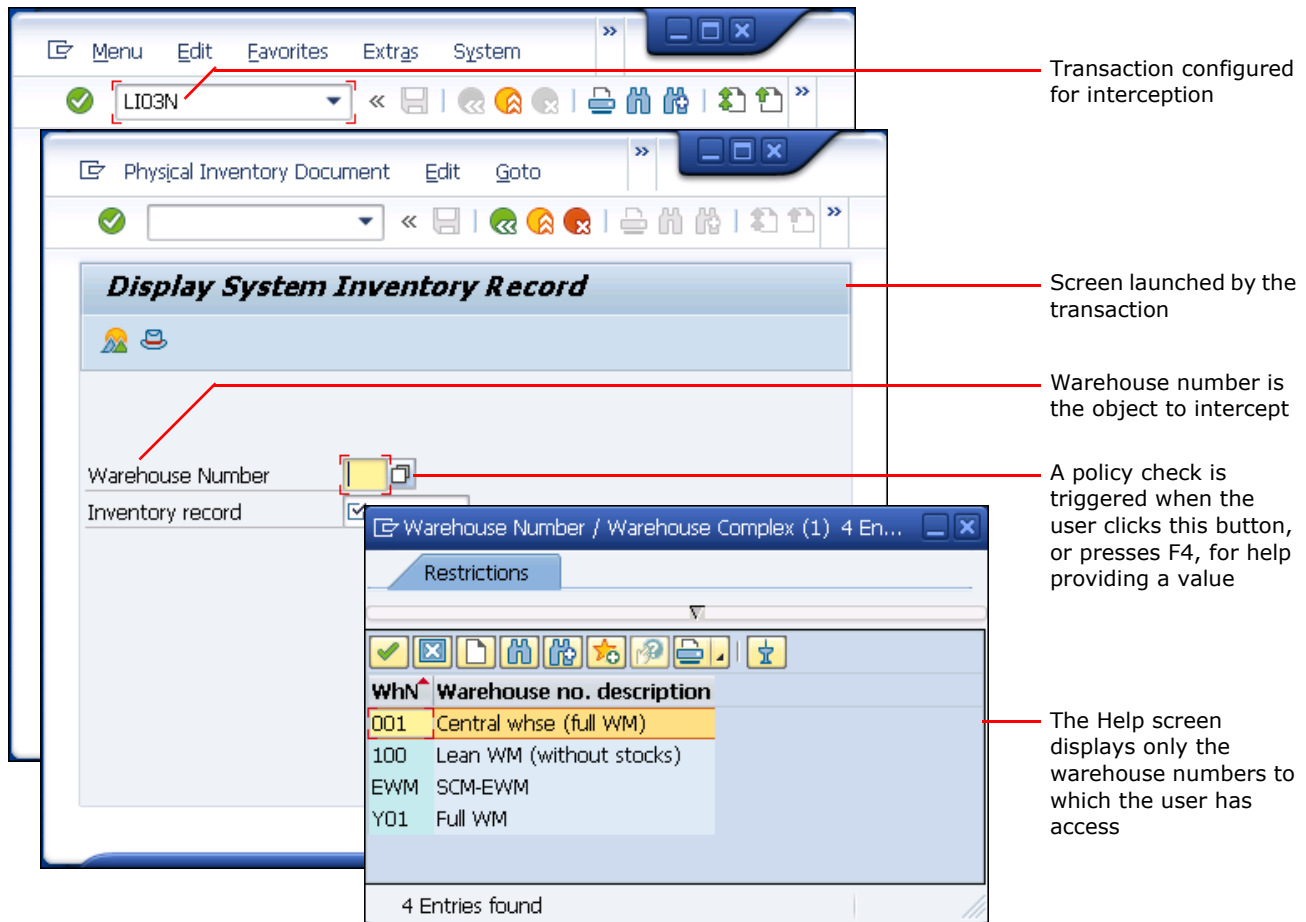


Figure 5-1: Example of policy check for Value (F4) help

Filtering the list of values that is displayed to each user provides a good user experience. If the values are not filtered, then users can select a value that they are not authorized to use, continue filling the rest of the form, and execute the transaction, only to then discover that they are denied access to that information.

## General Steps

To configure policy checks for F4 help, you define two enhancement implementations. These implementations combined with the business object-specific implementations and transactions that you configure for common interception determine which objects, and for what transactions, the F4 help policy checks apply to.

For example, to enable policy checks for F4 help for Materials, you must complete these configuration procedures:

- Configure the security identifier for Materials in the EPVAL table. For more information, see [Configuring Security Identifier/Composite Key Value Tables \(EPVAL\)](#) on page 86.



- Configure the security identifier for Materials in the OPTCFG table. For more information, see [Defining How Multiple Security Classifications Should Be Applied](#) on page 116.
- Define the enhancement implementation for Materials. For more information, see [Implementation Details](#) on page 294 > [MATNR\\_CI](#) on page 296.
- Specify the transactions to intercept in TXFLTR. For more information, see [Configuring the Transactions or Functions to Intercept](#) on page 123.
- Define the enhancement implementations for F4 Help. For details, see [Implementation Details](#) on page 294 > [NXL\\_F4\\_ALLIDT](#) on page 297.

---

## Configuring Policy-based Data Segregation

It is typical for companies to place restrictions on where data is stored based on regulations, programs, contracts, and so on. Common requirements include, for example, restricting ITAR data to servers in the US, or segregating classified data and unclassified data into different servers. Using Dynamic Authorization Management for SAP, you can control what data is stored in which DMS content server. You can do the following:

- Restrict the locations into which a user can check in different types of data. This ensures, for example, that classified data is properly stored and managed in secure locations.
- Restrict the locations into which a user can check out different types of data. This prevents, for example, classified data from being moved to unauthorized locations.

Dynamic Authorization Management for SAP provides two ways to implement those controls. Choose one of the following techniques:

- Write access control policies that specify what data can be checked into which specific content servers, and what data can be checked out into which specific locations. When a user attempts to check in or check out data to a particular location, the policy either allows or denies the action.
- Write filter policies that provide the user with a filtered list of approved content servers when the user checks in or checks out a particular class of data.

For examples of these types of policies, see [Designing Data Segregation Policies](#) on page 231.

This section describes the steps for configuring policy-based data segregation, which you must perform before you can write data segregation policies. The following configuration steps can be performed in any order.

- [Defining the BADI Implementation](#) on page 141
- [Configuring the Data Segregation Obligations](#) on page 143
- [Configuring the Check-in and Check-out Actions](#) on page 144

### Defining the BADI Implementation

Depending on your system (ECC or EasyDMS) and whether you are writing access control policies or filter policies, perform one of the following procedures.

## BADI Implementation for Filter Policies (ECC)

### Procedure

- 1 In the SAP interface, enter transaction `SE19`. The *BADI Builder* appears.
- 2 In **Create Implementation**, select a BADI type, and enter `DOCUMENT_STORAGE01`. Click **Create Impl.**
- 3 Enter an implementation name. Click **OK**.
- 4 In **Implementation Short Text**, enter a description.
- 5 Select the **Interface** tab.
- 6 Double-click the `BEFORE_LIST_STORAGECAT` method, and enter the following code:

```
INCLUDE /nxlecc/trans_process_doc_lsct.
```

- 7 Save and activate the method.
- 8 Double-click the `BEFORE_CHECKIN` method, and enter the following code:

```
IF NOT source_file IS INITIAL.  
    INCLUDE /nxlecc/trans_process_doc_bfck.  
ENDIF.
```

- 9 Save and activate the method.
- 10 Save and activate the implementation.

## BADI Implementation for Filter Policies (EasyDMS)

### Procedure

- 1 In the SAP interface, enter transaction `SE19`. The *BADI Builder* appears.
- 2 Create or edit the implementation for `DOCUMENT_MAIN01`.
- 3 Open the BADI Implementation Class.
- 4 In the `BEFORE_SAVE` method, enter the following code:

```
INCLUDE /NXLEDMS/TRANS_PROCESS_DOC_BFS.
```

- 5 Save and activate the method.
- 6 Save and activate the implementation.

**Note:** All `DOCUMENT_MAIN01` enhancements must be configured in the same `Z_DOCUMENT_MAIN01`.

## BADI Implementation for Access Control Policies (ECC)

### Procedure

- 1 In the SAP interface, enter transaction SE19. The *BADI Builder* appears.
- 2 Create or edit the implementation for DOCUMENT\_STORAGE01.
- 3 In the BEFORE\_PHYSICAL\_CHECKIN method, enter the following code:

```
INCLUDE /nextlabs/doc_enhn_check_tag.
INCLUDE /nxlecc/trans_process_doc_atck.
```

- 4 Save and activate the method.
- 5 In the BEFORE\_PHYSICAL\_CHECKOUT method, enter the following code:

```
INCLUDE /nextlabs/doc_enhn_check_tag.
INCLUDE /nxlecc/trans_process_doc_bfco.
```

- 6 Save and activate the method.
- 7 Save and activate the implementation.

## Configuring the Data Segregation Obligations

To write policies that provide users with a list of approved locations at check-in or check-out time, the following obligations must be added to Policy Studio:

- Data Segregation – Whitelist
- Data Segregation – Blacklist

You use these obligations in policies to specify which storage locations data can or cannot be stored in. The Data Segregation – Whitelist obligation specifies the approved storage locations, and the Data Segregation – Blacklist obligation specifies the unapproved storage locations. When writing a policy, you specify either the whitelisted storage locations, or the blacklisted storage locations, but not both. At check-in or check-out time, the user selects from the list of locations.

You add the definitions of these obligations, shown below, to the Policy Server's configuration file, `configuration.xml`. You must replace the storage category values (highlighted in bold) in the definition with the storage categories in your DMS system. Add as many values as needed.

```
<!-- [Data Segregation-Blacklist] -->
    <Obligation>
    <DisplayName>Data Segregation - Blacklist</Display-
Name>
    <RunAt>PEP</RunAt>
    <Name>RSTRCT</Name>
    <Arguments>
    <Argument>
    <Name>Storage category</Name>
    <Value>ZUS_LOC</Value>
```

```

        <Value>ZUS_LOC2</Value>
    </Argument>
</Arguments>
</Obligation>

    <!-- [Data Segregation-Whitelist] -->
    <Obligation>
    <DisplayName>Data Segregation - Whitelist</Display-
Name>

    <RunAt>PEP</RunAt>
    <Name>ALLOWD</Name>
    <Arguments>
    <Argument>
    <Name>Storage category</Name>
    <Value>ZUS_LOC</Value>
    <Value>ZUS_LOC2</Value>
    </Argument>
    </Arguments>
    </Obligation>

```

For more information about configuring SAP obligations, see [Configuring SAP Obligations](#) on page 70.

## Configuring the Check-in and Check-out Actions

To create data segregation policies, the following SAP actions must be added to Policy Studio:

- Check-in
- Check-out

You add the definitions of these actions, shown below, to the Policy Server’s configuration file, `configuration.xml`.

```

<Action>
    <Name>CHECK_IN</Name>
    <DisplayName>Check_in</DisplayName>
    <ShortName>CN</ShortName>
    <Category>Transform</Category>
</Action>

<Action>
    <Name>CHECK_OUT</Name>
    <DisplayName>Check_OUT</DisplayName>
    <ShortName>CO</ShortName>
    <Category>Transform</Category>
</Action>

```

For more information about configuring SAP actions, see [Configuring SAP Actions](#) on page 69.

---

## Configuration for Integrated Rights Management (IRM)

Dynamic Authorization Management for SAP offers Integrated Rights Management (IRM), which provides application integration with the Compliant Enterprise Rights Management Server (RMS) and NextLabs Rights Management Client (RMC). Rights Management Server automatically injects classification values into originals uploaded or stored in SAP. Based on how you define policies, RMS can also automatically apply NextLabs Encryption to files.

**Note:** To access NextLabs Encrypted files on endpoints, Rights Management Client must be installed and configured. Consult the *Rights Management Product Documentation* for more detailed information on Rights Management Client and Rights Management Server.

For an example policy using IRM, see [Example Policy: IRM Encryption and Tagging](#) on page 228.

### Installing Rights Management Server

The Rights Management Server must be installed on the same host as the NextLabs Policy Controller. After it is installed, RMS runs as its own Windows Service.

#### Procedure

- 1 Stop the Policy Controller through **Services > Control Center Enforcer Service > Stop**.
- 2 Locate the install.bat files supplied by NextLabs Technical Support, and locate the files on the C: drive where you plan to install RMS.
- 3 Launch a command prompt as Administrator. Change directories to where the install.bat file is located and run the file.
- 4 After the installation is complete, you can view the new service in the *Services* screen. It will not be started.

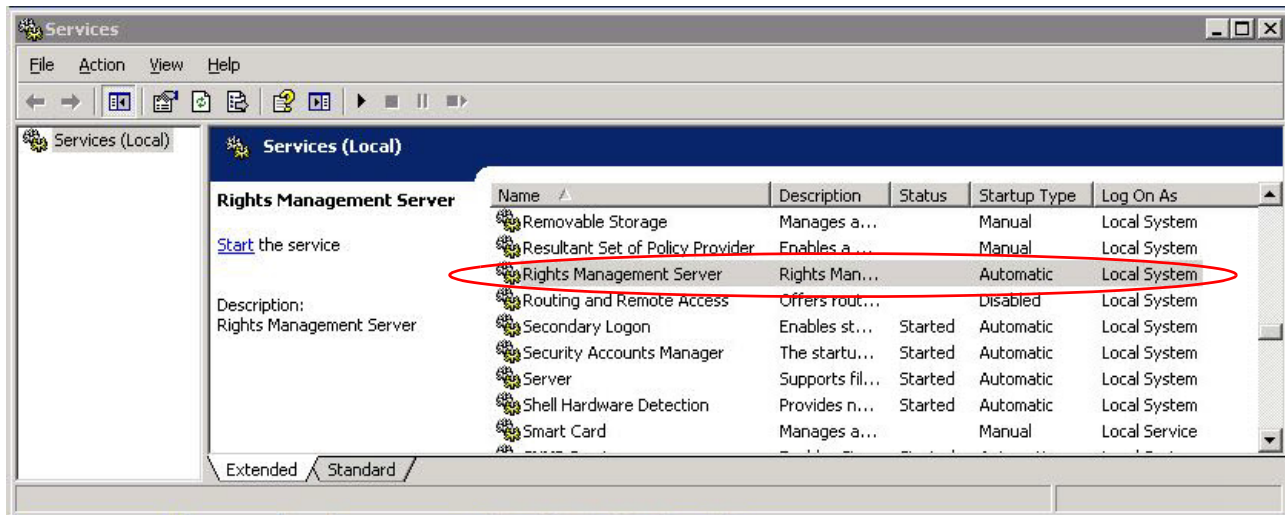


Figure 5-2: Rights Management Server

- Restart the Control Center Enforcer Service.
- Start the Rights Management Server.

## Configuring the IRM Conversion Directory

Integrated Rights Management (IRM) requires a location to temporarily store files and information while batch processes are being performed. This location must be configured on the same host as the Policy Controller.

### Before You Begin

The SAP user running the IRM process must have permission to write and read files within this directory.

### Procedure

- If not already done, on the same host as the Policy Controller, create a folder in the c: drive for conversion (in our example, this is simply `C:\Conversion`). This is the root folder, and may include additional subfolders.
- In SAP interface, enter the `/nFile` transaction to access the *Logical File Path Definition* screen.
- In the list of logical paths, locate the entries associated with IRM. You will be configuring the default setting for physical paths for each of these logical paths:
  - `/NEXTLABS/IRM_DOWNLOAD`
  - `/NEXTLABS/IRM_TAG`
  - `/NEXTLABS/IRM_UPLOAD`
  - `/NEXTLABS/IRM_ID`

4 Select /NEXTLABS/IRM\_DOWNLOAD. Click **Assignment of Physical Paths to Logical Path**.

5 In the Physical path field, define the path to the conversion directory, using the following format:

```
\\<hostname>\Conversion\<PARAM_2>\IN\<FILENAME>
```

where <hostname> implicitly includes the host's C:\ drive. (In other words, the path above references C:\Conversion.) If additional folders appear under the root conversion folder, you must also include them in the path. <PARAM\_2>\IN\<FILENAME> must be written exactly as is.

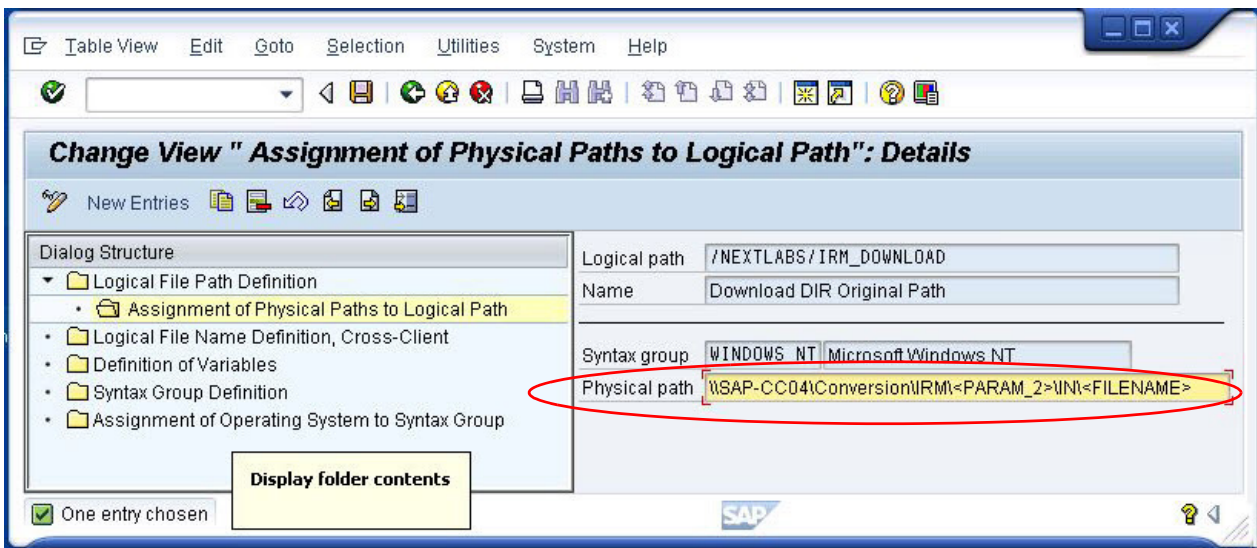


Figure 5-3: Configuring Logical Path for IRM\_DOWNLOAD

6 Click **Logical File Path Definition** to return to the list of logical paths.

7 Select /NEXTLABS/IRM\_TAG. Click **Assignment of Physical Paths to Logical Path**.

8 In the Physical path field, define the path to the conversion directory using the following format:

```
\\<hostname>\Conversion\<PARAM_2>\TAG\<FILENAME>
```

where <hostname> implicitly includes the C:\ drive. In other words, the path above points to C:\Conversion.) If additional folders appear under the root conversion folder, you must also include them in the path. <PARAM\_2>\TAG\<FILENAME> must be written exactly as is.

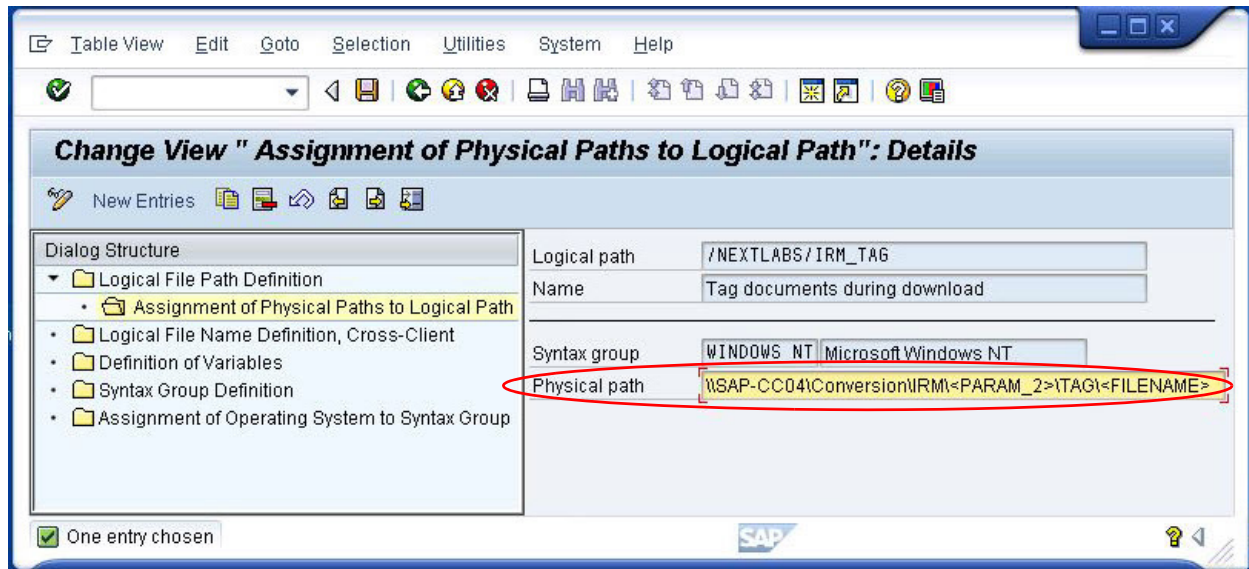


Figure 5-4: Configuring Logical Path for IRM\_TAG

- 9 Click **Logical File Path Definition** to return to the list of logical paths.
- 10 Select /NEXTLABS/IRM\_UPLOAD. Click **Assignment of Physical Paths to Logical Path**.
- 11 In the **Physical path** field, define the path to the conversion directory, using the following format:

```
\\<hostname>\Conversion\<PARAM_2>\OUT\<FILENAME>
```

where <hostname> implicitly includes the C:\ drive. (In other words, the path above references C:\Conversion.) If additional folders appear under the root conversion folder, you must also include them in the path. <PARAM\_2>\OUT\<FILENAME> must be written exactly as is.



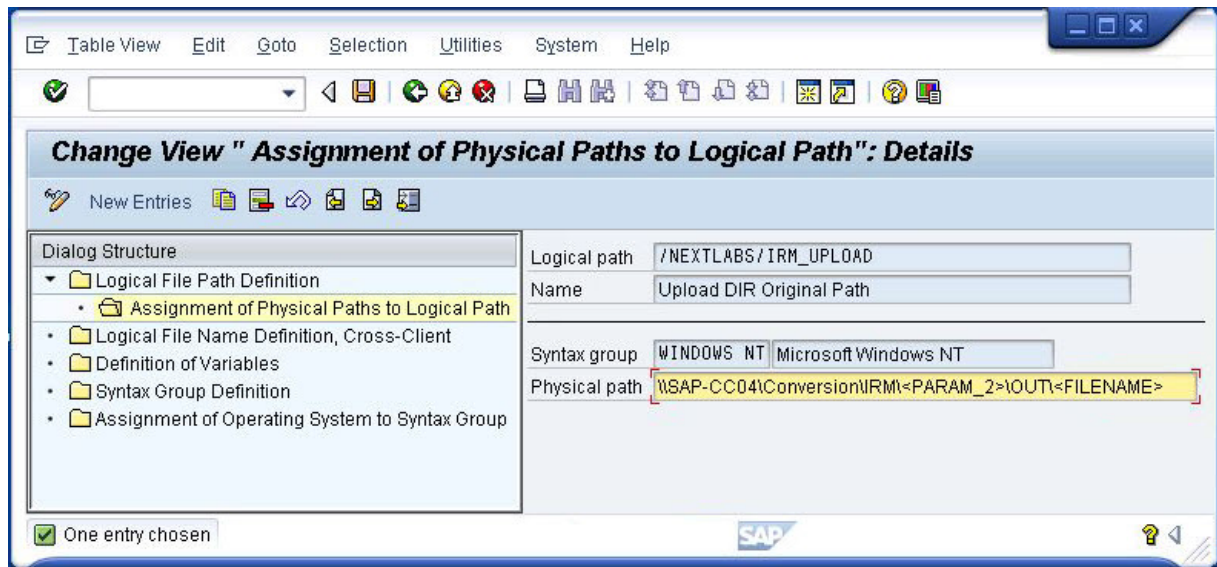


Figure 5-5: Configuring Logical Path for IRM\_UPLOAD

- 12 Select `/NEXTLABS/IRM_ID`. Click **Assignment of Physical Paths to Logical Path**.
- 13 In the Physical path field, define the `<hostname>` to point to the conversion directory. The definition implicitly includes the `C:\` drive. In other words, the path above points to `C:\Conversion`.) If there are additional folders under the root conversion folder, you must also include them in the path.
- 14 Click **Save** to save your work.

## Configuring IRM Selection Criteria (Filter)

You can determine which files should be sent to the IRM queue for policy evaluation. Depending on whether you are configuring for SAP ECC or for SAP cFolders, you should use different screens.

### Configuring IRM Selection Criteria for SAP ECC

Use the *IRM Selection Criteria* screen to configure IRM for SAP ECC. If you do not set a filter, or apply a wildcard (\*), all originals are sent to the queue for evaluation. In many cases, an organization may wish to limit which originals are being processed, because only certain objects are considered sensitive or controlled, and/or to avoid an impact on system performance.

#### Procedure

- 1 In the SAP interface, enter transaction `SE38`. The *ABAP Editor* screen appears.
- 2 In the Program field, enter `/NEXTLABS/NIRM`.
- 3 Select **Variants** and click **Execute** (or press [F8]).

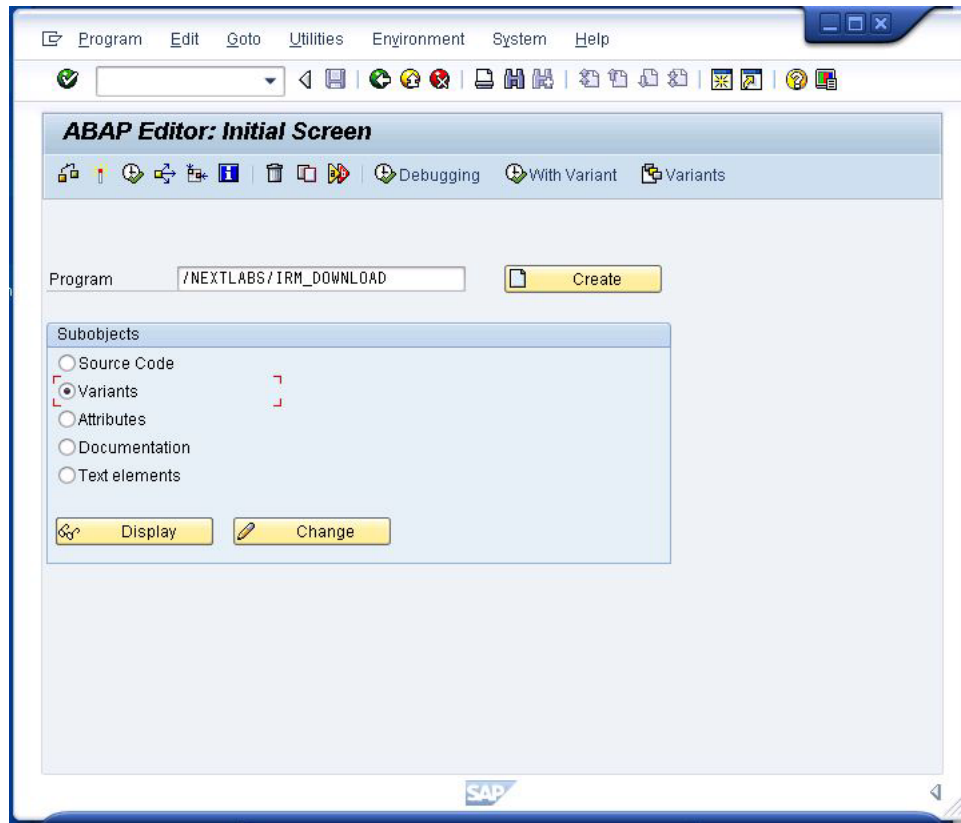


Figure 5-6: Modifying IRM\_DOWNLOAD Variants

- 4 For Documents and Materials, as well as for their Compound Keys (if configured), specify the range of objects that should be included in the evaluation queue.

**Note:** You can click the arrows next to each criteria to enter a list of multiple items. You can also use partial wildcards (DOC-12\*) to create filter criteria here.

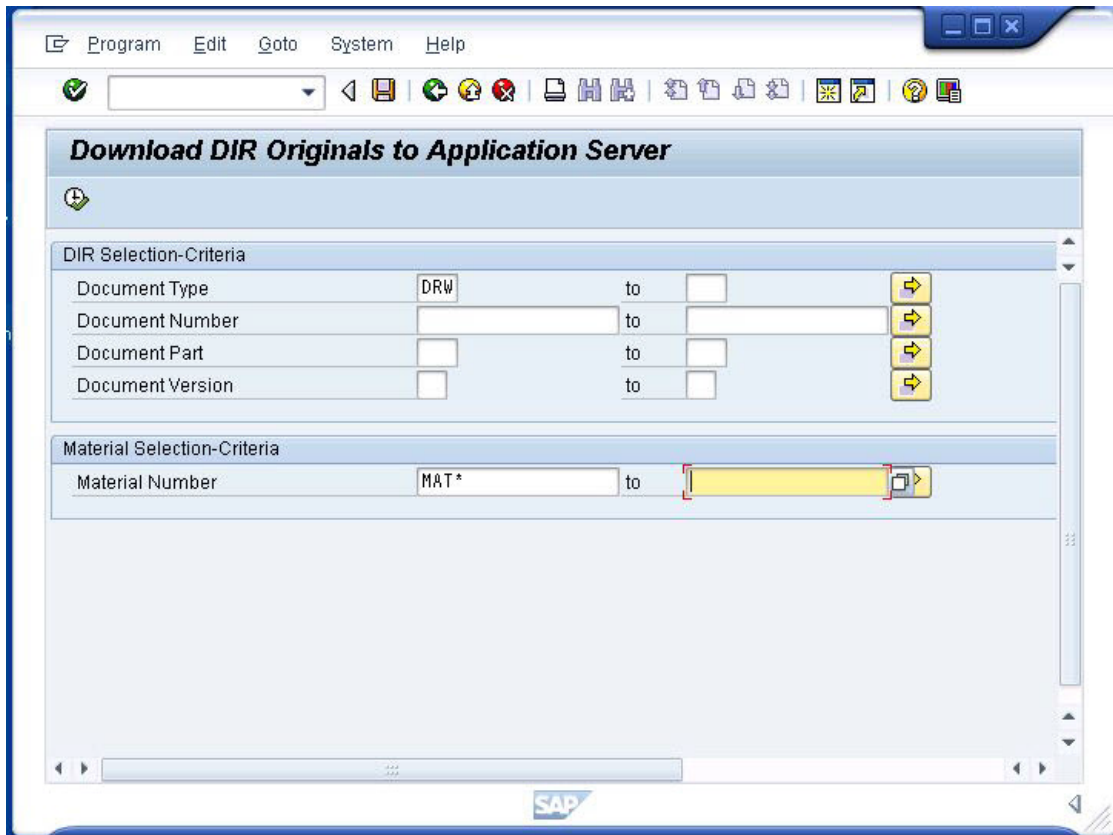


Figure 5-7: Defining DIR and Material Selection Criteria

5 Click Save.

### Configuring IRM Selection Criteria for SAP cFolders

If you are configuring selection criteria for IRM for SAP cFolders, you use the `/NEXTLABS/IRM_CFX` program.

#### Procedure

- 1 In the SAP interface, enter transaction `SA38`. The *ABAP Program Execution* screen appears.
- 2 Enter `/NEXTLABS/IRM_CFX`. Click **Execute**.



Figure 5-8: IRM\_CFX Program

3 Enter selection criteria for the cFolders Documents:

- (Required) Select a cFolders Document range by the description. You have the option to do this using partial wildcards or by selecting individual documents. You can also select all cFolders documents by entering a full wildcard (\*) here.

**Note:** This field is mandatory. You must specify at least one selection criteria for the IRM program to initiate.

- (Optionally) Specify cFolders Documents by range of **Created On** dates.
- (Optionally) Specify cFolders Documents using the **Created By** property.
- (Optionally) Specify cFolders Documents by range of **Changed On** dates.
- (Optionally) Specify cFolders Documents using the **Changed By** property.

4 A warning appears if the number of documents to be processed is higher than the threshold specified in the *CONCFG* screen. If you want this warning to be ignored (continue processing documents anyway) select **Ignore Performance Warning**.

**Note:** For more information on configuring this threshold, see [Changing Connection Configuration Settings](#) on page 96.

**NextLabs:NXLAFX:IRM:Run cFolder Documents for IRM Process**

cFolder Document

Created On (Date)  to

Created By  to

Last Changed On (Date)  to

Last Changed By  to

Ignore Performance Warning If Number of Records to be Processed is higher than  
Configured in Transaction SM30 (/NEXTLABS/CONCFG) as IRM\_UPLOAD\_NO\_OF\_RECORDS\_WARNG

Information:  
Here cFolder Document is Mandatory Field, Enter specific document (and/or Range)  
or Enter \* for considering all documents.  
Once it finishes the Execution, Log can be found in Transaction SLG1.  
For Transaction SLG1, Select /NEXTLABS/ as Object and IRM as Sub Object.

Figure 5-9: Selecting cFolders Documents for IRM Processing

## Defining Background Jobs for IRM

IRM runs as two batch processes, `IRM_DOWNLOAD` for SAP ECC and `IRM_CFX` for cFolders. If your implementation includes both, both must be configured as background jobs. The background jobs should be scheduled so updates occur at the necessary frequency to address business and authorization requirements.

**Note:** For recommendations on how to schedule these background jobs, contact NextLabs Professional Services.

### Defining the `IRM_DOWNLOAD` Background Job for SAP ECC

#### Procedure

- 1 In the SAP interface, enter transaction `SM36`. The *Define Background Job* screen appears.
- 2 Click the *Job Wizard* button to launch the job wizard.

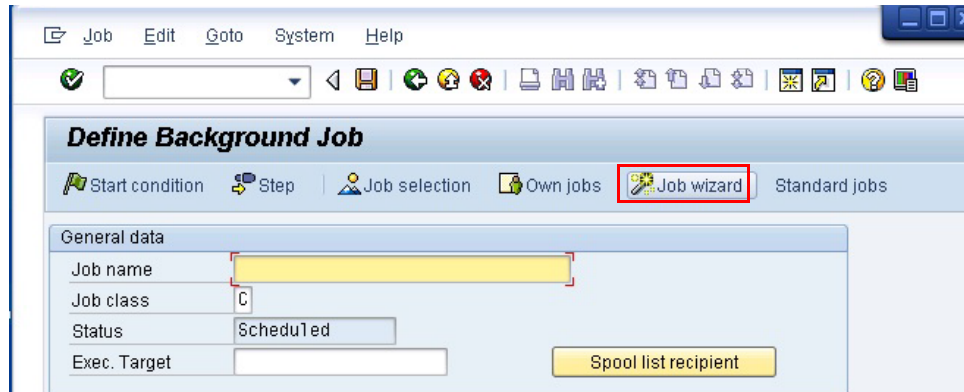


Figure 5-10: Job Wizard

3 Enter background information about the job:

- The Job Name can be anything you specify.
- Select the priority (Job Class) for the batch process to reflect your business requirements. Because this is a frequently running process, we select Low Priority. However, if you want the job to take priority over other batch processes, because of the high risk of exposing sensitive data, select a higher priority for the job.
- TargetServer is not required.

4 Click **Continue**.

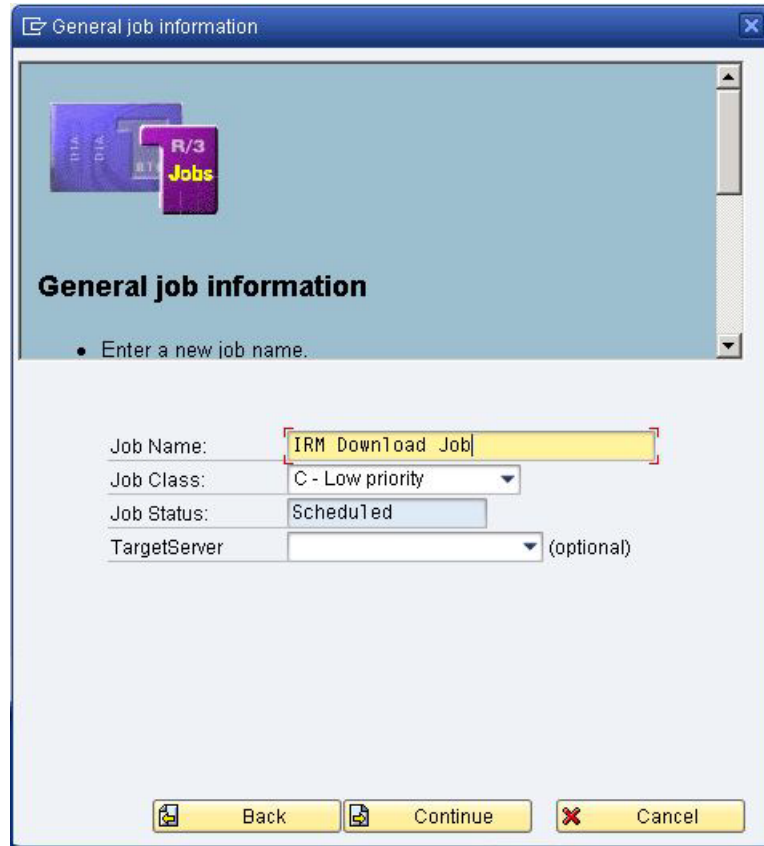


Figure 5-11: General Job Information

5 Select **ABAP program step** and click **Continue**.

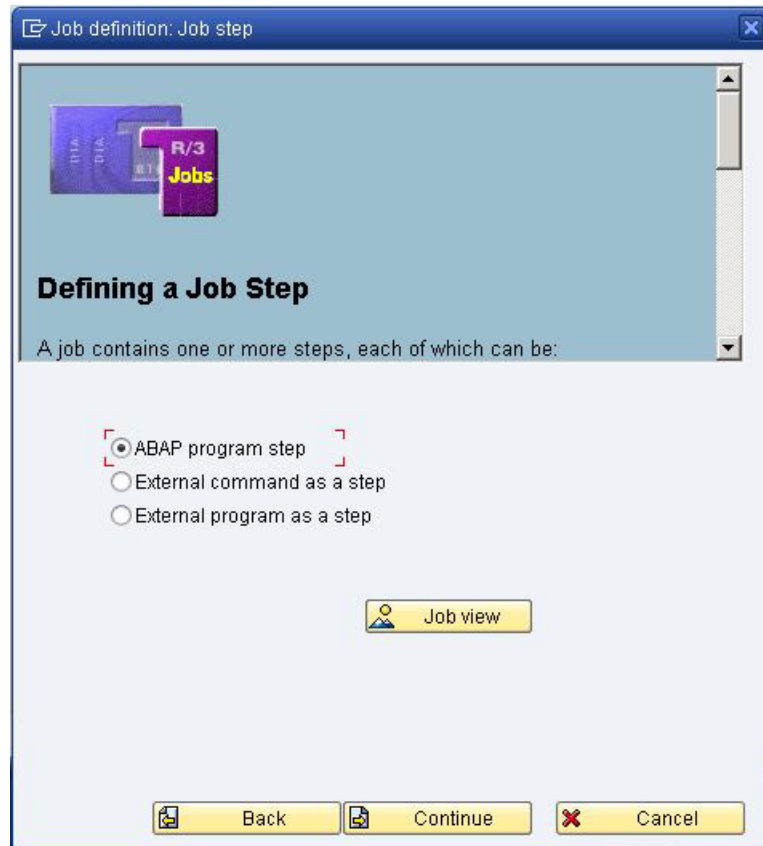


Figure 5-12: Defining a Job Step

6 Enter the following in ABAP program name field: /NEXTLABS/NIRM, and click **Continue**.



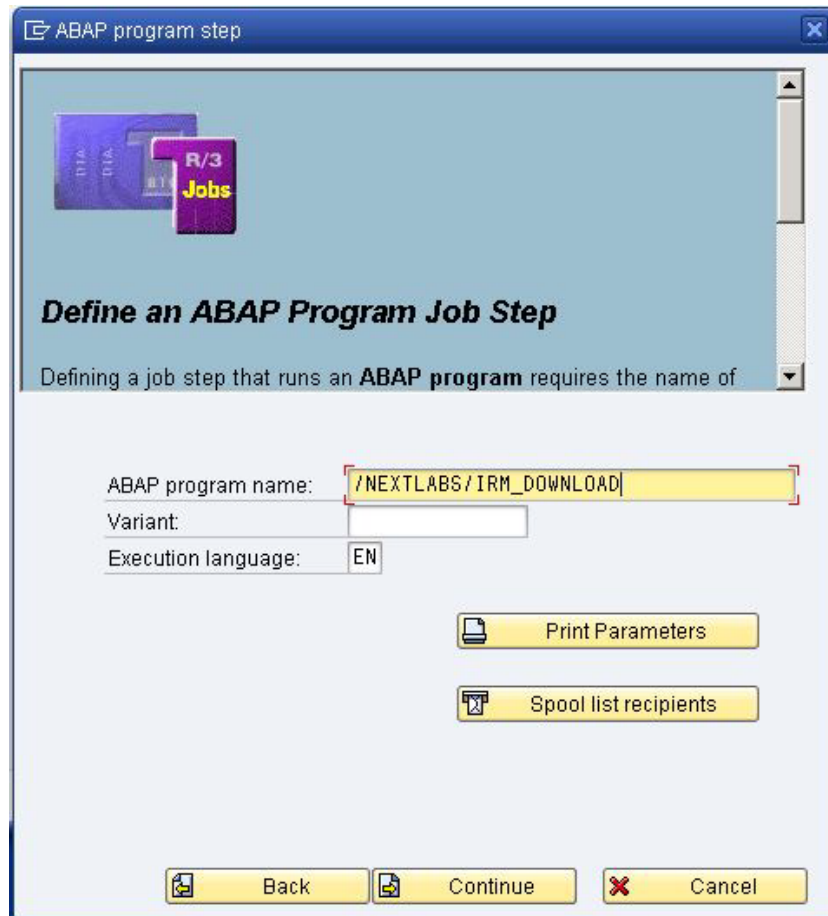


Figure 5-13: Job Step IRM\_DOWNLOAD

7 In the *Multi-step Job* screen, do not select **Add additional steps** and click **Continue**.

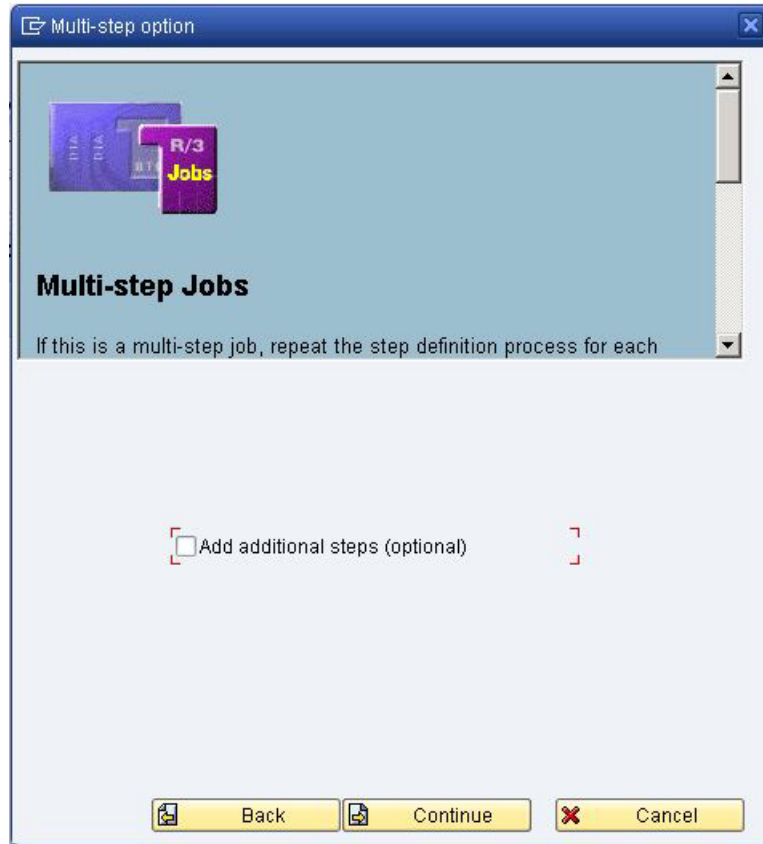


Figure 5-14: Multi-step Job Screen

- 8 In the scheduling screen, select the frequency for running the batch process, based on your business requirements, then click **Continue**.

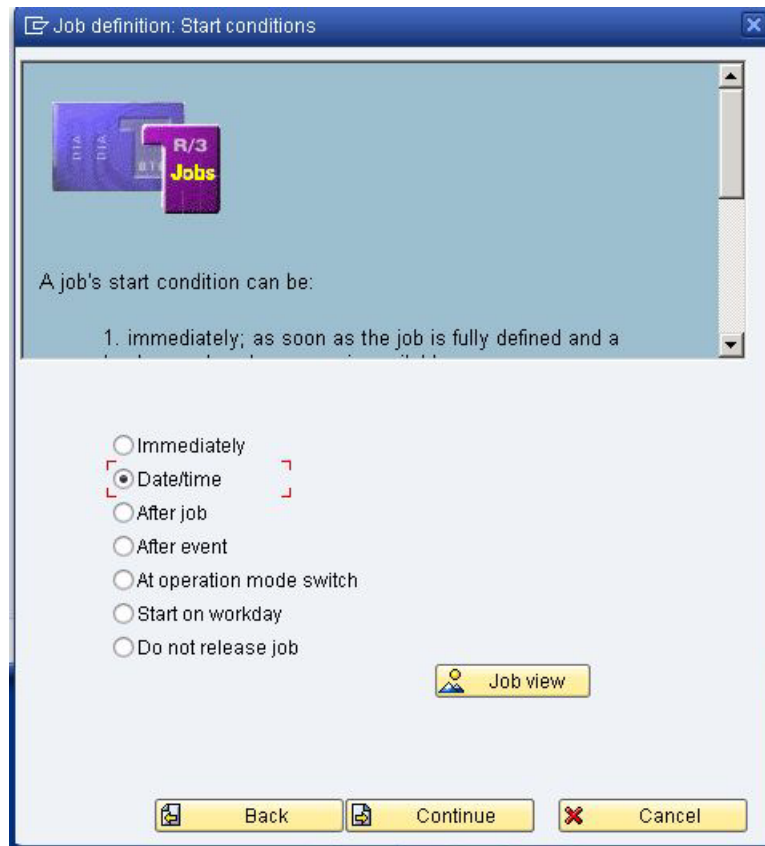


Figure 5-15: Scheduling the Job

A message appears stating that the job has been defined successfully.

9 Click **Complete**.

### Defining the IRM\_CFX Background Job for SAP cFolders

#### Procedure

- 1 In the SAP interface, enter transaction `SM36`. The *Define Background Job* screen appears.
- 2 Click the *Job Wizard* button to launch the job wizard.

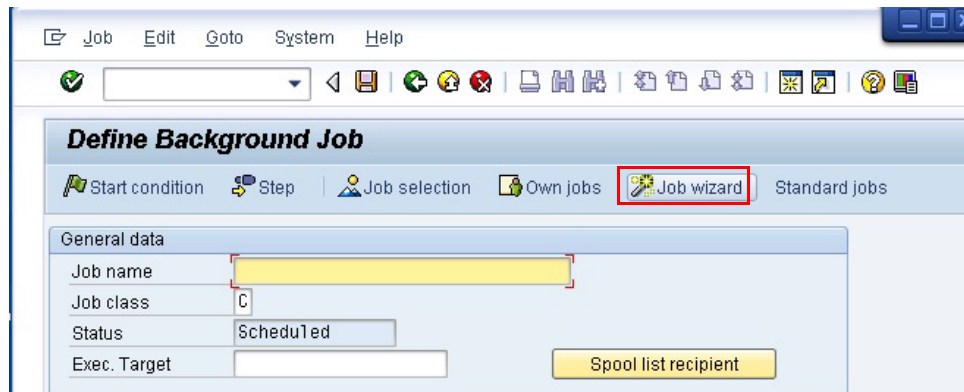


Figure 5-16: Job Wizard

3 Enter background information about the job:

- The Job Name can be anything you specify.
- Select the priority (Job Class) for the batch process to reflect your business requirements. Because this is a frequently running process, we select Low Priority. However, if you want the job to take priority over other batch processes, because of the high risk of exposing sensitive data, select a higher priority for the job.
- TargetServer is not required.

4 Click **Continue**.

5 Select **ABAP program step** and click **Continue**.

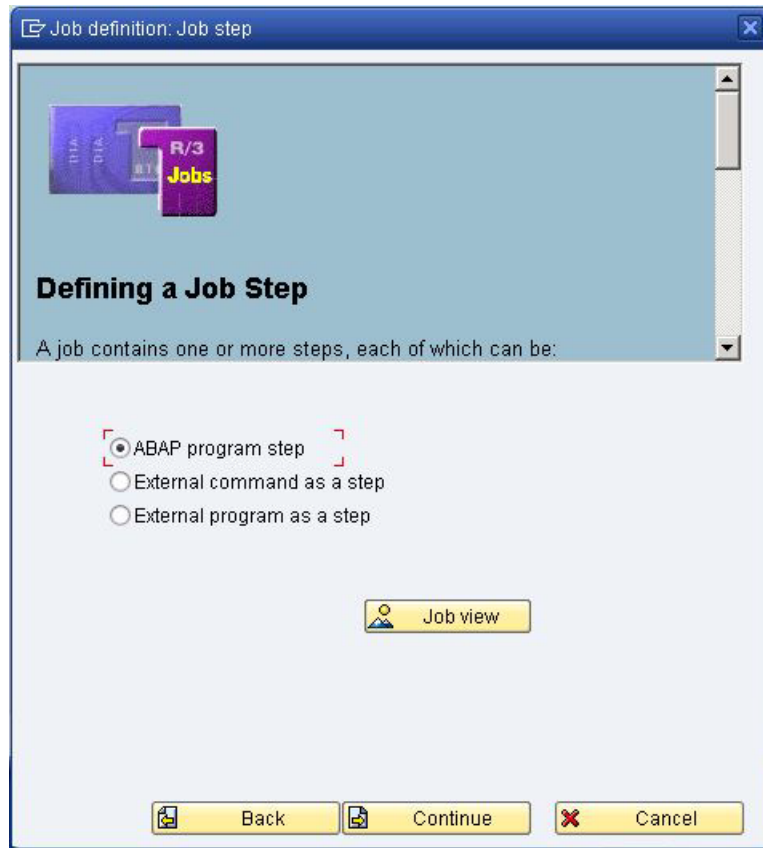


Figure 5-17: Defining a Job Step

6 Enter the following in ABAP program name field: /NEXTLABS/IRM\_CFX, and click **Continue**.

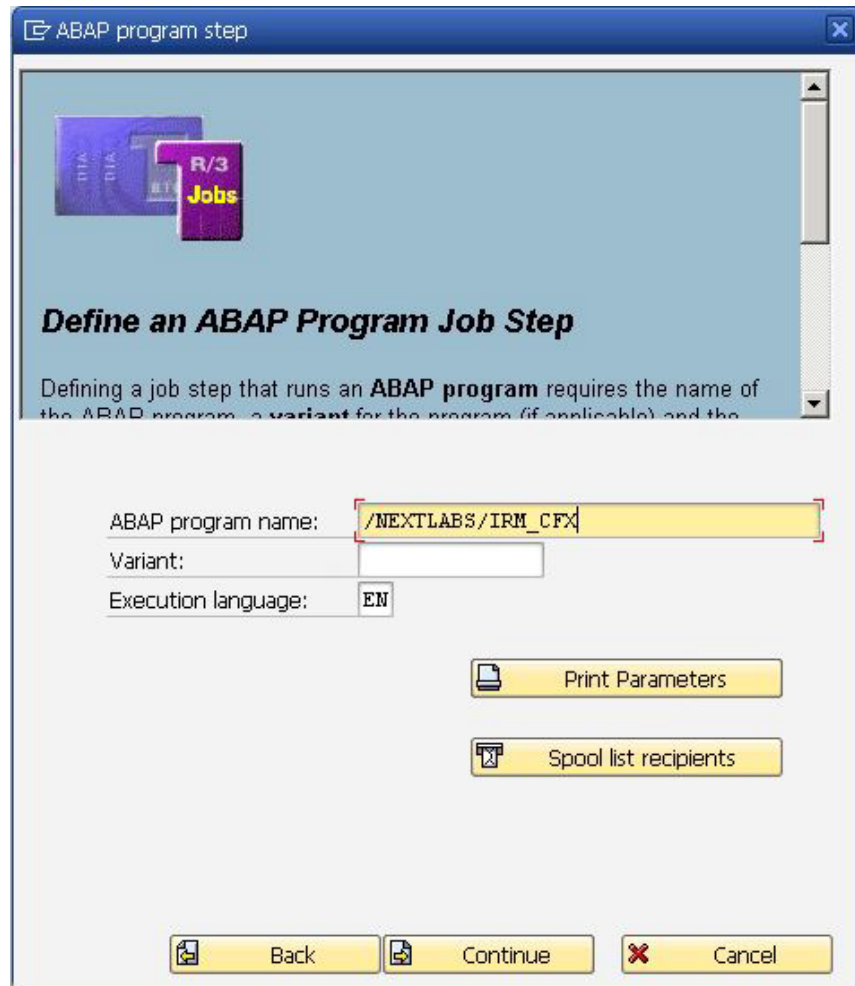


Figure 5-18: Defining a background job for IRM\_CFX

7 In the *Multi-step Job* screen, do not select **Add additional steps** and click **Continue**.

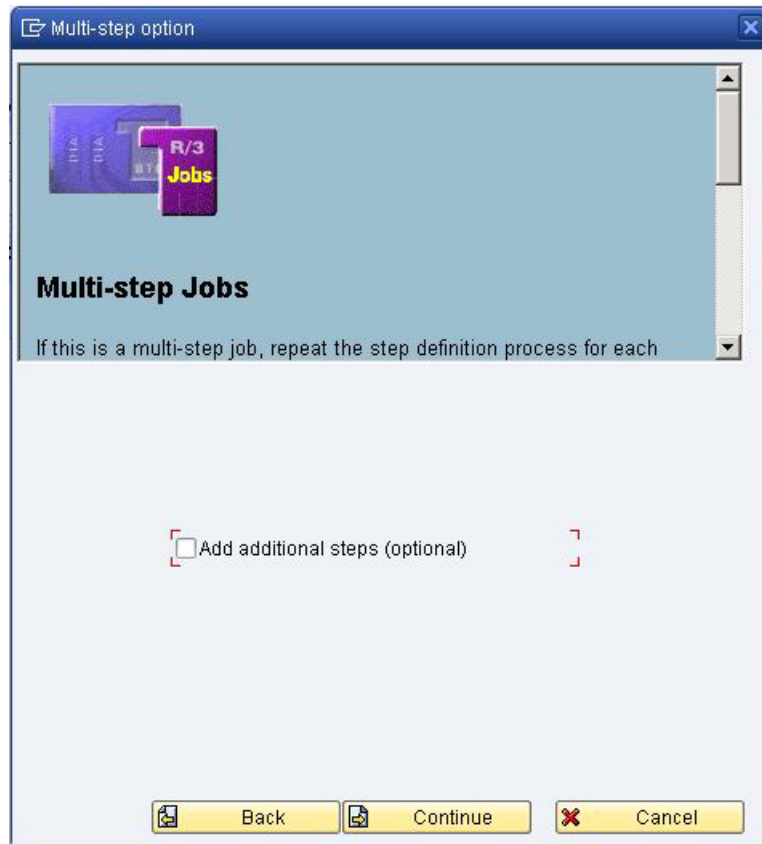


Figure 5-19: Multi-step Job Screen

- 8 In the *Scheduling* screen, select the frequency for running the batch process, based on your business requirements, then click **Continue**.

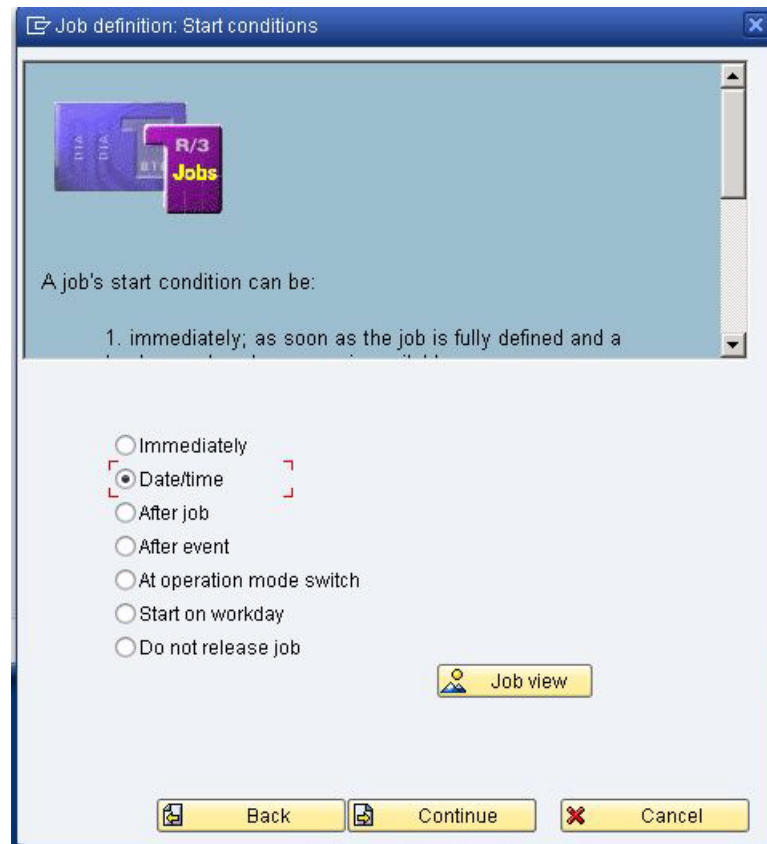


Figure 5-20: Scheduling the Job

A message appears stating that the job has been defined successfully.

9 Click **Complete**.

## Configuring the IRM Obligation

You also must insert obligations for Integrated Rights Management (IRM) in the Policy Server's configuration.xml file. The obligation required for IRM is called Integrated Rights Management. The same obligation is used for SAP ECC and SAP cFolders.

**Note:** Other Obligations pertinent to the general SAP Policy Model must also be configured using this same procedure. For more information, see [Configuring SAP Obligations](#) on page 70.

### Procedure

1 Use any text editor to open the SAP Obligations\_FullBuild.xml file, supplied by NextLabs Professional Services.



- 2 Open the main configuration file, `configuration.xml`, on the Control Center host. It is located at `<installDirectory>\server\configuration`.
- 3 Copy and paste the Integrated Rights Management obligation from the `SAP Obligations_FullBuild.xml` file into the main configuration file. It must be placed in the `<Obligations>` section.
- 4 Save your changes in the file and restart your Policy Server.
- 5 After the system restarts, test the configuration by opening Policy Studio and creating a New policy. The new obligations should appear in the drop-down list of Custom Obligations.

**Note:** For example policies that use this, see [Designing Integrated Rights Management Policies](#) on page 228.

## Configuring Encryption Keys

If you use IRM to apply NextLabs Encryption to files, a Shared Key must be created using Key Management Server (usually installed with the Control Center, or on the same host as the ICENet server). After this key is created, it is automatically sent to endpoints on the next heartbeat.

**Note:** The default key ring name is `NL_SHARE_DEFAULT`. If you create a Key Ring with any other name, it must be registered with all endpoints.

If you have distributed ICENet servers, you only need to generate Shared Key Rings and Keys on one host. Shared Keys and Key Rings are automatically distributed across all ICENet server locations where Key Management Server is installed.

For more information on Key Management and NextLabs Encryption, consult the Rights Management User's Guide.

### Procedure

- 1 In the Command prompt on the device where Key Management is installed, change directory to `<Install Dir>\Nextlabs\Policy Server\tools\keymanagement`.
- 2 Run the following command to create a shared key ring, where `keyRingName` is `NL_SHARE_DEFAULT`:

```
keymanagement.bat -u <username> -w <password>
-createKeyRing -keyRingName NL_SHARE_DEFAULT
```

**Note:** The Shared Key Ring can be no larger than 16 characters. If you create a key ring name that is longer than this, no error is display, but the key ring will not be created.

- 3 Run the following command to create a shared key on the newly created key ring:

```
keymanagement.bat -u <username> -w <password>
-generateKey -keyRingName NL_SHARE_DEFAULT
```

---

## Configuring the Read Tags Feature

You must configure these settings before you use the Read Tags feature.

- [Configuring the RFC Connection for Read Tags](#)
- [Configuring SAP SDK Service and SAPJCo-EDRM Properties Files](#)
- [Configuring SAP Data Handling and Connection Settings for Read Tags](#)
- [Adding Custom Classification Values](#)
- [Mapping Classification Values to the Policy Controller](#)
- [Implementation Reference for Read Tags](#)

For more information about using the Read Tags feature, see [Reading Tags](#).

### Configuring the RFC Connection for Read Tags

Use this procedure to configure an RFC connection for the Read Tags feature to work.

- 1 In the SAP interface, enter the `SM59` transaction code. The Configuration of RFC Connections screen appears.
- 2 Select the TCP/IP connections folder and then click **Create**.
- 3 Set the Connection type to T.
- 4 Enter an RFC Destination name, for example, `NEXTLABS_READTAG`.  
Make note of the RFC Destination name, because you will need to enter the RFC destination name into the NEXTLABS/CONCFG table when [Configuring SAP Data Handling and Connection Settings for Read Tags](#).
- 5 In the **Technical Settings** tab, select **Registered Server Program**.
- 6 Enter a Program ID, for example, `NXL_CONNECT_TO_READTAGS`.
- 7 Save the configuration.

#### Next steps

[Configuring SAP SDK Service and SAPJCo-EDRM Properties Files](#)

### Configuring SAP SDK Service and SAPJCo-EDRM Properties Files

Use this procedure to configure the SAP SDK Service properties file and the SAPJCo-EDRM properties file.

**Note:** The configuration procedure in this section assumes you have already installed and configured the NextLabs Java Connector, which is the communication interface between the Policy Controller and the SAP Agent. The steps here are additional configuration steps that are specific to NextLabs EDRM.

## Procedure

- 1 Stop the Policy Controller.
- 2 Edit the SAP SDK Service properties file to ensure that the Read Tags feature works correctly.  
This is the same file that had been configured during the [Manual Installation of the Java Connector for the Policy Controller for Java](#).

a. Open the `SAPJavaSDKService.properties` file using a text editor.

b. Locate the line `server_prefix=SERV1_;` and add `SERV1_;` at the end of the line as shown in the following example:

```
server_prefix=SERV1_;SERV1_;
```

c. Add the following code below the `server_prefix=SERV1_;` line:

```
# START of EDRM Settings
#destination data provider Connection details
SERVRMI_jco.client.ashost=[!CLIENT_HOST!]
SERVRMI_jco.client.sysnr=[!CLIENT_SYSNR!]
SERVRMI_jco.client.client=[!CLIENT_ID!]
SERVRMI_jco.client.user=[!CLIENT_USER!]
SERVRMI_jco.client.passwd=[!CLIENT_PASSWD!]
SERVRMI_jco.client.lang=en
#comment below two lines if connection pool is not required
SERVRMI_jco.destination.peak_limit=3
SERVRMI_jco.destination.pool_capacity=3
#server data provider Connection details
SERVRMI_jco.server.gwhost=[!GATEWAY_HOST!]
SERVRMI_jco.server.gwserv=[!GATEWAY_SERV!]
SERVRMI_jco.server.progid=[!GATEWAY_PRGID!]
SERVRMI_jco.server.connection_count=02
# NOTE: To add more destination server settings; add SERV1_ or
SERV2_
etc to server_prefix and respective block of SERV1_ or SERV2_
properties.
# END of SERV1 Settings
```

d. Replace the following variables, including the brackets and the exclamation points, with the appropriate values for your system:

- `[!CLIENT_HOST!]`: The hostname of your server, that is, the FQDN for the host where SAP ECC is installed.

- [!CLIENT\_SYSNR!]: The system to use, that is, the system number.
- [!CLIENT\_ID!]: The client ID to use. For example: 100
- [!CLIENT\_USER!]: The user name of the account. For example: developer
- [!CLIENT\_PASSWD!]: The password of the account. You must encrypt the password by using the `mkpassword.bat` utility that is available when you install NextLabs Policy Server. For more information, see *NextLabs Control Center Installation Guide*.
- [!GATEWAY\_HOST!]: The host name of the gateway server.
- [!GATEWAY\_SERV!]: The gateway service name.
- [!GATEWAY\_PRGID!]: The Program ID that was defined when [Configuring the RFC Connection for Read Tags](#), step 6. That is, `NXL_CONNECT_TO_READTAGS`.

e. Save the `SAPJavaSDKService.properties` file.

3 Extract the `SAPJCo-EDRM` Plug-in zip file to a temporary location.

4 Copy the `SAPJCo-EDRM.jar` file to the following location:

```
<install_dir>/Policy Controller/jservice/jar/sap/
```

5 Copy the `SAPJCo-EDRM.properties` file to the following location:

```
<install_dir>/jservice/config/
```

6 Open the `SAPJCo-EDRM.properties` file using a text editor.

The following is an example of the `SAPJCo-EDRM.properties` file.

```
-----
name = SAPJCo-RMAPI

jar-path = <install_dir>/Policy Controller/jservice/jar/sap/SAPJCo-EDRM.jar

friendly_name = SAP Java EDRM API
description = SAP Java EDRM API
category = EDRM SERVICE

#SAP module names
rmapi_handler=/NEXTLABS/READ_API_AGENT

#server destination details prefix
server_prefix=SERVRMI_

#Delimiter
delimiter = |
-----
```

7 In the `SAPJCo-EDRM.properties` file, ensure that `jar-path` contains the correct location of the `SAPJCo-EDRM.jar` file. That is, the `SAPJCo-EDRM.jar` file must be pointing to the following location:

```
<install_dir>/Policy Controller/jservice/jar/sap/
```

- 8 Save the file and start the Policy Controller.

### Next steps

[Configuring SAP Data Handling and Connection Settings for Read Tags](#)

## Configuring SAP Data Handling and Connection Settings for Read Tags

Use this procedure to configure SAP data handling and connection settings for the Read Tags feature to work.

- 1 In the SAP interface, enter the `SM30` transaction code. The Maintain Table Views screen appears.
- 2 In the Table/View field, enter `/NEXTLABS/CONCFG`.
- 3 Click **Change**. The Change View “NextLabs: Default Configuration Maintenance”: Overview screen appears.
- 4 Click **New Entries**. The New Entries: Overview of Added Entries screen appears.
- 5 In the Activity Name field, enter `READTAG_AGENT_RFC_NAME`.
- 6 In the Activity Type field, enter the same RFC destination name that you created while Configuring the RFC Connection for Read Tags.  
For example, `NEXTLABS_READTAG`.
- 7 Click **Save**.

### Next steps

[Adding Custom Classification Values](#)

## Adding Custom Classification Values

For more information about adding custom classification values to the Security Classification Maintenance table, see [Adding New Classification Values](#).

### Next steps

[Mapping Classification Values to the Policy Controller](#)

## Mapping Classification Values to the Policy Controller

For more information about mapping the classification values configured in the Security Classification Maintenance table to the Policy Controller, see [Mapping Security Fields \(SECM PG\)](#).

### Next steps

[Implementation Reference for Read Tags](#)

## Implementation Reference for Read Tags

This section describes the implementations required for the Read Tags feature.

Topics:

- [Enhancement Implementation for Read Tags](#)
- [BADI Implementation to Configure a Temporary File Location for Read Tags](#)
- [BADI Implementation for Dynamic User or Resource Attribute](#)

### Enhancement Implementation for Read Tags

[Table 5-1](#) lists the required enhancement implementation for systems using the Read Tags feature.

*Table 5-1: Required Enhancement Implementation for Read Tags*

Type	T-Code	BADI Name/Program	Method	Code
BADI	SE19	DOCUMENT_MAIN01	AFTER_SAVE	INCLUDE /NXLECC/READ_TAGS.

### BADI Implementation to Configure a Temporary File Location for Read Tags

You can build and activate a custom BADI if you do not want to design and implement a policy to get the temporary file location.

Use this procedure to implement a custom logic to get the temporary storage location for the Read Tags feature.

#### Procedure

- 1 In the SAP interface, enter transaction `SE19`. The Business Add-Ins screen appears.
- 2 In the Create Implementation section, select **New BADI-Enhancement Spot** and enter build and activate a custom BADI.
- 3 Click **Create Impl.**
- 4 Enter an implementation name.
- 5 In the Implementation Short Text field, enter a description, and then click **OK**.
- 6 In the BADI Implementation field, enter a name.  
For example: `Z_READ_TEMP_LOC`

**Note:** The BADI Implementation should begin with `Z`.

- 7 In Short Text, enter a description.

- 8 Enter the implementation class and BADI definition `/NXLECC/ENH_READ_BADI` and then click **Continue**.
- 9 Click the implementing class. The `GET_DYN_SERVER_LOC` method is available for inserting code to get the server temporary location for the document storage category.
- 10 Double-click the method and write the code for the method.
- 11 Save and activate the method.
- 12 Save and activate the BADI implementation.

## BADI Implementation for Dynamic User or Resource Attribute

A typical use case to use this implementation would be to enforce control on user or resource attributes that are not enrolled or that cannot be enrolled into NextLabs Control Center.

For example, you want to enforce access control based on physical location or any other attribute that can vary dynamically. ABAP developers can insert dynamic look-ups into policy checks to retrieve user and resource information from a designated store at the point of policy evaluation. The mechanisms that collect and store the user or resource attributes must be determined and developed by the customer. For example, a program can be written to automatically gather user or resource attributes (such as physical computer location) and store it as a session variable. Or, a program can be written to create a user interface that prompts users to enter attributes.

Use this procedure to configure the BADI method to collect attributes from the storage location you designate.

### Procedure

- 1 In the SAP interface, enter transaction `SE19`. The Business Add-Ins screen appears.
- 2 In the Create Implementation section, select the Classic BAdI option, and enter the BADI Name `/NEXTLABS/ENH_DYNATT`.
- 3 Click **Create Impl.**
- 4 Enter an implementation name.
- 5 In the Implementation Short Text field, enter a description, and then click **OK**.
- 6 Click the Interface tab. The following GET and SET methods are available:
  - `GET_DYN_USER_ATT`: Retrieves user attributes written to a storage location (defined by the customer). Used to pass user attributes to the requesting policy check.
  - `SET_DYN_USER_ATT`: Writes user attributes to a storage location (defined by the customer).
  - `GET_DYN_RESOURCE_ATT`: Retrieves resource attributes written to a storage location (defined by the customer). Used to pass resource attributes to the requesting policy check.
  - `SET_DYN_RESOURCE_ATT`: Writes resource attributes written to a storage location (defined by the customer).

- 7 Write code to retrieve attributes from a specified location, or to set attributes.
- 8 Save and Activate the Method.
- 9 Save and activate the BADI implementation.

---

## Reading Tags

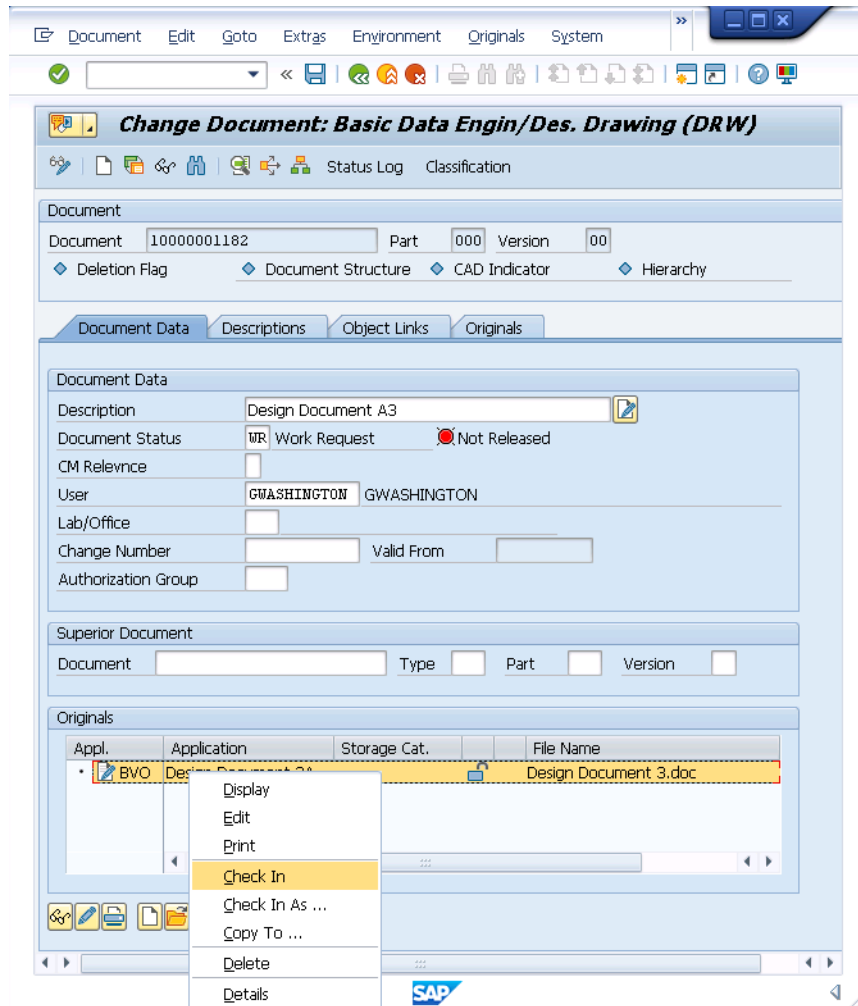
The Read Tags feature reads the pre-existing document tags when you upload a document into SAP and then inserts those tags into the corresponding columns of the Security Classification Maintenance table. Only after you check-in and save a document, and based on policy, the Read Tags feature automatically inserts the tags into the corresponding columns of the Security Classification Maintenance table.

This example procedure lists the steps to use the Read Tags feature in SAP interface.

### Procedure

- 1 In the SAP interface, enter the `CV01N` transaction code. The Create Document screen appears.
- 2 Press ENTER. The Create Document screen appears.
- 3 In the Description field, enter a description for the DIR.
- 4 Click **Open Original**.
- 5 Locate the document (with pre-existing tags) you want to upload and then click **Open**. The file is imported into the SAP system.
- 6 In the Originals section, right-click the document that you imported and then click **Check In**.

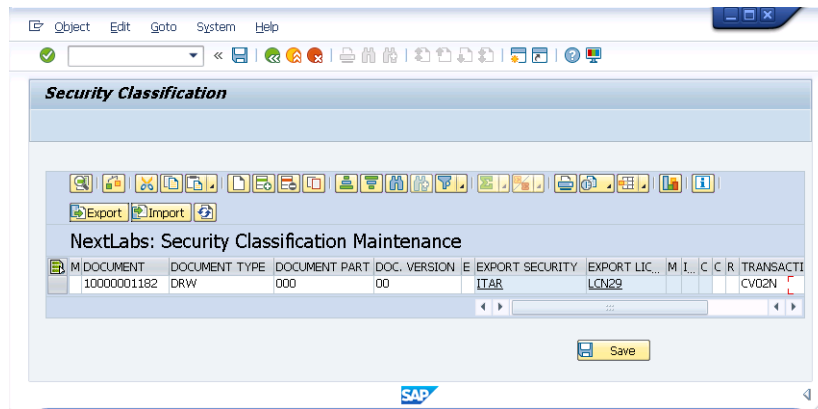




A pop-up window appears.

- 7 Select a storage category and press ENTER.
- 8 Click **Save**. A message appears stating that the document (with document number) is created.
- 9 To verify if the pre-existing tags are inserted into the Security Classification Maintenance table, perform the following steps:
  - a In the SAP interface, enter the `/nextlabs/sec_cls` transaction code. The NextLabs: Security Classification Maintenance screen appears.

- b In the Document field, enter the document number and click **Modify/Display**. The Security Classification screen appears.



## Designing Read Tags Policies

This section provides the following example Read Tags policies:

- [Example Policy: Temporary File Location](#)
- [Example Policy: Read Tags](#)

### Example Policy: Temporary File Location

The Read Tags feature uses this temporary file location to process the documents.

To configure a Temporary File Location policy, shown in [Figure 5-21](#), perform these general steps:

- 1 Create a Document policy.
- 2 Set the enforcement type to **Allow**.
- 3 Select the **Get Location** action component and drag it into the Action field. This is the only valid action.
- 4 In the Obligations area, specify these values:
  - Name: **SAP–EDRM–Temporary File Location for Processing Originals**
  - Storage Category: **<storage\_category\_name>**
  - File Location: **<file\_location>**
- 5 **Submit** and **Deploy** the policy.

## Document Policy

SAP-EDRM-Get-Tmp-File-Location

---

User +

Computer +

Application +

Perform the Following

Action -

**Get Location**  
Action Component

On Resources

Target +

Moved, Renamed or Copied:

+

Conditions

Connection Type +

Heartbeat +

Date/Time Start: +

End: +

Recurrence Time: +

Day: +

Condition Expression +

Subpolicy

Subpolicy -

Subpolicy

Obligations

On Allow, Monitor  Log

Display User Alert

Send Email

Custom Obligation

Name -

Storage Category -

File Location -

Name -

Storage Category -

File Location -

*Figure 5-21: Example Policy: Temporary File Location*

When an authorized user runs the Read Tags feature, the system uses this temporary file location to process the files.

### Example Policy: Read Tags

To create a Read Tags policy, shown in [Figure 5-22](#), perform these general steps:

1. Create a Document policy.
2. Set the enforcement type to **Allow**.
3. Select the **Save** action component and drag it into the Action field. This is the only valid action.
4. In the Obligations area, select these values:
  - (*Mandatory*) Name: **SAP-EDRM-Maintain Classification from Tags**
  - (*Mandatory*) Read Tags: **Read Tags from Original**

## Document Policy

EDRM-Read-Tag

---

**Enforcement** Allow

**Subject**

User

Computer

Application

**Perform the Following**

Action 

Save  
Action Component

**On Resources**

Target 
Moved, Renamed or Copied:

**Conditions**

Connection Type

Heartbeat

Date/Time
Start:   
End:

Recurrence
Time:   
Day:

Condition Expression

**Subpolicy**

Subpolicy

**Obligations**

On Allow, Monitor

- Log
- Display User Alert
- Send Email
- Custom Obligation

Name

Read Tags

Figure 5-22: Example Policy: Read Tags

When an authorized user uploads a document into the SAP system, the system reads the pre-existing tags from the document and inserts the tags into the Security Classification Maintenance table.



# 6 Using Classifications and Policies

---

This section explains how to use classifications and policies

Topics:

- [About Classifications and Policies](#)
- [What Can Dynamic Authorization Management Do?](#)
- [About SAP Policies](#)
- [Applying Security Classifications](#)
- [Designing SAP Access Control Policies](#)
- [Designing Access Control Policies for SAP BW](#)
- [Designing Integrated Rights Management Policies](#)
- [Designing Data Segregation Policies](#)
- [Verifying the Storage Location of Data](#)

---

## About Classifications and Policies

Classifications and policies used in Dynamic Authorization Management for SAP include:

- Policy Based Security Classifications (PBSC)
- Access control policies
- Integrated Rights Management (IRM) policies
- Data segregation policies

While this section provides some examples, keep in mind that policies are highly flexible and this section cannot cover every possible design. Depending on how you define security classifications, categorize users and resources, and author policies, you can use Entitlement Manager for SAP to apply broad access control across groups of people and data (for example, preventing all non-US users from accessing any data marked with a common flag, such as ITAR). Or, you can create granular, data-level policies that regulate access controls based on (1) Security Classification and/or Access Control Context (ACC), (2) SAP transaction or UI function type for Materials, Documents, and/or another Identifier you define, and (3) for a narrowly defined user group (for example, salespersons in the Southwest region).

While not exploring every option, this section describes the procedures you can use to make the most of this flexibility, across [The SAP Authorization Workflow](#) on page 13 (classification, access control, Integrated Rights Management (IRM)).

**Note:** For detailed information about using Policy Studio to create, deploy, and manage policies, refer to the Control Center *Policy Studio User's Guide*.

---

## What Can Dynamic Authorization Management Do?

As is discussed in more detail in the Introduction, the SAP Agent triggers a policy check based on transaction events in SAP ECC, UI function events in SAP PLM, user actions in SAP EasyDMS, or user access to classified BW objects in SAP Business Explorer (BEx) Analyzer. Different user behaviors may trigger policy checks.

The core functionality to execute a policy check and classify SAP business objects (discussed in the section [Key Benefits](#) on page 12) is included in the base Dynamic Authorization Management product. The policy checks that are supported depend on:

- The Entitlement Packs installed. Each Entitlement Pack supports a set of transactions or functions for policy checks.
- The transactions and business objects configured for interception and policy checks.

The following sections describe when policy checks occur for the supported transactions or functions for each Entitlement Pack.

### Entitlement Pack for ECC

The Entitlement Pack for ECC includes support for over three hundred transactions for the most common SAP ECC components, including Material Master (MM), Production Planning (PP), and Document Management System (DMS). For the complete list of ECC transactions supported for interception, see [Transactions](#) on page 277.

For most transactions, this is the typical sequence that results in a policy check:

- 1 The user enters a transaction that has been configured for interception.
- 2 In the initial screen that appears, the user enters a value for a business object, such as Material, Customer, or Vendor, that has been configured for security classification.
- 3 The user finishes entering information and executes the transaction.

[Figure 6-1](#) shows an example of when a policy check is triggered. As a result of the policy check, the user is either denied or permitted access to information about the specified Material.



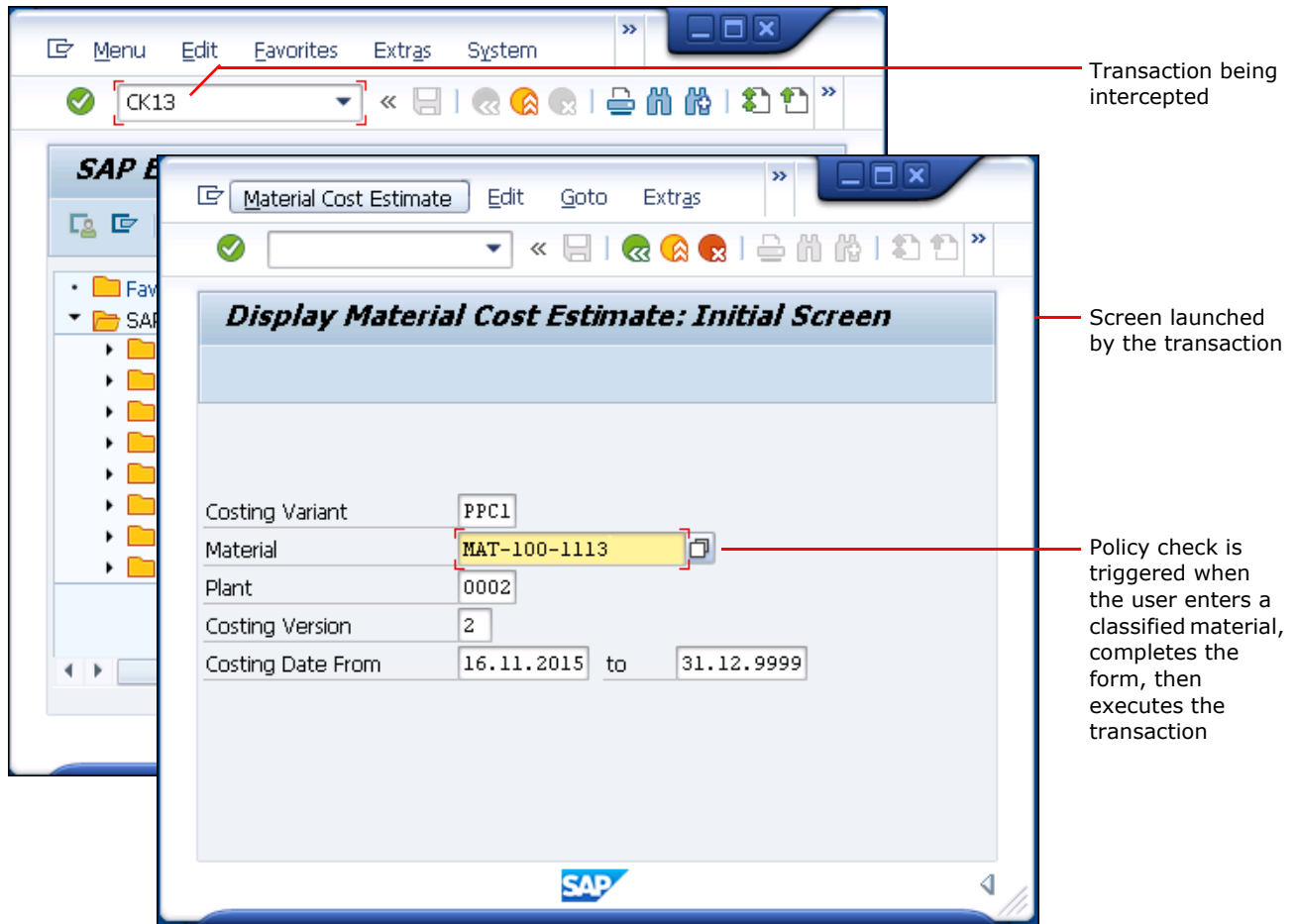


Figure 6-1: Example of intercepted transaction and policy check

### Entitlement Pack for PLM

Table 6-1 lists supported UI functions in SAP PLM, along with specific SAP PLM user events that trigger a policy check.

Table 6-1: Points of Policy Check for Supported SAP PLM UI Functions

PLM UI Functions	Point(s) of Policy Check
Create Material	<p>In the initial screen, when the user:</p> <ul style="list-style-type: none"> <li>Enters a Material name, selects an Industry Sector and Material Type, and clicks Start.</li> <li>Enters a Material in the Copy From Material field.</li> </ul> <p>In the <i>Create Material</i> screen, when the user:</p> <ul style="list-style-type: none"> <li>Attempts to add a Document to the Material.</li> <li>Finishes entering information and clicks Save.</li> </ul>

Table 6-1: Points of Policy Check for Supported SAP PLM UI Functions

PLM UI Functions	Point(s) of Policy Check
<b>Change Material</b>	In the initial screen, when the user selects a Material and clicks Start. In the <i>Change Material</i> screen, when the user: <ul style="list-style-type: none"> <li>• Attempts to delete the Material.</li> <li>• Attempts to add a Document to the Material.</li> <li>• Finishes making changes and clicks Save.</li> </ul>
<b>Display Material</b>	In the initial screen, when the user selects a Material and clicks Start. In the <i>Display Material</i> screen, when the user: <ul style="list-style-type: none"> <li>• Attempts to view a Document associated with the Material.</li> <li>• Selects Edit to change from Read Only to Edit mode. While in Edit mode, all the policy checks associated with Change Material apply.</li> </ul>
<b>Create Document</b>	In the initial screen, when the user selects: <ul style="list-style-type: none"> <li>• Document Type and clicks Start.</li> <li>• Enters an existing Document to Copy From, and clicks Start.</li> </ul> In the <i>Create Document</i> screen, when the user: <ul style="list-style-type: none"> <li>• Attempts to delete the Document.</li> <li>• Finishes making changes and clicks Save.</li> </ul>
<b>Change Document</b>	In the initial screen, when the user selects a Document and clicks Start. In the <i>Change Document</i> screen, when the user: <ul style="list-style-type: none"> <li>• Attempts to delete the Document.</li> <li>• Finishes making changes and clicks Save.</li> </ul>
<b>Display Document</b>	In the initial screen, when the user selects a Document and clicks Start. In the <i>Display Document</i> screen, when the user selects Edit to change from Read Only to Edit mode. While in Edit mode, all the policy checks associated with Change Document apply.
<b>Create Material BOM</b>	In the initial screen, when the user selects Material, Plant, BOM Usage, and clicks Start. In the <i>Create Material BOM</i> screen, when the user: <ul style="list-style-type: none"> <li>• Selects Copy From Existing BOM and supplies an existing BOM.</li> <li>• Adds an existing Material or Document to a BOM.</li> <li>• Finishes making changes and clicks Save.</li> </ul>
<b>Change Material BOM</b>	In the initial screen, when the user clicks selects Material, Plant, BOM Usage, and clicks Start. In the <i>Change Material BOM</i> screen, when the user: <ul style="list-style-type: none"> <li>• Clicks Copy From Existing BOM and selects an existing BOM.</li> <li>• Selects an Item and clicks Display Item Details.</li> <li>• Finishes making changes and clicks Save.</li> </ul>
<b>Display Material BOM</b>	In the initial screen, when the user selects Material, Plant, BOM Usage, and clicks Start. In the <i>Display Material BOM</i> screen, when the user: <ul style="list-style-type: none"> <li>• Attempts to display an Document or Material linked to the BOM.</li> <li>• Selects an Item from the BOM and clicks show Item Details.</li> <li>• Clicks Edit to change from Read Only to Edit mode. While in Edit mode, all the policy checks associated with Change Material BOM apply.</li> </ul>

*Table 6-1: Points of Policy Check for Supported SAP PLM UI Functions*

PLM UI Functions	Point(s) of Policy Check
Create Change Number	In the initial screen, when the user: <ul style="list-style-type: none"> <li>• Enters a new Change Number and clicks Start.</li> <li>• Attempts to create a Change Number based on a Profile or existing Change Number (Template), and clicks Start.</li> </ul> In the <i>Create Change Number</i> screen, when the user: <ul style="list-style-type: none"> <li>• Attempts to link Documents to the Change Number.</li> <li>• Finishes making changes and clicks Save.</li> </ul>
Change Change Number	In the initial screen, when the user: <ul style="list-style-type: none"> <li>• Selects an existing Change Number and clicks Start.</li> </ul> In the <i>Change Change Number</i> screen, when the user: <ul style="list-style-type: none"> <li>• Attempts to delete a Change Number.</li> <li>• Attempts to link Documents to the Change Number.</li> <li>• Finishes making changes and clicks Save.</li> </ul>
Display Change Number	In the initial screen, when the user selects the Change Number and clicks Start. In the <i>Display Change Number</i> screen, when the user: <ul style="list-style-type: none"> <li>• Attempts to display an Object linked to the BOM.</li> <li>• Clicks Edit to change from Read Only to Edit mode. While in Edit mode, all the policy checks associated with Change Change BOM Master.</li> </ul>

## Entitlement Pack for EasyDMS

Table 6-2 lists supported functions in SAP EasyDMS, along with specific user events that will trigger a policy check.

*Table 6-2: Points of Policy Check for EasyDMS Functions*

EasyDMS Functions	Point(s) of Policy Check
View Filtering	When a user first accesses EasyDMS
Display	Depending on configuration, when a user double-clicks a file or presses ENTER when the file is highlighted. When a user right-clicks a document in EasyDMS and attempts to perform the following actions: <ul style="list-style-type: none"> <li>• Display in SAP interface or PLM Web UI</li> <li>• View the file</li> <li>• Cut and/or copy the file</li> </ul> When a user attempts to perform the following action using the toolbar: <ul style="list-style-type: none"> <li>• View File</li> <li>• Print original</li> <li>• Email file</li> </ul>

*Table 6-2: Points of Policy Check for EasyDMS Functions (Continued)*

EasyDMS Functions	Point(s) of Policy Check
Change	<p>Depending on configuration, when a user double-clicks a file or presses ENTER when the file is highlighted.</p> <p>When a user right-clicks a file and attempts to select the following actions:</p> <ul style="list-style-type: none"> <li>• SAP Properties &gt; Technical Details &gt; Change (Document or in PLM Web UI)</li> <li>• SAP Properties &gt; Edit</li> <li>• Change the file in SAP interface or PLM Web UI</li> <li>• Change Status</li> <li>• Edit File</li> <li>• Rename File</li> <li>• Change Description</li> <li>• Create New Version</li> </ul> <p>When a user attempts to perform the following action using the toolbar:</p> <ul style="list-style-type: none"> <li>• Create new version of the file</li> <li>• SAP Properties</li> <li>• Edit File</li> </ul>
Email	When a user right-clicks the file and selects Send File.
Print	When a user clicks Print Original in the toolbar.

## Entitlement Pack for cFolders

[Table 6-3](#) lists supported functions in SAP cFolders, along with specific user events that will trigger a policy check.

*Table 6-3: Points of Policy Check for EasyDMS Functions*

cFolders Functions	Point(s) of Policy Check
Change	<ul style="list-style-type: none"> <li>• When a user changes the description of a document</li> <li>• When a user attempts to upload a document that has been downloaded and modified</li> </ul>
Display	When a user clicks a link to a document and attempts to download or open it
Deletion	When a user selects a document and clicks Delete.
Filter	<ul style="list-style-type: none"> <li>• When View Filter is enabled and users click on a folder and attempts to view the contents; Documents the viewer is not authorized to see will not display</li> <li>• When a user attempts to route a notification to other users in cFolders; If the other users are not authorized to view the document, the notification will be blocked</li> <li>• When a user attempts to view search results, if the user is not authorized to view a document in the search result, it will be filtered out</li> </ul>
Insertion	When a user clicks Create, enters a new description for a document, browses for a file and attaches it, then clicks Save; the policy check occurs upon Save
Copy	When a user selects an existing document, and clicks Copy

## Entitlement Pack for BW

SAP BW is a data consolidation and business intelligence system that provides reporting and analysis of business data across SAP applications and external data sources. The Entitlement Pack for BW enables you to secure classified data accessed through SAP Business Explorer (BEx) Analyzer, a reporting tool that analysts use to work with data in the BW database.

As described in [Implementation Reference for SAP BW](#) on page 305, you can configure access control so that BEx Analyzer displays all objects and restricts access when an object is selected, or filters what objects the user sees in the first place.

In this release, access control is supported for BEx Analyzer only (other BEx tools are not supported), and for these BW objects: InfoArea, InfoProvider, and InfoObject. The following is a brief description of these objects. For more information, see the SAP documentation.

- **InfoObject:** An object used for business analysis. InfoObjects are divided into characteristics and key figures. You can think of InfoObjects as table fields. Examples of characteristics are customers, products, and other attributes that can be measured. Examples of key figures are revenue, quantity, and other numeric measures.
- **InfoProvider:** An object from which queries can be created and run in BEx Analyzer to provide data in a report. In this release, the only InfoProvider objects supported for access control are DataSource Objects (DSOs) and InfoCubes.
- **InfoArea:** An object, akin to a folder, used to organize InfoProvider objects.

[Table 6-4](#) lists the functions, the BW objects that a function can apply to, the actions that will trigger a policy check, and the result of the policy check.

*Table 6-4: Points of Policy Check for BW Functions*

BW Function	BW Object	Point(s) of Policy Check and Resulting Action
ANALYZER_FILTER	InfoArea	When a user clicks Open Query, then clicks the InfoAreas button to display its contents. BEx Analyzer displays only the InfoAreas that the user is authorized to see.
ANALYZER_FILTER	InfoProvider	When displaying a list of InfoProviders. BEx Analyzer displays only the InfoCubes and DSOs that the user is authorized to see.
ANALYZER_FILTER	InfoObject	When a user runs a query to generate a report. The report shows only the data that the user is authorized to see. For example, if the user is not permitted to view Customer123 data, that data is excluded from the report.
ANALYZER_DISPLAY	InfoProvider	When a user tries to access a classified InfoCube or DSO, or run a query created from a classified InfoCube or DSO. The action is allowed if the user is authorized to access the InfoCube or DSO; the action is blocked if the user is not authorized to access the InfoCube or DSO.
ANALYZER_DISPLAY	InfoObject	When BEx Analyzer generates a report (after the user runs a query). The report displays data only if the entire result set contains data the user is authorized to see. If the result set contains any data that user is not authorized to see, the report displays nothing.

---

## About SAP Policies

This section provides information about some policy characteristics that are unique to Dynamic Authorization Management for SAP, including:

- [Policy Based Security Classification](#) on page 186
- [About the SAP Resource String](#) on page 187
- [About SAP Policy Messages](#) on page 187

**Note:** For more detailed information about NextLabs policies in general, see the *Policy Studio User's Guide*. For some example SAP policies, see [Designing SAP Access Control Policies](#) on page 204 and [Designing Integrated Rights Management Policies](#) on page 228.

### Policy Based Security Classification

For Policy Based Security Classification (PBSC), attached originals on a Material or Document record can be used to classify the parent Material or Document. Which originals will be queued (and thus evaluated for possible automatic classification) is determined by PBSC filters configured during setup. Originals included in the filter will be queued on the upload (save) or modification of originals in SAP ECC, SAP PLM, SAP EasyDMS, or SAP cFolders. Policy evaluation then determines how classification records should be updated in a scheduled batch process.

## About the SAP Resource String

When you design SAP policies in Policy Studio, you create an SAP Resource component and define an SAP resource string to identify which applications, transactions/UI functions, and business objects you are trying to target.

The resource string uses the following syntax: `**sap://Server/System/Client/Application/Transaction Code or UI Function Name/Business Object`. Wildcards are supported, so for an example where we want to apply across all Applications, Transactions, and Business Objects, the string would be:

```
**sap://server/system/client/**/*
```

Depending on which Entitlement Packs you have purchased, valid settings for `Application` are:

- `BW` (for SAP BW)
- `ECC` (for SAP ECC)
- `EasyDMS` (for SAP EasyDMS)
- `cFolders` (for SAP cFolders)
- `PLM` (for SAP PLM)
- `*` (wildcard to specify all systems)

## About SAP Policy Messages

There are two options for user alerts that display when a NextLabs policy is enforced: a NextLabs User Alert defined in Policy Studio, or an SAP Message Class that is defined in SAP and referenced in a Policy Studio policy. To reference a message within an SAP Message Class, specify both the class and the message ID number in Policy Studio.

**Note:** For more information on how to add user alerts to policies, see [Designing SAP Access Control Policies](#) on page 204.

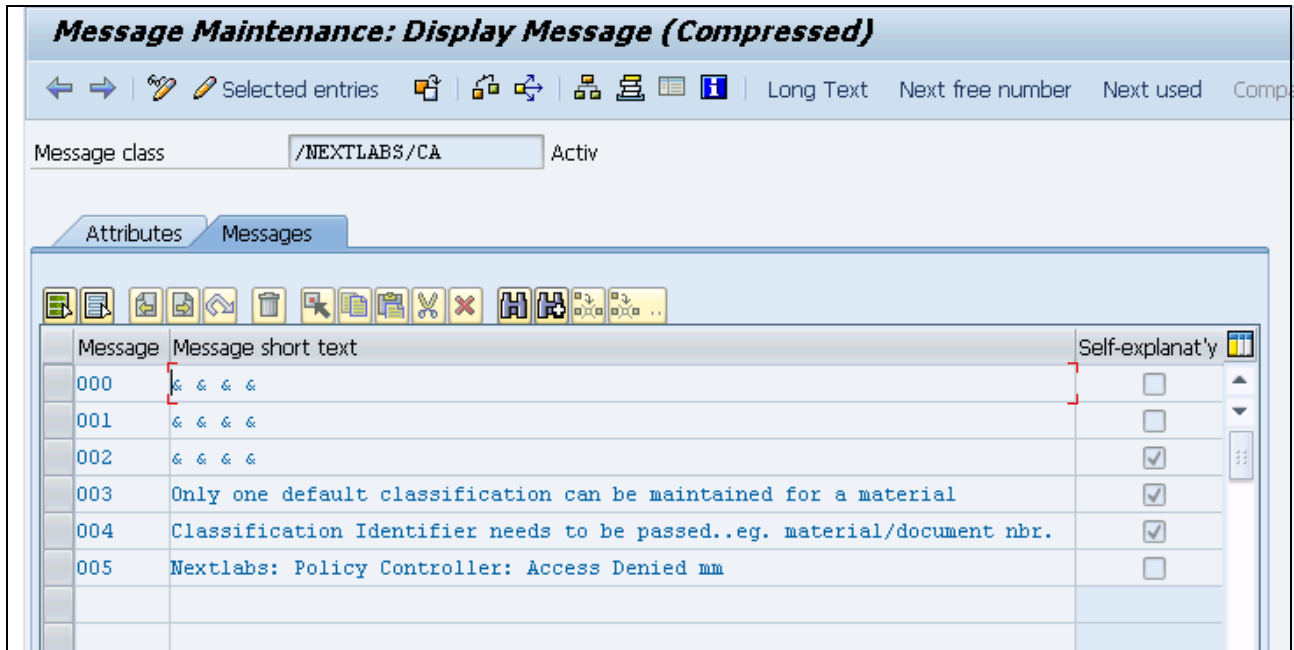


Figure 6-2: SAP Messages in the NextLabs Message Class

## About Custom Obligations

As with any NextLabs product, you can design policies for Dynamic Authorization Management for SAP ECC to automatically trigger a process that occurs upon policy evaluation. With other NextLabs products, this process could be any executable a customer writes.

With Dynamic Authorization Management for SAP, these processes can be written in ABAP code. An example implementation might be to have a certain transaction execute when a user is denied access to a resource.

**Note:** For more information on creating custom obligations for Dynamic Authorization Management for SAP, see [Custom Obligations](#) on page 271.

## Applying Security Classifications

Security Classifications are maintained as rows in the *Security Classification Maintenance* screen, where you classify business objects by Material, by Document, by the versions ([About Compound Classification Keys](#) on page 189) of Materials or Documents, or by another Identifier that you define. The *Security Classification Maintenance* screen can reside within SAP ECC and/or SAP cFolders, or both. This section provides the following information about applying classifications:

- [About Classification Data](#) on page 189
- [About Compound Classification Keys](#) on page 189



- [Applying Security Classifications Manually](#) on page 194
- [Applying Security Classifications Based on Policy](#) on page 200
- [Updating Classifications for Files Exported to cFolders](#) on page 203

## About Classification Data

The identifiers (the column headings) are stored in the table `/NEXTLABS/SECIDT` and the classification values (the values beneath the headings) are stored in the table `/NEXTLABS/SECCLS`. The link between the two tables is the `Reference Id`.

Some columns in the *Security Classification* screen are standard and others can be customized to address a particular need or business practice. For example, if you are using the Entitlement Pack for BW, the columns, INFOAREA and INFOPROVIDER, are standard. However, if you want to apply access control on a particular InfoObject, (for example, customers or materials), you must create and configure the identifier for that object. For more information on how to customize classifications, see [Custom Security Classification Identifiers](#) on page 255.

There are some configuration best practices that may apply depending on how many records you intend to maintain for the Security Classification Maintenance table. For more information about these best practices, see [Recommended Configuration for Implementations with Many Classifications \(more than 40,000 rows\)](#) on page 102.

## About Compound Classification Keys

Businesses typically use SAP to manage multiple versions of business objects. As a document evolves into multiple versions, different versions may require different access controls. For instance, in an export control use case, certain versions of a document may not be subject to export regulations, whereas others are. Dynamic Authorization Management for SAP provides compound classification key support, so you can design policies to reference the version of business objects. In the [Figure 6-3](#), columns in the Security Classification Maintenance table can reference Document Type, Part, and Version, but these values can be configured to represent your business requirements.

When multiple compound classification keys are present, a single record (row) can include classification values for multiple keys. For example, you could configure a classification where for Document DOC-100, Doc Type = DRW, and Doc Version = Internal, Export Compliance should be "ITAR."

**Note:** Compound Keys are supported in the back-end data system SAP ECC only. There is no support in SAP cFolders. For more information on how to reference compound keys in policy, see [Example Policy: Access Control Based on Compound Key](#) on page 213.

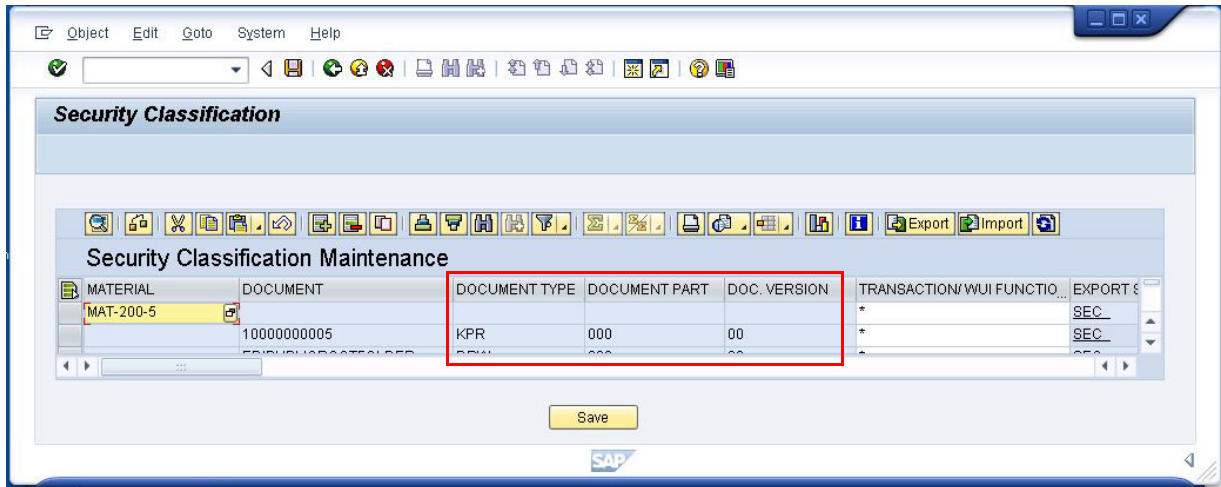


Figure 6-3: Compound Classification Key Support

## Inheritance of Security Classifications and Access Control Contexts

Certain SAP business objects may be associated with one or more other business objects. This raises practical issues for policy designers: how should policies be enforced when multiple classification and context values are present?

In the case where a business object is associated with other business objects, you can configure the evaluation to occur based on either Access Control Context, Security Classification, or both. The inheritance settings determine how parent classifications will be inherited in SAP EasyDMS, SAP ECC, and SAP PLM. This configuration setting controls only the main (parent) business object. The associated (child) business object is always evaluated by Security Classification. The associated object is never evaluated by Access Control Context. For more information, see [Defining How Security Classifications and Access Control Contexts Should Be Applied](#) on page 112 and [Defining How Multiple Security Classifications Should Be Applied](#) on page 116.

## Best Practices for Retrieving Security Classification Records

If you have a large number of records in the *Security Classification Maintenance* screen, it is best to follow search techniques that target data you want to retrieve in the most efficient manner. The following tips ensure that your searches are optimized and return results quickly and without error:

- For the best results, always specify a Security Identifier and/or Composite Key for a search, then supply additional search items (such as classification fields). A search based on classification values only may result in too many records or slow search returns.
- If you want to retrieve Classification values for a set of Security Identifiers or Composite Keys, you can use a partial wildcard to specify the set. In other words, you can search on MAT\* and Classification Value ITAR to retrieve a list of all material records classified as ITAR.

- If you want to run a search based on classification fields (all ITAR records, regardless of Security Identifier and/or Composite Key), it is recommended that you limit search records using another search criteria, such as date range or number of records.

## Viewing Security Classification Records

You can use the *Security Classification Maintenance Selection* screen to determine which records to view.

### Procedure

- 1 In the SAP interface, enter transaction `/nextlabs/sec_cls`. The *Security Classification Selection* screen appears.
- 2 Do any of the following:
  - Enter selection criteria to specify which records you want to view.
  - Create a new security record (in which case, you do not need to define selection criteria).
  - Change the selection options you use to specify records you want to view.

Dynamic selections			
Material			➔
Customer		to	➔
Document		to	➔
Document Type		to	➔
Document Part		to	➔
Document version		to	➔
Maximum No. of Hits	200		

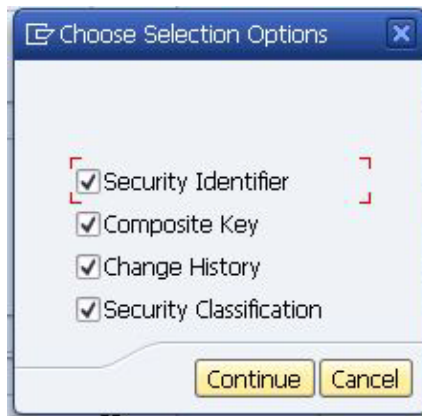
Figure 6-4: Security Classification Maintenance Selection Screen: Change Selection Options

- 3 If desired, click **Change Selection Options** to view different selection options other than the ones displayed. This determines the options you can use in the *Security Classification Maintenance Selection* screen.

The choices are currently Security Identifier, Composite Key, Change History, and Security Classification.

**Note:** The selection options that display by default are configured in the CONCFG table. For more information, see [Configuring SAP Data Handling and Connection Settings](#) on page 94.

4 Click Continue.



*Figure 6-5: Changing Selection Options*

5 Specify the range of records you wish to view, by Material, Document, or other property.

**Note:** In [Figure 6-6](#), because we selected additional selection options, we now have more options to use for selection. A warning message might display if a user enters selection options that retrieve a large number of records. The number of records that trigger this warning is configured in CONCFG. For more information, see [Configuring SAP Data Handling and Connection Settings](#) on page 94.

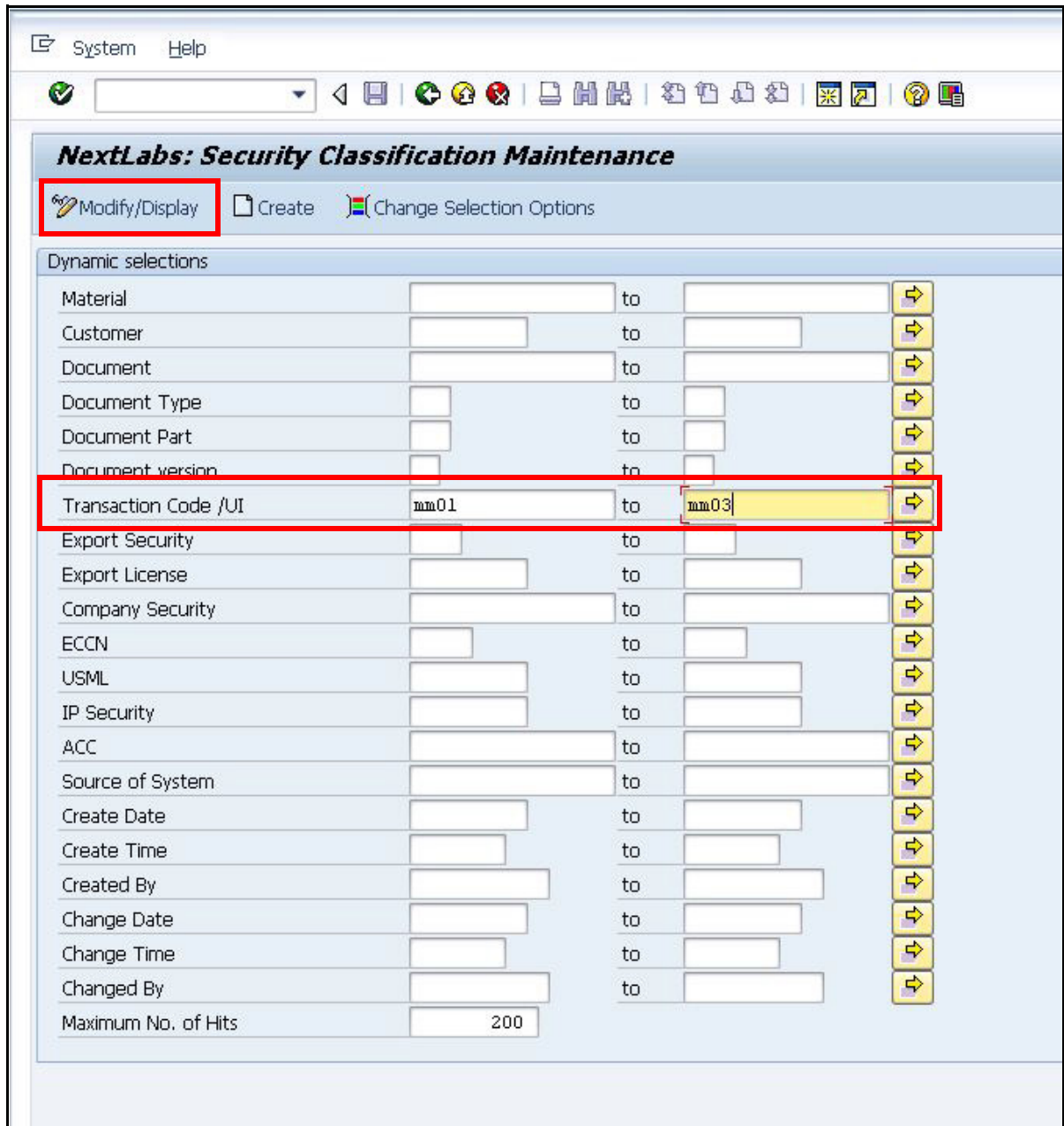


Figure 6-6: Additional Options in the Selection Screen

6 Click **Modify/Display**. The classification records that match your selection criteria display.

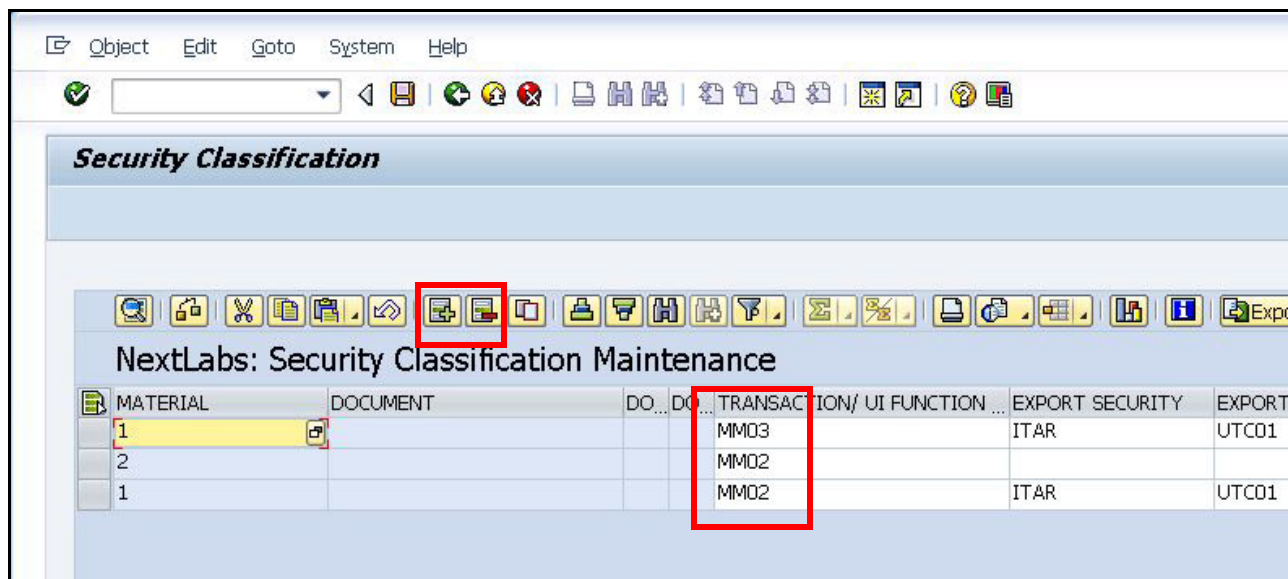


Figure 6-7: Classification Values Returned Based on Selection Options

## Applying Security Classifications Manually

In the *Security Classification Maintenance* screen, you can enter new rows to assign new classifications, or upload multiple rows at once by importing them from an Excel spreadsheet. Note the flexibility of how classifications can be manually assigned: they can be assigned by specific transaction code or for all transactions at once. You can also use partial wildcards to apply classifications to a whole family of materials or documents at once.

**Note:** There are best practices around using wildcards for customer implementations that include a large number of security classification records. For more information, see [Recommended Configuration for Implementations with Many Classifications \(more than 40,000 rows\)](#) on page 102.

You can also add, delete, and modify classifications. Values in these columns are mapped to NextLabs Policy Controllers. They can be referenced in policies you design in Policy Studio.

If your implementation includes both the Entitlement Pack for ECC and cFolders, you can enter classification values for ECC and cFolders in the same table, if the two SAP modules reside on the same system. If they are hosted on separate systems, you access the *Security Classification Maintenance* table in the two different systems. Figure 6-8 demonstrates how the *Security Classification Maintenance* table appears if the Entitlement Packs for SAP ECC and cFolders are configured on the same system.

SAP ECC Identifiers      SAP cFolders Identifiers      Compound Keys      Classification

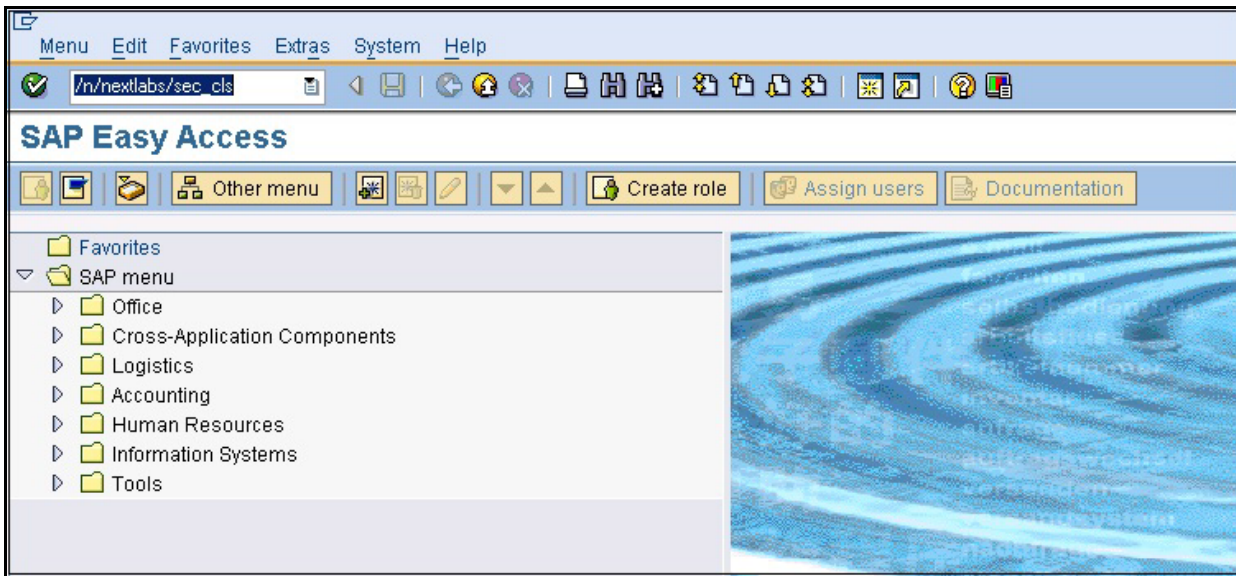
MATERIAL	DOCUMENT	DOC ID/DESC	MAT ID/DESC	DOCU_ DO...	DO...	TRANSACTION/ UI FUNCTION.	EXPORT SECURITY
			Export_cFolder_TEST_MAT-112			*	EAR
			TEST_MAT-112 TEST material f...			*	EAR
	10000000107		TEST_MAT-112 TEST material f...	DRW	000 12	*	ITAR
		Content Analysis				*	ITAR
		demo1				*	ITAR
		Export_cFolders_70				*	EAR
	10000000070			DRW	000 01	*	
	10000000070			DRW	000 02	*	ITAR
		Content Doc				*	ITAR
		TestDoc				*	ITAR
		Export_cFolders_30				*	EAR
		DRW_0000000000000000100...				*	EAR
		DRW_0000000000000000100...				*	
	10000000030			DRW	000 05	*	EAR
	10000000079			DRW	000 01	*	ITAR
	10000000079			DRW	000 02	*	ITAR

Figure 6-8: Security Classification Maintenance Screen

**Procedure**

- 1 In the SAP interface, enter transaction /nextlabs/sec\_cls.





*Figure 6-9: Access the Security Classification Screen*

The *Security Classification Selection* screen appears.

- 2 Since we are creating new records, click **Create** without specifying any selection criteria.



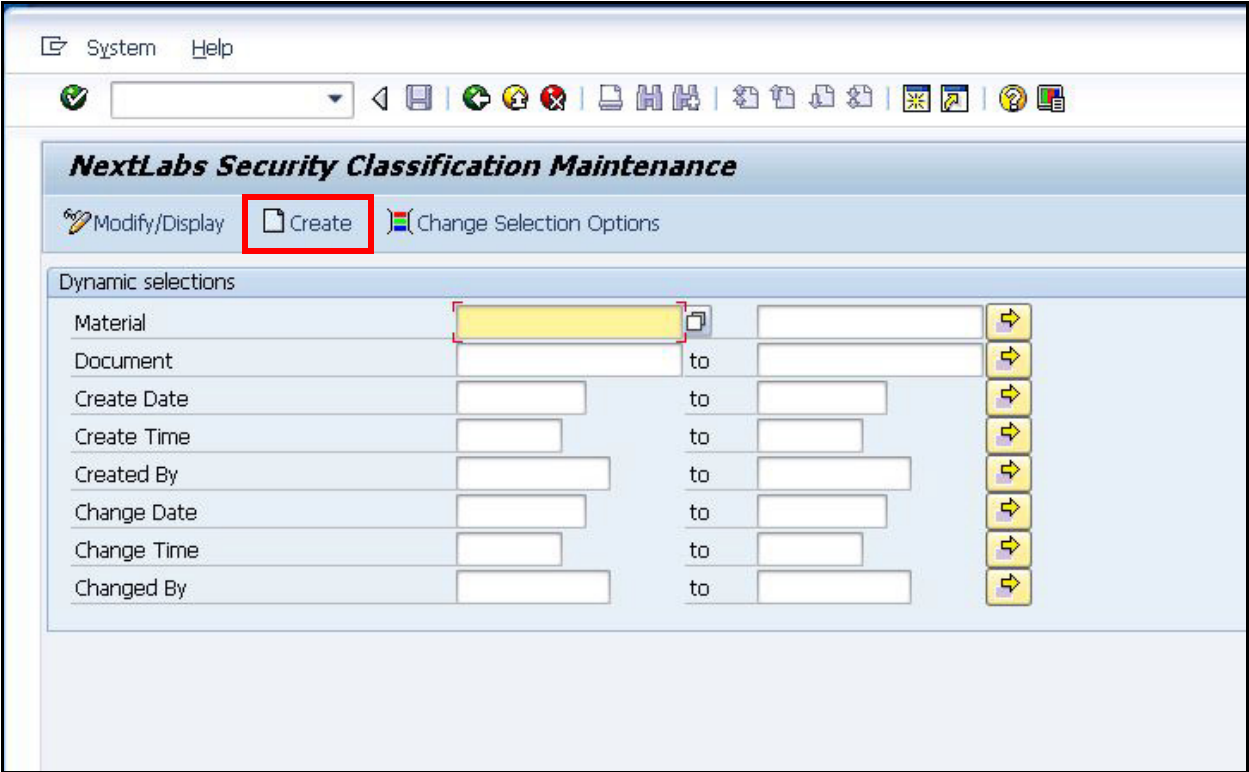


Figure 6-10: Security Classification Maintenance Selection Screen: Create

The Security Classification screen displays no records.

3 Click the Insert Row Icon in the toolbar to add a new Security Classification record.

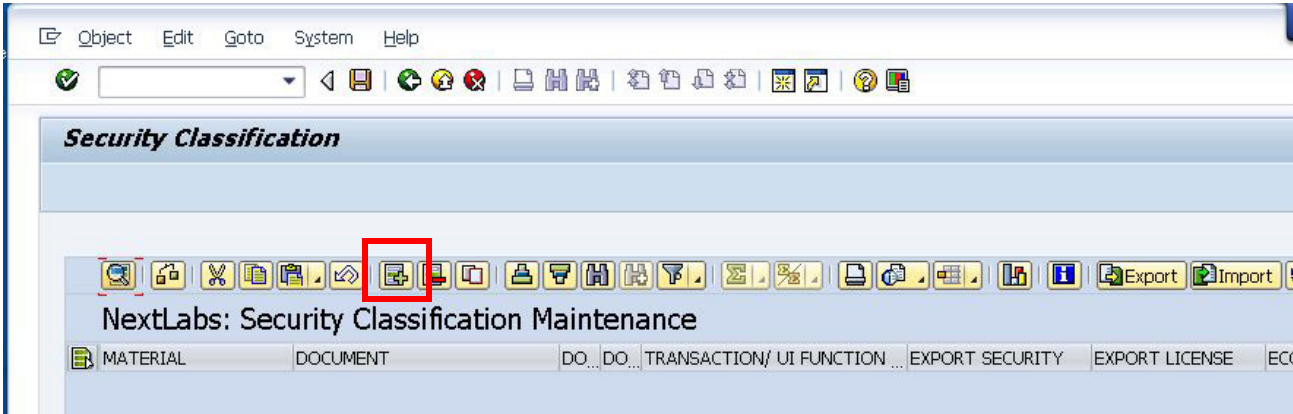


Figure 6-11: Inserting a row

- 4 In the example shown in the previous figure, for each row, you can create a security classification for *either* a Material *or* a Document, *either* for SAP ECC *or* cFolders data. Enter the Material *or* Document number, or retrieve a valid number by clicking the pop up icon that appears on the right side of the field when it is active.

**Note:** While you add a row to classify only one Material or Document at a time, multiple rows might apply to a single transaction: Materials can be the children of Documents, BOMs typically include multiple records, and Engineering Workbenches (CEWB) can include multiple Materials, Documents, BOMs, and Routings (which include multiple Materials). Part of system configuration is to define how the Security Classifications should be enforced when multiple Identifiers are present for a given transaction. See [Defining How Multiple Security Classifications Should Be Applied](#) on page 116.

- 5 In the Transaction column, enter a Material or Document transaction or UI Function code. For example, you can apply a classification to the viewing of the Material (transaction code MM03), or to the modification of a Document (CV02N). You can enter a wildcard (\*) here to apply a classification to all transactions at once.
- 6 In the Export Security column, enter the Export Security for the Material or Document. In our example, we classify material N-100-1 as ITAR data.
- 7 If there are Export Licenses applicable to the Material or Document, enter them in the Export Licence column. This is a multi-value field. Double-click it to access the pop-up window where you can enter multiple licenses, if applicable.

**Note:** Defining Export Licenses is another way to fine-tune access controls. For example, you can restrict access to ITAR data to only authorized users, which can be users who are US citizens, or users in countries where Export Licenses apply.

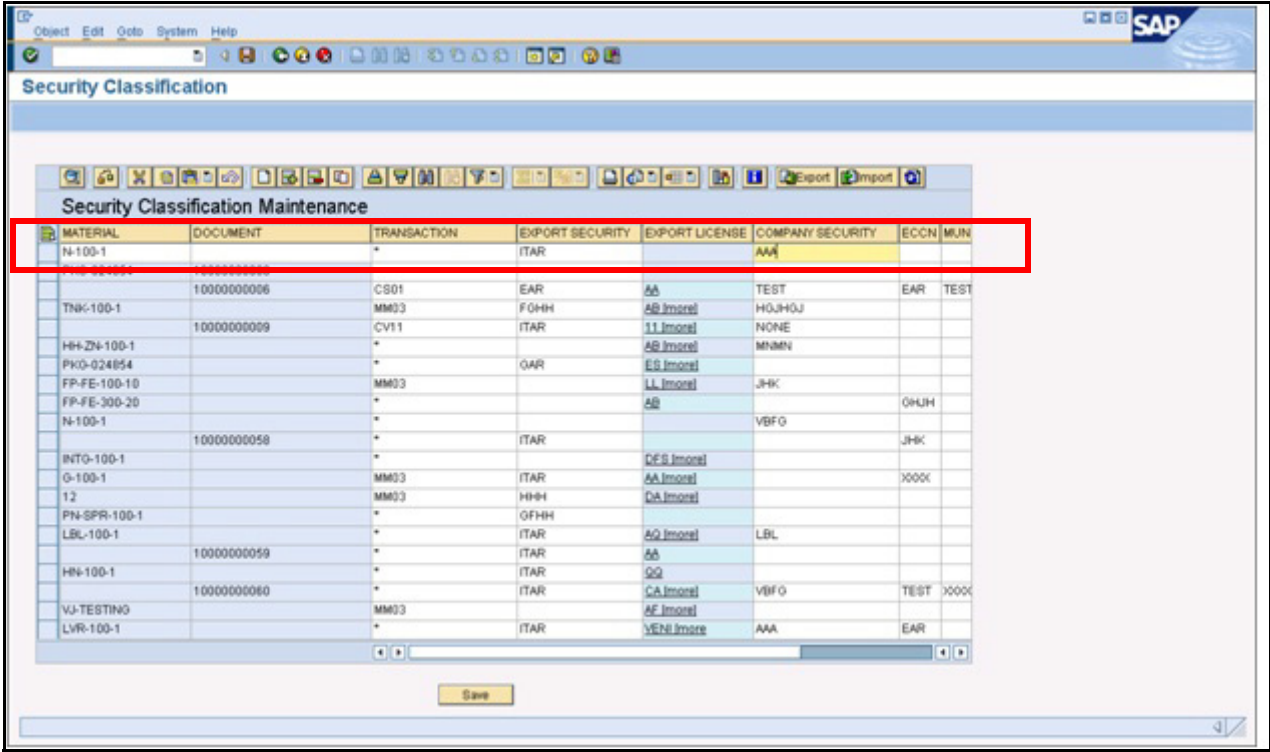


Figure 6-12: Applying Classifications (SAP ECC only in this example)

- 8 Enter pertinent classification information in the remaining columns, as needed. Remember that the columns that display here depend on how you have defined Identifiers during your system’s configuration. When you are finished, click Save.

## Applying Security Classifications Based on Policy

You can also add classification values to the Security Classification Maintenance table automatically based on originals that are associated with Material or Document records. PBSC can collect originals and update classifications anytime documents are saved (attached the first time and/or edited) within SAP ECC, SAP PLM, SAP EasyDMS, and SAP cFolders.

PBSC policies can evaluate these originals in a variety of ways to determine how classifications should be applied: based on existing classifications (tags) that are present in the file, based on file characteristics (type, location, creator, and so on), and/or based on content analysis (the presence of certain keywords within the file).

All PBSC policies have the following requirements:

- Allow policies
- Use the Save action
- Include the custom obligation “SAP Set Classification Value”

### Example Policy: Security Classification Based on Content Analysis

This example policy performs content analysis on originals, and based on the presence of the keyword (ITAR), adds new entries to the Security Classification Maintenance table. The policy is as follows:

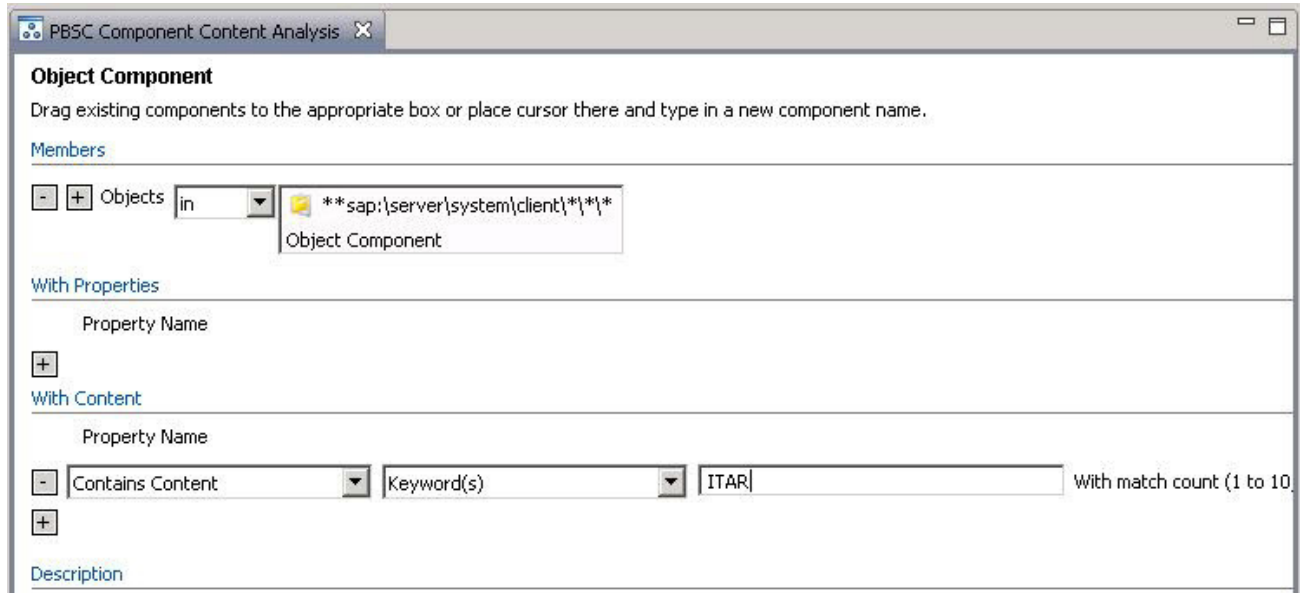
*For SAP Documents with keyword “ITAR,” within SAP ECC, SAP PLM, SAP EasyDMS, or SAP cFolders on Save, Run Obligation: SAP Set Classification Value “EXPORT SECURITY” to “ITAR.”*

**Note:** For more thorough instructions on the general procedure for creating a policy in Policy Studio, see [Example Policy: Access Control Based on Classification and ACC](#) on page 205. What follows is a high-level example meant to illustrate the unique features of PBSC policies.

Perform these general steps:

- Create a Document policy. Set the enforcement type to Allow.
- For this example use case, we want the classification to occur no matter who the user is, so we do not specify a Subject component.
- Select the “Save” Action and drag it into the Action field. This is the only valid action for this type of policy.
- Create an SAP Resource component that includes SAP ECC, SAP PLM, SAP EasyDMS, and SAP cFolders as applications. The resource string uses the following syntax: `**sap://Server/System/Client/Application/Transaction Code or UI Function Name/Business Object`. For our example, we want to apply across all Applications, Transactions, and Business Objects, so our string is: `**sap://server/system/client/**/*`

In the With Content drop-down menu, select Contains Content, Keywords, ITAR.



*Figure 6-13: SAP Resource Component for PBSC Content Analysis Policy*

- Drag the resource component into the policy.
- Select the **SAP Set Classification Value** custom obligation. For the field, select Export Security, enter the value “ITAR,” and select the mode of update (either Append existing classifications, or Overwrite existing ones).

**Note:** If you design a policy with two Overwrite classifications enforcing for the same record, preference is given via alphabetical order of the classification value (EAR displays rather than ITAR). If you write a policy where an Append classification is being written to a field with single cardinality, the existing value will be overwritten. If two values are being written to the same record from within the same policy, where one is Append, and the other is Overwrite, Overwrite will take precedence.

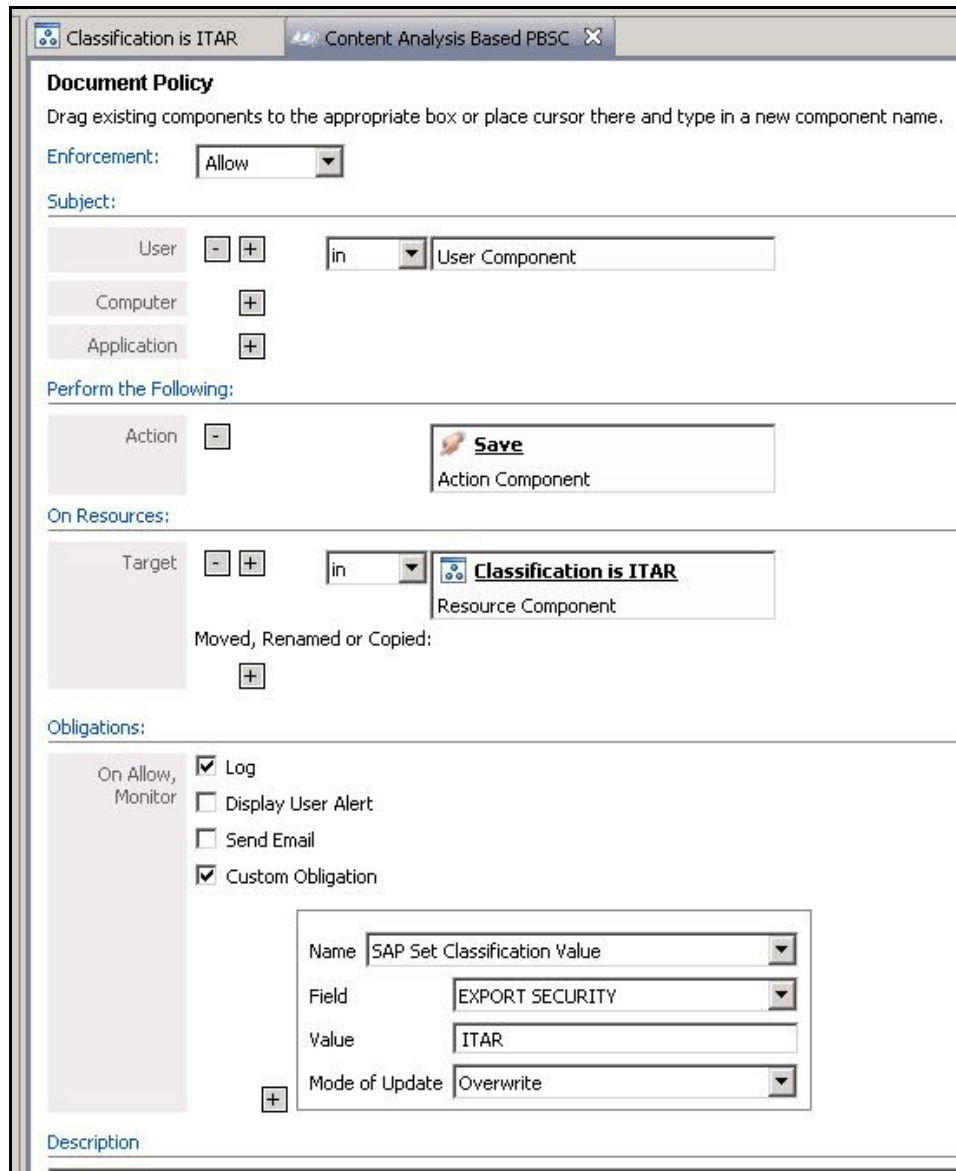


Figure 6-14: Example PBSC Content Analysis Policy

After this policy is deployed, when any user uploads a document containing the keyword "ITAR," a new classification record (row) for the parent Material or Document record will be added to the Security Classification Maintenance table, with the column "Export Security" set to "ITAR."

## Updating Classifications for Files Exported to cFolders

As is discussed in more detail in the section [About SAP ECC and cFolders Integration](#) on page 22, classification values applied to originals in SAP ECC can persist when these files are exported to SAP cFolders. This behavior is automatic and requires no policies or configuration. However, it may be the case that classification settings in SAP ECC change after export (for instance, because a user updates the file with content that triggers PBSC to reclassify it). In this case, the exported document in cFolders would not have the most up-to-date classification. To address this scenario, you can use the transaction `/NEXTLABS/UPDCLS` to send the new classification values to files exported to cFolders.

### Procedure

- 1 In the SAP interface, enter transaction `/NEXTLABS/UPDCLS`.
- 2 Optionally, enter the RFC Destination. This is only necessary if the record of the destination (which was captured during the initial export of the files) is wrong. The recorded destination would be wrong if the cFolders location had been changed since the initial export event.
- 3 Click **Execute**.

The program will detect any new classification values for exported files, and send those values to the Security Classification Maintenance table in cFolders. You will be prompted to check the log files in `SLG1`.

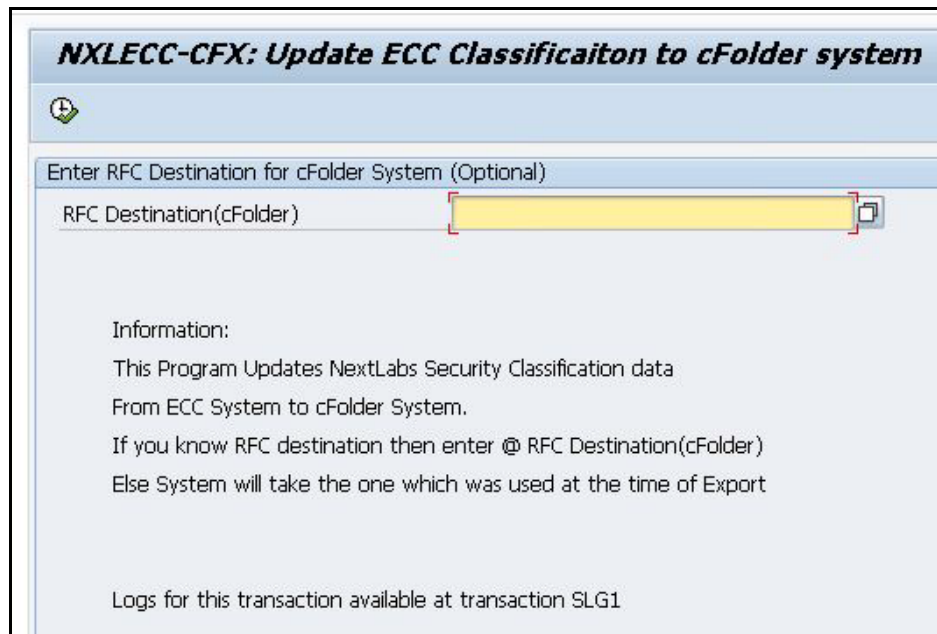


Figure 6-15: Updating Classification Values for Documents Exported to cFolders

---

## Designing SAP Access Control Policies

After you have applied Security Classifications to files, you can reference these values in access control policies. Recall that access control can be based off values in the Security Classification Maintenance table, as well as your PLM Access Control Context schema. Access control policies can also be applied to Compound Keys (versions) for documents and materials. Another alternative is to reference resource attribute values in access control policies.

This section provides the following example access control policies:

- [Example Policy: Access Control Based on Classification and ACC](#) on page 205
- [Example Policy: Access Control Based on Compound Key](#) on page 213
- [Example Policy: Access Control Based on Resource Attributes](#) on page 215
- [Example Policy: View Filtering \(EasyDMS Only\)](#) on page 220



## Example Policy: Access Control Based on Classification and ACC

This section provides an example policy that restricts access to business objects based on both the Security Classification and Access Control Context. The basic structure of the policy is:

*Allow Only US Citizens to Run PLM transactions on Business Objects with security classification "ITAR" and ACC "Project Destiny."*

### Procedure

#### Create the Policy

- 1 In the policy tree, select the folder in which you want the new policy to be stored, then click the New Policy button. (All policies must be saved in some folder.)

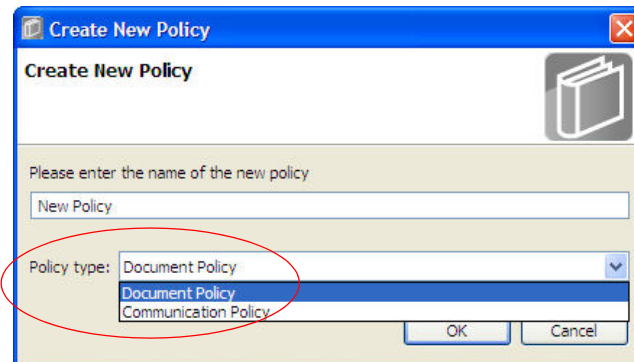


Figure 6-16: Selecting a Policy Type

- 2 In the dialog box, type a name for the new policy. In the Policy type drop down menu, select **Document Policy**. All policies for Entitlement Manager for SAP are Document policies.

#### Define the Policy Effect

- 3 In the Enforcement field at the top of the Subject area, specify what Dynamic Authorization Management for SAP does if the event described in the policy occurs:
  - **Deny:** Do not permit specified users to perform the action, but allow all others to do so.
  - **Allow:** Permit the specified users to perform the action. Allowing a set of users to perform an action does not mean that others are blocked from performing that action.

In this policy, select Deny.

#### Define the Policy Subject: Users

- 4 Click the **User Component** tab in the *Resources* pane to access the list of User Components. For our policy, we click **New** to define a New user component specifically for ITAR-restricted data in SAP. In [Figure 6-17](#), we select Users in the group SAP, and then specify the

Property: “Country Name = US.” Select the **not in** operator. We want to deny users who are not in this group the capability to perform the action.

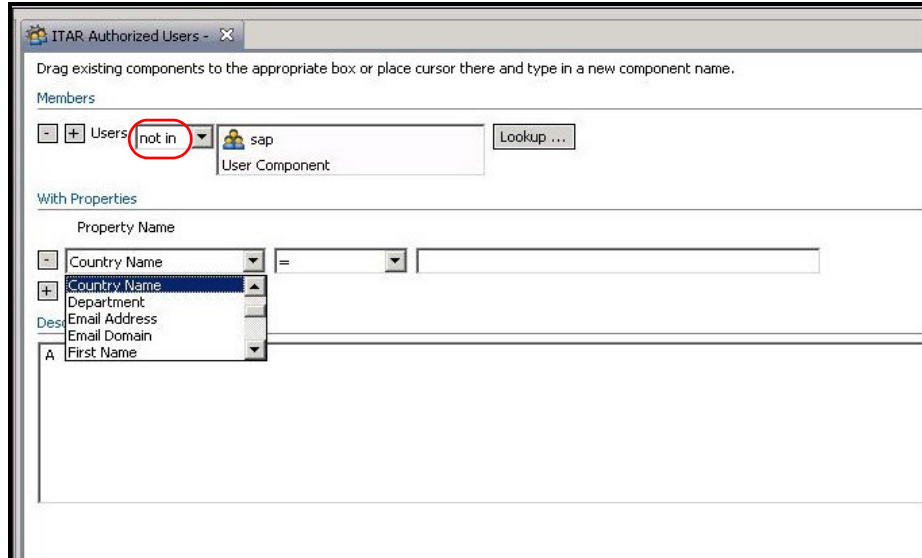


Figure 6-17: Creating a New User Component

- 5 After the new User component is created, select it from the list of components and drag it into the User policy field (as shown in [Figure 6-18](#)).

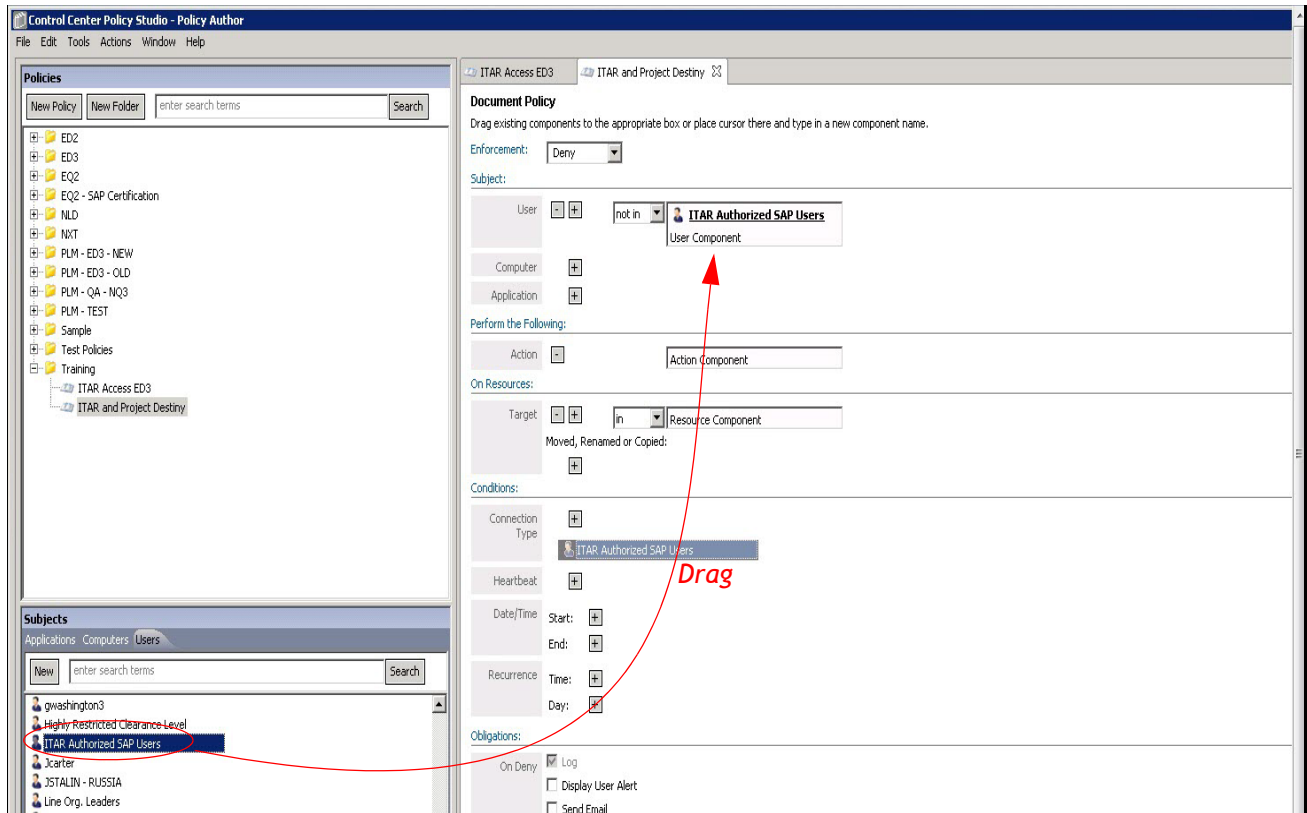


Figure 6-18: Adding a User Component

### Define the Policy Action

- Expand the Actions component panel and drag a Run action component into the policy.

**Note:** Based on the configuration we performed in [Mapping Transaction Codes and UI Functions to Actions \(ACTIONS\)](#) on page 90, other valid actions besides “Run” could be “Delete” and “SAP Copy From.”

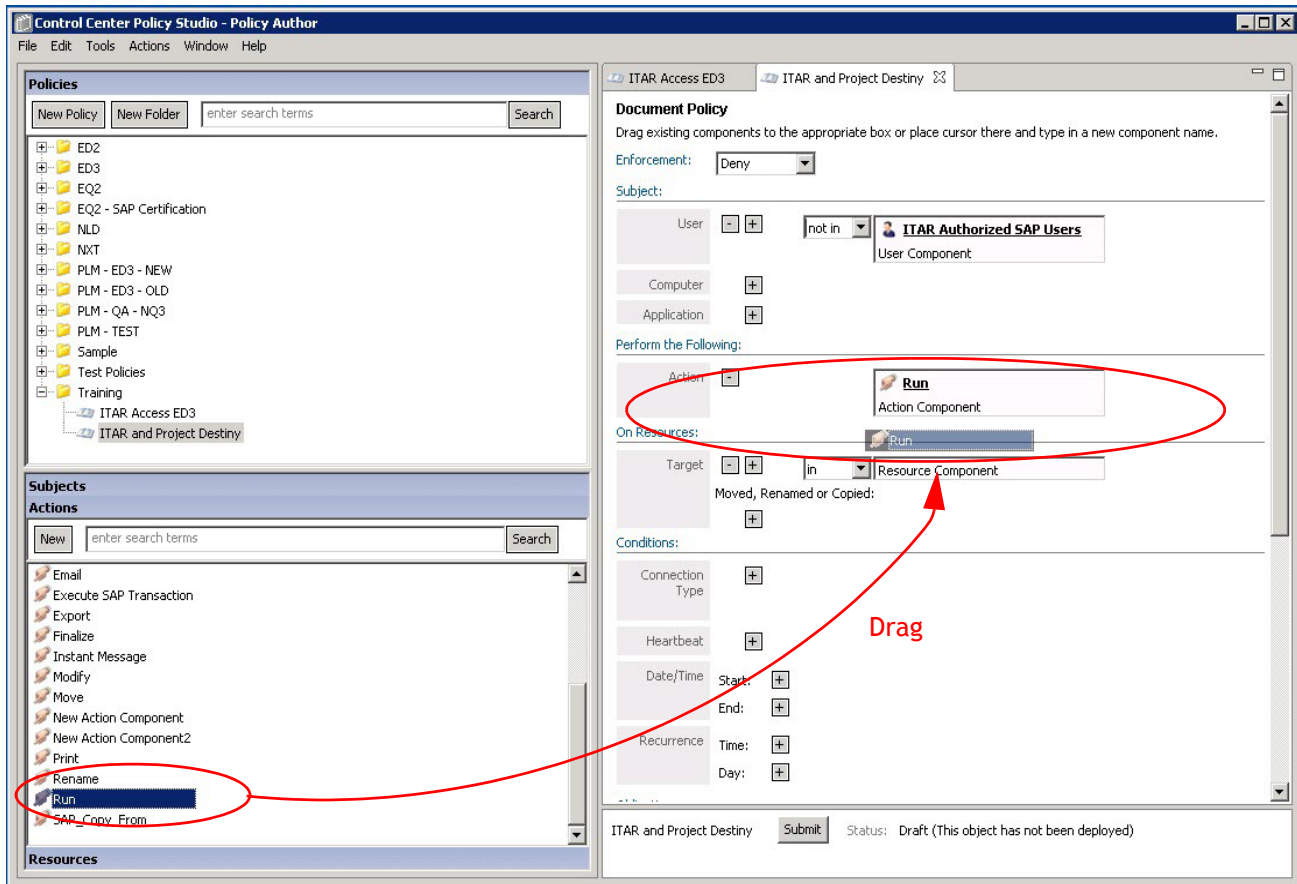


Figure 6-19: Adding an Action to a Policy

## Define the Policy Resource

7 In the *Resources* pane, click the **SAP** tab. This is where you can create Resource components that reference business objects, SAP ECC transactions, and SAP PLM UI functions. In our example, we are creating a Resource that should: refer to all PLM web UI functions, for all object classified "ITAR" and associated with the ACC "Project Destiny." Follow these steps:

- Click **New** to create a new SAP Resource.
- In the new Resource component, define the path to the SAP object using the following syntax: `**sap://Server/System/Client/Application/Transaction Code or UI Function Name/Business Object`.

In our example, we enter the path `**sap://Server1/System200/*/PLM /*/*`. This resource will apply to all UI Functions and all Business Objects accessed through SAP PLM.

Valid settings for Application are:

- BW (for SAP BW)

- ECC (for SAP ECC)
  - EasyDMS (for SAP EasyDMS)
  - cFolders (for SAP cFolders)
  - PLM (for SAP PLM)
  - \* (wildcard for all systems)
- To reference an Access Control Context (ACC) within SAP PLM, add an additional property line. Click the plus button (+) and select (or type) “Access Control Context.” Select “is” and select (or type) “Project Destiny.”

Based on the configuration of our system, we are passing Access Control Context values, not full hierarchy paths. To pass a full path, enter it using forward slashes, for example: /Nextlabs\_Root/Engineering/ProjectA. Wildcards can be used to designate all children at a given level of the hierarchy, for example: /Nextlabs\_Root/Engineering/\*.

**Note:** For more information on configuring how ACC values are passed to the Policy Controller, see [Configuring Access Control Context Settings \(PLM Only\)](#) on page 121.

- To reference a classification in the Security Classification table in SAP ECC, add an additional property line. Click the plus button (+), and select (or type) “Export Control.” Select “is” and select (or type) “ITAR.” (These correspond with an Identifier and Value defined in the Security Classification table.)
- Enter a Description of the Resource component and click **Save**. It will now display in the SAP Resource components tab.

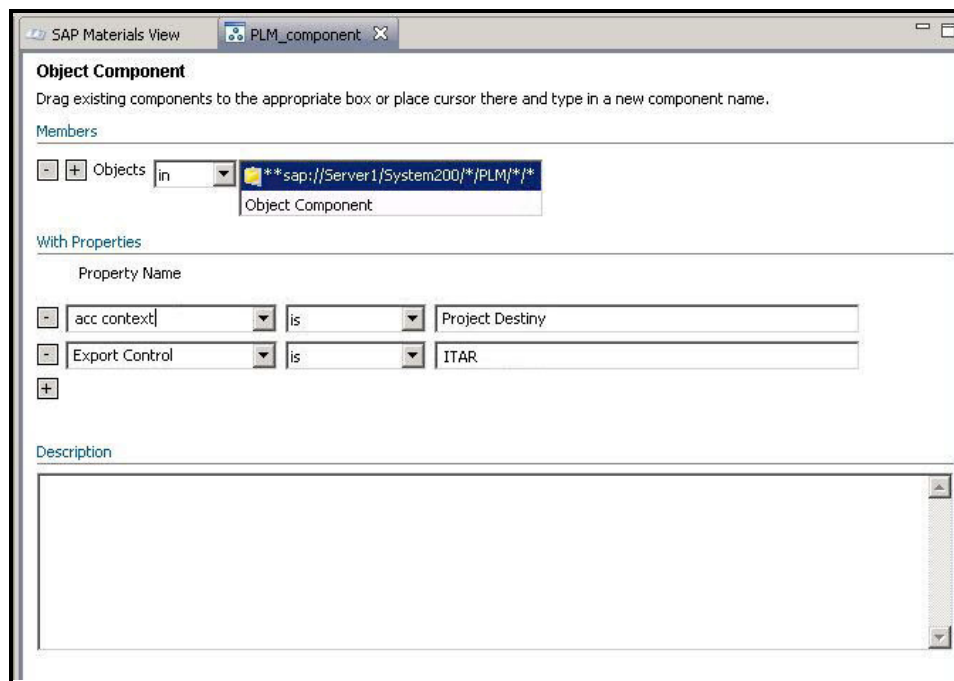


Figure 6-20: Creating a New SAP Resource

## 8 Drag the new SAP Resource component into the Policy.

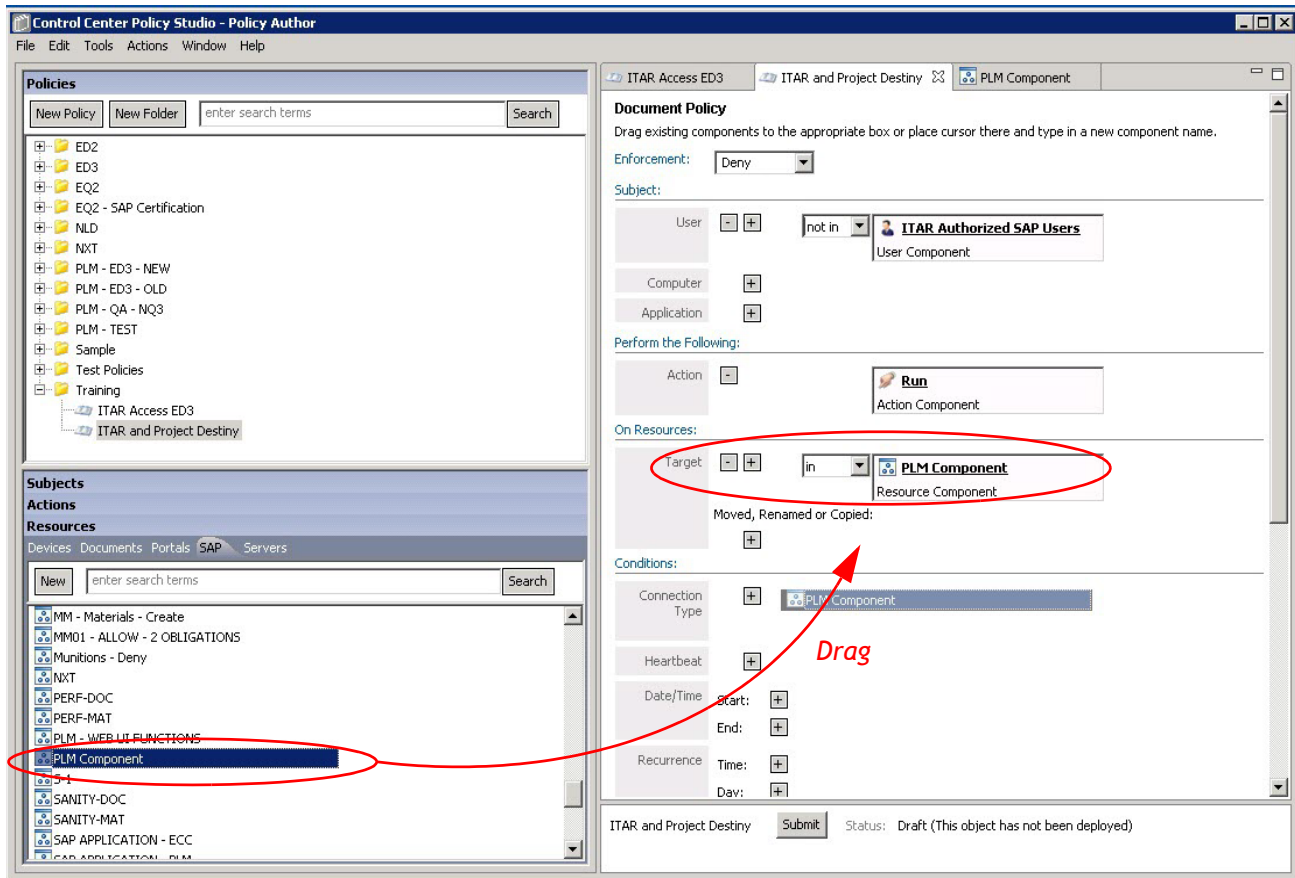


Figure 6-21: Adding a New SAP Resource to a Policy

## Define the Policy Obligations

9 In the Obligations section, define a user message. This is the notification that is shown to users who try to open or use the protected data but are denied.

- Select the custom obligation **SAP User Message** to enter user alert content in Policy Studio. You can select a language from the Language Code drop down menu if you want the message to display in a language other than English.
- Select the custom obligation **SAP Message Class Display** to use a message that is maintained in a message class in SAP. You must specify the message class and message number.

**Note:** Obligations are configured as part of initial installation. See [Configuring SAP Obligations](#) on page 70.

The screenshot shows the 'Obligations' configuration window. It is divided into two main sections: 'On Deny' and 'On Allow, Monitor'.  
 In the 'On Deny' section, the following options are checked: 'Log', 'Custom Obligation'. 'Display User Alert' and 'Send Email' are unchecked.  
 In the 'On Allow, Monitor' section, the following options are checked: 'Log'. 'Display User Alert', 'Send Email', and 'Custom Obligation' are unchecked.  
 A '+ ' button is located between the two sections.  
 A 'MessageText' dropdown menu is open, showing a list of language codes: 'EN:English' (highlighted), 'DE:German', 'ZH:Chinese', 'ZF:Chinese trad.', and 'SR:Serbian'.  
 The 'Name' field is set to 'SAP User Message' and the 'Language Code' is set to 'EN:English'.

Figure 6-22: SAP User Message with Language Code drop down

### Submit and Deploy the Policy

- 10 Provide a description of this policy's UI function in the Description field. This will help other designers understand the purpose of this policy.
- 11 Click **Submit** to save your changes to this policy.
- 12 Deploy the policy, which simply means distributing so that it will be enforced. To do this, click the **Deploy** button in the bottom Status bar. This will open the *Deploy* screen.

Specify the date and time you want to deploy the policy, or select the Deploy Immediately radio button, and click OK. When a policy is deployed, all the components included in its definition are deployed as well.

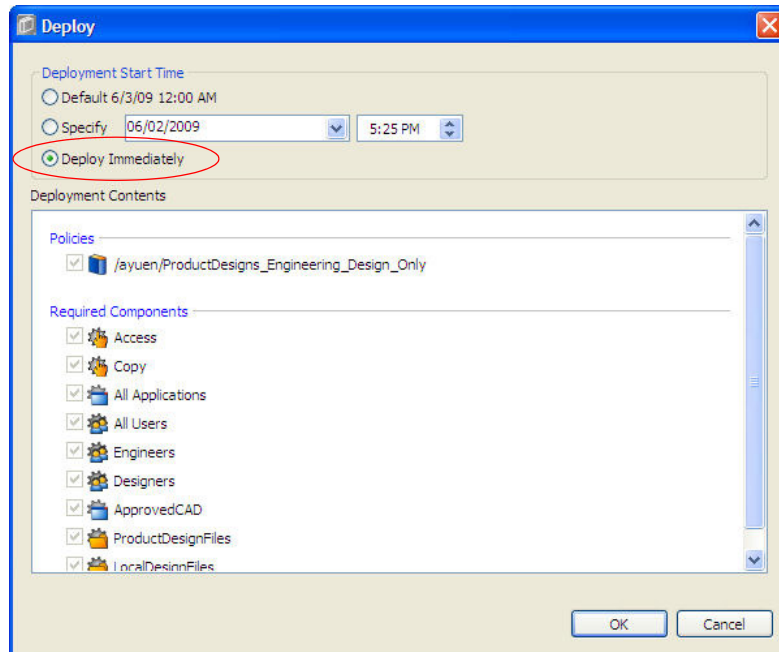


Figure 6-23: Deploying Policy Objects

At the point the policy becomes active, Entitlement Manager for SAP will start enforcing it against user actions on host PCs and laptops. In our example policy, if unauthorized users attempt to access data tagged as “ITAR” and “Project Antares” from within SAP PLM, they are denied access and receive an “Access Denied” message. When authorized users access this data, they receive an educational message associated with the policy.



## Example Policy: Access Control Based on Compound Key

You can define policies that allow or deny access to business objects based on how Compound Keys are classified in the Security Classification Maintenance table. (See [About Compound Classification Keys on page 189](#).)

A common example would be where an organization has a certain type of document that typically includes sensitive or controlled information, such as engineering drawings. You could design a policy that checks the security classification for all documents of type "DRW," but not for other types of documents which require no (or require different) access controls. Another use case might be where an organization maintains different versions of the same document, localized for different countries, each of which is subject to different license restrictions.

The following policy applies to the first use case: *For Documents of Type DRW, where Classification is "ITAR," deny users not in US and Engineering team to access files.*

**Note:** For more thorough instructions on the general procedure for creating a policy in Policy Studio, see [Example Policy: Access Control Based on Classification and ACC on page 205](#). What follows is a high-level example meant to illustrate the unique features of Compound Key policies.

### Procedure

- 1 Create a Document policy and set the enforcement to Deny.

For this example use case, our subject contains two attributes: citizenship is US, and team membership is Engineering. We drag a user component for each in an "And" relationship, and select "not in" for both.

- 2 Select the "Run" Action and drag it into the Action field.
- 3 Create an SAP Resource component that includes SAP ECC, SAP PLM, and SAP EasyDMS as applications. The resource string uses the following syntax: `**sap://Server/System/Client/Application/Transaction Code or UI Function Name/Business Object`. For our example, we want to apply across all Applications, Transactions/UI Functions, and Business Objects, so our string is: `**sap://server/system/client/**/*`
- 4 In the With Properties drop-down menu, enter "DOKAR" (for document type) and "DRW" (for drawing). To reference other compound keys, you would enter "DOKTL" for document part, or "DOKVR" for document version. (See [Adding Composite Keys and Classification Values on page 78](#) for more information on how to configure these values.)
- 5 Click "+" to add another property attribute. Enter "Export Control" is "ITAR."

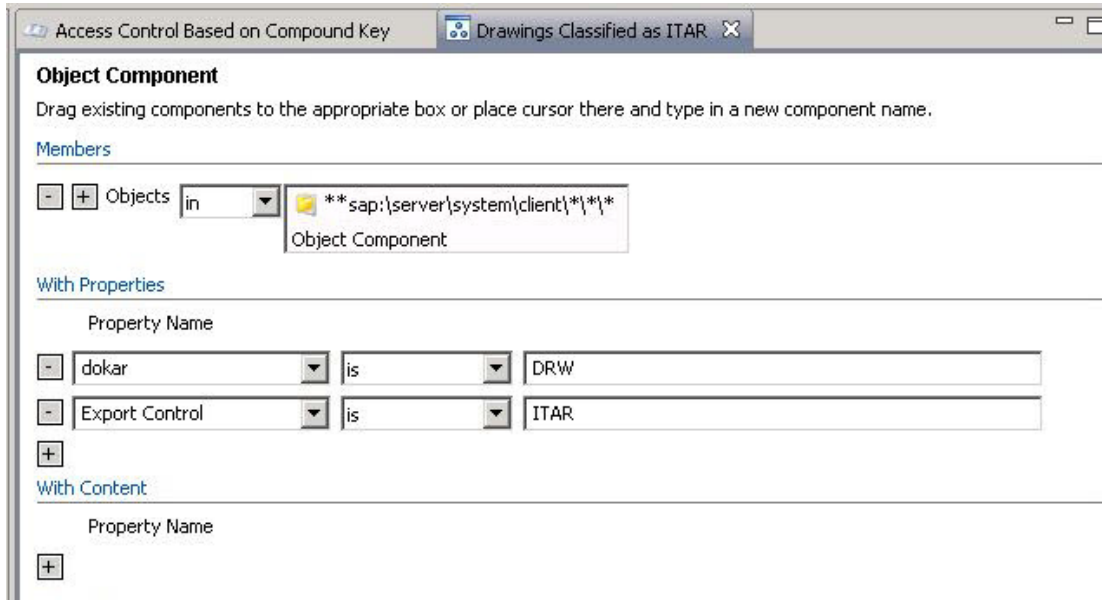


Figure 6-24: SAP Resource Component for Compound Key Policy

6 Drag the resource component into the policy.

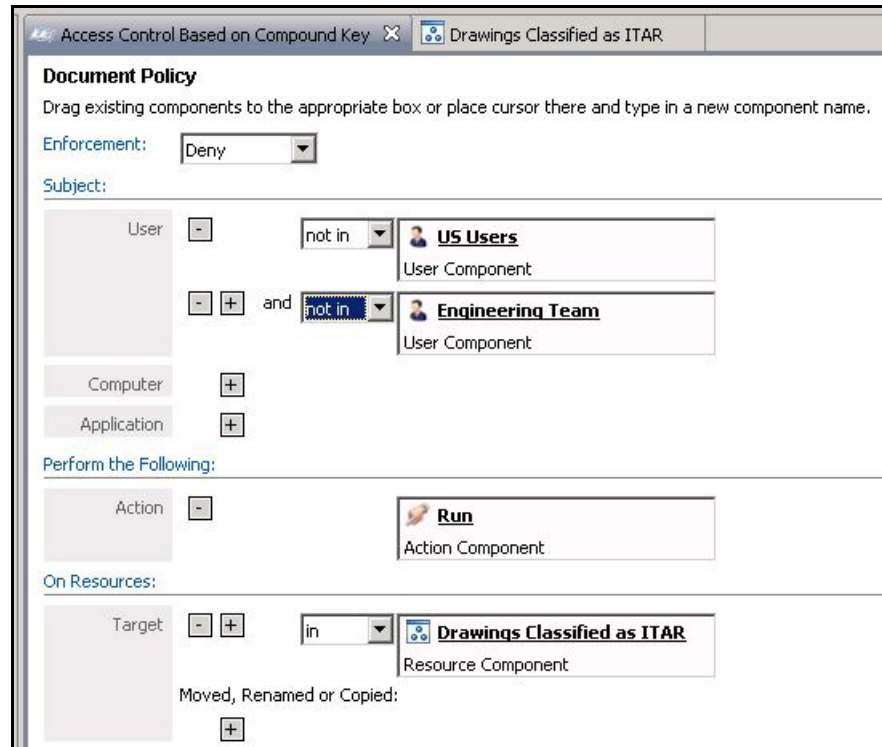


Figure 6-25: Example PBSC Content Analysis Policy

After this policy is deployed, when any user attempts to access engineering drawings that are classified “ITAR,” using any transaction or UI function within SAP ECC, SAP PLM, or SAP EasyDMS, a policy check will evaluate the user’s citizenship and department and allow or deny access accordingly.

### Example Policy: Access Control Based on Resource Attributes

The previous examples showed how to write policies that used security classification values to control user access to business objects. Objects with the “ITAR” classification were restricted to authorized users.

This example shows another way to write an access control policy—one that uses resource attribute values instead of security classification values to control access to data. Note, however, that in many cases, particularly when combining multiple resources or values in a policy, it is easier to maintain and use security classification values.

This example policy restricts access to data about US plants to US users only. A top-level policy, shown in [Figure 6-26](#), specifies the following:

- The enforcement type is set to Deny.
- No user component is specified, which means that all users are denied.
- The action component is Run.

- Two resource components are specified, designating the resources to which to target access control.
  - The first resource component, **SAP SYSTEM - ED6**, specifies any resource in the ED6 server, across all systems, clients, applications, functions and business objects:  
`**sap://ED6/**/**/**/**`
  - The second resource component, **US Plant**, refines access control to any data for plant 0001. In this component definition, the property (attribute) `plant` is set to the value `0001`. To control access to other objects, you can substitute `plant` with, for example, `material` or `document` or `warehouse number`. For more information about the attributes available for use in policy components, see [About SAP Resource Attribute Names](#) on page 218.

Both resource components form AND conditions, which specify that all data for Plant 0001 in the ED6 server are subject to access control.

- A subpolicy, **Allow US Users**, which specifies the exception to the top-level policy. It specifies that US users are authorized to access the data for plant 0001.

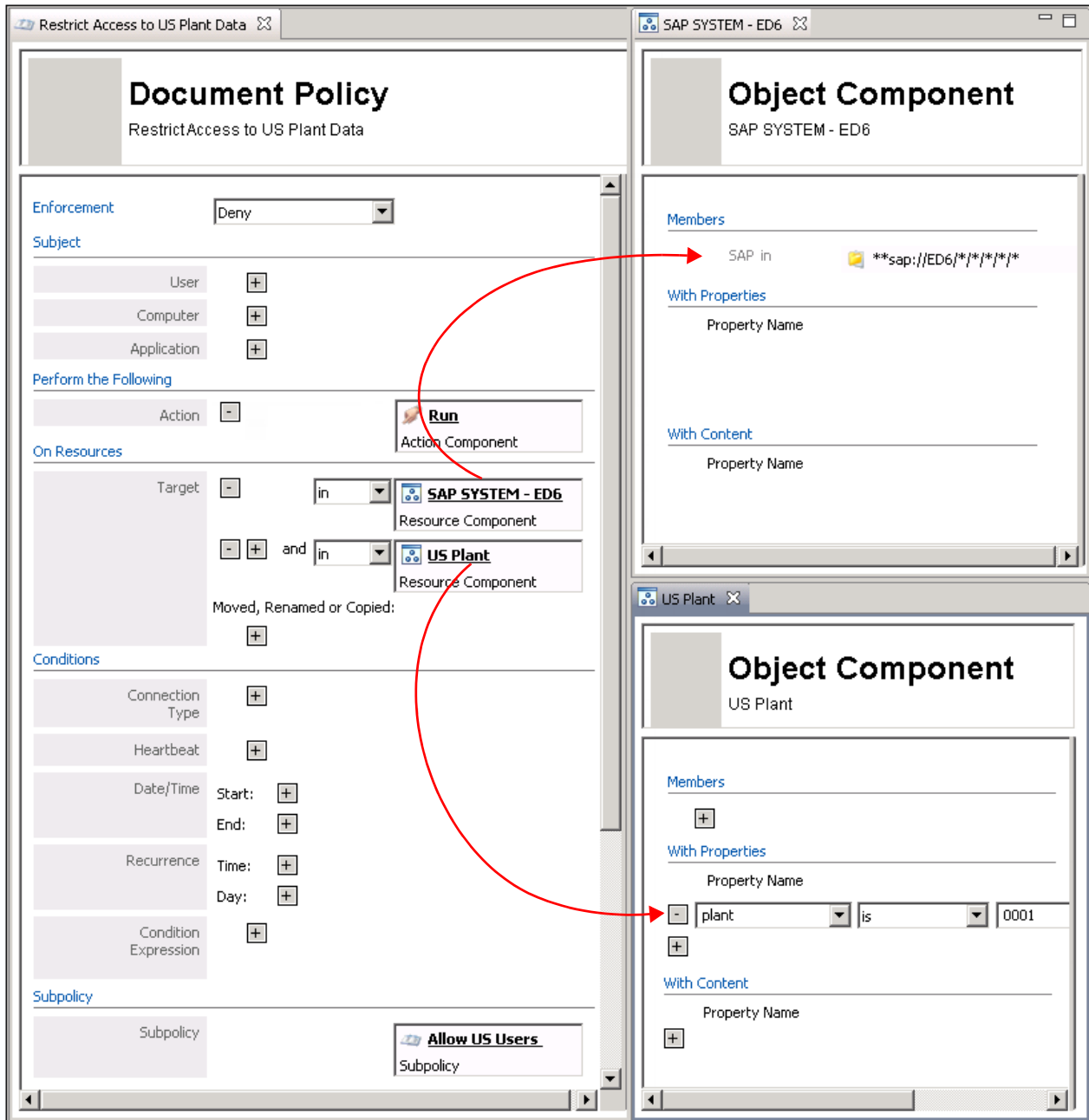


Figure 6-26: Policy to restrict access to Plant 0001 data in ED6 server

Figure 6-27 shows the definition of the subpolicy, which specifies the exceptions to the top-level policy. The subpolicy states that US users can access the resources specified in the top-level policy. US users are defined as any user whose `countrycode` attribute value is `US`.

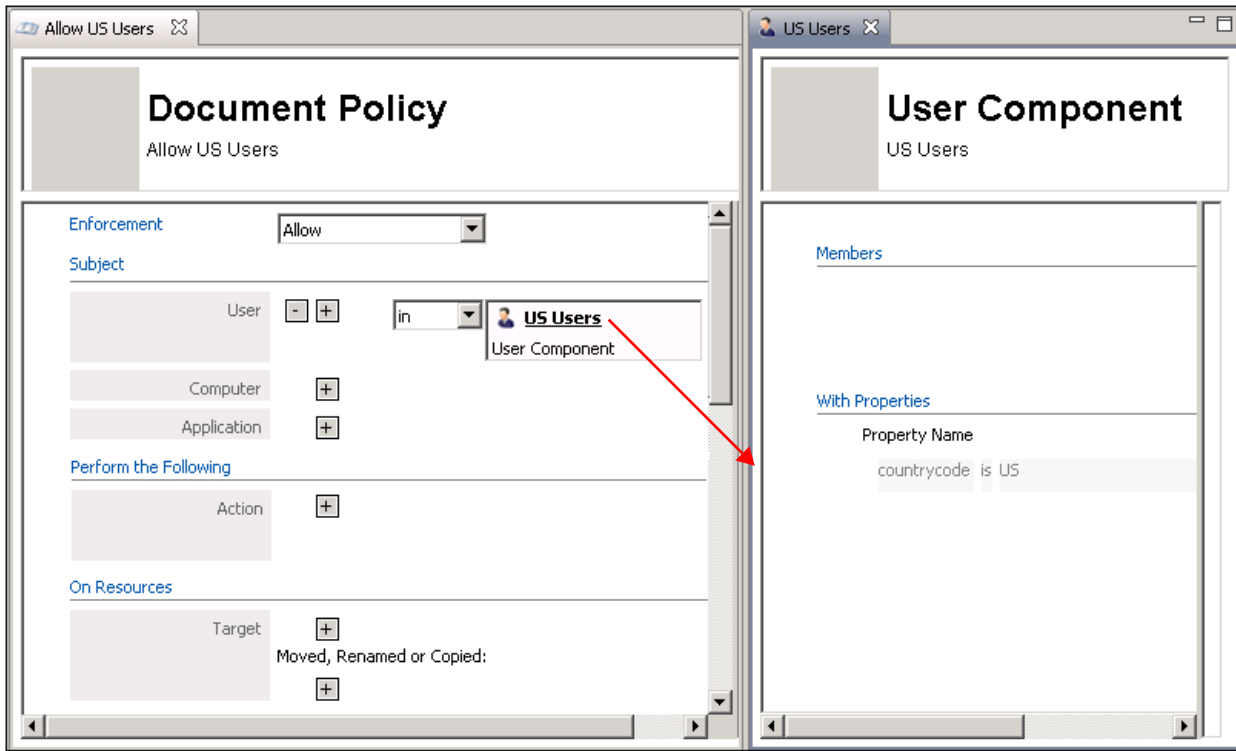


Figure 6-27: Subpolicy specifies that US users can access Plant 0001 data

### About SAP Resource Attribute Names

As the example above showed, you can use resource attributes to control access to data. The attribute names available for use in resource component definitions are derived from the /NEXTLABS/SECIDT (Security Identifiers) table. The Long field label for each identifier in this table is the attribute name you use. The following procedure shows how to discover the available attribute names.

#### Procedure

- 1 In the SAP interface, enter transaction SE11. The /NEXTLABS/SECIDT table, shown in [Figure 6-28](#), appears.

**Dictionary: Display Table**

Transp. Table: /NEXTLABS/SECIDT Active  
 Short Description: NextLabs:Base:Security identifiers

Attributes | Delivery and Maintenance | **Fields** | Entry help/check | Currency/Quantity Fields

Srch Help | Predefined Type | 1 / 29

Field	Key	Ini...	Data element	Data Ty...	Length	Deci...	Short Description
MANDT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	MANDT	CLNT	3		0 Client
REF_ID	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	/NEXTLABS/REFID	CHAR	10		0 Reference ID
.INCLUDE	<input type="checkbox"/>	<input type="checkbox"/>	/NEXTLABS/SECID...	STRU	0		0 NextLabs:ECC: Identifier Structure
MATNR	<input type="checkbox"/>	<input type="checkbox"/>	MATNR_D	CHAR	18		0 Material number, without search help
DOCNUM	<input type="checkbox"/>	<input type="checkbox"/>	DOKNR	CHAR	25		0 Document number
NXL_KUNNR	<input type="checkbox"/>	<input type="checkbox"/>	KUNNR	CHAR	10		0 Customer Number
NXL_LIFNR	<input type="checkbox"/>	<input type="checkbox"/>	LIFNR	CHAR	10		0 Account Number of Vendor or Creditor
NXL_RECIP	<input type="checkbox"/>	<input type="checkbox"/>	/NEXTLABS/RECIPE	CHAR	40		0 Recipe/Group/Routings
NXL_EQUNR	<input type="checkbox"/>	<input type="checkbox"/>	EQU NR	CHAR	18		0 Equipment Number
NXL_LGNUM	<input type="checkbox"/>	<input type="checkbox"/>	LGNUM	CHAR	3		0 Warehouse Number / Warehouse Complex
NXL_WERKS	<input type="checkbox"/>	<input type="checkbox"/>	WERKS_D	CHAR	4		0 Plant
NXL_VSTEL	<input type="checkbox"/>	<input type="checkbox"/>	VSTEL	CHAR	4		0 Shipping Point/Receiving Point
NXL_BWART	<input type="checkbox"/>	<input type="checkbox"/>	/NXLECC/BWART	CHAR	3		0 Movement Type (Inventory Management)
NXL_BUKRS	<input type="checkbox"/>	<input type="checkbox"/>	BUKRS	CHAR	4		0 Company Code
NXL_LGORT	<input type="checkbox"/>	<input type="checkbox"/>	LGORT_D	CHAR	4		0 Storage Location
NXL_TPLST	<input type="checkbox"/>	<input type="checkbox"/>	TPLST	CHAR	4		0 Transportation planning point
NXL_BWLVS	<input type="checkbox"/>	<input type="checkbox"/>	/NXLECC/BWLVS	NUMC	3		0 Movement Type for Warehouse Management
.INCLUDE	<input type="checkbox"/>	<input type="checkbox"/>	/NEXTLABS/SECID...	STRU	0		0 NextLabs:cFolder: Identifier Structure

Figure 6-28: /NEXTLABS/SECIDT table displaying the security identifiers

- 2 To discover the attribute name to use in a resource component, find the associated identifier in the **Field** column. In our example, NXL\_WERKS is the identifier for Plant.
- 3 Double-click the name in the **Data element** column. In our example, this is WERKS\_D, as shown in the previous figure.
- 4 In the *Display Data Element* screen, select the **Field Label** tab. The Long field label of every identifier is passed to the Policy Controller for use in policy evaluations. If a Long label is not defined, the Medium label is used. If that is not defined, the Field name is used.

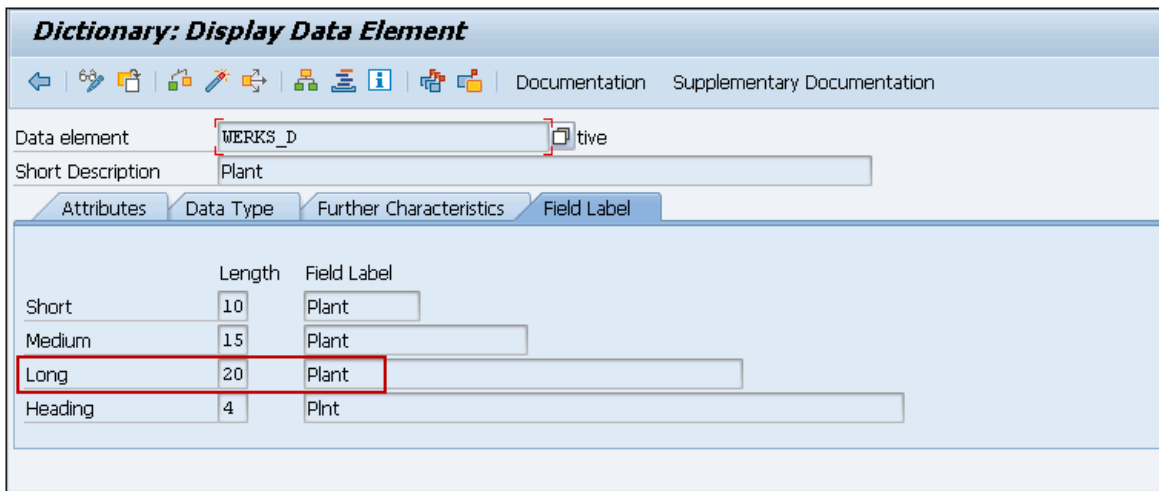


Figure 6-29: The Long field label is used as the attribute name in a policy

### Example Policy: View Filtering (EasyDMS Only)

You can create policies that automatically filter out documents users are not authorized to view when users log in to EasyDMS. That way, not only will users not be able to *access* restricted documents, they will not even be aware of their existence. The basic structure of this example policy is:

*For SAP business objects classified ITAR, run “Filter Document.”*

**Note:** In order for View Filtering to take effect, you must first configure it in the EasyDMS Configuration table (see [Configuring View Filtering \(EasyDMS Only\)](#) on page 102).

To create a View Filtering policy, perform these general steps:

- Create a Document policy. Enforcement can be set to allow or deny.
- Select a user component for the group the policy should impact, for example, Non-US employees.
- Select the “Run” Action component and drag it into the Action field
- For the Resource, create an SAP component with `EASYDMS` as the SAP application and `FILTER_DOCUMENT` as the function in the resource string, as in [Figure 6-30](#). You can also apply properties to the resource. This component applies to all documents that are classified ITAR.



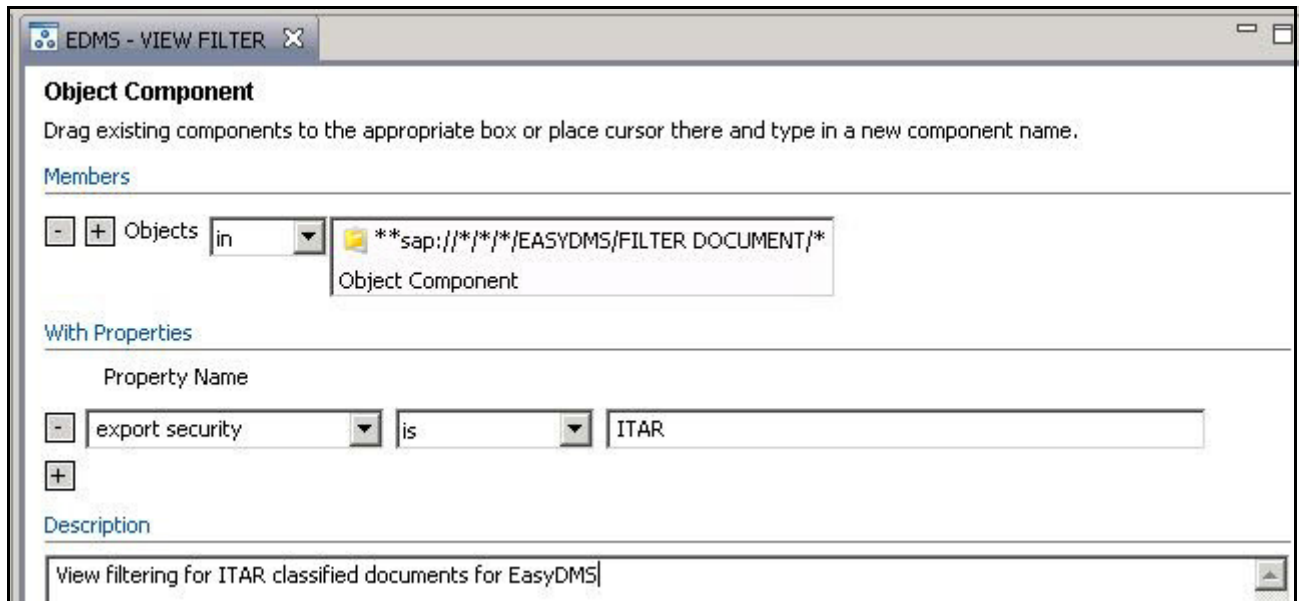


Figure 6-30: SAP Resource Component for View Filtering

- Since this policy enforcement is designed to be transparent to users, there should be no display message obligation.

After this policy is deployed, when non-US users log in to EasyDMS, documents that are classified as ITAR are automatically filtered out of view.

## View Filtering in cFolders

View Filtering in cFolders is a configuration that can be enabled and disabled in the *Connection Configuration* screen. If View Filtering is enabled, existing access control policies control not just who can download or open files by clicking on links, but what files will display to users. Unauthorized users will not be able to see links to documents they cannot access.

**Note:** For more information on enabling and disabling View Filtering, see [Configuring SAP Data Handling and Connection Settings](#) on page 94.

## Designing Access Control Policies for SAP BW

After Security Classifications are applied to BW objects—InfoAreas, InfoProviders, or custom InfoObjects—you can reference these values in access control policies. [Figure 6-31](#) shows examples of security classifications defined for BW objects in the Security Classification table. INFOPROVIDER and INFOAREA (shown as column headings) are standard identifiers for the BW Entitlement Pack. CUSTOMER ID is a custom identifier created to apply access control to an InfoObject that provides customer ID data.

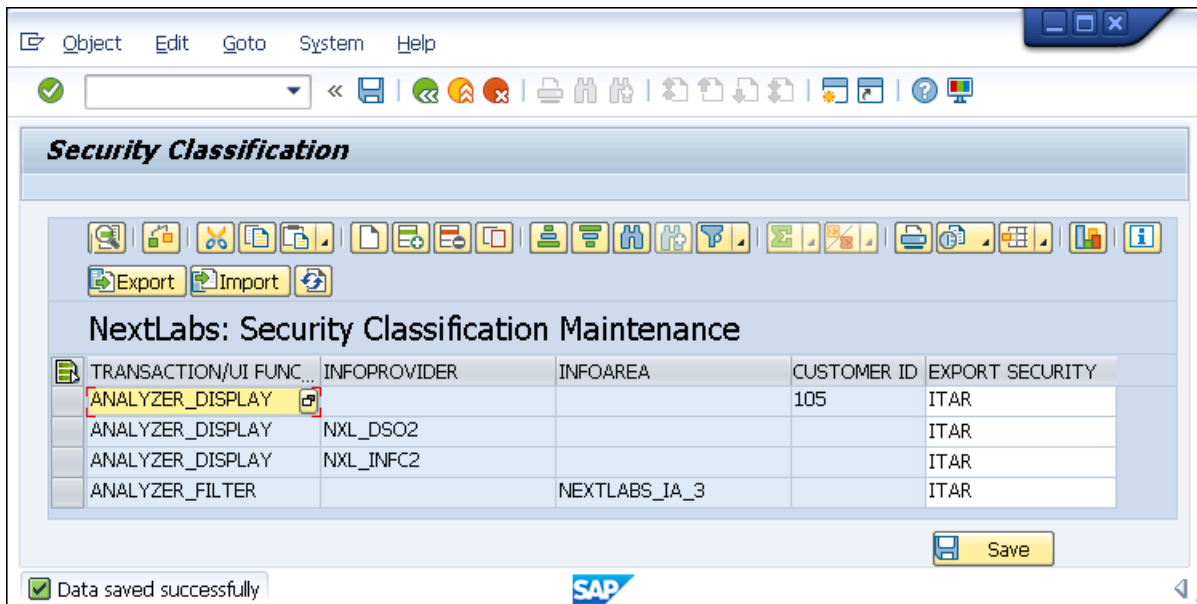


Figure 6-31: Security classifications for BW objects

The InfoObject, Customer ID 105 is classified as ITAR data. Similarly, the InfoProviders, NXL\_DSO2 and NXL\_INFC2, and the InfoArea, NEXTLABS\_IA\_3, are also classified as ITAR. The sample policy, described next, uses this classification information.

**Note:** Specifying ANALYZER\_DISPLAY for InfoProviders and InfoObjects and ANALYZER\_FILTER for InfoArea assume that an SAP developer has defined the appropriate enhancement implementations to support this functionality. For information about the enhancement implementations, see [Implementation Reference for SAP BW](#) on page 305.

### Example Policy: Restrict Access to Classified Data

This policy restricts access to BW objects classified as ITAR. Only users authorized to use ITAR data can access the data. A top-level policy, shown in the next figure, specifies the following:

- The enforcement type is set to Deny.
- No user component is specified, which means that all users are denied.
- No action component is specified, which means that all actions are denied.
- Two resource components are specified, designating the resources to which to target access control.
  - The first resource, **SAP - ITAR**, specifies any data where Export Security is set to ITAR.
  - The second resource, **SAP - ANALYZER - ACCESS**, specifies any ITAR data in a particular server, system, client, SAP application, function, and business object. In this example, the target is all business objects impacted by the ANALYZER\_DISPLAY function in the BW application in the NQ8 server and nq8 system:

**\*\*sap://NQ8/nq8\*/BW/ANALYZER\_DISPLAY/\***

- A subpolicy, **Allow US Users**, which specifies the exception to the top-level policy. It specifies that only US users are authorized to access the ITAR data.
- An obligation that displays an SAP message, "Access denied! Classified object." when the policy enforcement is Deny.

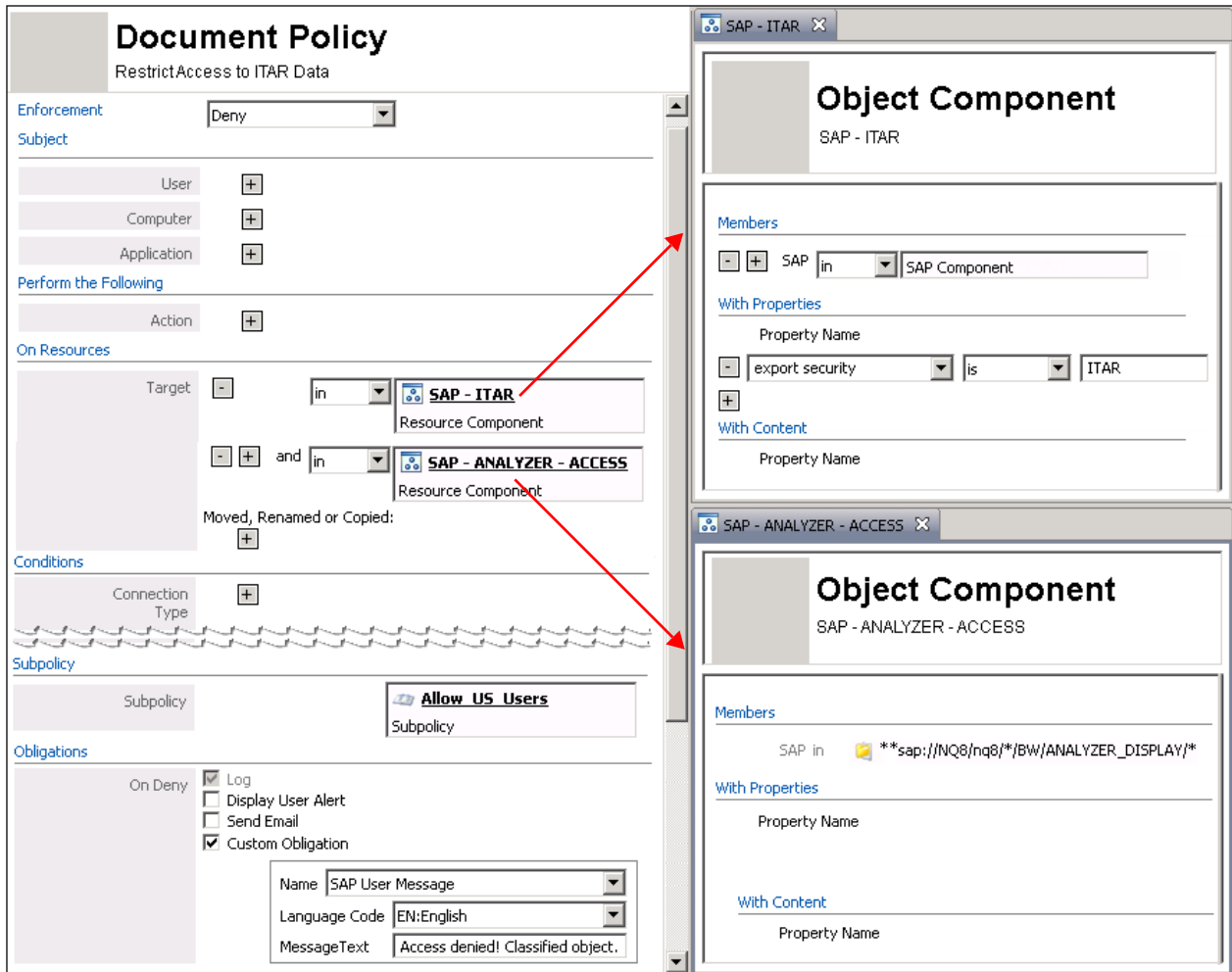
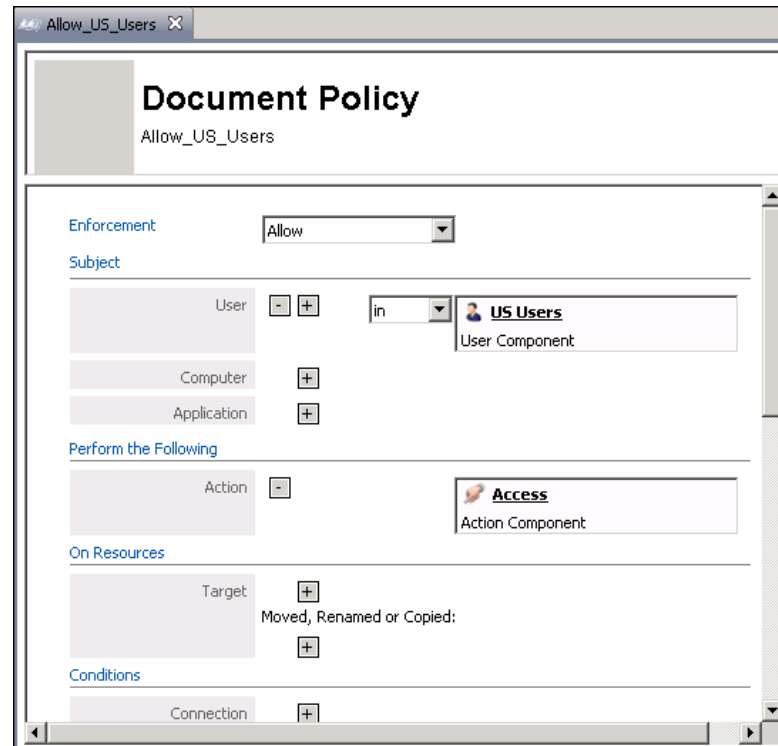


Figure 6-32: Policy to restrict access to ITAR data in SAP BW

Figure 6-33 shows the definition of the subpolicy. Because the subpolicy specifies the exceptions to the top-level policy, only the User and Action components need to be defined. The subpolicy states that US users can access the ITAR data in the resources specified in the top-level policy.



*Figure 6-33: Subpolicy specifies which users and what actions are permitted*

When deployed, the policy works in conjunction with the classifications defined in the Security Classification table to control access to ITAR classified objects. The following use cases demonstrate how access control in this example works:

- When a user selects **Open Query** in BEx Analyzer, InfoArea NEXTLABS\_IA\_3 is visible only to users that are members of the US Users component.
- All InfoProviders—classified or not—are visible to all users. [Figure 6-34](#) shows an example of InfoProviders listed in an InfoArea. NextLabs DSO2 is classified as ITAR in the Security Classification table.

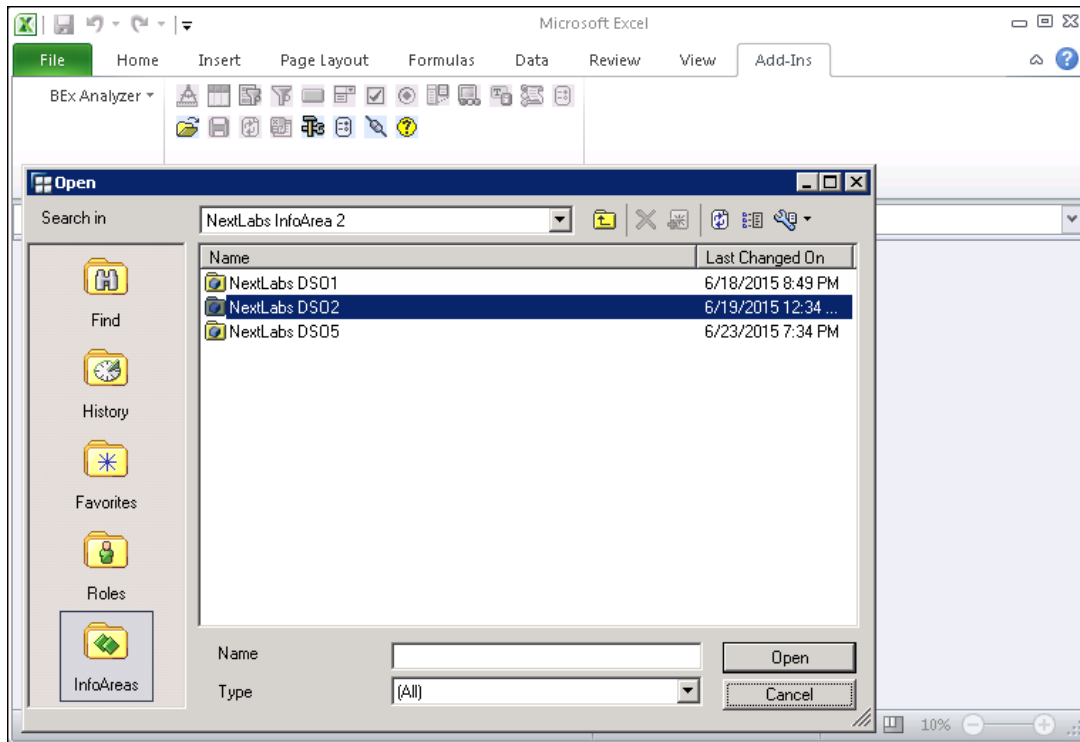


Figure 6-34: Open dialog displaying all InfoProviders in a selected InfoArea

- When the user selects the classified InfoProvider NextLabs DSO2, then runs the DSO, or an InfoCube or query that is derived from the DSO, Dynamic Authorization Management allows only members of the US Users component to view the generated report, as shown in [Figure 6-35](#).

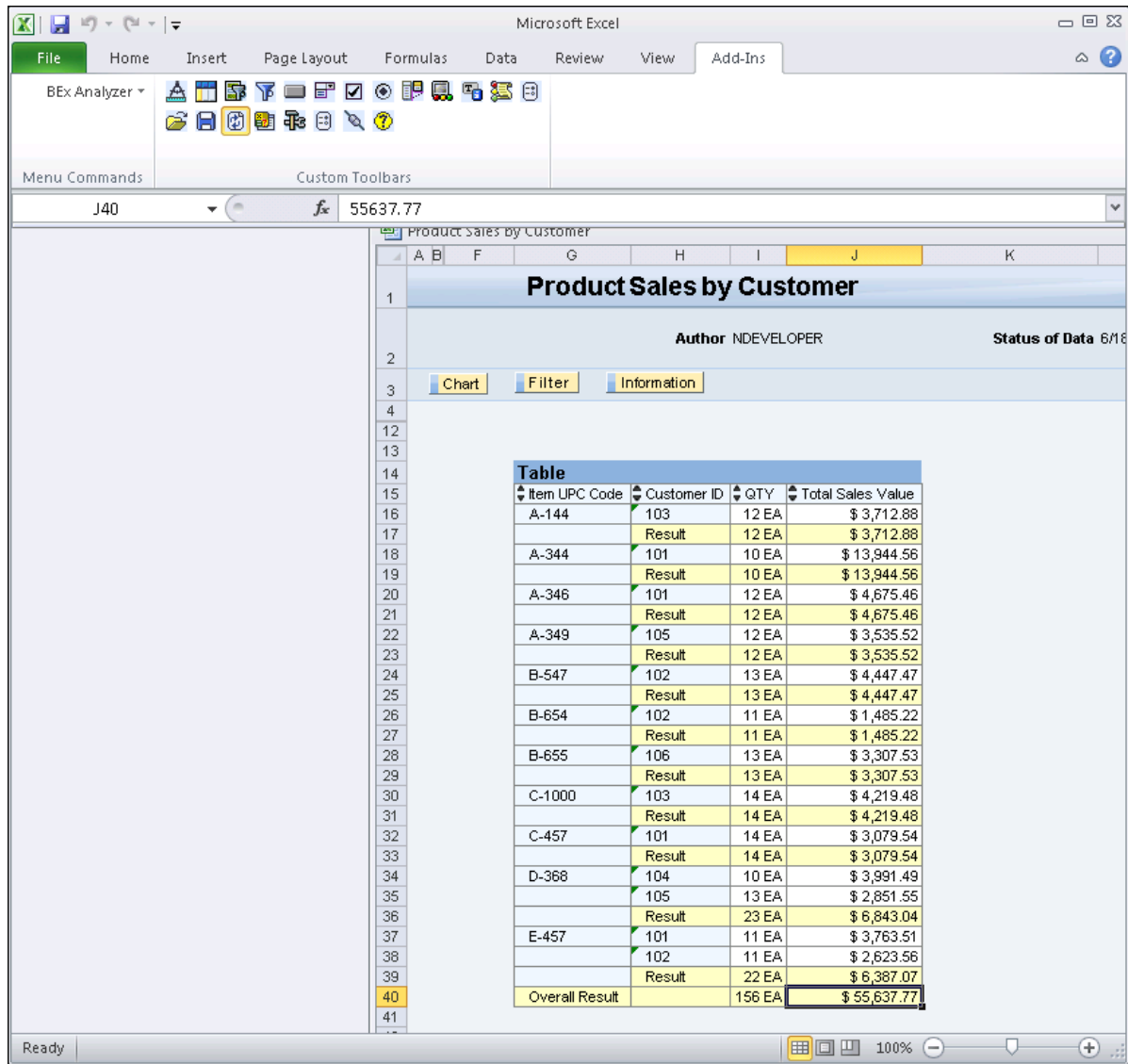


Figure 6-35: The report generated from the classified InfoProvider is displayed only to members of the US Users component

Non-members are denied access to the report; instead, they see the message defined in the policy's On Deny obligation, as shown in Figure 6-36.

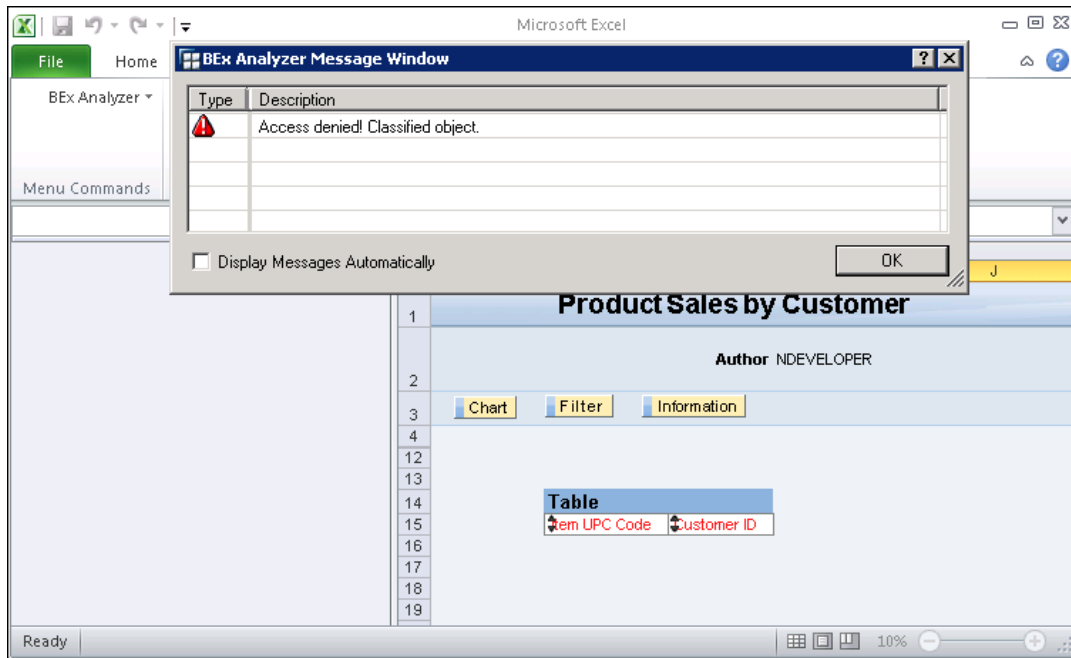


Figure 6-36: Access denied message displayed to users not authorized to view classified data

### Example Policy: Filter Access to Classified Data

The previous policy showed how to restrict access to classified objects. All users could see all the InfoProviders in the *Open* screen in BEx Analyzer. Access to classified InfoProviders was either permitted or blocked when a user tried to open those objects. The policy targeted the ANALYZER\_DISPLAY function to enforce this type of access control.

The alternate way to control access to classified objects is to filter the list of available objects so that users can see only the objects that they are authorized to access. To do so, you need to make the following security classification and policy changes:

- In the Security Classification table, for each entry, replace the ANALYZER\_DISPLAY function value with ANALYZER\_FILTER.
- In the SAP - ANALYZER - ACCESS resource component, change the resource string to the following:

```
**sap://NQ8/nq8/*/BW/ANALYZER_FILTER/*
```

**Note:** These changes assume that an SAP developer has defined the appropriate enhancement implementations to support the filtering functionality. For information about the enhancement implementations, see [Implementation Reference for SAP BW](#) on page 305.

---

## Designing Integrated Rights Management Policies

The goal of Integrated Rights Management (IRM) policies is to ensure that originals downloaded out of SAP remain protected, either because classification values have been injected into document metadata, or because files have been encrypted using NextLabs Encryption, or both. Dynamic Authorization Management for SAP enables this protection through its Integration with NextLabs Rights Management Server (RMS).

After IRM applies tags to originals, other NextLabs products can enforce policies that govern whether users can distribute, print, duplicate or otherwise manipulate files that are downloaded at the endpoint. When IRM applies NextLabs Encryption, originals can only be accessed by authorized users within a Policy Domain, meaning that unauthorized users will never be able to decrypt or view data.

Both encryption and tagging will occur during a batch process that evaluates queued originals. This batch process is scheduled during configuration.

**Note:** For more information on IRM configuration, see [Configuration for Integrated Rights Management \(IRM\)](#) on page 145. For more information on how NextLabs Encrypted files behave and how users interact with them, consult the *Rights Management User Guide*.

All Integrated Rights Management policies have the following requirements:

- Monitor polices (they do not allow or deny access)
- Use the Upload action
- Include the custom obligation “Integrated Rights Management”

### Example Policy: IRM Encryption and Tagging

A typical use case for an IRM policy would be where you want to both tag and encrypt originals anytime they are uploaded through SAP ECC, SAP PLM, SAP EasyDMS, or SAP cFolders using any transaction or UI function, where classification status (in the Security Classification Maintenance table) is “Export Control” is “ITAR.”

The policy would be: *For any resource where Export Control is ITAR, on Upload, run custom obligation Integrated Rights Management.*

**Note:** For more thorough instructions on the general procedure for creating a policy In Policy Studio, see [Example Policy: Access Control Based on Classification and ACC](#) on page 205. What follows is a high-level example meant to illustrate the unique features of IRM policies.

Perform these general steps:

- Create a Document policy. Set the enforcement to Allow.
- For this example use case, we want the policy to apply regardless of the user. We do not specify a subject.
- Select the “Upload” Action and drag it into the Action field.



- Create an SAP Resource component that includes SAP ECC, SAP PLM, SAP EasyDMS, and SAP cFolders as applications. For our example, we want to apply across all Applications, Transactions/UI Functions, and Business Objects, so our string is: `**sap://server/system/client/**/*`

In the With Properties drop down menu, enter “Export Control” is “ITAR.”

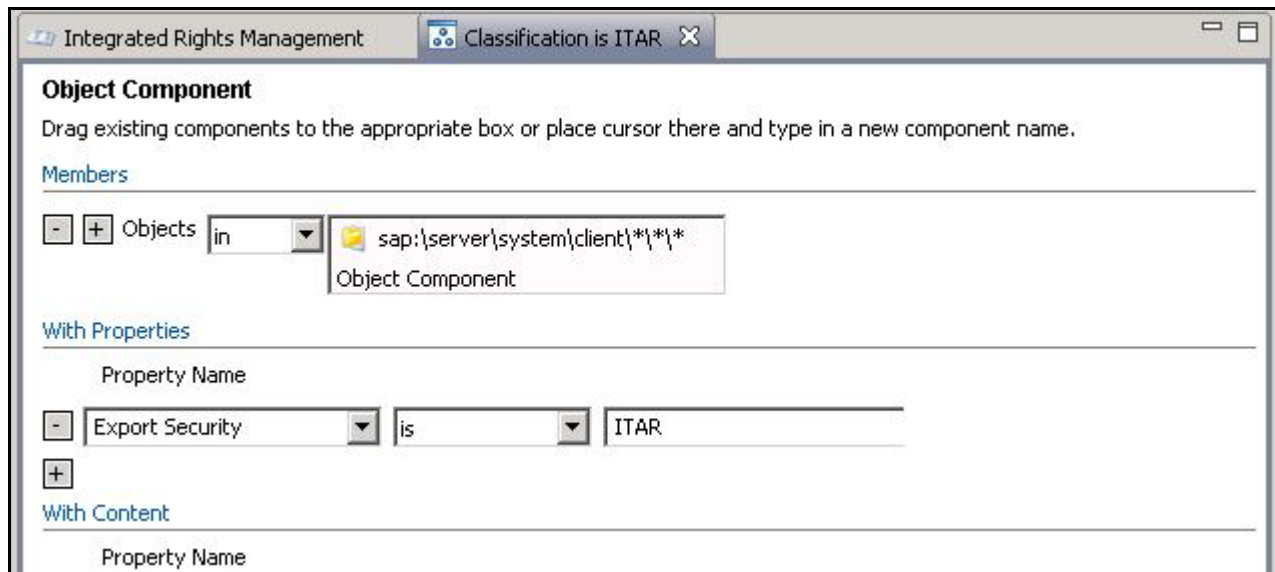
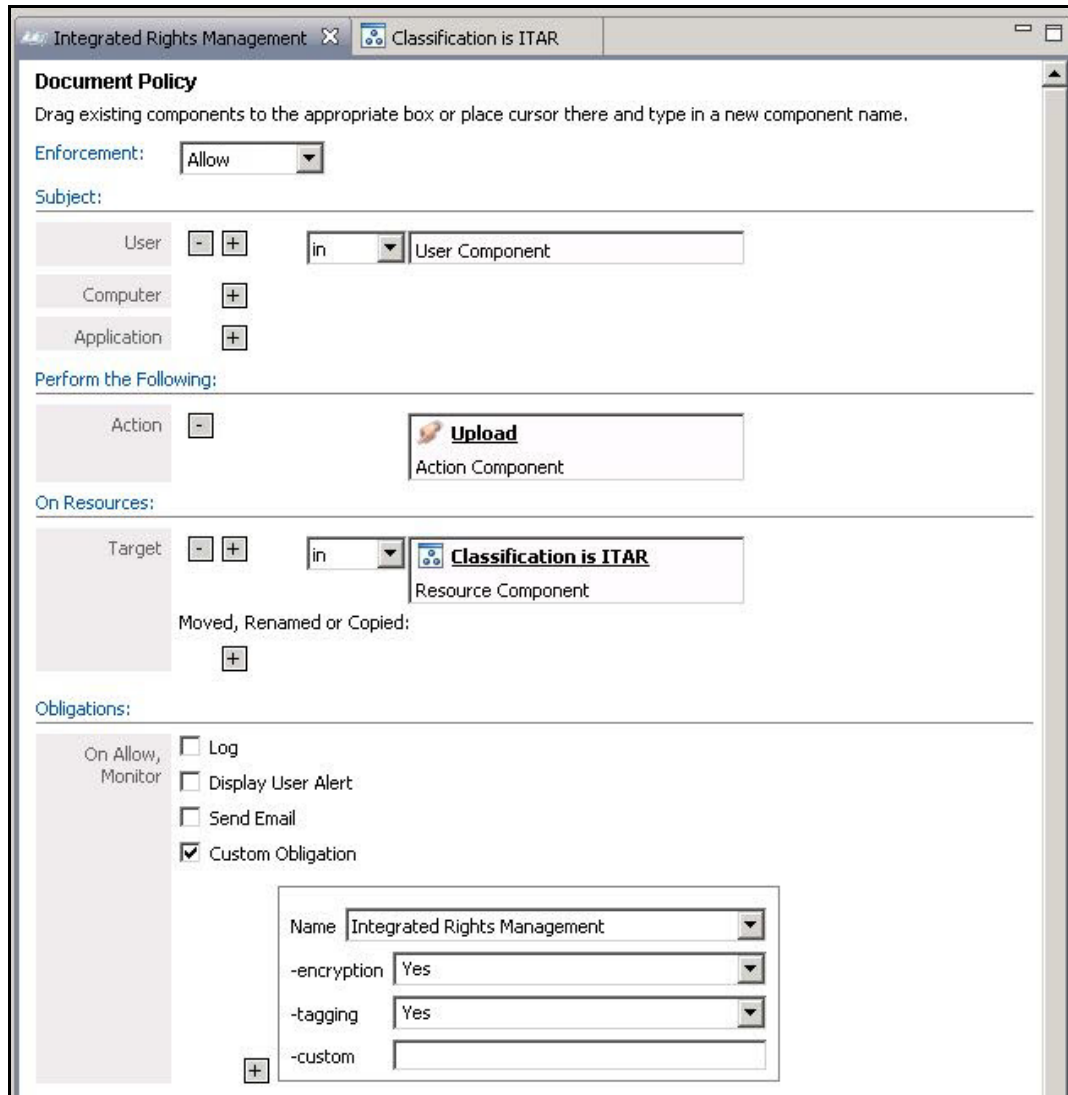


Figure 6-37: SAP Resource Component for IRM Policy

- Drag the resource component into the policy.
- Click Custom Obligation, and select “Integrated Rights Management” from the drop-down menu.
- To apply NextLabs Encryption, set **-encryption** to yes.
- To tag originals with all the classification values associated with the parent Material or Document in the Security Classification Maintenance table, set **-tagging** to yes. If you want to apply custom tags only (in the next step), set **-tagging** to no.
- To add custom tags (which can be *other than*, or *in addition to* the classification values associated with the parent Material or Document in the Security Classification Maintenance table), define them in the **-custom** field using the following format: `tag1=value1, value2;tag2=value1, value2`. For example, "export control=itar, ear;IP=yes".



*Figure 6-38: Example IRM Policy*

After this policy is deployed, when any user uploads a file to a business object that is classified "ITAR," through SAP ECC, SAP PLM, SAP EasyDMS, or when a file is created in SAP cFolders, the original will automatically be sent to the IRM conversion queue, along with the pertinent classifications in the Security Classification Maintenance table. On a schedule determined by your configuration, Rights Management Server (RMS) runs a batch process to tag and encrypt the original.

**Note:** A filter can be applied to restrict which documents are routed to the IRM queue. See [Configuring IRM Selection Criteria \(Filter\)](#) on page 149. For information on scheduling batch processes, see [Defining Background Jobs for IRM](#) on page 153.

---

## Designing Data Segregation Policies

Data segregation policies control what data is stored in which DMS content server. Create these policies to restrict storage of a class of data to a secure location, or to prevent a class of data from being stored in unauthorized locations. Before creating these policies, you must perform the configuration tasks described in [Configuring Policy-based Data Segregation](#) on page 141. You must also have applied security classifications to data. This section provides the following example data segregation policies:

- [Example Policy: Restrict Storage Locations for Check-in of Classified Data](#) on page 231
- [Example Policy: Segregate Classified Data and Other Data](#) on page 233
- [Example Policy: Restrict Download Locations for Check-out of Classified Data](#) on page 235

The first and second example policies work together. They show how to provide the user with a filtered list of approved storage locations when the user checks in classified or unclassified data. For this type of policy, observe the following guidelines:

- The enforcement type must be Allow.
- The action must be Check-in.
- You must use either the Data Segregation – Whitelist obligation or the Data Segregation – Blacklist obligation to determine which storage locations to display to the user. You cannot use both in a single policy. The Whitelist obligation specifies the locations to include in the list that the user sees. The Blacklist obligation specifies the locations to exclude from the list.

### Example Policy: Restrict Storage Locations for Check-in of Classified Data

This policy example provides the user with the choice of two approved storage locations when the user checks in data classified as ITAR.

- The enforcement type is set to Allow. You must use Allow if you want to filter the storage locations that the user sees.
- The action component is Check-in.
- The resource component specifies any data where Export Security is set to ITAR.
- The Data Segregation – Whitelist obligation is used to specify each storage location.

**Document Policy**  
 Drag existing objects to the appropriate box or place cursor there and type in a new object name.

Enforcement: Allow

Subject:

User +  
 Computer +  
 Application +

Perform the Following:

Action - Check in  
 Action Component

On Resources:

Target - + in Export Security: ITAR  
 Resource Component

Moved, Renamed or Copied: +

Conditions:

Connection Type +  
 Heartbeat +  
 Date/Time Start: + End: +  
 Recurrence Time: + Day: +  
 Advanced Condition +

Subpolicy:

Subpolicy Subpolicy

Obligations:

On Allow, Monitor  Log  
 Display User Alert  
 Send Email  
 Custom Obligation

+ Name Data Segregation - Whitelist storage category ZQA\_US2

- + Name Data Segregation - Whitelist storage category ZQA\_DBUSA3

Figure 6-39: Policy to restrict locations in which to check in ITAR data

The previous policy restricts the storage of ITAR data to two designated locations, which the user selects at check in. The policy, however, does *not* prevent other types of data from also being checked into those designated locations. To segregate ITAR data from other types of data, you need to create another policy, that provides the user with an alternate list of locations in which to store non-ITAR data. See [Example Policy: Segregate Classified Data and Other Data](#) on page 233.

### **Example Policy: Segregate Classified Data and Other Data**

This policy works in concert with the previous policy. It provides the user with a list of locations in which he can store non-ITAR data. The previous policy provides the list of locations in which the user can store ITAR data.

- The enforcement type is set to Allow.
- The action component is Check-in.
- The first resource component specifies any data in the SAP NQ3 system.
- The second resource component specifies any data that is *not* classified as ITAR.
- The Data Segregation – Blacklist obligation is used to specify the locations to exclude from the list. In this example, the excluded locations are the ones designated for ITAR data. When the user checks in non-ITAR data, he sees storage locations that are *not* designated for ITAR data.

**Document Policy**  
 Drag existing objects to the appropriate box or place cursor there and type in a new object name.

Enforcement:

Subject:

User

Computer

Application

Perform the Following:

Action   **Check in**  
 Action Component

On Resources:

Target    **SAP SYSTEM - NQ3**  
 Resource Component

and   **Export Security: ITAR**  
 Resource Component

Moved, Renamed or Copied:

Conditions:

Connection Type

Heartbeat

Date/Time Start:   
 End:

Recurrence Time:   
 Day:

Advanced Condition

Subpolicy:

Subpolicy

Obligations:

On Allow, Monitor  Log  
 Display User Alert  
 Send Email  
 Custom Obligation

storage category

storage category

Figure 6-40: Policy to designate locations in which to check in non-ITAR data

### Example Policy: Restrict Download Locations for Check-out of Classified Data

This policy example denies the checking out of ITAR data into any location that is not an SAP classified location.

- The enforcement type is set to Deny.
- The action component is Check-out.
- The first resource component specifies any data where Export Security is set to ITAR.
- The second resource component specifies any location that is *not* an SAP classified location.

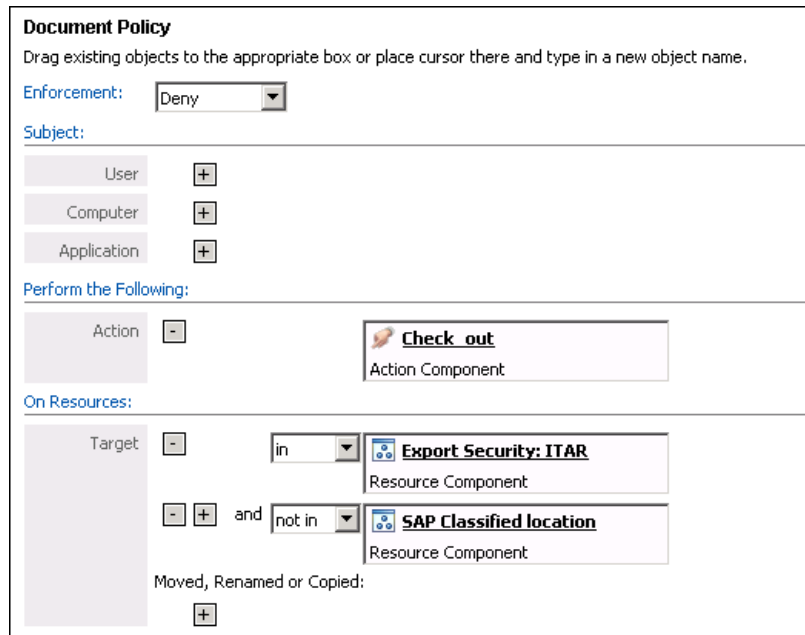


Figure 6-41: Policy to restrict locations in which to check out ITAR data

## Verifying the Storage Location of Data

As discussed in [Designing Data Segregation Policies](#) on page 231, you can create and deploy policies to control what data is stored in which DMS content servers. For example, a policy can restrict the storage of data classified as ITAR to specific servers in the US, and another policy can segregate unclassified data from ITAR-classified data by storing each type of data in different servers.

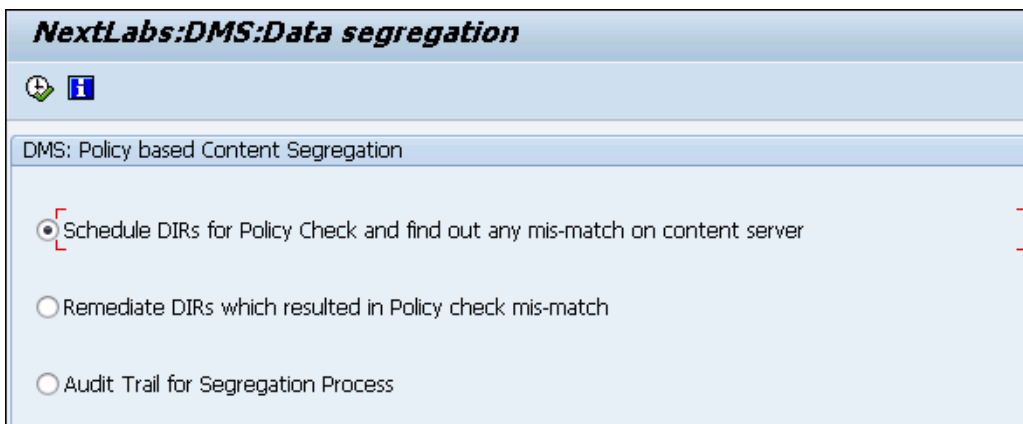
As you might expect, these data segregation policies take effect only on data created, stored or moved after the policies are deployed. Data that already exists in the system remain where they are, and this legacy data might not be stored in the correct locations, as specified by the policies. To address this issue, Dynamic Authorization Management for SAP provides a utility that does the following:

- 1 Check the DIRs in your system against the policies to verify if originals are stored in the correct locations.
- 2 Remediate any mismatch between an original and its storage location by moving the original to the correct location. The utility moves data to the new location as long as it is in a different content server.
- 3 Generate an audit log that tracks remediation actions.

You can run the utility as needed or on a schedule. If you are running it for the first time, all DIRs in the system are included in the evaluation. Otherwise, the evaluation includes only new DIRs or DIRs whose classifications have changed since the utility was last run.

**Procedure**

- 1 In the SAP interface, enter transaction /NXLECC/DMS\_DSEG. The *DMS Data Segregation* screen appears, as shown in [Figure 6-42](#).



*Figure 6-42: Data Segregation utility*

- 2 Execute the first option to check the DIRs against the data-segregation policies.
- 3 In the Job Wizard, specify the desired job options, then click **Complete** to create the job.
- 4 Back in the *DMS Data Segregation* screen, execute the second option to display and fix any mismatches between data and storage location that the utility finds.
- 5 Optionally, in the Selection options, filter the data to display, then click **Execute**.

The utility displays the list of originals that are not in the correct locations. [Figure 6-43](#) shows an example. The **Orignl SCG** column shows the current location of the original. The **Policy Based SCG** column shows the correct location, as specified in the policy.



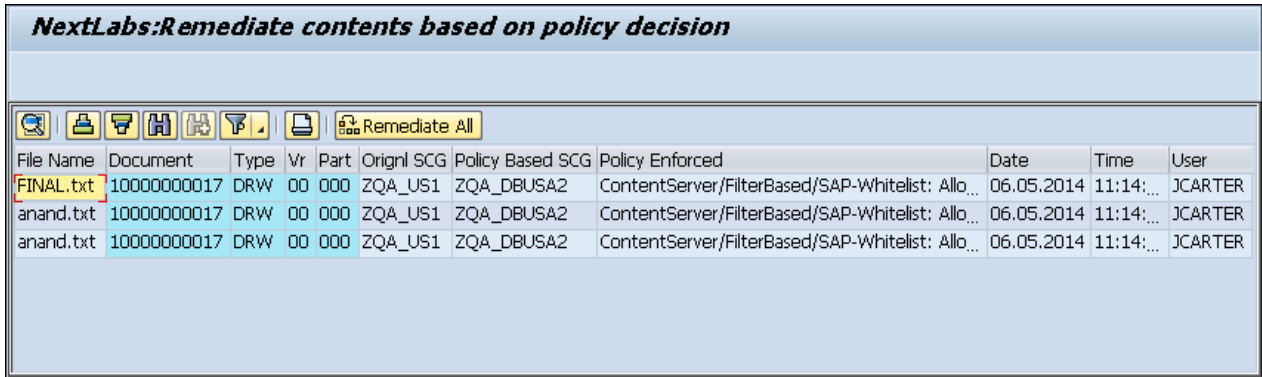


Figure 6-43: Remediation screen

- 6 Choose **Remediate All** to move data to the correct locations.
- 7 In the Job Wizard, specify the desired job options, then click **Complete** to create the job.
- 8 Choose **Back** twice to return to the utility’s main screen.
- 9 Execute the third option to view the audit log.
- 10 Optionally, in the Selection options, filter the data to display. Choose **Execute**.

The utility displays the list of originals that were selected for remediation. [Figure 6-44](#) shows the last four columns of the log. The **Status Message** column indicates whether the data was moved or not. The message, *Both storage type share same content table, not possible to transfer*, means that the data was not moved because the original storage location and the target storage location are both in the same content server. You can, however, manually move the data to a storage location on the same content server, if the policy permits it.

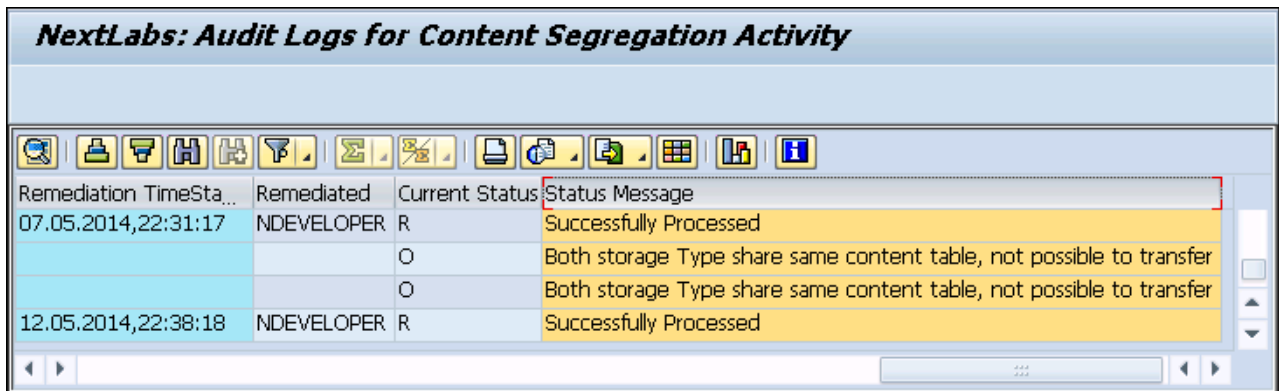


Figure 6-44: Audit log displaying remediation information



This section describes the administrative procedures associated with Entitlement Manager for SAP, including maintenance procedures related to Security Classifications in SAP, as well as maintenance procedures related to the NextLabs Policy Server and Control Center.

Topics:

- [Maintenance for Dynamic Authorization Management for SAP ECC](#)
- [Viewing NextLabs Log Information in SAP](#)
- [Configuration and Management](#)
- [About Service Account Permissions](#)
- [About Bundle Encryption](#)
- [Managing Enforcer Policies](#)
- [Managing Event Logging](#)
- [Load Balancing the Policy Controller](#)

## Maintenance for Dynamic Authorization Management for SAP ECC

The procedures necessary for maintaining SAP Security Classifications are already discussed in different areas of this manual. [Table 7-1](#) provides cross-references to sections where you can find more information on each procedure.

*Table 7-1: Links to Maintenance Procedures Discussed in this Guide*

Procedure	See section...
You may need to add or edit the Identifiers that display in the <i>Security Classification</i> screen. You add an append structure to the classification table to create a new custom identifier. Then you map the new identifier to the NextLabs Policy Controller.	See <a href="#">Custom Security Classification Identifiers</a> on page 255 and <a href="#">Mapping Security Fields (SECM PG)</a> on page 84.
You may need to enter new Security Classification values to Materials and Document, or change the classification settings of existing ones.	See <a href="#">Applying Security Classifications</a> on page 188.

Table 7-1: Links to Maintenance Procedures Discussed in this Guide (Continued)

Procedure	See section...
You may need to change the settings that determine how Identifiers should be prioritized when multiple Identifiers are present (i.e., when multiple Document records are present on a Material or for BOMs or other transactions that may have multiple business objects that may have different classification settings).	See <a href="#">Defining How Multiple Security Classifications Should Be Applied</a> on page 116.
You may need to reconfigure whether a Policy Check occurs for all transactions, or only on transactions for business objects that have security classification data.	See <a href="#">Configuring Policy Checks Based on Transaction/UI Function</a> on page 109.
You may need to reconfigure whether a Policy Check occurs for transactions where multiple identifiers are present, but where there is no default prioritization setting to determine how multiple security Identifiers should be prioritized.	See <a href="#">Configuring SAP Data Handling and Connection Settings</a> on page 94.
You may need to change how Time Out Errors are handled for communication errors between the SAP Agent and the Policy Controller.	See <a href="#">Configuring SAP Data Handling and Connection Settings</a> on page 94.

## Viewing NextLabs Log Information in SAP

The following NextLabs events are tracked by SAP logging:

- Export of classification data from the Security Classification Maintenance table
- The running of IRM programs
- The running of PBSC programs
- The updating of classifications for documents (using /NEXTLABS/UPD\_SCL) that were exported from SAP ECC to cFolders

You can use transaction `SLG1` to view these logs for troubleshooting or administrative procedures.

### Procedure

- 1 In the SAP interface, enter transaction `SLG1`. *The Analyze Application Log* screen appears.
- 2 To target NextLabs log information, enter `/NEXTLABS/*` in the Program field.
- 3 Specify other filter criteria for the event you want to view in the log.

**Analyze Application Log**

Object \*  
 Subobject \*  
 External ID \*

**Time Restriction**  
 From (Date/Time) 31.03.2016 00:00:00  
 To (Date/Time) 31.03.2016 23:59:59

**Log Triggered By**  
 User \*  
 Transaction code \*  
 Program /NEXTLABS/\*

**Log Class**  
 Only very important logs  
 Only important logs  
 Also less important logs  
 All logs

**Log Creation**  
 Any  
 Dialog  
 In batch mode  
 Batch input

**Log Source and Formatting**  
 Format Completely from Database  
 Format Only Header Data from Database  
 Format Completely from Archive

*Figure 7-1: Analyze Application Log*

4 Click **Execute**. The logging events display.

The screenshot shows the 'Display logs' application window. The main area contains a table with the following data:

Date/Time/User	Nu...	External ID	Object txt	Sub-object text	T	Program	Mode	Log nu...
29.03.2016 11:07:54 BOBAMA	9	DMS_READ_TAGS	NextLabs Application Read Log ...	Read functionality		/NEXTLABS/SAP...	Dialog proce...	000000000C
29.03.2016 11:57:58 BOBAMA	9	DMS_READ_TAGS	NextLabs Application Read Log ...	Read functionality		/NEXTLABS/SAP...	Dialog proce...	000000000C
29.03.2016 13:13:50 BOBAMA	9	DMS_READ_TAGS	NextLabs Application Read Log ...	Read functionality		/NEXTLABS/SAP...	Dialog proce...	000000000C
29.03.2016 13:16:51 BOBAMA	5	DMS_READ_TAGS	NextLabs Application Read Log ...	Read functionality		/NEXTLABS/SAP...	Dialog proce...	000000000C
29.03.2016 13:20:11 BOBAMA	9	DMS_READ_TAGS	NextLabs Application Read Log ...	Read functionality		/NEXTLABS/SAP...	Dialog proce...	000000000C
29.03.2016 13:21:10 BOBAMA	9	DMS_READ_TAGS	NextLabs Application Read Log ...	Read functionality		/NEXTLABS/SAP...	Dialog proce...	000000000C
29.03.2016 14:24:11 RGREEN	9	DMS_READ_TAGS	NextLabs Application Read Log ...	Read functionality		/NEXTLABS/SAP...	Dialog proce...	000000000C
29.03.2016 14:29:13 GWASHIN	9	DMS_READ_TAGS	NextLabs Application Read Log ...	Read functionality		/NEXTLABS/SAP...	Dialog proce...	000000000C
29.03.2016 14:33:59 RGREEN	3	DMS_READ_TAGS	NextLabs Application Read Log ...	Read functionality		/NEXTLABS/SAP...	Dialog proce...	000000000C
29.03.2016 14:35:06 RGREEN	9	DMS_READ_TAGS	NextLabs Application Read Log ...	Read functionality		/NEXTLABS/SAP...	Dialog proce...	000000000C
29.03.2016 14:37:11 RGREEN	5	DMS_READ_TAGS	NextLabs Application Read Log ...	Read functionality		/NEXTLABS/SAP...	Dialog proce...	000000000C
29.03.2016 14:52:43 RGREEN	9	DMS_READ_TAGS	NextLabs Application Read Log ...	Read functionality		/NEXTLABS/SAP...	Dialog proce...	000000000C
29.03.2016 15:12:13 GWASHIN	15	DMS_READ_TAGS	NextLabs Application Read Log ...	Read functionality		/NEXTLABS/SAP...	Dialog proce...	000000000C
29.03.2016 15:22:17 GWASHIN	15	DMS_READ_TAGS	NextLabs Application Read Log ...	Read functionality		/NEXTLABS/SAP...	Dialog proce...	000000000C

Below the table, the 'Type Message Text' section shows the following details:

- DIR: DRW DOC-100-122 000 00 TCODE: CV02
- Original Id : 000C2938FAC91ED58DBB24313B0C3ECF File Name: C:\Users\Administrator\Desktop\test.docx Logid: 107374254403
- IN Folder Created
- Successfully Checked out the file \\SAPGTS-CC01\CONVERSION\IRM\SAR\US1\NXL\_DRWDOC-100-12200000\IN\test.docx
- Reading of tags completed successfully
- No of relevant Tags in this Original document is: 1
- IN Folder Deleted
- No of Unique Tags to be Updated is :1
- Successfully Updated DOC-100-122

Figure 7-2: Display Logs for /NEXTLABS/\* Program

## Configuration and Management

No configuration changes are required for running enforcers once they have been installed. There are some configuration controls available through Administrator, and there are some minimal management activities that enforcer administrative personnel may need to perform from time to time.

### Configuration Tools

The only configuration settings available for enforcers are the enforcer profile settings, and the log file management settings. For details, see [Managing Enforcer Policies](#) on page 248 and [Managing Event Logging](#) on page 249.

## Management Activities

Because enforcers are designed to run continuously with no complications, there are very few management functions required from system administrators. These include stopping enforcers that are currently running, restarting enforcers after they have been stopped, and monitoring the status of currently running enforcers.

### Stopping and Starting Dynamic Authorization Managements

Because Dynamic Authorization Managements are designed to resist tampering, no user can stop them through the standard services manager, the Windows Task Manager, changing registry settings, or any other standard Windows procedure. Dynamic Authorization Managements can be stopped only with a special, password-protected utility.

As an additional tamper-resistance measure, no user can even view the contents of an Dynamic Authorization Management's installation directory while Dynamic Authorization Management is running. This means that you must stop Dynamic Authorization Management before you can perform certain tasks, such as examining log files in the logs directory.

Administrators can manually stop each individual Dynamic Authorization Management at the local host using a utility called **StopEnforcer.exe**, installed on each enforcer host at *C:\Program Files\NextLabs\Policy Controller\public\_bin*. You can run this utility from a link under **Start > All Programs > NextLabs**. This utility requires the administrative password set for whatever profile is assigned to Dynamic Authorization Management. This means you need to know which profile is in use, and what its password is, before you can stop an Dynamic Authorization Management.

After you stop an Dynamic Authorization Management with this utility, you can restart it again manually with the standard Services manager in Windows Control Panel, or by rebooting the host.

### Monitoring Enforcers

You can monitor the status of all enforcers in the network—on file servers and on PCs—by opening Administrator, going to the Status tab, and clicking the Policy Enforcer Status link. By default, this tab displays the status of all enforcers in the system that are displaying warnings. To show all enforcers with or without warnings, uncheck the Enforcers with Warnings Only checkbox (A). To view only desktop server enforcers, select *All Desktop Enforcers* from the Show combo-box list at the left (B). If you are interested in the status of enforcers on a specific enforcer host or host group, you can also filter by host name by typing it into the Search By Host field (C) and clicking the Search button.

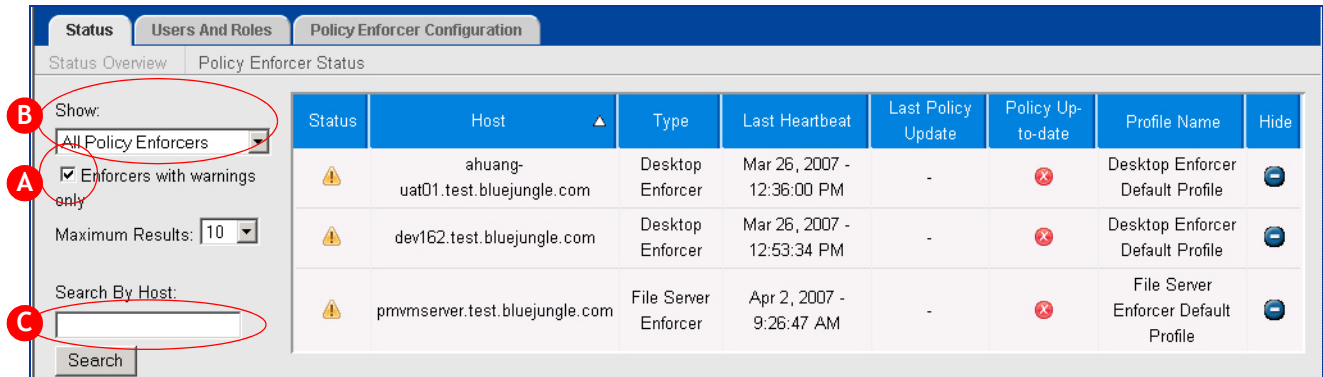


Figure 7-3: Monitoring the Status of Enforcers

Descriptions of the contents of the enforcer status grid are provided in Table 7-2.

Table 7-2: Information on Policy Enforcer Status

Column	Description
<b>Status</b>	Indicates the current status of this enforcer, which may be either of the following: Green light = Clear: the policy enforcer is sending normal heartbeats. Exclamation point = Warning: the policy enforcer has not sent a heartbeat in the last 24 hours.
<b>Host</b>	Name of the machine where the policy enforcer is installed.
<b>Type</b>	Indicates the policy enforcer type: File Server Enforcer or Desktop Enforcer.
<b>Last Heartbeat</b>	Time stamp of the last heartbeat generated by this policy enforcer. If the policy enforcer is running normally, this time should correspond to the configured heartbeat interval. However, keep in mind that this does not necessarily indicate a problem, since certain policy enforcers—in particular, those on laptop computers used by remote personnel or computers that are turned off when not in use—might not be able to send a heartbeat for an extended period of time even though they are operating normally.
<b>Last Policy Update</b>	Tells when a new or modified policy or policy component was last deployed to this policy enforcer.
<b>Policy Up to Date</b>	A check mark appears if the policy enforcer has received the latest version of the policies that are targeted for deployment to it.
<b>Profile Name</b>	Tells which policy enforcer profile is assigned to this host. This profile determines behavior such as logging and heartbeat frequency.
<b>Hide</b>	Click to remove this host from the display. This is useful when the policy enforcer software has been uninstalled, and you therefore no longer need to monitor that host. If a policy enforcer is ever reinstalled on this host, the host will reappear on the list. If you click Hide by mistake on a host with an active policy enforcer, it will reappear automatically the next time the policy enforcer sends a heartbeat.

### Uninstalling, Repairing, or Modifying Policy Controllers and Enforcers

Policy Controllers and enforcers use the standard Repair/Remove procedure common to all Windows applications. Because of the Policy Controller’s tamper-resistance features, an authorized administrator must first stop it before removing it. It is recommend that the same Administrator remove the Policy Controller as installed it.



Policy Controllers and enforcers are listed separately in the Add/Remove Programs list, and the process for uninstalling, repairing, or modifying is common to all Policy Controllers.

**Important:** You must uninstall the enforcer first, and the Policy Controller second. The sequence must not be reversed.

### Procedure

- 1 Open the Windows Control Panel.
- 2 Launch the Add or Remove Programs utility.
- 3 Select the enforcer.
- 4 Click the Change button.

When this wizard runs for an enforcer, it offers the following options:

- **Modify:** Available only for the Policy Controller; enables you to reset any of the options that were selected during the initial installation.
- **Repair:** Runs an automatic diagnostic procedure that detects and restores any missing or damaged system files, for the currently installed version.
- **Remove:** Uninstalls the current Policy Controller. You use this option when you want to remove the enforcer for any reason—for example, to upgrade to a new version.

## About Service Account Permissions

All enforcers run as Windows services, and are assigned a default user account at installation. This account must have read, execute and create permission for the folder where the enforcer is installed. For example, by default the Windows Desktop Enforcer is installed in the C:\Program Files\NextLabs\Desktop Enforcer folder, and is assigned the Local System user account, as shown in [Figure 7-4](#).

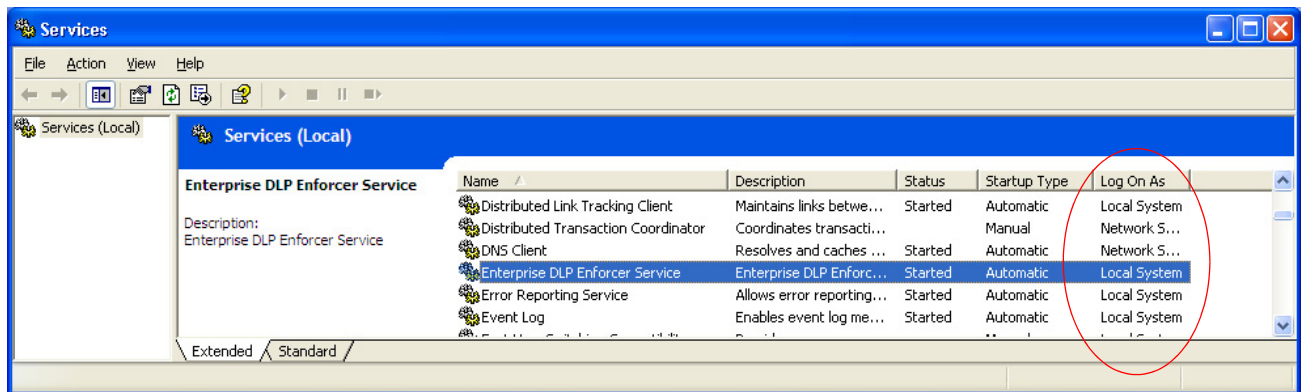


Figure 7-4: Enforcer Service User Accounts

You must take care not to change the permission levels of the install directory, wherever it may be, or of the services user accounts, in such a way that the account does not have read, execute and create permission for the installation folder. If you do this, the account will not be able to access the service, and it will not restart automatically if it ever stops. For example, if you change the Local System user permission to extend only to Program Files without any child directories, the service will not be able to restart.

---

## About Bundle Encryption

The Policy Server continuously updates enforcers, of all kinds, with any newly-defined or modified policies relevant to them. Each enforcer periodically sends a heartbeat message to the ICENet Server, which then checks whether any new or changed policies are in queue to be sent to that enforcer. If there are, it sends them, in the form of a file called *bundle.bin*; this file is referred to as a *policy bundle*. Each enforcer only retains one bundle at a time, which contains its most recent enforcement instructions. If the Policy Server sends another bundle with updated instructions—additional new policies to enforce, for example—that will overwrite the previous bundle.

All policy bundles sent from the Policy Server to enforcers are encrypted using standard SSL protocols. When they arrive at the enforcer, the enforcer authenticates them with digital certificates to ensure that they were indeed created by the Policy Server, and that they have not been modified by any other processes. This protects against the possibility of anyone deploying spoof policies designed to open security holes in your enterprise.

## Authentication Failure

Whenever a bundle file arrives at an enforcer client and cannot be authenticated, a Level 3 document activity event is written to the Windows Event Log:

```
policy bundle authentication failed
```

This event will also be displayed by Reporter, if your query includes Level 3 events. The most likely cause of such failure is that the file is corrupted in some way; in such cases you should examine the file contents.

## Decrypting the Bundle: Policy Controller

Bundle files are encrypted, but administrators can decrypt them for troubleshooting purposes. For this purpose, a special utility called *Decrypt.exe* is available in the *public\_bin* directory of each host where an enforcer is installed.

### Procedure

- 1 Stop the enforcer on the host where the encrypted bundle file is located. You cannot decrypt any bundles while the enforcer is running.
- 2 Open a command prompt and run the utility, supplying the arguments shown below.

```
decrypt -b <path\file> -f <OutputFile.txt> -e <InstallPath>
```

In this command:

- **-b** is the complete path and name of the encrypted bundle file. By default, the path is C:\Program Files\NextLabs\Desktop Enforcer. The file name will always be *bundle.bin*. This argument is not mandatory; if it is not present, the enforcer will assume the default path and file name.
- **-f** is the name of the output (decrypted) file. This argument is also not mandatory; if it is not present, the utility will call the output file *bundle.out*, and place it in the same path as *bundle.bin*.
- **-e** is the actual installation directory for the enforcer, which the utility uses to load the security keys from the keystore on the file system. If the enforcer is installed in this default path (C:\Program Files\NextLabs\Desktop Enforcer), and the Decrypt utility is running from its default path, this argument is not needed.

For example:

```
decrypt -b "C:\Program Files\Info Security\Dynamic Authorization Management for
SAP\Windows Desktop Enforcer\bundle.bin" -f bundle.txt -e "C:\Program Files\Info
Security\Dynamic Authorization Management for SAP\Windows Desktop Enforcer"
```

If you are using default values for install paths and file names, no arguments are required, and you can use the following command:

```
decrypt
```

If the Decrypt utility has been moved to a non-default location, only the **-e** argument is required.

To display the utility's help screen, type the following command:

```
decrypt -h
```

- 3 When the utility starts, it prompts you for the standard utility password, which is the same as the password required to stop the enforcer.

After the utility runs, the output text file is available for analysis.

## Decrypting the Bundle: Policy Controller for Java

As discussed in more detail in NextLabs Enforcer User's Guides, the Policy Server continuously updates enforcers, of all kinds, with any newly-defined or modified policies relevant to them. Each enforcer periodically sends a heartbeat message to the ICENet Server, which then checks whether any new or changed policies are in queue to be sent to that enforcer. If there are, it sends them, in the form of a file called *bundle.bin*; this file is referred to as a *policy bundle*. Each enforcer only retains one bundle at a time, which contains its most recent enforcement instructions. If the Policy Server sends another bundle with updated instructions—additional new policies to enforce, for example—that will overwrite the previous bundle.

All policy bundles sent from the Policy Server to enforcers are encrypted using standard SSL protocols. When they arrive at the enforcer, the enforcer authenticates them with digital certificates to ensure that they were indeed created by the Policy Server, and that they have not been modified by any other processes. This protects against the possibility of anyone deploying spoof policies designed to open security holes in your enterprise.

Bundle files are encrypted, but administrators can decrypt them for troubleshooting purposes. For this purpose, a special utility is included with the PolicyControllerJava.zip installation file. `Decrypt.bat` (for Windows) and `decrypt.sh` (for Solaris) are available in `<tomcat-home>\nextlabs\dpc` folder.

**Note:** To run this utility on Solaris, users must have executable permission for the file (`decrypt.sh`).

### Procedure

- 1 Run the appropriate utility, depending on your installation environment:
  - For Windows, enter `decrypt` in the command prompt.
  - For Solaris, enter `./decrypt.sh` in the command shell.
- 2 When the utility starts, it prompts you for the standard utility password, which is the same as the password required to stop the enforcer.
- 3 After the utility runs, the output text file, `bundle.out`, exists in the same folder as the `decrypt` utility.

---

## Managing Enforcer Policies

All policy enforcers are governed by a number of configuration settings that control such aspects as logging behavior, heartbeat rates, tamper-prevention passwords, and network configuration. These are assigned default values when you first install an enforcer, but they can be changed manually at any time. To simplify this, Administrator enables you to create named sets of configuration settings, which you can then assign to one or more enforcers in your network. These are referred to as *enforcer profiles*, and you manage them in Administrator, with the settings on the Policy Enforcer Configuration tab.



The settings controlled by enforcer profiles include the following:

- Which ICENet Server the enforcer will use to communicate with the Policy Server
- How often the enforcer sends heartbeat signals to the Policy Server to indicate it is operating normally

For more details on defining and using enforcer profiles, see the *NextLabs Control Center Administrator's Guide*.

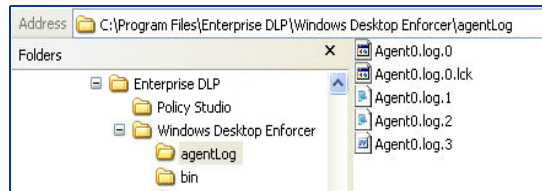
## Managing Event Logging

All enforcers maintain a set of local log files, which can be useful for troubleshooting or when communicating with NextLabs Technical Support. Both kinds of enforcers save their log files at

```
<InstallDir>\agentLog
```

Because of the tamper-resistance features, you must stop any enforcer before you can view or open its log files.

Enforcers maintain a log file called *Agent0.log.0* until the file reaches its specified maximum size, the file is saved as *Agent0.log.1*, and current logging continues in the original file name. Every time the file reaches its maximum size it is closed and saved, and the ending integer in all existing files is incremented. That is, the file ending in 0 will always contain the latest information, and the one ending in the highest integer will contain the oldest information.



### Logging Settings

You can configure the limit on the number and size of log files each enforcer maintains by editing the file *logging.properties* in the following directory:

```
<InstallDir>\config
```

The following properties control the number and size of log files maintained by each policy enforcer:

- **java.util.logging.FileHandler.count:** Specifies the maximum number of log files that can be archived at any given time. When this maximum is reached, the oldest file is discarded so that a new file can be started. Default = 10.
- **java.util.logging.FileHandler.limit:** Specifies the maximum size of each log file, in bytes. When this limit is reached, the current file is archived and a new log file is started. Default = 500K.

### Changing Logging Levels

You can configure enforcers to the following levels of event logging, in order of increasing verbosity:

- Severe
- Warning
- Info

- Fine
- Finest

By default, the logging level is set to Severe, but if you wish you can change this individually for each policy enforcer. It is controlled by three parameters in the logging.properties file:

- java.util.logging.ConsoleHandler.level
- com.bluejungle.level
- .level

You should always set `java.util.logging.ConsoleHandler.level` and `com.bluejungle.level` to the same value. The `.level` parameter represents a default level that will apply when no level has been defined for a logging component. [Figure 7-5](#) shows an example of this file.

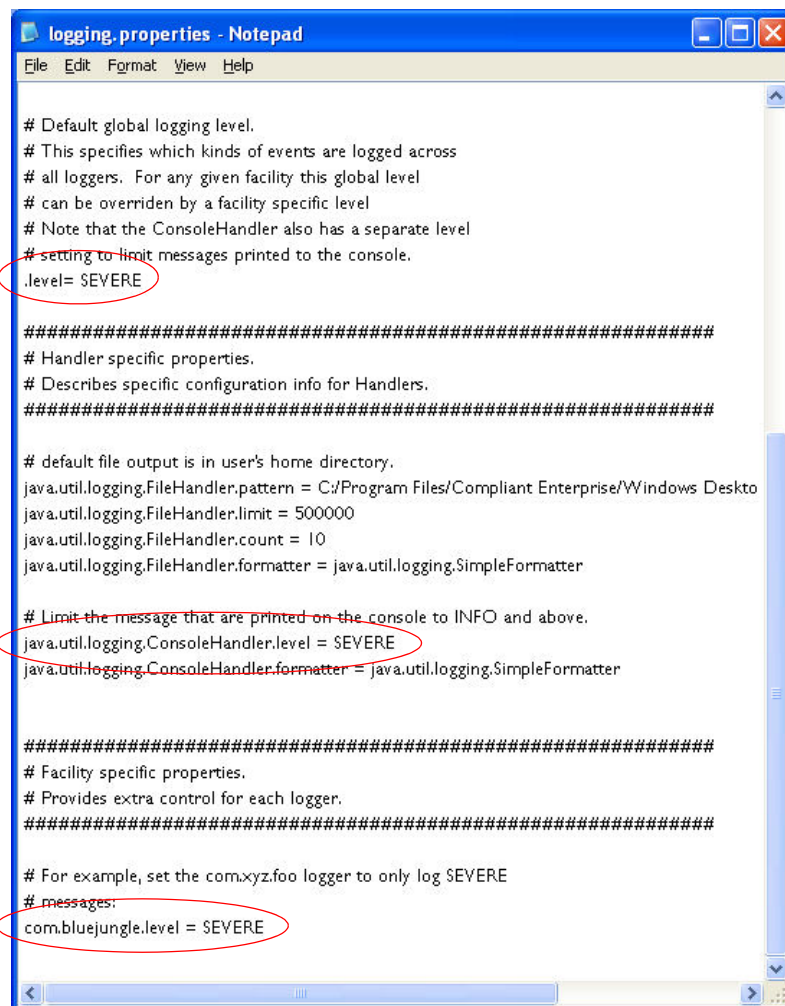


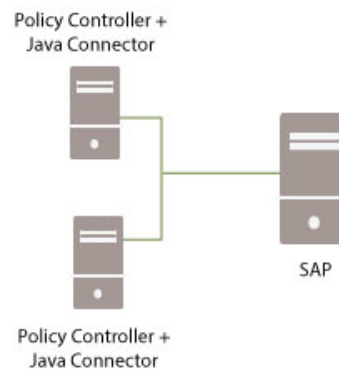
Figure 7-5: Changing Enforcer Logging Levels

## Load Balancing the Policy Controller

As is discussed in more detail in [Functional Integration During a Policy Check](#) on page 19, the Policy Controller is the component of the NextLabs system that receives policy evaluation queries from the SAP Agent, evaluates them, and returns a policy decision. Customers may choose to load balance this component to enhance system performance or implement redundancy in their system.

NextLabs supports load balancing for the following configurations:

- Server Policy Controller on Windows with the Java Connector
- Policy Controller for Java on Windows with the Java Connector
- Policy Controller for Java on Solaris with the Java Connector



*Figure 7-6: Load Balancing the Policy Controller*

There is no special configuration required for on the Policy Controller side. You only need to follow the standard procedures to install the Policy Controller on multiple hosts, then configure the Java Connector to point to the SAP server, following the procedures appropriate for the operating system.

**Note:** For more information about installing Policy Controllers, see *Installing Policy Controllers* section in the *NextLabs Control Center 7.7 Installation Guide*.

On the SAP server side, you must use the load balancing configuration options provided in the SAP Gateway Monitor (transaction SMGW). There are several options for how to configure load balancing in SMGW. The example instructions below explain one configuration.

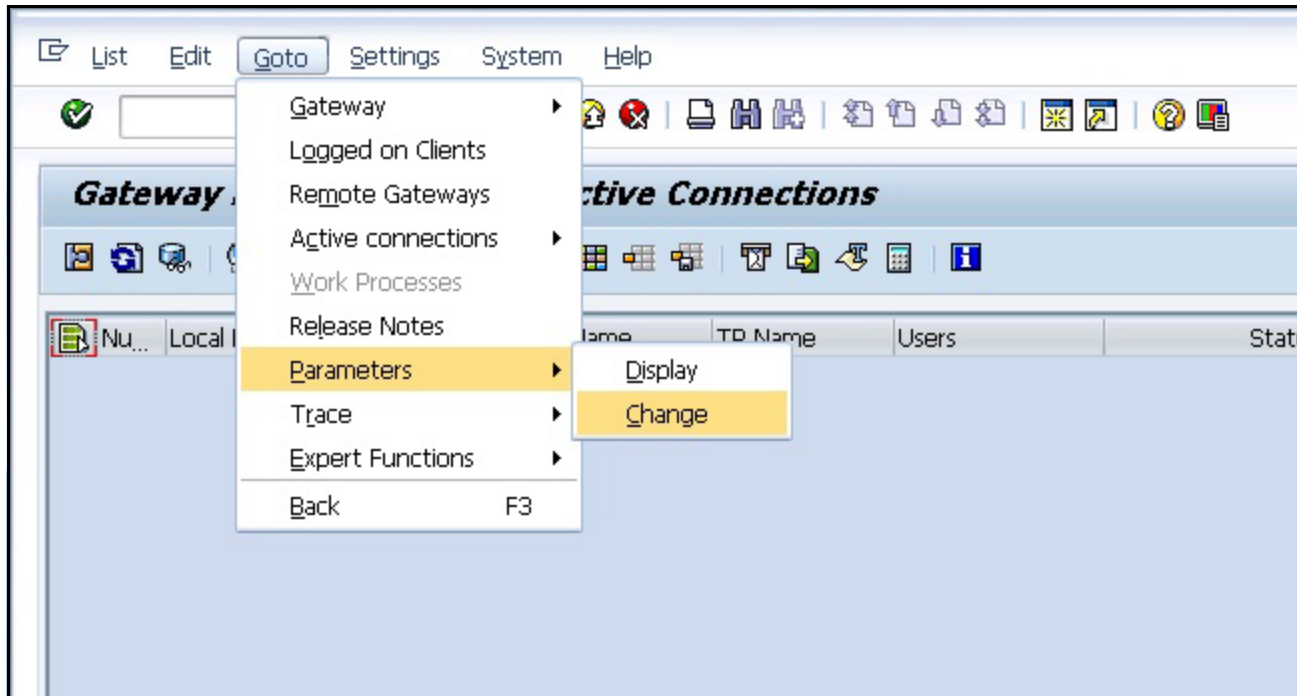
**Note:** For more information on other load balancing configurations in SAP, consult SAP documentation.

### Example: Load Balancing Configuration

The following procedure provides an example configuration for load balancing the Policy Controller using properties available in SAP Gateway Monitor.

### Procedure

- 1 Install and configure the Policy Controllers and Java Connectors on host to be load balanced. The Java SDK Properties files on both hosts should point to the same SAP server.
- 2 In the SAP interface, enter transaction `SMGW`. The SAP Gateway Monitor appears.
- 3 Select **Goto > Parameters > Change**.



*Figure 7-7: Changing Gateway Connection Parameters*

- 4 There are several Gateway parameters that can be used to configure load balancing. In this example, we use the `gw/reg_1b_level = 1` parameter for load balancing. In the level parameter, the following values are possible:
  - 0: No load balancing; the first free registered program is used.
  - 1: The program with the lowest counter is used. Every time a registered program is assigned a request, the counter is increased by 1.
  - 2: The program with the lowest load is used. The load is determined as defined by profile parameter: `gw/reg_1b_ip`.



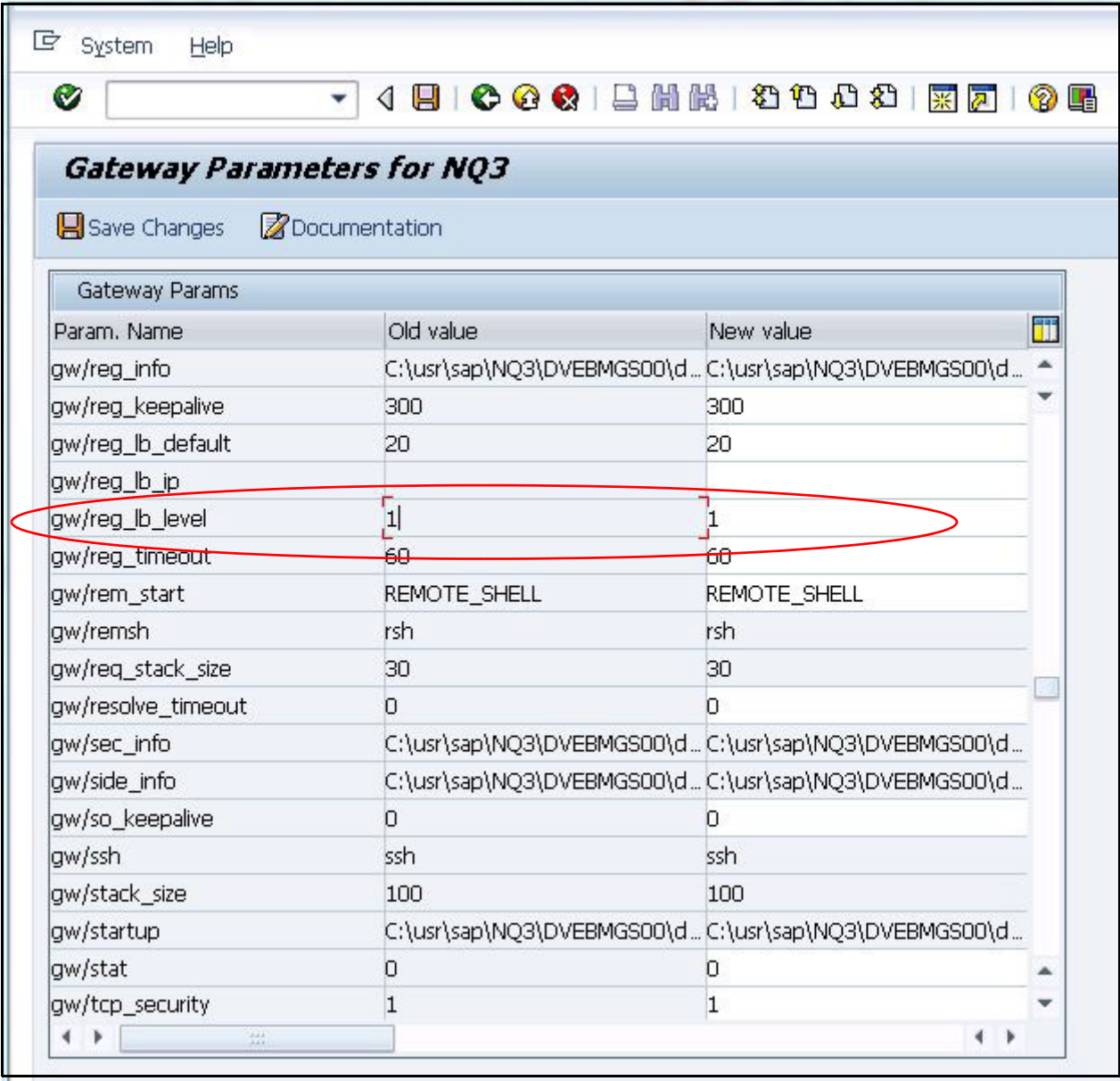


Figure 7-8: Using the gw/reg\_1b\_level in the SAP Gateway Monitor Parameters for Load Balancing

**Note:** For more information on how to configure other parameters for load balancing in this screen, consult SAP documentation.

5 Save the Parameter change.



This section explains the custom enhancements available for the NextLabs Dynamic Authorization Management for SAP.

Topics:

- [Custom Security Classification Identifiers](#)
- [Custom Enhancement Activations](#)
- [Dynamic User and Resource Attributes](#)
- [Custom Obligations](#)

The procedures in this section reference standard ABAP procedures and assume that users have ABAP development expertise. For more information on how to perform ABAP procedures, see the SAP documentation.

**Note:** The only modifications supported for the NextLabs namespace are officially released NextLabs product code. Customers should not store other modifications to code in the NextLabs namespace because it can result in installation and upgrade issues.

---

## Custom Security Classification Identifiers

One kind of custom enhancement you can perform is adding custom identifiers to the *Security Classification Maintenance* screen. This step may be necessary if your implementation hooks into a custom transaction. For example, you might want to configure a new business object as a primary Custom Identifier, such as a sales order.

### Configuring Custom Security Classification Identifiers

If your system includes the Entitlement Pack for BW, and you want to apply access control at the InfoObject, or data, level, you must create a custom identifier for each data object. For example, to control access to classified customer and materials data stored in BW, you need to create an identifier for customer and another for materials. An example of configuring a BW identifier is provided in [Adding a Custom Identifier](#) on page 256.

**Note:** For more information on creating a custom transaction, see [Custom Enhancement Activations](#) on page 261.

Security Identifiers are stored in the structures /NEXTLABS/SECIDT\_ECC (for SAP ECC), /NEXTLABS/SECIDT\_CFX (for SAP cFolders), /NEXTLABS/SECIDT\_BW (for SAP BW), or /NXLS4H/SECIDT\_S4HANA (for SAP S/4HANA). To create a custom identifier, add an append structure to the target structure, and then add the custom component (the new Identifier you want to create) to the append structure.

**Note:** To prevent custom enhancement activations from being overwritten by a NextLabs upgrade, you should not add your Append Structures to the NextLabs namespace. You should create them in your customer namespace.

What additional Security Identifiers would you like to display, other than Material or Document?

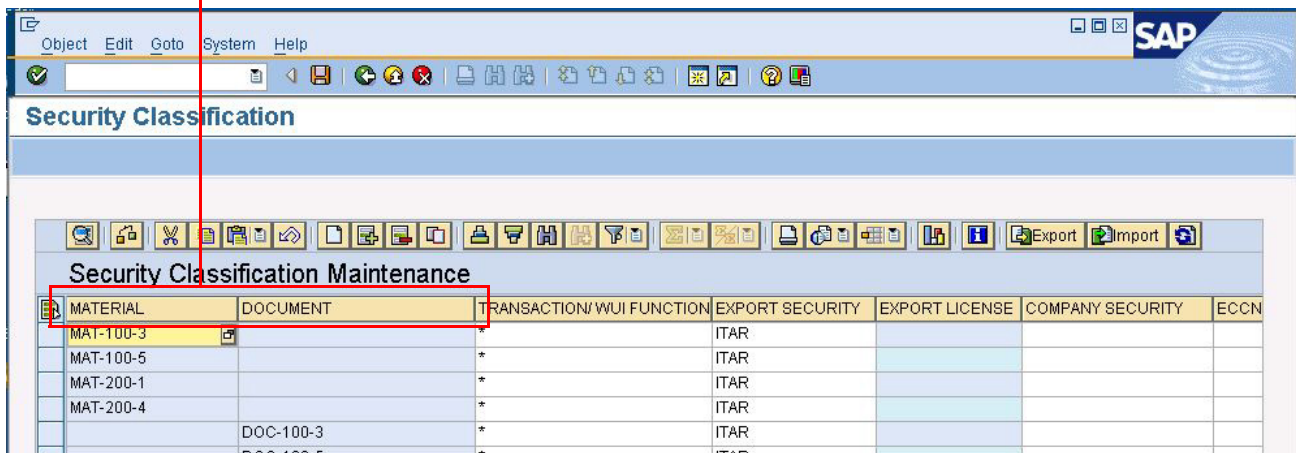


Figure 8-1: Customizing the SAP Security Classification Screen

### Adding a Custom Identifier

The first step to add custom identifiers to the Security Classification Maintenance table is to create append structures for the Include structures:

/NEXTLABS/SECIDT\_ECC (for SAP ECC), /NEXTLABS/SECIDT\_CFX (for SAP cFolders), /NEXTLABS/SECIDT\_BW (for SAP BW), or /NXLS4H/SECIDT\_S4HANA (for SAP S/4HANA).

**Note:** A similar procedure is used to add Security Classification fields to the Security Classification Maintenance table. This procedure (which is part of a normal system configuration) is discussed in [Adding Composite Keys and Classification Values](#) on page 78.

### Procedure

- 1 In the SAP interface, enter transaction SE11.
- 2 In the **Data Type** field, enter the structure name /NEXTLABS/SECIDT\_ECC, /NEXTLABS/ SECIDT\_CFX, /NEXTLABS/SECIDT\_BW, or /NXLS4H/SECIDT\_S4HANA, and click **Display**. The identifiers that are pre-configured display. [Figure 8-2](#) and [Figure 8-3](#) show examples of default identifiers in SECIDT\_ECC and SECIDT\_BW.

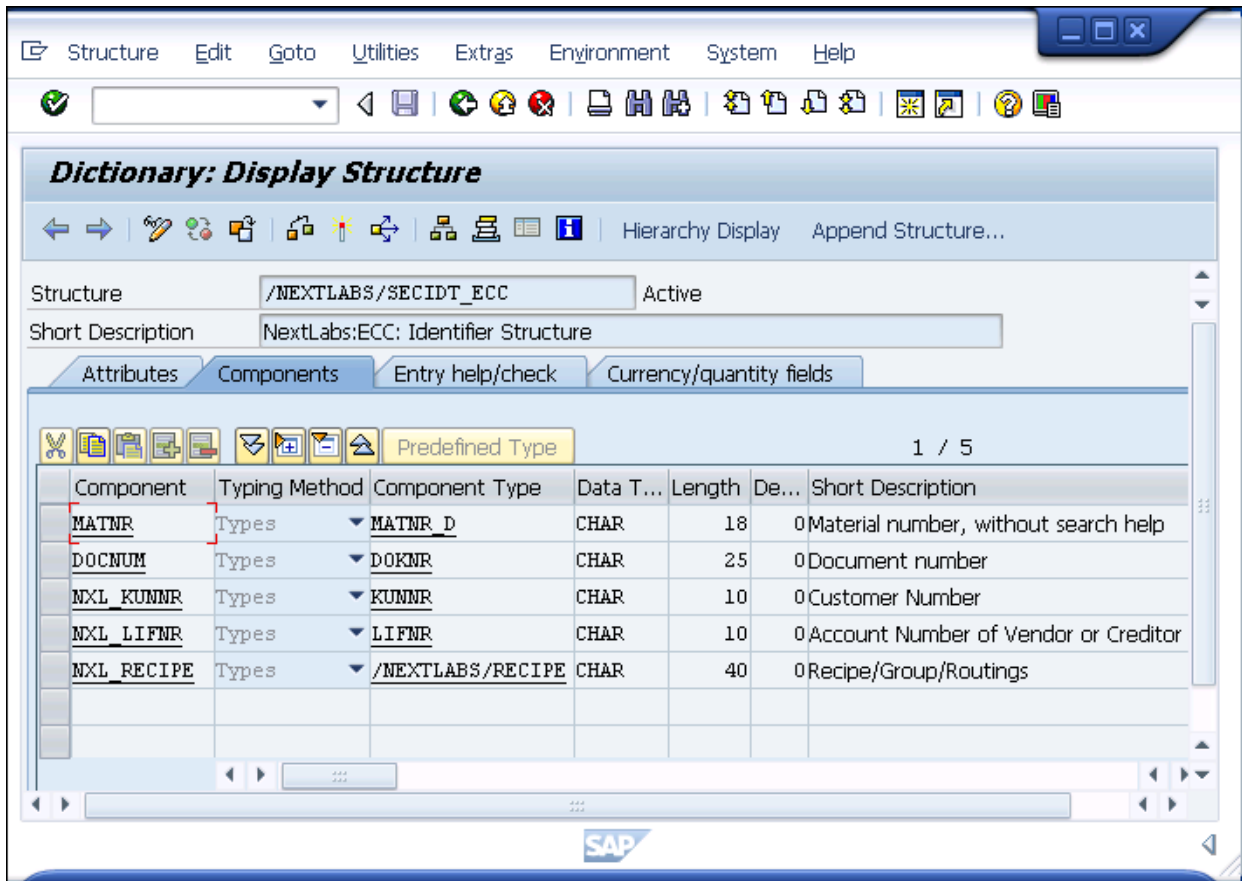


Figure 8-2: Default identifiers in /NEXTLABS/SECIDT\_ECC

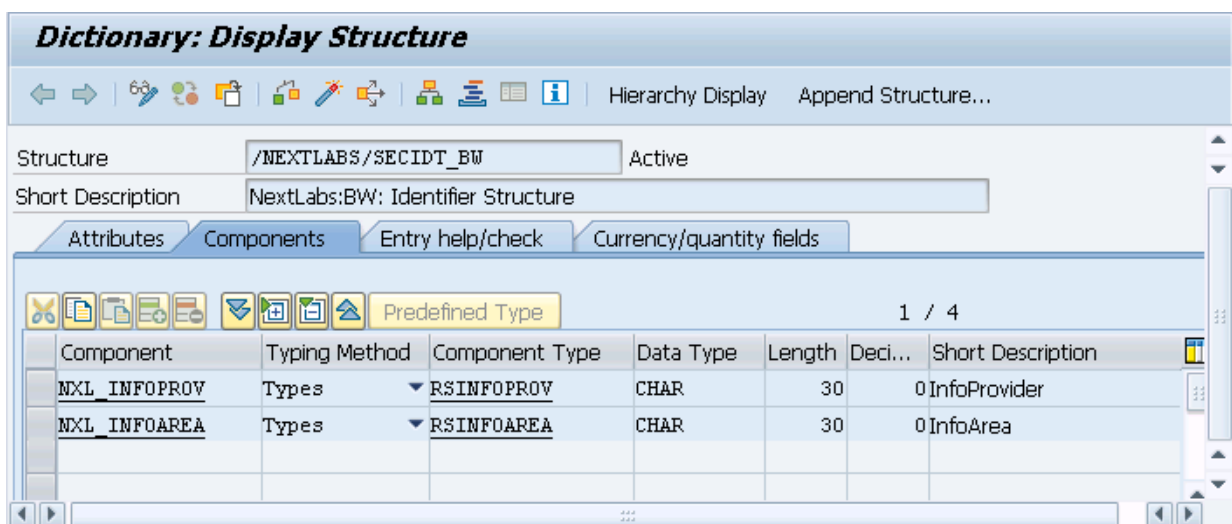


Figure 8-3: Default identifiers in /NEXTLABS/SECIDT\_BW

- 3 Click **Append Structure** on the toolbar.
- 4 The message "No append defined..." may display. Click **OK**.
- 5 Enter an Append Structure name. Click **OK**.

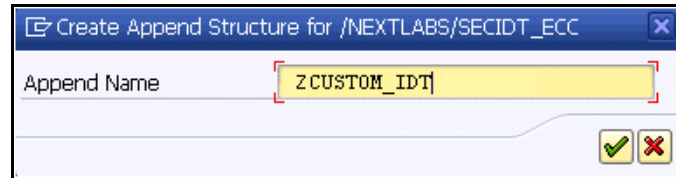


Figure 8-4: Name Append Structure

- 6 Enter a Description and add the Component.

SECIDT\_ECC example: Figure 8-5 shows an example of adding KUNNR of type KUNNR in SECIDT\_ECC.

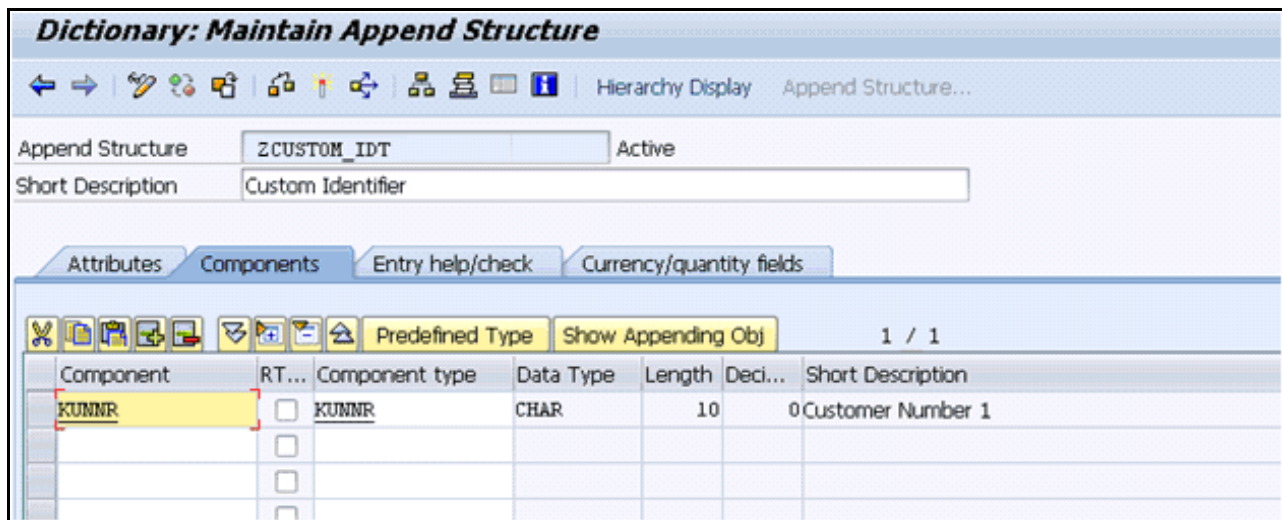


Figure 8-5: Defining a new identifier for SECIDT\_ECC

SECIDT\_BW example: Figure 8-6 shows an example of adding a component (identifier) named /BIC/CUST\_ID of type /BIC/OICUST\_ID in SECIDT\_BW. These values must match values in other sources, as described next.

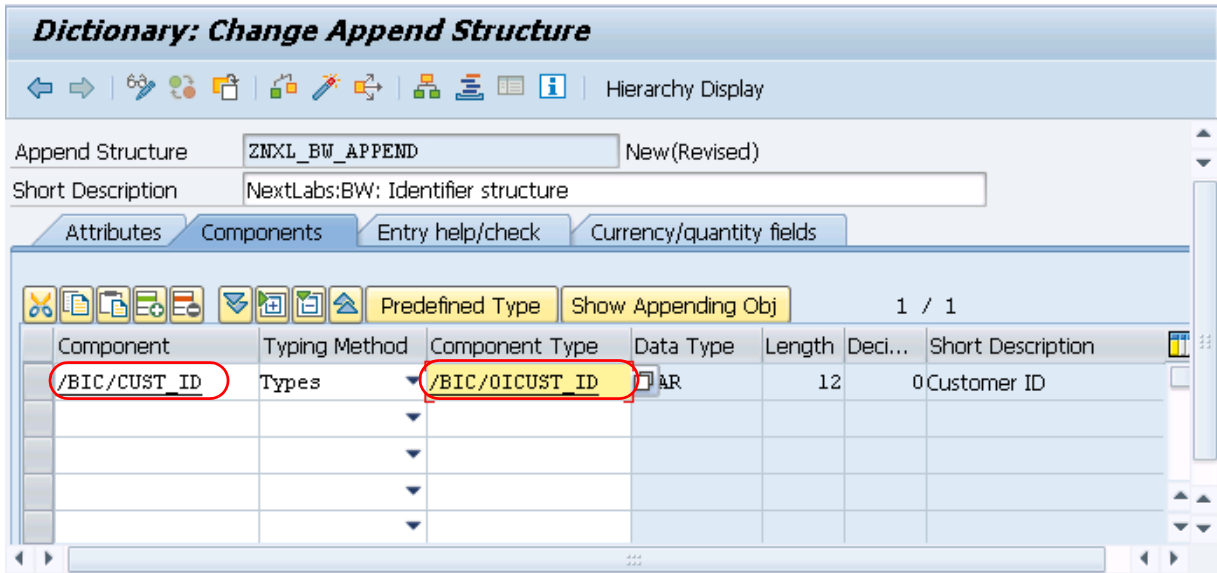


Figure 8-6: Defining a new identifier for SECIDT\_BW

The value you enter in **Component Type** must match the **Data Element** value for the InfoObject defined in the Data Warehousing Workbench (transaction RSA1), as shown in [Figure 8-7](#).

Also take note of the **SID Table** value highlighted in the figure. You need this value for the next configuration procedure.

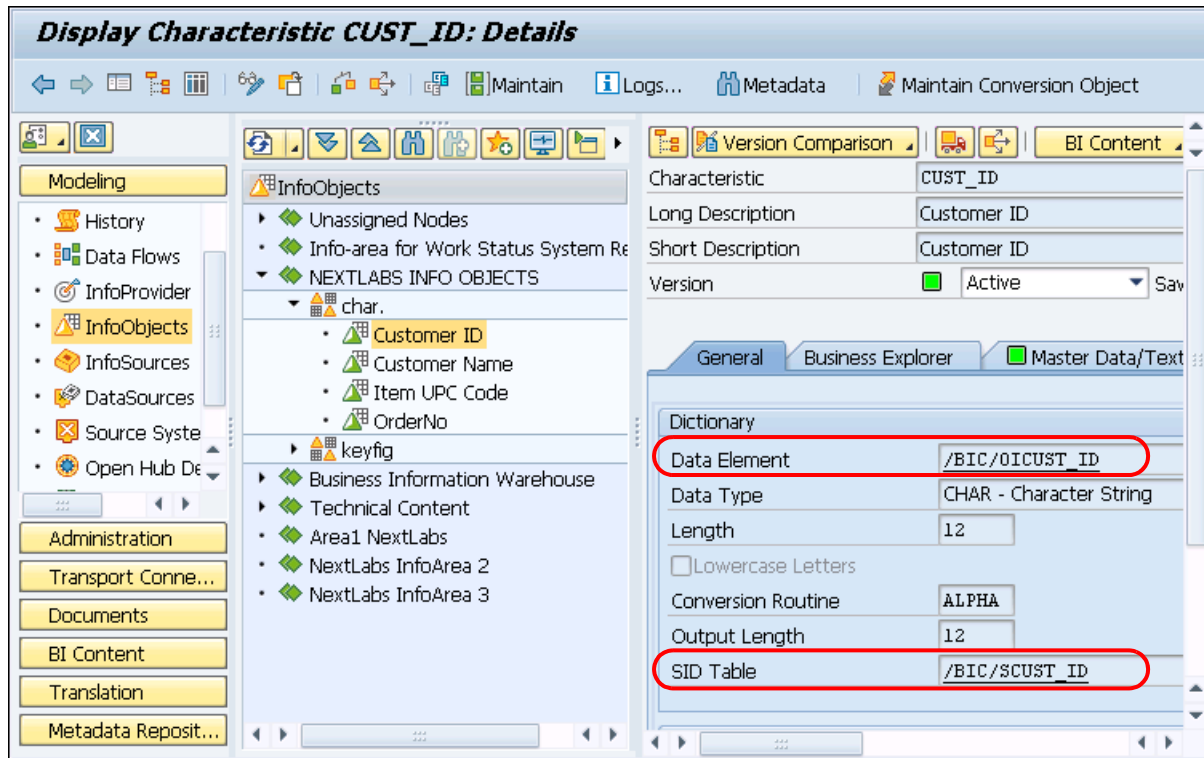


Figure 8-7: Details about the Customer ID InfoObject

The value you enter in Component must match the field name in the SID table. Use transaction SE11 to open the SID table (/BIC/SCUST\_ID in this example) to get the field value. Figure 8-8 shows the field name that you must enter as the Component value in SECIDT\_BW.



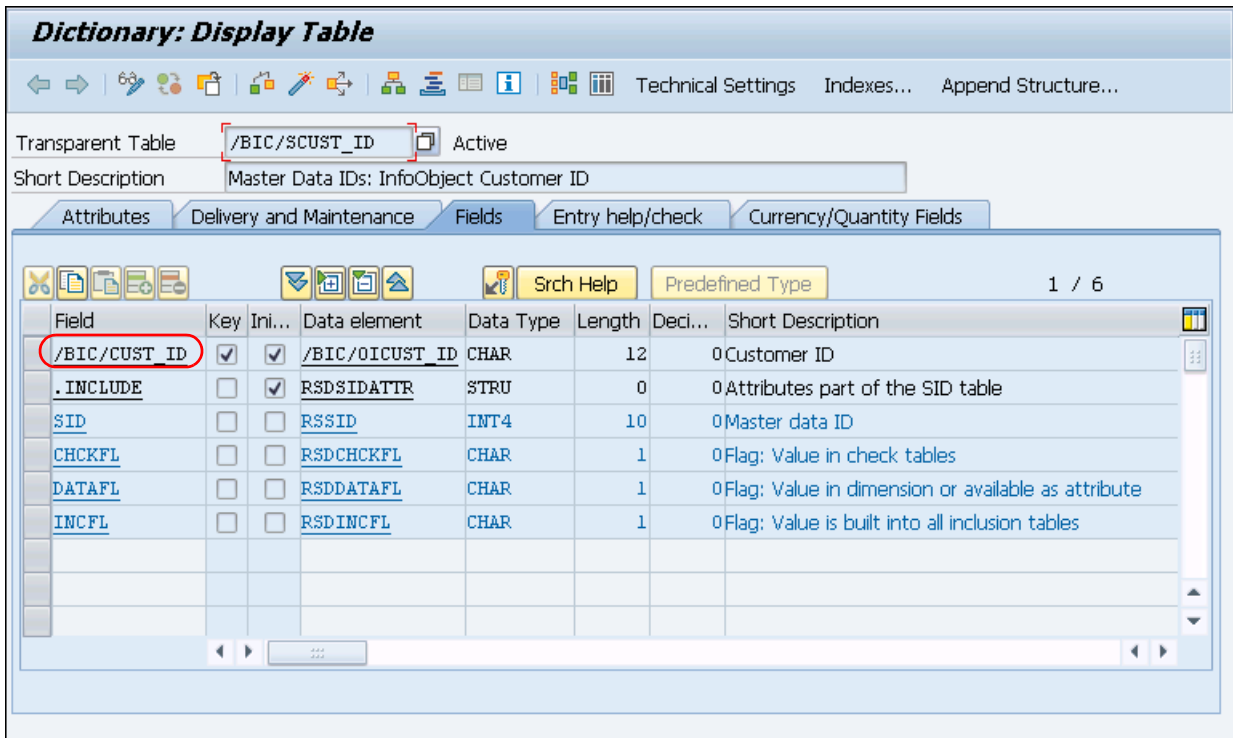


Figure 8-8: Fields in the SID table

- 7 Save and Activate the Append Structure.
- 8 After adding the identifier to the append structure, the next steps are the following:
  - Activate the security identifier for the transactions to intercept for policy checks. For more information, see [Defining How Multiple Security Classifications Should Be Applied](#) on page 116.
  - Specify the table from which the security identifier gets its values. For more information, see [Configuring Security Identifier/Composite Key Value Tables \(EPVAL\)](#) on page 86. For the identifier created in SECIDT\_BW, the value table you specify in EPVAL is the SID Table value for the InfoObject defined in the Data Warehousing Workbench, as shown in the previous figure.

## Custom Enhancement Activations

Another kind of custom enhancement is to build custom enhancement activations, meaning interception points into SAP transactions, other than the ones NextLabs supports. This section describes how to create a custom implementation of the NextLabs ABAP framework for a new transaction, using the example of adding NextLabs policy check into the SAP transaction CA0X - Routing.

The procedure is similar for other custom enhancements, but the details differ in regards to which Includes and Identifiers you need to change.

**Note:** To prevent custom enhancement activations from being overwritten by a NextLabs upgrade, you should not store them in the NextLabs namespace. You should maintain them in your customer namespace.

## Example: Creating a Custom Enhancement Activation Based on Routing

### Procedure

- 1 Locate a NextLabs transaction Include that you want to base your transaction on, in this example, `/NEXTLABS/TRANS_PROCESS_CA0X`.
- 2 Open the transaction, and identify all the Include programs that require modification.

Typically, the main transaction Include has several include programs, some of which are generic across SAP transactions (and thus do not need to be modified), and some of which are transaction-specific and must be modified.

The Includes you must modify typically have a Security Identifier that must be passed to the correct component of the Security Identifier structure. For example, the transaction you are using as your example may reference a material, but your transaction may need to reference a document or another Identifier.

Because you want to modify a new version of each of these includes, you will need to create a z version of it in your customer namespace. For our example transaction `TRANS_PROCESS_CA0X`, we will need a Z version of the following includes:

- The transaction include: `/NEXTLABS/TRANS_PROCESS_CA0X`

```

ENHANCEMENT-POINT CP_DYNP_TITLE_FILL_01 SPOTS ES_SAPLCP04 STATIC .
*$$$-Start: CP_DYNP_TITLE_FILL_01-----$$$$
ENHANCEMENT 1 ZROUTING_ENHANCE.           "active version
DATA: v_rattr TYPE REF TO /nextlabs/check_instance.
      v_rattr = /nextlabs/check_instance=>get_instance_rou( ).
IF v_rattr IS NOT INITIAL.
  include /NEXTLABS/TRANS_PROCESS_CA0X.
endif.
ENDENHANCEMENT.

```

Figure 8-9: Include for `/NextLabs/TRANS_PROCESS_CA0X`

- The main include: `/nextlabs/rout_data_init`

```

Include /NEXTLABS/TRANS_PROCESS_CA0X Active
8 * Author : NextLabs *
9 * Date : 2010-07-28 *
10 *-----*
11 * Description: *
12 * This Program will be called from the User-exit/enhancement *
13 * and checks the policy associated with the Routing transaction *
14 * and apply the decision *
15 *-----*
16 * Tables Used : SELECT UPDATE INSERT DELETE *
17 * N/A *
18 * Input/Output Files: *
19 * Global Classes used: *
20 * N/A *
21 * Includes: *
22 * N/A *
23 *-----*
24 * Program History *
25 *-----*
26 * Date Change Request SAP Rel Description *
27 *-----*
28 * 2010-07-28 6.0/6.3 Initial creation *
29 *-----*
30 *-----*
31 *** Data Declarations ***
32 *-----*
33 INCLUDE /nextlabs/global_data_init.
34 INCLUDE /nextlabs/rout_data_init.
35 *-----*
36 *** Begin Processing logic ***
37 *-----*
38 IF NOT s_secidt-matnr IS INITIAL.
39 INCLUDE /nextlabs/prog_generic.
40 INCLUDE /nextlabs/func_clasfn.
41 ENDIF.
42 INCLUDE /nextlabs/check_def_param.
    
```

1

Figure 8-10: Identifying Includes that Require Modification

- 3 Perform the following steps to create a copy of each Include program you plan to modify in your customer namespace:
  - Enter transaction SE38.
  - Create a new program name for the new include program, for example: <your namespace>/ZTRANS\_PROCESS\_CA0X.
  - Select the option to base the new program on an existing program, where the name of the program is the include in the NextLabs namespace, for example: /NEXTLABS/TRANS\_PROCESS\_CA0X.
  - Repeat steps for all other Includes you intend to modify. In our example, this means creating a z version of /nextlabs/rout\_data\_init.
- 4 Locate the main zinclude program for the custom transaction (Z/TRANS\_PROCESS\_CA0X). For each Include you plan to modify within it, remove the /NextLabs/ include and insert the Z version you created in se38. In our example, we would be removing the include /nextlabs/rout\_data\_init and inserting zrout\_data\_init.
- 5 Activate all newly created Z Includes.
- 6 Locate the Enhancement Spot where you will insert the new enhanced transaction, as in Figure 8-11.

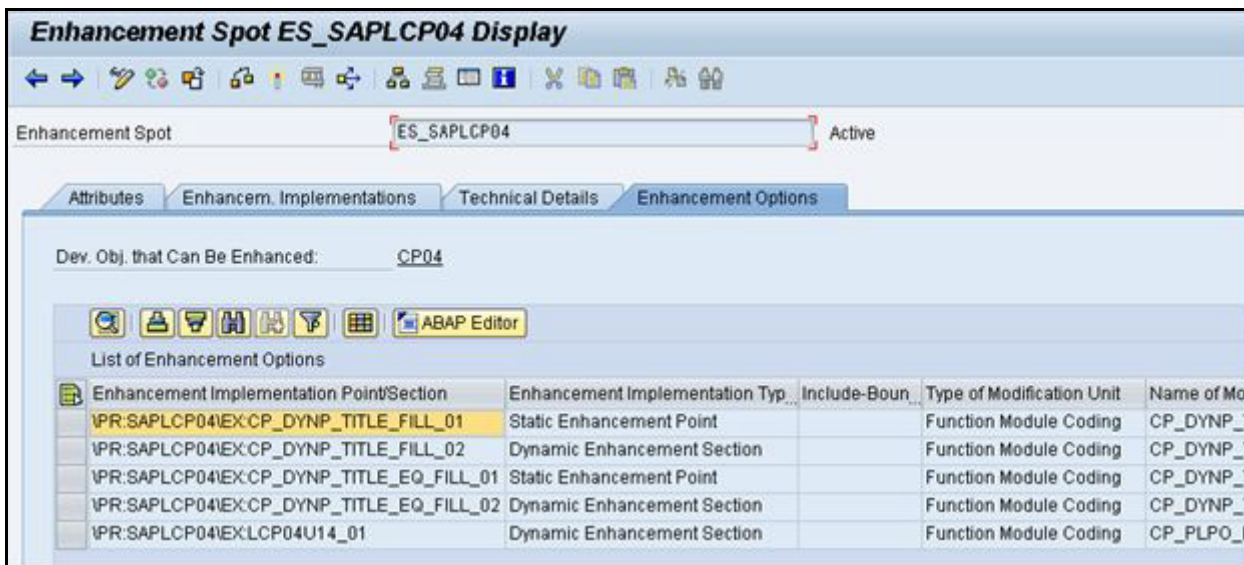


Figure 8-11: Enhancement Spot

- 7 Create an enhancement implementation in your customer namespace, not in the NextLabs namespace.
- 8 Insert the new z (customer namespace) version of the transaction Include code into the User Exit/BADI. For example:

```
include ZTRANS_PROCESS_CA0X
```

- 9 Double-click the include program (`include ZTRANS_PROCESS_CA0X`). The z version should display for all includes you plan to modify.

**Note:** The actual includes that display, and the code that must be modified, will vary based on the customer enhancement you are creating.

```

29 | *****
30 | *****
31 | *** Data Declarations ***
32 | *****
33 | INCLUDE /nextlabs/global_data_init.
34 | INCLUDE zrout data init.
35 | *****
36 | *** Begin Processing logic ***
37 | *****
38 | IF NOT s secidt-matnr IS INITIAL.
39 |     INCLUDE /nextlabs/prog_generic.
40 |     INCLUDE /nextlabs/func clasfn.

```

*Figure 8-12: Generic and Transaction-Specific Code*

- 10 To modify the transaction-specific Include program, double-click the Include program to display it.
- 11 Change the Security Identifier to reference the correct identifier. In the example below (Item 2, line 38), we are referencing material, but this might need to change for another transaction. Ensure all references are to the correct Security Identifier for the transaction.



```

Include ZROUT_DATA_INIT Active
1  *-----*
2  *& Include      /NEXTLABS/ROUT_DATA_INIT
3  *-----*
4  *****
5  * Program for Routing (CA01/02/03) Transaction NextLabs Policy Check
6  *-----*
7  * OBJECT   : /NEXTLABS/MAT_INIT
8  * Author   : NextLabs
9  * Date     : 2010-08-18
10 *-----*
11 * Description:
12 * This Program defines Routing Transaction global data that will be used
13 * through the policy Evaluation Process
14 *-----*
15 * Tables Used :                SELECT UPDATE INSERT DELETE
16 * N/A
17 * Input/Output Files:
18 * Global Classes used:
19 * N/A
20 * Includes:
21 * N/A
22 *****
23 * Program History
24 *-----*
25 * Date      Change Request SAP Rel  Description
26 *-----*
27 * 2010-08-18    6.0/6.3  Initial creation
28 *****
29 *****
30 *** Data Declarations ***
31 *****
32
33
34 data s_matnr type matnr.
35
36 * Initialization
37 s_matnr = matnr.
38 s_secidt-matnr = matnr.

```

Figure 8-13: Transaction-Specific Modification of Includes

12 Save and activate.

### Example: Generic Include

The include `/nextlabs/global_data_init` contains the Global data declaration for any NextLabs transaction interception code. It defines the data needed across the NextLabs execution through the transaction execution process. This include is standard across transactions and does not have to be changed for other customer BADI implementations.

```

ABAP Editor: Display Include /NEXTLABS/GLOBAL_DATA_INIT
-----
Include: /NEXTLABS/GLOBAL_DATA_INIT Active
-----
25 * Date      Change Request SAP Rel  Description
26 *-----*
27 * 2010-09-30      6.0/6.3  Initial creation
28 *****
29
30 [ ] *****
31 *** Data Declarations ***
32 *****
33 * Constants
34   CONSTANTS: c_default(1) TYPE c VALUE '*',
35              c_ecc(3)   TYPE c VALUE 'ECC',
36              c_run(3)  TYPE c VALUE 'RUN',
37              c_edms(7) TYPE c VALUE 'EASYDMS'.
38
39 * structure declarations
40   DATA: s_secidt TYPE /nextlabs/secidt,
41          i_return TYPE STANDARD TABLE OF bapiret2,
42
43          i_return_cp TYPE STANDARD TABLE OF bapiret2,
44          nxl_s_return_cp TYPE bapiret2,
45
46          s1_tcode TYPE /nextlabs/uifuncname.
47   DATA s_generic_attr TYPE /nextlabs/generic_attr.
48   DATA: ret_tcode TYPE REF TO /nextlabs/check_instance.
49
50   DATA: nxl_s_transtruct TYPE typename,
51          nxl_s_sectrans  TYPE /nextlabs/char2000.
52
53 [ ] *****
54 *** Initialize Variables ***
55 *****
56   CLEAR s_secidt.
57   CLEAR s_generic_attr.
58   CLEAR nxl_s_transtruct.
59   CLEAR nxl_s_sectrans.
60   REFRESH i_return.
    
```

Figure 8-14: Include `/nextlabs/global_data_init`

---

## Dynamic User and Resource Attributes

There may be use cases where customers want to enforce controls on user or resource attributes that are not (or cannot be) enrolled into the NextLabs Control Center. For example, a customer may need to enforce access controls based on physical location or another attribute that can vary dynamically.

To address this requirement, Dynamic Authorization Management for SAP supports the implementation of dynamic user and resource attributes. ABAP developers can insert dynamic look-ups into policy checks to retrieve user and resource information from a designated store at the point of policy evaluation. The mechanism that collects and stores the user or resource attribute are to be determined and developed by the customer. For example, you could build a program to automatically gather a user attribute (such as physical computer location) and store it as a session variable. Or, you could create a user interface that prompts users to enter attributes.

This section provides information about how to configure the BADI method to collect these attributes from the storage location you designate. It does not provide guidelines for the solution you develop to collect and store those attributes (which are dependent on the particulars of your environment and business requirements).

### Configuring Enhancement Activations for Dynamic Attributes

This section provides step-by-step instructions for the using the BADI Definition for dynamic user and resource attributes, including creating the implementation, the supported methods, and the structure for parameters for the method.

#### Procedure

- 1 In the SAP interface, enter transaction SE19. The *Business Add-Ins* screen appears.
- 2 Select a BADI type and enter the BADI Name `/NEXTLABS/ENH_DYNATT`. Click **Create Implementation**.
- 3 Enter an Implementation Name.
- 4 In Implementation Short Text, enter a description.
- 5 Click the **Interface** tab.



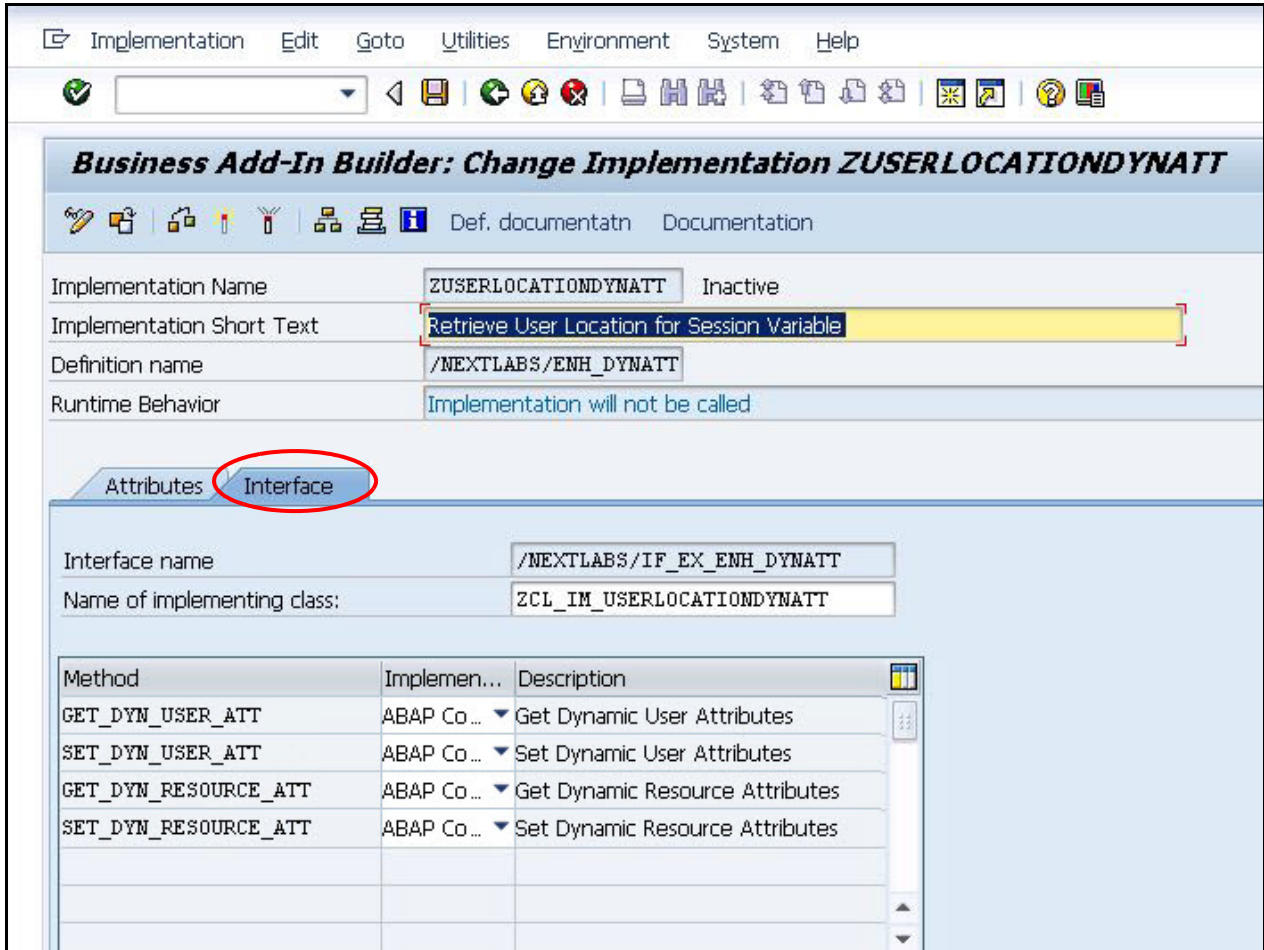


Figure 8-15: Methods for /NEXTLABS/ENH/DYNATT

6 GET and SET methods are available, as described next.

Table 8-1: Available Methods

Method	Explanation
GET_DYN_USER_ATT	Retrieves user attributes written to a storage location (defined by the customer).
SET_DYN_USER_ATT	Writes user attributes to a storage location (defined by the customer).
GET_DYN_RESOURCE_ATT	Retrieves resource attributes written to a storage location (defined by the customer).
SET_DYN_RESOURCE_ATT	Writes resource attributes to a storage location (defined by the customer).

- 7 Select the method for which to write code, then enter the code. Refer to [Table 8-2](#) for supported parameters.

*Table 8-2: Parameters for GET and SET methods*

Parameter	Description
I_USER_ID	User ID (Import parameter)
I_S_GENERIC_ATTR	Generic attributes for transaction interception (Import parameter) <ul style="list-style-type: none"> <li>• IPADD: IP Address</li> <li>• ADDON: NextLabs Add on</li> <li>• Action: NextLabs Action name</li> <li>• ACC_HIER_T: Access Control Context hierarchy</li> <li>• PBSC_ATTR: PBSC Attributes</li> <li>• SECENH: Composite key information</li> <li>• FILTER: Bypass message processing flag</li> <li>• USERID: User ID</li> </ul>
I_S_TCODE	Transaction code or UI function code (Import parameter)
I_S_SECIDT	Security identifiers work area (Import parameter)
E_USER_ATTR_T	User attribute table (Export parameter)
E_RESOURCE_ATTR_T	Resource attribute table (Export parameter)

The following sample code in GET\_DYN\_USER\_ATT shows an example implementation.

```

**Start
**Attribute name - Location
s_attrs-key = 'Location'
Call Function 'ZGet_Location'
Importing
    i_userID = syuname
Exporting
    e_location = s_attrs_value
APPEND s_attrs TO e_user_attr_t.

```

- 8 After you have finished entering code, **Save** and **Activate** the method.
- 9 **Save** and **Activate** the Implementation.

## Referencing Dynamic Attributes in Policies

Dynamic attributes can be referenced the same way as any attribute in a Policy Studio policy. Reference the User Key and Value in the Properties field of the User components.

---

## Custom Obligations

All NextLabs products support obligations, which are programs that can be configured to run after a NextLabs policy evaluation. Dynamic Authorization Management for SAP has obligations that can be associated with policies (for example, see [Configuring SAP Obligations](#) on page 70). Like other NextLabs products, Dynamic Authorization Management for SAP also enables you incorporate custom-defined obligations with a NextLabs policy.

Dynamic Authorization Management for SAP has the unique feature of supplying a BADI Definition ABAP developers can use to write obligations and associate them with NextLabs policies. This feature provides additional flexibility to extend a policy enforcement event to include any process that can be written in ABAP code, for example, to automate work flows (such as an approval process), to run a custom-designed program (for example, to scramble long text fields), or to automatically execute a standard SAP transaction.

This section explains how to configure a BADI definition, the framework NextLabs provides, including the base method and its parameters, and provides an example. Keep in mind, however, that developers can include any ABAP code they want and the information in this section is not exhaustive.

### Supported Platforms

Custom ABAP obligations are only supported for SAP version 6.0 and Dynamic Authorization Management for SAP version 6.6 and higher.

### Configuring Enhancement Activations for Custom Obligations

This section provides step-by-step instructions for the using the BADI Definition for Custom Obligations, including creating the implementation, the supported methods, and the structure for parameters for the method.

#### Procedure

- 1 In the SAP interface, enter transaction `SE19`. The *Business Add-Ins* screen appears.
- 2 In the Create Implementation area, select a BADI type and enter the BAdI Name `/NEXTLABS/ENH_CUSOBL`.

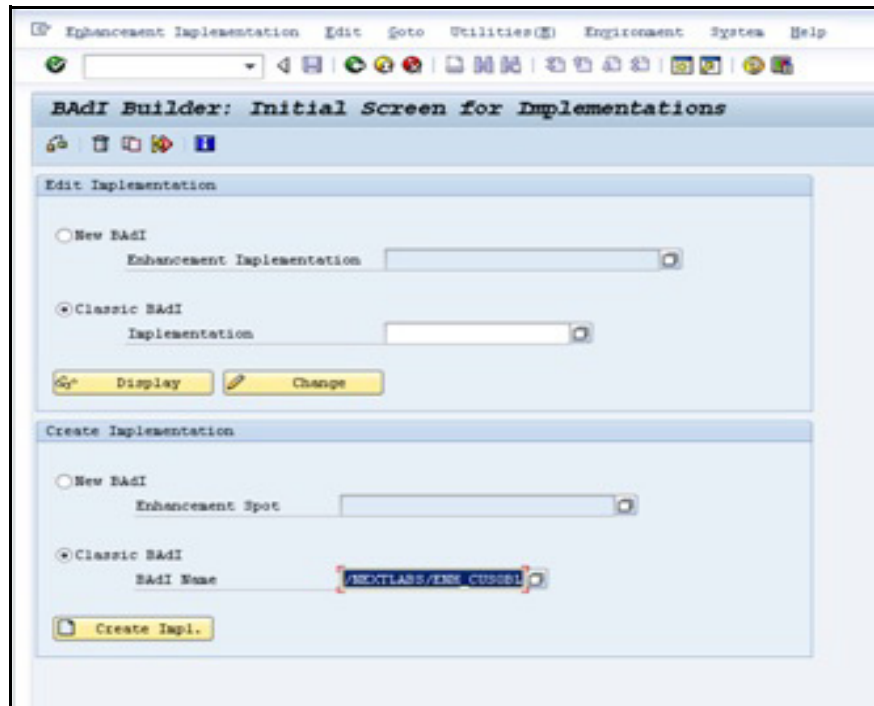


Figure 8-16: Creating an Implementation

- 3 Click **Create Impl.** The system will prompt you for the Implementation Name. Enter an Implementation name and click **OK**.

**Note:** The Implementation Name in this example is for demonstration purposes only. Follow the naming conventions recommended for your system.

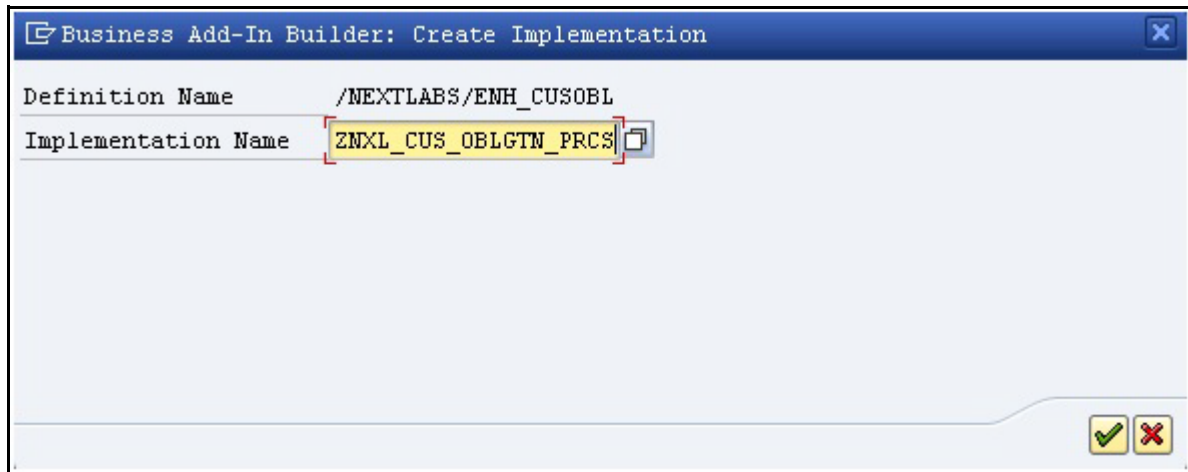


Figure 8-17: Naming the Implementation

4 Enter a description for the BADI and click the Interface tab.

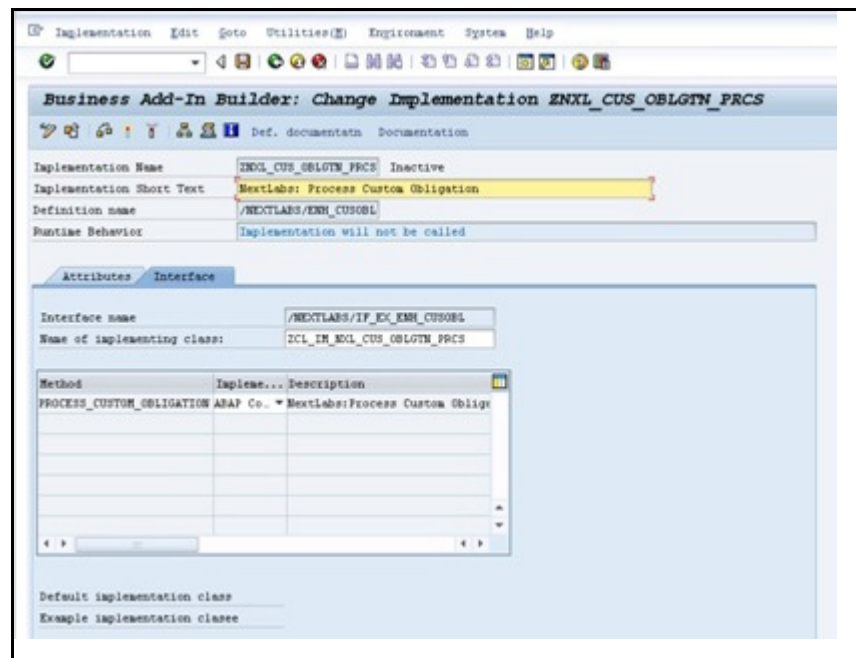


Figure 8-18: Accessing the Interface Tab

5 Locate and double-click the `PROCESS_CUSTOM_OBLIGATION` method. This is the primary method for this BADI. The following sections describe the import and export parameters and their components.

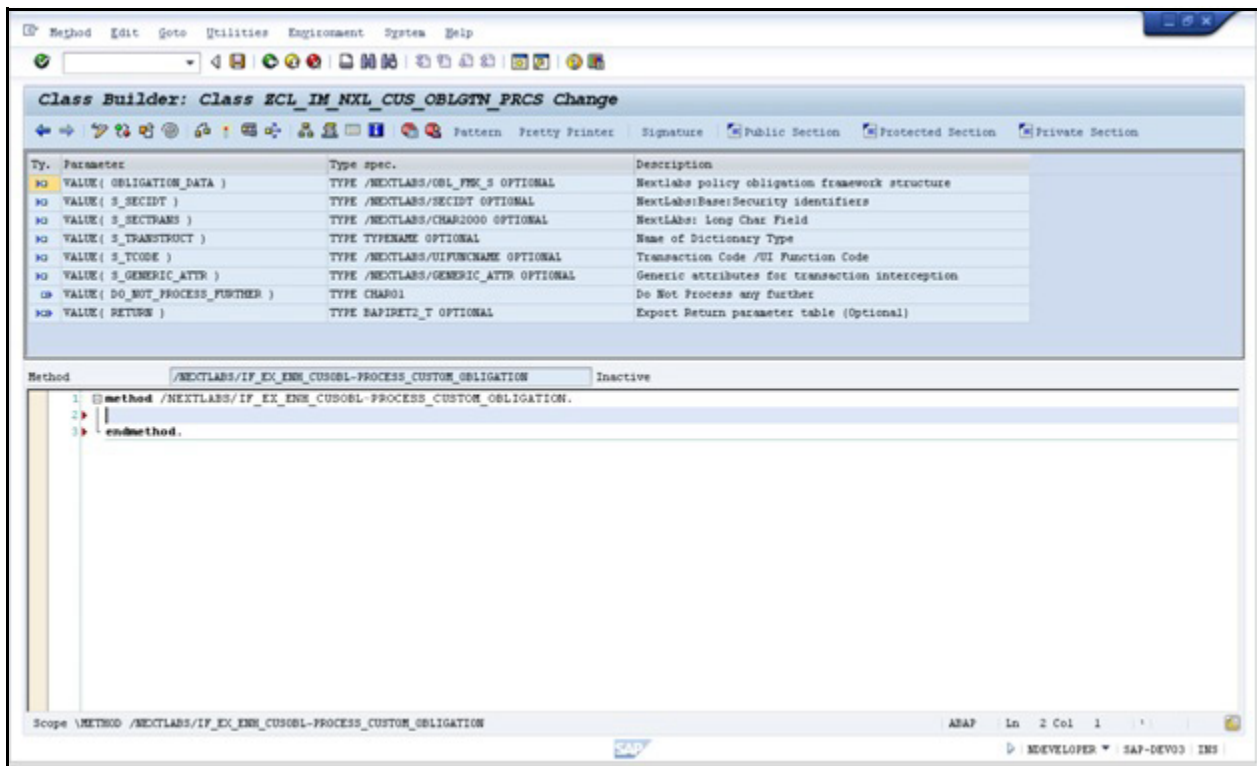


Figure 8-19: Process\_Custom\_Obligation Method

- 6 Define the obligation using the import and export parameters described below ([Process Custom Obligation Method](#) on page 274).
- 7 After entering the code, Save and Activate the method.
- 8 Save and Activate the BADI implementation.

## Process Custom Obligation Method

The following sections describe the Import and Export parameters and their components for the `PROCESS_CUSTOM_OBLIGATION` method.

### Import Parameters

[Table 8-3](#) and [Table 8-4](#) describe the available import parameters and their components for the method `PROCESS_CUSTOM_OBLIGATION`.

**Table 8-3: Import Parameters for Method PROCESS\_CUSTOM\_OBLIGATION**

Parameter	Description	Type	Components
OBLIGATION_DATA	To retrieve required information on what policy was triggered and what obligation data was passed.	/NEXTLABS/OBL_FMK_S	<ul style="list-style-type: none"> <li>Response: Policy evaluation response (Allow or Deny)</li> <li>OBL_TYPE: Policy Obligation type (S for Standard Obligation, C for Custom Obligation)</li> <li>POLICY_OBL_NAME: Policy Obligation Name (SAPMSG, SAPMCL, CUSTOM)</li> <li>POLICY_ID: Policy Name</li> <li>OBL_COUNTER: Policy Obligation Counter (order of policy in the list)</li> <li>OBL_DETAIL: Policy Obligation Details (key value/pair information specific to the obligation)</li> </ul>
S_SECIDT	To retrieve Security identifier data	/NEXTLABS/SECIDT	This parameter supplies Security Identifier information, such as: <ul style="list-style-type: none"> <li>MATNR: Material Number</li> <li>DOCNUM: Document Number</li> </ul>
S_SECTRANS	To retrieve Dynamic Structure Data	/NEXTLABS/CHAR2000	Dynamic Structure Data
S_TRANSSTRUCT	To retrieve Dynamic Structure Name	TYPENAME	Dynamic Structure Name
S_TCODE	To retrieve Transaction code	/NEXTLABS/UIFUNCNAME	Transaction code for the request
S_GENERIC_ATTR	To retrieve specified attribute in the policy evaluation	/NEXTLABS/GENERIC_ATTR	<ul style="list-style-type: none"> <li>IPADD: IP Address</li> <li>ADDON: NextLabs Add on</li> <li>Action: Action name</li> <li>ACC_HIER_T: Access Control Context</li> <li>PBSC_ATTR: PBSC Attributes</li> <li>SECENH: Composite Key information</li> <li>FILTER: Filter Flag for this request</li> <li>USERID: User ID</li> </ul>

### Export Parameters

Table 8-4 describes the export parameters and their components (where applicable) for the method PROCESS\_CUSTOM\_OBLIGATION.

*Table 8-4: Export Parameters for Method PROCESS\_CUSTOM\_OBLIGATION*

Parameter	Description	Type
DO_NOT_PROCESS_FURTHER	If no additional obligations (standard or custom) should be triggered from this execution onwards, set this flag and NextLabs will ignore other obligations that might be present.	CHAR01
RETURN	Sets return table to be used. For example, in the filter records example ( <a href="#">Example: Custom Obligation</a> on page 276) RETURN contains the obligation parameters set from the Policy Obligation. This table is used in the main calling program to process the obligation.	BAPIRET2_T

### Example: Custom Obligation

The following code provides an example of a custom obligation that filters records that display in a table (obligation name SAPFLT).

```

CONSTANTS badi_c_allow(5) TYPE c VALUE 'allow'.
DATA badi_return TYPE bapiret2.

**Begin Logic
CASE obligation_data-policy_obl_name.
WHEN 'SAPFLT'."Filter Records
IF obligation_data-response <> badi_c_allow. "Check Obligation Overall Response
badi_return-type = 'E'.
badi_return-message = 'Filter Records:ITAR:01B'.
APPEND badi_return TO return.
CLEAR badi_return.
do_not_process_further = 'X'.
ENDIF.
ENDCASE.
ENDMETHOD.

```

### Configuring a Custom Obligation

You must create a custom obligation and insert it into the configuration.xml file located with the NextLabs Control Center. This step enables you to assign the obligation to a policy in Policy Studio. These procedures are documented in the Control Center documentation.



# A

# Implementation Reference for ECC

This section provides an implementation reference to the SAP ECC transactions that can be intercepted.

Topics:

- [Transactions](#)
- [Implementations](#)

---

## Transactions

[Table A-1](#) provides an alphabetical list of the ECC transactions supported for interception by the SAP Dynamic Authorization Management.

For each transaction, the table includes the following information:

- Security Identifier indicates which object is subject to policy checks.
- Interception Type indicates the type of interception.
  - Common Interception: the interception of multiple transactions through a common, or central, interception point, and the triggering of policy checks for a specific object. With common interception, you define a few enhancement implementations to intercept many transactions that have a particular object, such as material, associated with those transactions.
  - Transaction Specific: the interception of a specific transaction. With this type of interception, you generally define one enhancement implementation per transaction or per set of related transactions (for example, one implementation to intercept MM01, MM02, and MM03). In some cases, policy checks may be triggered for multiple objects per transaction.
- Implementation ID provides a link to the implementation details.

*Table A-1: Supported transactions*

Transaction Code	Description	Security Identifier	Interception Type	Implementation ID
C201	Create Master Recipe	NXL_RECIPE	Common Interception	<a href="#">NXL_RECIPE_CI</a> on page 297
C202	Change Master Recipe	NXL_RECIPE	Common Interception	<a href="#">NXL_RECIPE_CI</a> on page 297

Table A-1: Supported transactions (Continued)

Transaction Code	Description	Security Identifier	Interception Type	Implementation ID
C203	Display Master Recipe	NXL_RECIPE	Common Interception	<a href="#">NXL_RECIPE_CI</a> on page 297
CA01	Create Routing	MATNR	Transaction Specific	<a href="#">CA01_MATNR</a> on page 295
CA01	Create Routing	NXL_RECIPE	Common Interception	<a href="#">NXL_RECIPE_CI</a> on page 297
CA02	Change Routing	MATNR	Transaction Specific	<a href="#">CA0X_MATNR</a> on page 295
CA02	Change Routing	NXL_RECIPE	Common Interception	<a href="#">NXL_RECIPE_CI</a> on page 297
CA03	Display Routing	MATNR	Transaction Specific	<a href="#">CA0X_MATNR</a> on page 295
CA03	Display Routing	NXL_RECIPE	Common Interception	<a href="#">NXL_RECIPE_CI</a> on page 297
CA11	Create Reference Operation Set	NXL_RECIPE	Common Interception	<a href="#">NXL_RECIPE_CI</a> on page 297
CA12	Change Reference Operation Set	NXL_RECIPE	Common Interception	<a href="#">NXL_RECIPE_CI</a> on page 297
CA13	Display Reference Operation Set	NXL_RECIPE	Common Interception	<a href="#">NXL_RECIPE_CI</a> on page 297
CC01	Create Change Master	MATNR	Transaction Specific	<a href="#">CC0X_ALLIDT</a> on page 295
CC01	Create Change Master	DOCNUM	Transaction Specific	<a href="#">CC0X_ALLIDT</a> on page 295
CC02	Change Change Master	MATNR	Transaction Specific	<a href="#">CC0X_ALLIDT</a> on page 295
CC02	Change Change Master	DOCNUM	Transaction Specific	<a href="#">CC0X_ALLIDT</a> on page 295
CC03	Display Change Master	MATNR	Transaction Specific	<a href="#">CC0X_ALLIDT</a> on page 295
CC03	Display Change Master	DOCNUM	Transaction Specific	<a href="#">CC0X_ALLIDT</a> on page 295
CEWB	PP: Engineering Workbench	MATNR	Transaction Specific	<a href="#">CEWB_MATNR</a> on page 295
CF11	PRT: Use of material in prod. order	MATNR	Common Interception	<a href="#">NXL_PRT_CI</a> on page 297
CF12	PRT: Use of document in prod. order	DOCNUM	Common Interception	<a href="#">NXL_PRT_CI</a> on page 297
CF13	PRT: Use of equipment in prod. order	NXL_EQUNR	Common Interception	<a href="#">NXL_PRT_CI</a> on page 297
CF16	PRT: Use of material in network	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
CF16	PRT: Use of material in network	MATNR	Common Interception	<a href="#">NXL_PRT_CI</a> on page 297
CF17	PRT: Use of document in network	DOCNUM	Common Interception	<a href="#">NXL_PRT_CI</a> on page 297
CF18	PRT: Use of piece of equipment in network	NXL_EQUNR	Common Interception	<a href="#">NXL_PRT_CI</a> on page 297
CF21	PRT: Use of material in orders	MATNR	Common Interception	<a href="#">NXL_PRT_CI</a> on page 297
CF22	PRT: Use of document in orders	DOCNUM	Common Interception	<a href="#">NXL_PRT_CI</a> on page 297
CF23	PRT: Use of piece of equipment in orders	NXL_EQUNR	Common Interception	<a href="#">NXL_PRT_CI</a> on page 297
CK11	Create Product Cost Estimate	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
CK11N	Create Material Cost Estimate	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
CK13	Display Product Cost Estimate	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
CK51	Create Order BOM Cost Estimate	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
CK53	Display Order BOM Cost Estimate	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
CK88	Partner Cost Component Split	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296

Table A-1: Supported transactions (Continued)

Transaction Code	Description	Security Identifier	Interception Type	Implementation ID
CKM3	Material Price Analysis	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
CKM3N	Material Price Analysis	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
CKM3OLD	Material Price Analysis	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
CKM3VERYOLD	Display Material Ledger Data	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
CKMCCC	Manual Change: Act. Cost Comp. Split	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
CKMCCD	ManChang: Display Actual CC Split	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
CKMCCS	Display Actual Cost Component Split	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
CKMD	Transactions for a Material	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
CKMDISPTAB	Technical View of ML Master Data	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
CKML_FPR1	Create Production Process	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
CKML_FPR3	Display Production Process	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
CKMLAVREXP	Analysis of Data Cumulation	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
CKMLAVRPERD	Display period values	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
CKMLQS	Valuated Quantity Structure(M-level)	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
CKW1	Create Production Lot Cost Est.	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
CKW3	Display Production Lot Cost Est.	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
CO02	Change Production Order	WERKS	Common Interception	<a href="#">PLANT_CO_02_03 on page 297</a>
CO03	Display Production Order	WERKS	Common Interception	<a href="#">PLANT_CO_02_03 on page 297</a>
CO06	Backorder Processing	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
CO09	Availability Overview	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
CO10	Production order with project	WERKS	Common Interception	<a href="#">PLANT_CO_02_03 on page 297</a>
COOIS	Production Order Information System	WERKS	Common Interception	<a href="#">PLANT_CO_02_03 on page 297</a>
CS01	Create Material BOM	MATNR	Transaction Specific	<a href="#">CS01_MATNR on page 295</a>
CS01	Create Material BOM	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
CS02	Change Material BOM	MATNR	Transaction Specific	<a href="#">CS0X_MATNR on page 295</a>
CS02	Change Material BOM	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
CS03	Display Material BOM	MATNR	Transaction Specific	<a href="#">CS0X_MATNR on page 295</a>
CS03	Display Material BOM	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
CS07	Allocate Material BOM to Plant	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
CS08	Change Material BOM - Plant Alloc.	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
CS09	Display Allocations to Plant	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
CS11	Display BOM Level by Level	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
CS12	Multilevel BOM	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
CS13	Summarized BOM	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
CS15	Single-Level Where-Used List	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>

Table A-1: Supported transactions (Continued)

Transaction Code	Description	Security Identifier	Interception Type	Implementation ID
CS40	Create Link to Configurable Material	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
CS41	Change Material Config. Allocation	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
CS42	Display Material Config. Assignment	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
CS61	Create Order BOM	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
CS62	Change Order BOM	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
CS63	Display Order BOM	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
CS71	Create WBS BOM	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
CS72	Change WBS BOM	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
CS73	Display WBS BOM	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
CS74	Create multi-level WBS BOM	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
CS75	Change multi-level WBS BOM	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
CS76	Display multi-level WBS BOM	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
CS80	Change Documents for Material BOM	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
CS83	Change documents for WBS BOM	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
CSP1	Multi-level WBS BOM explosion	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
CSP2	WBS BOM multi-level BOM	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
CSP3	WBS BOM - summarized BOM	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
CU44	Material Configuration Overview	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
CUCK	Copy Config. Material	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
CUMODEL	Transaction for Displaying Model	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
CV01N	Create Document	DOCNUM	Transaction Specific	<a href="#">CV0XN_DOCNUM on page 295</a>
CV02N	Change Document	DOCNUM	Transaction Specific	<a href="#">CV0XN_DOCNUM on page 295</a>
CV03N	Display document	DOCNUM	Transaction Specific	<a href="#">CV0XN_DOCNUM on page 295</a>
DRPM	Deployment for Material	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
IE01	Create Equipment (Plant Maintenance module)	NXL_EQUNR	Common Interception	<a href="#">NXL_PRT_CI on page 297</a>
IE02	Change Equipment	NXL_EQUNR	Common Interception	<a href="#">NXL_PRT_CI on page 297</a>
IE03	Display Equipment	NXL_EQUNR	Common Interception	<a href="#">NXL_PRT_CI on page 297</a>
IE08	Create Equipment (Customer Service module)	NXL_EQUNR	Common Interception	<a href="#">NXL_PRT_CI on page 297</a>
IE10	Multiple Equipment Entry	NXL_EQUNR	Common Interception	<a href="#">NXL_PRT_CI on page 297</a>
IE25	Create Production Resource/Tool	NXL_EQUNR	Common Interception	<a href="#">NXL_PRT_CI on page 297</a>
IE31	Create Fleet Object	NXL_EQUNR	Common Interception	<a href="#">NXL_PRT_CI on page 297</a>
IH05	Material Structure	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
INV_DISPLAY_MAT	Display Material Master/Price Analysis	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
IP02	Change Maintenance Plan	WERKS	Transaction Specific	<a href="#">PLANT_IP_IW_03 on page 297</a>

Table A-1: Supported transactions (Continued)

Transaction Code	Description	Security Identifier	Interception Type	Implementation ID
IP10	Schedule Maintenance Plan	WERKS	Transaction Specific	<a href="#">PLANT_IP_IW_03</a> on page 297
IP42	Add strategy-controlled plan	IWERK	Transaction Specific	<a href="#">PLANT_IP_IW_01</a> on page 297
IQ01	Create Material Serial Number	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
IQ04	Create Material Serial Number: List Entry	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
IW21	Create PM Notification - General	WERKS	Transaction Specific	<a href="#">PLANT_IP_IW_02</a> on page 297
IW51	Create Service Notification-General	WERKS	Transaction Specific	<a href="#">PLANT_IP_IW_02</a> on page 297
KKAS	Calculate Work in Process: Individual Processing	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
KKAT	Display Work in Process: Individual Processing	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
KKF1	Create CO Production Order	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
KKF6	Create Product Cost Collector	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
KKF7	Change Product Cost Collector	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
KKF8	Display Product Cost Collector	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
LB01	Create Transfer Requirement	BWLVS	Transaction Specific	<a href="#">LB01_MVTYPE_WM</a> on page 295
LB01	Create Transfer Requirement	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LB02	Change transfer requirement	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LB03	Display Transfer Requirement	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LB10	TRs for Storage Type	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LB11	TRs for Material	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LB12	TRs and Posting Change for Mat.Doc.	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LB13	TRs for Requirement	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LI01N	Create System Inventory Record	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LI02N	Change System Inventory Record	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LI03N	Display System Inventory Record	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LI04	Print System Inventory Record	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LI11N	Enter Inventory Count	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LI12N	Change inventory count	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LI13N	Display Inventory Count	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LI14	Start Inventory Recount	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LI20	Clear Inventory Differences WM	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LI21	Clear Inventory Differences in MM-IM	BUKRS	Transaction Specific	<a href="#">LI21_BUKRS</a> on page 295
LI21	Clear Inventory Differences in MM-IM	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LICC	Cycle Counting per Quant	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LICC	Cycle Counting per Quant	WERKS	Transaction Specific	<a href="#">LICC_WERKS01</a> on page 295

Table A-1: Supported transactions (Continued)

Transaction Code	Description	Security Identifier	Interception Type	Implementation ID
LP21	WM replenishment for fixed bins	BWLVS	Transaction Specific	<a href="#">LP21_MOVE_TYPE</a> on page 295
LP21	WM replenishment for fixed bins	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LS01N	Create Warehouse Master Record	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LS02N	Change Warehouse Master Record	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LS03N	Display Warehouse Master Record	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LS04	Display Empty Storage Bins	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LS06	Block Storage Bins	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LS07	Block Quants	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LS08	Block Storage Bins by Aisle	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LS11	Change several stor.bins simulate.	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LS22	Change Quants	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LS23	Display Quants	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LS24	Display Quants for Material	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LS25	Display Quants per Storage Bin	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LS26	Warehouse stocks per material	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LT01	Create Transfer Order	BWLVS	Transaction Specific	<a href="#">LT01_MVTYPE_WM</a> on page 295
LT01	Create Transfer Order	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LT01	Create Transfer Order	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
LT02	Clear Inventory using Transfer Order	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
LT03	Create TO for Delivery Note	BUKRS	Transaction Specific	<a href="#">LT03_COMP_CODE</a> on page 296
LT03	Create TO for Delivery Note	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LT04	Create TO from TR	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LT05	Process Posting Change Notice	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LT06	Create TO for Material Document	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LT0A	Pre-plan storage units	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
LT0G	Return delivery to stock	BWLVS	Transaction Specific	<a href="#">LT0G_WHNO_MVTYPE</a> on page 296
LT0G	Return delivery to stock	LGNUM	Transaction Specific	<a href="#">LT0G_WHNO_MVTYPE</a> on page 296
LT0R	Request replenishment manually	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
LT10	Create Transfer Order from List	BWLVS	Transaction Specific	<a href="#">LT10_MOVE_TYPE</a> on page 296
LT10	Create Transfer Order from List	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LT11	Confirm Transfer Order Item	BUKRS	Transaction Specific	<a href="#">LT11_ALLIDT</a> on page 296
LT11	Confirm Transfer Order Item	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LT11	Confirm Transfer Order Item	BWART	Transaction Specific	<a href="#">LT11_ALLIDT</a> on page 296
LT11	Confirm Transfer Order Item	WERKS	Transaction Specific	<a href="#">LT11_ALLIDT</a> on page 296

Table A-1: Supported transactions (Continued)

Transaction Code	Description	Security Identifier	Interception Type	Implementation ID
LT12	Confirm transfer order	BUKRS	Transaction Specific	<a href="#">LT11_ALLIDT</a> on page 296
LT12	Confirm transfer order	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LT15	Cancelling transfer order	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LT21	Display Transfer Order	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LT22	Display Transfer Order / Stor. Type	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LT23	Display Transfer Orders by Numbers	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LT24	Display Transfer Order / Material	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LT24	Display Transfer Order / Material	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
LT27	Transfer order for storage unit	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LT31	Print TO Manually	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LU02	Change Posting Change Notice	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LU03	Display Posting Change Notice	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LU04	Selection of Posting Change Notices	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LX01	List of Empty Storage Bins	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LX02	Stock list	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LX03	Bin Status Report	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LX04	Capacity load utilization	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LX09	Overview of All Transf.Requirements	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LX10	Activities per Storage Type	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LX12	Document Overview: Landscape Format	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LX13	Analysis of differences	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LX14	Matl mvmt frequency	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LX16	Selection of Bins for Continuous Inv	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LX17	List of Inventory Differences	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LX18	Statistics of Inventory Differences	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LX22	Process Inventory from Overview	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LX25	Inventory Status	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LX27	Stock levels by shelf life exp.date	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
LX29	Fixed bin supervision	LGNUM	Common Interception	<a href="#">WHNUM_ID</a> on page 298
MAHD2	Change Alternative Historical Data	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
MAHD3	Display Alternative Historical Data	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
MB01	Post Goods Receipt for PO	WERKS	Transaction Specific	<a href="#">MB01_PLANT</a> on page 296
MB01	Post Goods Receipt for PO	BWART	Transaction Specific	<a href="#">MB01_MVTTYPE</a> on page 296
MB02	Change Material Document	BWART	Transaction Specific	<a href="#">MB0203_PLANT_MVTTYPE</a> on page 296
MB02	Change Material Document	WERKS	Transaction Specific	<a href="#">MB0203_PLANT_MVTTYPE</a> on page 296



Table A-1: Supported transactions (Continued)

Transaction Code	Description	Security Identifier	Interception Type	Implementation ID
MB03	Display Material Document	BWART	Transaction Specific	<a href="#">MB0203_PLANT_MVTTYPE on page 296</a>
MB03	Display Material Document	WERKS	Transaction Specific	<a href="#">MB0203_PLANT_MVTTYPE on page 296</a>
MB1B	Transfer Posting	BWART	Transaction Specific	<a href="#">MB01_MVTTYPE on page 296</a>
MB1B	Transfer Posting	WERKS	Transaction Specific	<a href="#">MB01_PLANT on page 296</a>
MB51	Material Doc. List	BUKRS	Transaction Specific	<a href="#">MB51_53_ALLIDT on page 296</a>
MB51	Material Doc. List	LGORT	Transaction Specific	<a href="#">MB51_53_ALLIDT on page 296</a>
MB51	Material Doc. List	WERKS	Transaction Specific	<a href="#">MB51_53_ALLIDT on page 296</a>
MB51	Material Doc. List	BWART	Transaction Specific	<a href="#">MB51_53_ALLIDT on page 296</a>
MB52	List of Warehouse Stocks on Hand	WERKS	Transaction Specific	<a href="#">MB52_PLANT on page 296</a>
MB53	Display Plant Stock Availability	WERKS	Transaction Specific	<a href="#">MB51_53_ALLIDT on page 296</a>
MB53	Display Plant Stock Availability	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
MB56	Analyze batch where-used list	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
MB5B	Stocks for Posting Date	WERKS	Transaction Specific	<a href="#">MB5B_PLANT on page 296</a>
MB5M	Material Shelf life List	WERKS	Transaction Specific	<a href="#">MB5M_PLANT on page 296</a>
MB5T	Stock in transit CC	BUKRS	Transaction Specific	<a href="#">MB51_53_ALLIDT on page 296</a>
MB5T	Stock in transit CC	WERKS	Transaction Specific	<a href="#">MB5T_PLANT on page 296</a>
MB90	Output Processing for Mat. Documents	LGORT	Transaction Specific	<a href="#">MB90_PLANT_SLOC on page 296</a>
MB90	Output Processing for Mat. Documents	WERKS	Transaction Specific	<a href="#">MB90_PLANT_SLOC on page 296</a>
MBST	Cancel Material Document	LGORT	Transaction Specific	<a href="#">MB0203_PLANT_MVTTYPE on page 296</a>
MBST	Cancel Material Document	WERKS	Transaction Specific	<a href="#">MB0203_PLANT_MVTTYPE on page 296</a>
MBST	Cancel Material Document	BWART	Transaction Specific	<a href="#">MB0203_PLANT_MVTTYPE on page 296</a>
MC74	Transfer Mat. to Demand Management	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
MC90	Tsfr.to Dm.Mgmt.: Mat.from any IS	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
MC9K	Maintain Available Capacity	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
MD02	MRP - Single-item, Multi-level -	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
MD03	MRP-Individual Planning-Single Level	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
MD04	Display Stock/Requirements Situation	WERKS	Transaction Specific	<a href="#">MD04_PLANT on page 296</a>
MD17	Collective Requirements Display	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
MD20	Create Planning File Entry	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
MD41	MPS - Single-item, Multi-level -	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>



Table A-1: Supported transactions (Continued)

Transaction Code	Description	Security Identifier	Interception Type	Implementation ID
MD42	MPS - Single-item, Single-level -	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
MD43	MPS - Single-item, Interactive -	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
MD44	MPS Evaluation	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
MD45	MRP List Evaluation	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
MD48	Cross-Plant Evaluation	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
MDP6	Modeling	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
MDPH	Planning Profile	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
MDPV	Planning variant: Initial screen	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
ME01	Maintain Source List	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
ME03	Display Source List	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
ME11	Create Purchasing Info Record	NXL_LIFNR	Common Interception	<a href="#">NXL_LIFNR_CI on page 297</a>
ME12	Change Purchasing Info Record	NXL_LIFNR	Common Interception	<a href="#">NXL_LIFNR_CI on page 297</a>
ME13	Display Purchasing Info Record	NXL_LIFNR	Common Interception	<a href="#">NXL_LIFNR_CI on page 297</a>
ME15	Flag Purch. Info Rec. for Deletion	NXL_LIFNR	Common Interception	<a href="#">NXL_LIFNR_CI on page 297</a>
ME21N	Create Purchase Order	NXL_LIFNR	Common Interception Transaction Specific	<a href="#">NXL_LIFNR_CI on page 297</a> <a href="#">ME2XN_LIFNR on page 296</a>
ME22N	Change Purchase Order	NXL_LIFNR	Common Interception Transaction Specific	<a href="#">NXL_LIFNR_CI on page 297</a> <a href="#">ME2XN_LIFNR on page 296</a>
ME23N	Display Purchase Order	NXL_LIFNR	Common Interception Transaction Specific	<a href="#">NXL_LIFNR_CI on page 297</a> <a href="#">ME2XN_LIFNR on page 296</a>
ME24	Maintain Purchase Order Supplement	NXL_LIFNR	Common Interception	<a href="#">NXL_LIFNR_CI on page 297</a>
ME31K	Create Contract	NXL_LIFNR	Common Interception	<a href="#">NXL_LIFNR_CI on page 297</a>
ME41	Create Request For Quotation	NXL_LIFNR	Common Interception	<a href="#">NXL_LIFNR_CI on page 297</a>
ME42	Change Request For Quotation	NXL_LIFNR	Common Interception	<a href="#">NXL_LIFNR_CI on page 297</a>
ME43	Display Request For Quotation	NXL_LIFNR	Common Interception	<a href="#">NXL_LIFNR_CI on page 297</a>
ME44	Maintain RFQ Supplement	NXL_LIFNR	Common Interception	<a href="#">NXL_LIFNR_CI on page 297</a>
ME51	Create Purchase Requisition	NXL_LIFNR	Common Interception Transaction Specific	<a href="#">NXL_LIFNR_CI on page 297</a> <a href="#">ME51_LIFNR on page 296</a>
ME52	Change Purchase Requisition	NXL_LIFNR	Common Interception	<a href="#">NXL_LIFNR_CI on page 297</a>
ME53	Display Purchase Requisition	NXL_LIFNR	Common Interception	<a href="#">NXL_LIFNR_CI on page 297</a>

Table A-1: Supported transactions (Continued)

Transaction Code	Description	Security Identifier	Interception Type	Implementation ID
ME54	Release Purchase Requisition	NXL_LIFNR	Common Interception	<a href="#">NXL_LIFNR_CI on page 297</a>
ME61	Maintain Vendor Evaluation	NXL_LIFNR	Common Interception	<a href="#">NXL_LIFNR_CI on page 297</a>
ME62	Display Vendor Evaluation	NXL_LIFNR	Common Interception	<a href="#">NXL_LIFNR_CI on page 297</a>
ME64	Evaluation Comparison	NXL_LIFNR	Common Interception	<a href="#">NXL_LIFNR_CI on page 297</a>
MEQ1	Maintain Quota Arrangement	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
MEQ3	Display Quota Arrangement	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
MF20	REM Cost Controlling	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
MGW0	Create Components for Set Material	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
MGW1	Display Components for Set Material	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
MGW2	Create Components for Display Matl	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
MGW3	Display Components for Display Matl	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
MGW4	Create Components for Prepack Matl	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
MGW5	Display Components for Prepack Matl	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
MGW6	Create Components for Full Product	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
MGW7	Display Components for Full Product	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
MGW8	Change Components for Set Material	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
MGW9	Change Components for Display Matl	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
MGWA	Change Components for Prepack	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
MGWB	Change Components for Full Product	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
MIGO	Goods Movement	BWART	Transaction Specific	<a href="#">MB51_53_ALLIDT on page 296</a>
MIGO	Goods Movement	WERKS	Transaction Specific	<a href="#">MB51_53_ALLIDT on page 296</a>
MIGO	Goods Movement	LGORT	Transaction Specific	<a href="#">MB51_53_ALLIDT on page 296</a>
MIGO_GR	Goods Movement	BWART	Transaction Specific	<a href="#">MB51_53_ALLIDT on page 296</a>
MIGO_GR	Goods Movement	WERKS	Transaction Specific	<a href="#">MB51_53_ALLIDT on page 296</a>
MIR7	Park Invoice	NXL_LIFNR	Common Interception	<a href="#">NXL_LIFNR_CI on page 297</a>
MIRA	Fast Invoice Entry	NXL_LIFNR	Common Interception	<a href="#">NXL_LIFNR_CI on page 297</a>
MIRO	Enter Incoming Invoice	NXL_LIFNR	Common Interception Transaction Specific	<a href="#">NXL_LIFNR_CI on page 297</a> <a href="#">MIRO_LIFNR on page 297</a>
MK01	Create vendor (Purchasing)	NXL_LIFNR	Common Interception	<a href="#">NXL_LIFNR_CI on page 297</a>
MK02	Change vendor (Purchasing)	NXL_LIFNR	Common Interception	<a href="#">NXL_LIFNR_CI on page 297</a>

Table A-1: Supported transactions (Continued)

Transaction Code	Description	Security Identifier	Interception Type	Implementation ID
MK03	Display vendor (Purchasing)	NXL_LIFNR	Common Interception	<a href="#">NXL_LIFNR_CI</a> on page 297
MK05	Block Vendor (Purchasing)	NXL_LIFNR	Common Interception	<a href="#">NXL_LIFNR_CI</a> on page 297
MK06	Mark vendor for deletion (purch.)	NXL_LIFNR	Common Interception	<a href="#">NXL_LIFNR_CI</a> on page 297
MK12	Change vendor (Purchasing), planned	NXL_LIFNR	Common Interception	<a href="#">NXL_LIFNR_CI</a> on page 297
MK18	Activate planned vendor changes (Pu)	NXL_LIFNR	Common Interception	<a href="#">NXL_LIFNR_CI</a> on page 297
MK19	Display vendor (purchasing), future	NXL_LIFNR	Common Interception	<a href="#">NXL_LIFNR_CI</a> on page 297
MLCCSPD	Cost Components for Price	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
MM01	Create Material	MATNR	Transaction Specific	<a href="#">MM0X_MATNR</a> on page 296
MM02	Change Material	MATNR	Transaction Specific	<a href="#">MM0X_MATNR</a> on page 296
MM02	Change Material	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
MM03	Display Material	MATNR	Transaction Specific	<a href="#">MM0X_MATNR</a> on page 296
MM03	Display Material	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
MM04	Display Material Change Documents	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
MM06	Flag Material for Deletion	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
MM12	Schedule Changing of Material &	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
MM15	Display Changes (Migration)	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
MM16	Schedule Material for Deletion	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
MM19	Display Material & at Key Date	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
MMAM	Change Material Type	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
MMBE	Stock Overview	LGORT	Transaction Specific	<a href="#">MMBE_ALLIDT</a> on page 296
MMBE	Stock Overview	WERKS	Transaction Specific	<a href="#">MMBE_ALLIDT</a> on page 296
MMSC	Enter Storage Locations Collectively	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
MMVH	Create Centrally: Decentral.Shipping	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
MOKS	CAP: Calculation Simulation	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
MP30	Execute Material Forecast	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
MP31	Change Material Forecast	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
MP32	Display Material Forecast	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
MRCHVW	Batch mgmt with reconciliation	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
MS02	Long-term plng: single-itm, mult-lvl	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
MS03	Long-term plng: singl-itm, singl-lvl	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
MS20	Planning File Entry: Long-Term Plnng	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
MS44	Flexible Evaluation Long-Term Plnng	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
MSC2N	Change Batch	WERKS	Transaction Specific	<a href="#">MSC3N_ALLIDT</a> on page 297

Table A-1: Supported transactions (Continued)

Transaction Code	Description	Security Identifier	Interception Type	Implementation ID
MSC2N	Change Batch	BWART	Transaction Specific	<a href="#">MSC3N_ALLIDT</a> on page 297
MSC2N	Change Batch	LGORT	Transaction Specific	<a href="#">MSC3N_ALLIDT</a> on page 297
MSC3N	Display Batch	WERKS	Transaction Specific	<a href="#">MSC3N_ALLIDT</a> on page 297
MSC4N	Display Change Documents for Batch	WERKS	Transaction Specific	<a href="#">MSC3N_ALLIDT</a> on page 297
MSK4	Display Vdr Consignment Change Docs	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
PCA1	Creating a Production Campaign	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
PK01	Create Control Cycle	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
PK02	Change Control Cycle	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
PK03	Display Control Cycle	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
PMEVC	Variant Configuration Modeling Envmt	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
POPT	Test Packing Instruction Master Data	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
QA01	Create Inspection Lot	WERKS	Transaction Specific	<a href="#">QA01_WERKS</a> on page 297
QA02	Change Inspection Lot	WERKS	Common Interception	<a href="#">QA_WERKS_CI</a> on page 297
QA03	Display inspection lot	WERKS	Common Interception	<a href="#">QA_WERKS_CI</a> on page 297
QA11	Record usage decision	WERKS	Transaction Specific	<a href="#">QA11_12_13_WERKS</a> on page 297
QA12	Change usage decision with history	WERKS	Transaction Specific	<a href="#">QA11_12_13_WERKS</a> on page 297
QA13	Display usage decision	WERKS	Transaction Specific	<a href="#">QA11_12_13_WERKS</a> on page 297
QA32	Change data for inspection lot	WERKS	Common Interception	<a href="#">QA_WERKS_CI</a> on page 297
QA33	Display data for inspection lot	WERKS	Common Interception	<a href="#">QA_WERKS_CI</a> on page 297
QAC1	Change insp. lot actual quantity	WERKS	Common Interception	<a href="#">QA_WERKS_CI</a> on page 297
QDL1	Create quality level	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
QE03	Display characteristic results	WERKS	Common Interception	<a href="#">QA_WERKS_CI</a> on page 297
QP01	Create Inspection Plan	NXL_RECIP	Common Interception	<a href="#">NXL_RECIP_CI</a> on page 297
QP02	Change Inspection Plan	NXL_RECIP	Common Interception	<a href="#">NXL_RECIP_CI</a> on page 297
QP03	Display Inspection Plan	NXL_RECIP	Common Interception	<a href="#">NXL_RECIP_CI</a> on page 297
QP08	Print task lists for material	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
QS61	Maintain material specification	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
QS62	Display material specification	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296

Table A-1: Supported transactions (Continued)

Transaction Code	Description	Security Identifier	Interception Type	Implementation ID
QS63	Maintain material spec: Planning	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
QS64	Display material spec: For key date	MATNR	Common Interception	<a href="#">MATNR_CI on page 296</a>
V-03	Create ordering party (Sales)	NXL_KUNNR	Common Interception	<a href="#">NXL_KUNNR_CI on page 297</a>
V-04	Create invoice recipient (Sales)	NXL_KUNNR	Common Interception	<a href="#">NXL_KUNNR_CI on page 297</a>
V-05	Create payer (Sales)	NXL_KUNNR	Common Interception	<a href="#">NXL_KUNNR_CI on page 297</a>
V-06	Create consignee (Sales)	NXL_KUNNR	Common Interception	<a href="#">NXL_KUNNR_CI on page 297</a>
V-07	Create one-time customer (Sales)	NXL_KUNNR	Common Interception	<a href="#">NXL_KUNNR_CI on page 297</a>
V-08	Create payer (Centrally)	NXL_KUNNR	Common Interception	<a href="#">NXL_KUNNR_CI on page 297</a>
V-09	Create ordering party (Centrally)	NXL_KUNNR	Common Interception	<a href="#">NXL_KUNNR_CI on page 297</a>
VA01	Create Sales Order	NXL_KUNNR	Common Interception	<a href="#">NXL_KUNNR_CI on page 297</a>
VA02	Change Sales Order	NXL_KUNNR	Common Interception	<a href="#">NXL_KUNNR_CI on page 297</a>
VA03	Display Sales Order	NXL_KUNNR	Common Interception	<a href="#">NXL_KUNNR_CI on page 297</a>
VA11	Create Inquiry	NXL_KUNNR	Common Interception	<a href="#">NXL_KUNNR_CI on page 297</a>
VA12	Change Inquiry	NXL_KUNNR	Common Interception	<a href="#">NXL_KUNNR_CI on page 297</a>
VA13	Display Inquiry	NXL_KUNNR	Common Interception	<a href="#">NXL_KUNNR_CI on page 297</a>
VA21	Create Quotation	NXL_KUNNR	Common Interception	<a href="#">NXL_KUNNR_CI on page 297</a>
VA22	Change Quotation	NXL_KUNNR	Common Interception	<a href="#">NXL_KUNNR_CI on page 297</a>
VA23	Display Quotation	NXL_KUNNR	Common Interception	<a href="#">NXL_KUNNR_CI on page 297</a>
VA41	Create Contract	NXL_KUNNR	Common Interception	<a href="#">NXL_KUNNR_CI on page 297</a>
VA42	Change Contract	NXL_KUNNR	Common Interception	<a href="#">NXL_KUNNR_CI on page 297</a>
VA43	Display Contract	NXL_KUNNR	Common Interception	<a href="#">NXL_KUNNR_CI on page 297</a>
Value Help	Input, or F4, help for any supported transaction	All configured identifiers	Common Interception	<a href="#">NXL_F4_ALLIDT on page 297</a>
VD01	Create Customer (Sales)	NXL_KUNNR	Common Interception	<a href="#">NXL_KUNNR_CI on page 297</a>
VD02	Change Customer (Sales)	NXL_KUNNR	Common Interception	<a href="#">NXL_KUNNR_CI on page 297</a>
VD03	Display Customer (Sales)	NXL_KUNNR	Common Interception	<a href="#">NXL_KUNNR_CI on page 297</a>

Table A-1: Supported transactions (Continued)

Transaction Code	Description	Security Identifier	Interception Type	Implementation ID
VD05	Block customer (sales)	NXL_KUNNR	Common Interception	<a href="#">NXL_KUNNR_CI</a> on page 297
VD06	Mark customer for deletion (sales)	NXL_KUNNR	Common Interception	<a href="#">NXL_KUNNR_CI</a> on page 297
VD54	Display Customer-Material Info	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
VEPR	Customs log	MATNR	Common Interception	<a href="#">MATNR_CI</a> on page 296
VL02N	Change Outbound Delivery	BWART	Transaction Specific	<a href="#">VL02_3N_ALLIDT</a> on page 297
VL02N	Change Outbound Delivery	WERKS	Transaction Specific	<a href="#">VL02_3N_ALLIDT</a> on page 297
VL02N	Change Outbound Delivery	VSTEL	Transaction Specific	<a href="#">VL02_3N_ALLIDT</a> on page 297
VL03N	Display Outbound Delivery	VSTEL	Transaction Specific	<a href="#">VL02_3N_ALLIDT</a> on page 297
VL06F	General delivery list - Outb.deliv.	VSTEL	Common Interception	<a href="#">VL06_SHIPPING_POINT</a> on page 297
VL06G	List of Outbound Dlvs for Goods Issue	VSTEL	Common Interception	<a href="#">VL06_SHIPPING_POINT</a> on page 297
VL06L	Outbound Deliveries to be Loaded	VSTEL	Common Interception	<a href="#">VL06_SHIPPING_POINT</a> on page 297
VL06O	Outbound Delivery Monitor	VSTEL	Common Interception	<a href="#">VL06_SHIPPING_POINT</a> on page 297
VL06P	List of Outbound Dlvs for Picking	VSTEL	Common Interception	<a href="#">VL06_SHIPPING_POINT</a> on page 297
VL09	Cancel Goods Issue for Delivery Note	VSTEL	Transaction Specific	<a href="#">VL09_SHIPPING_POINT</a> on page 298
VL71	Output from Outbound Deliveries	VSTEL	Transaction Specific	<a href="#">VL71_SHIPPING_POINT</a> on page 298
VT01N	Create Shipment	TPLST	Transaction Specific	<a href="#">VT01/2N_TRAN_PL_PT</a> on page 298
VT01N	Create Shipment	VSTEL	Transaction Specific	<a href="#">VT01N_SHIPPING_POINT</a> on page 298
VT02N	Change Shipment	TPLST	Transaction Specific	<a href="#">VT02N_TRAN_PL_PT</a> on page 298
VT02N	Change Shipment	VSTEL	Transaction Specific	<a href="#">VT02N_SHIPPING_POINT</a> on page 298
VT03N	Display Shipment	TPLST	Transaction Specific	<a href="#">VT01/2N_TRAN_PL_PT</a> on page 298
VT70	Output for Shipments	TPLST	Transaction Specific	<a href="#">VT70_TRAN_PL_PT</a> on page 298
XD01	Create Customer (Centrally)	NXL_KUNNR	Common Interception	<a href="#">NXL_KUNNR_CI</a> on page 297
XD02	Change Customer (Centrally)	NXL_KUNNR	Common Interception	<a href="#">NXL_KUNNR_CI</a> on page 297
XD03	Display Customer (Centrally)	NXL_KUNNR	Common Interception	<a href="#">NXL_KUNNR_CI</a> on page 297
XD05	Block customer (centrally)	NXL_KUNNR	Common Interception	<a href="#">NXL_KUNNR_CI</a> on page 297

Table A-1: Supported transactions (Continued)

Transaction Code	Description	Security Identifier	Interception Type	Implementation ID
XD06	Mark customer for deletion (centr.)	NXL_KUNNR	Common Interception	<a href="#">NXL_KUNNR_CI on page 297</a>
XD07	Change Customer Account Group	NXL_KUNNR	Common Interception	<a href="#">NXL_KUNNR_CI on page 297</a>
XK01	Create Vendor (Centrally)	NXL_LIFNR	Common Interception Transaction Specific	<a href="#">NXL_LIFNR_CI on page 297</a> <a href="#">XK01_LIFNR on page 298</a>
XK02	Change vendor (centrally)	NXL_LIFNR	Common Interception	<a href="#">NXL_LIFNR_CI on page 297</a>
XK03	Display vendor (centrally)	NXL_LIFNR	Common Interception	<a href="#">NXL_LIFNR_CI on page 297</a>
XK05	Block Vendor (Centrally)	NXL_LIFNR	Common Interception	<a href="#">NXL_LIFNR_CI on page 297</a>
XK06	Mark vendor for deletion (centrally)	NXL_LIFNR	Common Interception	<a href="#">NXL_LIFNR_CI on page 297</a>

## Implementations

This section provides the information for defining the enhancement implementations for the transactions listed in the previous section, [Transactions](#) on page 277. Information is organized by Implementation ID in alphabetical order.

**Note:** You need only configure enhancement implementations for the transactions you want to intercept and the business objects for which you require access control.

There are four types of enhancement implementations:

- BADI
- Explicit Enhancement
- Implicit Enhancement
- User Exit

The step-by-step procedure is provided once for each type of implementation. Then, in [Implementation Details](#) on page 294, only the pertinent details are provided for each Implementation ID, for example, the transaction to run, the program or function module to modify, the enhancement point, and the code to insert.

### BADI Enhancements

BADI enhancement implementations are based on object-oriented concepts of interfaces, classes and methods. You add custom code to methods in an interface. The following is the procedure for creating a BADI enhancement implementation.

### Procedure

- 1 In the SAP interface, enter transaction `SE18`. The *BADI Builder* appears.
- 2 Enter the BADI name, and click **Display**.
- 3 Select **Implementation > Create**.
- 4 In **Implementation Name**, enter a name for the implementation, and click **Continue**.
- 5 In **Implementation Short Text**, provide a description of the implementation.
- 6 Click the **Interface** tab.
- 7 Double-click the method in which to add the custom code.
- 8 Save the implementation.
- 9 Insert the code in the method.
- 10 Save and activate the method.
- 11 Click the **Back** button and activate the implementation.

### Explicit Enhancements

Explicit enhancements are predefined enhancements sections provided explicitly by SAP. The following is the procedure for creating an explicit enhancement.

### Procedure

- 1 In the SAP interface, enter transaction `SE38`. The ABAP Editor appears.
- 2 In the next screen, in **Program**, enter the name of the program. Click **Display**.
- 3 Click the **Enhance** button on the tool bar.
- 4 Locate the enhancement point in which to insert your code. Right-click the line that contains the text "ENHANCEMENT-POINT" or "ENHANCEMENT-SECTION," and select **Enhancement Operations > Create Implementation**.
- 5 Select an existing enhancement implementation or create a new one. To create a new enhancement implementation:
  - a Click the **Create Enhancement Implementation** button.
  - b Enter a name for the enhancement implementation and a description.
  - c Save the enhancement implementation, then select it, if necessary.
- 6 Insert code in the Enhancement block.
- 7 Activate the implementation.



## Implicit Enhancements

Implicit enhancement implementations are added at the beginning and end of a program, method, function module, or subroutine. The following is the procedure for creating an implicit enhancement implementation.

### Procedure

- 1 In the SAP interface, enter one of the following transactions:
  - SE37: Displays the Function Builder
  - SE38: Displays the ABAP Editor
  - SE24: Displays the Class Builder
- 2 In the next screen, in **Function Module** or **Program** or **Object**, enter the name of the function module or program or class. Click **Display**.

If you specified a class, double-click the method to which to add an enhancement implementation.

- 3 Click the **Enhance** button on the tool bar.
- 4 In the main menu, select **Edit > Enhancement Operations > Show Implicit Enhancement Options**.

Enhancement points are indicated by an arrow on the left of the line number and by "\*\*\*\*\*" in the line, as shown in the following example.



- 5 Locate the specific enhancement point in which to insert your code, for example, at the beginning of a particular form, or at the end of a particular function module. Right-click the line and select **Enhancement Operations > Create Implementation**.
- 6 Select **Code** as the enhancement type.
- 7 Select an existing enhancement implementation or create a new one. To create a new enhancement implementation:
  - a Click the **Create Enhancement Implementation** button.
  - b Enter a name for the enhancement implementation and a description.
  - c Save the enhancement implementation, then select it, if necessary.
- 8 Insert code in the Enhancement block.
- 9 Activate the implementation.

## User Exit

A user exit is a location where you can access SAP program components and include your program enhancements. The following is the procedure for implementing a user exit enhancement.

### Procedure

- 1 In the SAP interface, enter transaction `CMOD`. The *Project Management of SAP Enhancements* screen appears.
- 2 In **Project**, enter a name for a new project, and click **Create**. Alternatively, enter the name of an existing project, and click **Display**.
- 3 In **Short text**, enter a project description. Click **Save**.
- 4 Click **Enhancement assignments** to assign an enhancement point to the project.
- 5 In the **Enhancement** column, enter the name of the enhancement to implement, then press Enter.
- 6 Click **Save**.
- 7 Click **Components**. A list of enhancements in the project appears. Each enhancement includes a function exit.
- 8 Double-click the function exit. The Function Builder appears.
- 9 Find the line `INCLUDE <include program>`. Double-click the name of the include program. A message appears if this include has not been previously implemented.
- 10 If the include has not been implemented, press Enter. A message prompts you to create an object. Click **Yes**.
- 11 In the include program, insert your code.
- 12 Save and activate the include program.
- 13 Navigate back to the screen that displays the enhancements in the project.
- 14 Activate the project.

### Implementation Details

[Table A-2](#) provides the pertinent details for each implementation ID. Step-by-step procedures are provided in the previous sections.

**Note:** To copy content in the table to use in the SAP application, it is more convenient to use the HTML version of this guide.

*Table A-2: Details for Implementation IDs*

Implementation ID	Type	T-Code	Program/Function Module/BADI Name	Form/Enhancement Point/Method	Location	Code
CA01_MATNR	Explicit Enh.	SE38	LCPCOU02	CP_CO_ALT_COPY_EXT_01		IF sy-tcode = 'CA01'. INCLUDE /NEXTLABS/ TRANS_PROCESS_CA01. ENDIF.
CA0X_MATNR	Explicit Enh.	SE38	LCP04U03	CP_DYNP_TITLE_FILL_01		DATA: v_rattr TYPE REF TO / nextlabs/check_instance. v_rattr = /nextlabs/ check_instance=>get_instance _rou( ). IF v_rattr IS NOT INITIAL. include /NEXTLABS/ TRANS_PROCESS_CA0X. ENDIF.
CC0X_ALLIDT	User exit	CMOD	PCCD0005	EXIT_SAPMC29C_006	Inside INCLUDE ZXC0AU06	INCLUDE /NEXTLABS/ TRANS_PROCESS_CC0X.
CEWB_MATNR (2 implementations)	BADI	SE18	EWB_SELECTION	BOM_PRE_SELECT		INCLUDE /nextlabs/ TRANS_PRCSS_CEWB_BOM.
				TSK_PRE_SELECT		INCLUDE /nextlabs/ TRANS_PRCSS_CEWB_TSK.
CS01_MATNR (2 implementations)	Explicit Enh.	SE38	LCSDIFCG	OKCODE_PRUEFEN_01		IF sy-tcode = 'CS01' AND okcode = 'CLWI'. INCLUDE /nextlabs/ trans_process_cs01_c. ENDIF.
	BADI	SE18	BOM_EXIT	START_SCREEN_CHECK		IF sy-tcode = 'CS01'. INCLUDE /NEXTLABS/ TRANS_PROCESS_CS01. ENDIF.
CS0X_MATNR	BADI	SE18	BOM_EXIT	BOM_DATA_CHECK		INCLUDE /NEXTLABS/ TRANS_PROCESS_CS0X.
CV0XN_DOCNUM	BADI	SE18	Document_Number01	DOCNUMBER_CHECK		INCLUDE /nextlabs/ doc_enhn_check_tag. INCLUDE /nextlabs/ trans_process_cv0xn.
LB01_MVTYPE_WM	Implicit Enh.	SE38	FL000F00	BERECHTIGUNG_BWLVS	End of form	INCLUDE /NXLECC/ TRANS_PROCESS_LB01_MVT.
LI21_BUKRS	Implicit Enh.	SE38	RLLI2110	LISTING	Start of form	INCLUDE /NXLECC/ TRANS_PROCESS_LI21_BUK.
LICC_WERKS01	Implicit Enh.	SE38	RLINV060	SELEKTION_IM	End of form	INCLUDE /NXLECC/ TRANS_PROCESS_LICC_WRK.
LP21_MOVE_TYPE	Implicit Enh.	SE38	RLLNACH1	T331_CHECK	End of form	INCLUDE /NXLECC/ TRANS_PROCESS_LP21_MMT.
LT01_MVTYPE_WM	Implicit Enh.	SE38	FL000F00	BERECHTIGUNG_BWLVS	End of form	INCLUDE /NXLECC/ TRANS_PROCESS_LT01_MVT.

Table A-2: Details for Implementation IDs (Continued)

Implementation ID	Type	T-Code	Program/Function Module/BADI Name	Form/Enhancement Point/Method	Location	Code
LT03_COMP_CODE	Implicit Enh.	SE38	ML03TI00_LF_LESEN	LF_LESEN	Start of form	INCLUDE /NXLECC/ TRANS_PROCESS_LT03_CCD.
LT0G_WHNO_MVTYP E	Implicit Enh.	SE38	RLLT0G00	TALF_TAB_FUELLE N	Start of form	INCLUDE /NXLECC/ TRANS_PROCESS_LT0G_MWH.
LT10_MOVE_TYPE	Implicit Enh.	SE38	RLS10034	USER_COMMAND	Start of form	INCLUDE /NXLECC/ TRANS_PROCESS_LT10_MMT.
LT11_ALLIDT	Implicit Enh.	SE38	ML03TF20_TA_LESEN	TA_LESEN	End of form	INCLUDE /NXLECC/ TRANS_PROCESS_LT11_ALL.
MATNR_CI	BADI	SE18	BADI_MATN1	MATN1_INPUT_002 MATN1_OUTPUT_002  In addition, two custom methods are required.		For detailed steps, see <a href="#">Implementation for Materials (Common Interception)</a> on page 298
MB01_MVTTYPE	Explicit Enh.	SE38	FM07MEB0 Main program: SAPFM071	BEWEGUNGSART_P RUEFEN_01		INCLUDE /NXLECC/ TRANS_PROCESS_MB01_MVT.
MB01_PLANT	Implicit Enh.	SE38	FM07MEW0	WERK_PRUEFEN	End of form	INCLUDE /NXLECC/ TRANS_PROCESS_MB01_PLT.
MB0203_PLANT_MVT TYPE	Implicit Enh.	SE37	MB_READ_MATERIAL_POSITION		End of function module	INCLUDE /NXLECC/ TRANS_PROCESS_MB02_3.
MB51_53_ALLIDT	Implicit Enh.	SE24	CL_MMIM_AUTH (class)	CHECK (method)	End of method	INCLUDE /NXLECC/ TRANS_PROCESS_MB51_53.
MB52_PLANT	Implicit Enh.	SE38	RM07MLBS	CHECK_AUTHORIZ ATION	End of form	INCLUDE /NXLECC/ TRANS_PROCESS_MB52.
MB5B_PLANT	Implicit Enh.	SE38	RM07MLBD_FOR M_02	F0000_CREATE_TA BLE_G_T_ORGAN	End of form	INCLUDE /NXLECC/ TRANS_PROCESS_MB5B.
MB5M_PLANT	Implicit Enh.	SE38	RM07MMHD	BERECHTIGUNG_P RUEFEN	End of form	INCLUDE /NXLECC/ TRANS_PROCESS_MB5M.
MB5T_PLANT	Implicit Enh.	SE38	RM07MTRB	DATENSELEKTION	End of form	INCLUDE /NXLECC/ TRANS_PROCESS_MB5T.
MB90_PLANT_SLOC	Implicit Enh.	SE38	MM70AMEA	MESSAGES_FILTER	End of form	INCLUDE /NXLECC/ TRANS_PROCESS_MB90.
MD04_PLANT	Implicit Enh.	SE38	MM61RF50_FCTA B_FILL_TC	FCTAB_FILL_TC	End of form	INCLUDE /NXLECC/ TRANS_PROCESS_MD04.
MM0X_MATNR	BADI	SE18	BADI_MATERIAL_ OD	SET_PROGRAM_FO R_OKCODE_ROUTN		INCLUDE /NEXTLABS/ TRANS_BUFFER_MM0X. INCLUDE /NEXTLABS/ TRANS_PROCESS_MM0X.
MMBE_ALLIDT	Implicit Enh.	SE38	RMMMBENFM	MARD_BESTAENDE	End of form	INCLUDE /NXLECC/ TRANS_PROCESS_MMBE.
ME2XN_LIFNR	Implicit Enh.	SE38	LAQRUCU10		End of function module	INCLUDE /NXLECC/ TRANS_PROCESS_ME2XN.
ME51_LIFNR	Implicit Enh.	SE38	MM06BFRF_RFT_ LESEN_KEY	RFT_LESEN_KEY	End of form	INCLUDE /NXLECC/ TRANS_PROCESS_ME51.

Table A-2: Details for Implementation IDs (Continued)

Implementation ID	Type	T-Code	Program/Function Module/BADI Name	Form/Enhancement Point/Method	Location	Code
MIRO_LIFNR	Implicit Enh.	SE38	LFACSU06		Start of function module	INCLUDE /NXLECC/ TRANS_PROCESS_MIRO.
MSC3N_ALLIDT	Implicit Enh.	SE38	LCHRGF04	D2000_OK_CODE	Start of form	INCLUDE /NXLECC/ TRANS_PROCESS_MSCXN.
NXL_F4_ALLIDT (2 implementations)	Implicit Enh.	SE38	WDTMFORS	DISPLAY_RESULT	Start of form	INCLUDE /NXLECC/ CI_VALUE_HELP1.
	Implicit Enh.	SE38	LSDSDF05	F4PROZ_STEP_DISPLAY	Start of form	INCLUDE /NXLECC/ CI_VALUE_HELP2.
NXL_KUNNR_CI (3 implementations)	Implicit Enh.	SE38	LV08DU01		Start of function module	INCLUDE /NXLECC/ CI_CUSTOMER2.
	Implicit Enh.	SE38	LVS01U19		Start of function module	INCLUDE /NXLECC/ CI_CUSTOMER1.
	Implicit Enh.	SE38	MF02DFR0 Main program: SAPMF02D	REF_KNA1_LESEN	Start of form	INCLUDE /NXLECC/ TRANS_PROCESS_XD01.
NXL_LIFNR_CI (2 implementations)	Implicit Enh.	SE38	LWY01U01		Start of function module	INCLUDE /NXLECC/ CI_VENDOR1.
	Implicit Enh.	SE38	MF02KFL0 Main program: SAPMF02K	LFA1_LESEN	Start of form	INCLUDE /NXLECC/ CI_VENDOR2.
NXL_PRT_CI	Implicit Enh.	SE38	LCFDBU26		Start of function module	INCLUDE /NXLECC/CI_PRT.
NXL_RECIPCI (2 implementations)	Implicit Enh.	SE38	LCPDIFP1	plan_prof_fill	Start of form	INCLUDE /NXLECC/ CI_GRP_TL_HDR_CHK.
	User exit	CMOD	CPAU0001	EXIT_SAPLCPAU_001	Inside Include ZXCPAU01	INCLUDE /NXLECC/ CI_GRP_RCP_RT_GRP.
PLANT_CO_02_03	Implicit Enh.	SE37	CO_ZF_AUTHORITY_CHECK		End of function module	INCLUDE /NXLECC/ CI_PLANT_CO_01.
PLANT_IP_IW_01	Implicit Enh.	SE37	IWERK_INST_AUTHORITY_CHECK		End of function module	INCLUDE /NXLECC/ CI_PLANT_IP_IW_01.
PLANT_IP_IW_02	Implicit Enh.	SE37	SWERK_INST_AUTHORITY_CHECK		End of function module	INCLUDE /NXLECC/ CI_PLANT_IP_IW_02.
PLANT_IP_IW_03	Implicit Enh.	SE38	LIWP3FZZ	MPOS_SELECT_F20	End of form	INCLUDE /NXLECC/ TRANS_PROCESS_IP_PLANT.
QA_WERKS_CI	Implicit Enh.	SE37	QAUT_INSPTYPE		End of function module	INCLUDE /NXLECC/ CI_PLANT_QA_01.
QA01_WERKS	Implicit Enh.	SE38	LQPL1F2X	WERK	End of form	INCLUDE /NXLECC/ TRANS_PROCESS_QA01_PLT.
QA11_12_13_WERKS	Implicit Enh.	SE38	MQEVAF24	LESE_TQ30	End of form	INCLUDE /NXLECC/ TRANS_PROCESS_QA01_PLT.
VL02_3N_ALLIDT	BADI	SE18	LE_SHP_DELIVERY_PROC	READ_DELIVERY		INCLUDE /NXLECC/ TRANS_PROCESS_VL02_3NA.
VL06_SHIPPING_POINT	Implicit Enh.	SE37	WS_LM_DATA_SELECTION		End of function module	INCLUDE /NXLECC/ CI_SHIPPINGPOINT.

Table A-2: Details for Implementation IDs (Continued)

Implementation ID	Type	T-Code	Program/Function Module/BADI Name	Form/Enhancement Point/Method	Location	Code
VL09_SHIPPING_POI NT	Implicit Enh.	SE38	RVV50L09	LIST_BODY_AUFBA UEN	Start of form	INCLUDE /NXLECC/ TRANS_PROCESS_VL09.
VL71_SHIPPING_POI NT	Implicit Enh.	SE38	SD70AV2A_MESS AGE_SELECTION	MESSAGES_FILTER	End of form	INCLUDE /NXLECC/ TRANS_PROCESS_VL71.
VT01/ 2N_TRAN_PL_PT	Implicit Enh.	SE38	LV56TF0B	BERECHTIGUNG_P RUEFEN	End of form	INCLUDE /NXLECC/ TRANS_PROCESS_VT01_2NT.
VT01N_SHIPPING_PO INT	Implicit Enh.	SE38	LV56LF01	LIEFERUNGEN_SEL EKTIEREN	End of form	INCLUDE /NXLECC/ TRANS_PROCESS_VT01N_VS.
VT02N_SHIPPING_PO INT	Implicit Enh.	SE38	MV56AF0S	SHIPMENT_READ_F OR_TRANSACTION	End of form	INCLUDE /NXLECC/ TRANS_PROCESS_VT02N_VS.
VT02N_TRAN_PL_PT	Implicit Enh.	SE38	MV56AF0B	BERECHTIGUNG_P RUEFEN	End of form	INCLUDE /NXLECC/ TRANS_PROCESS_VT01_2NT.
VT70_TRAN_PL_PT	Implicit Enh.	SE38	LV56NF02	AUTHORITY_CHEC K_SHIP	End of form	INCLUDE /NXLECC/ TRANS_PROCESS_VT70.
WHNUM_ID (3 implementations)	Implicit Enh.	SE38	FL000F00	BERECHTIGUNG_B WLVS	End of form	INCLUDE /NXLECC/ CI_WHNUM1.
				BERECHTIGUNG_L GNUM	End of form	INCLUDE /NXLECC/ CI_WHNUM1.
				BERECHTIGUNG_L GTYP	End of form	INCLUDE /NXLECC/ CI_WHNUM1.
XK01_LIFNR	Implicit Enh.	SE38	MF02KFR0 Main program: SAPMF02K	REF_LFA1_LESEN	Start of form	INCLUDE /NXLECC/ TRANS_PROCESS_XK01.

### Implementation for Materials (Common Interception)

The BADI implementation to trigger policy checks for Materials through a common interception point requires creating two custom methods in addition to inserting code in predefined methods.

#### Procedure

- 1 Create a BADI implementation, using the steps described in [BADI Enhancements](#) on page 291.
- 2 The following are the details:
  - BADI name: BADI\_MATN1
  - First method: MATN1\_INPUT\_002; code to insert:  
`INCLUDE /NXLECC/CI_MAT_MATN1_INPUT_002.`
  - Second method: MATN1\_OUTPUT\_002; code to insert:  
`INCLUDE /NXLECC/CI_MAT_MATN1_OUT_002.`
- 3 Create a new method, using the following steps:
  - Double-click the name of the implementing class.
  - Make sure you are in Change mode.

- Add a method named `NEXTLABS_POLICY_CHECK`.
  - In **Level**, select Instance Method.
  - In **Visibility**, select Public.
  - In **Description**, enter a description, such as NextLabs Policy Check.
- 4 Select the method, and click the **Parameter** button. Specify the following information:
- In **Parameter**, enter `MATNR`.
  - In **Type**, select Importing.
  - Select the **Pass by value** check box.
  - Select the **Optional** check box.
  - In **Typing Method**, select Type.
  - In **Associated type**, enter `C`.
- 5 Click **Save**, then click **Back**.
- 6 Click **Exception**.
- 7 In Exception, enter `nextlabs_check`. Enter a description.
- 8 Click **Save**, then click **Back**.
- 9 Double-click the `NEXTLABS_POLICY_CHECK` method, which you just created, and enter the following code:
- ```
INCLUDE /NXLECC/CI_MAT_NXL_PLC_CHK1.
```
- 10 Save and activate the method.
- 11 Click **Back** twice.
- 12 Create another new method:
- Double-click the name of the implementing class.
  - Make sure you are in Change mode.
  - Add a method named `NEXTLABS_POLICY_CHECK_OUTPUT`.
  - In **Level**, select Instance Method.
  - In **Visibility**, select Public.
  - In **Description**, enter a description, such as NextLabs Policy Check.
- 13 Select the `NEXTLABS_POLICY_CHECK_OUTPUT` method, and click the **Parameter** button. You will create one importing parameter and two exporting parameters.
- For the importing parameter, specify the following information:
    - In **Parameter**, enter `MATNR`.
    - In **Type**, select Importing.
    - Select the **Pass by value** check box.
    - Select the **Optional** check box.
    - In **Typing Method**, select Type.
    - In **Associated type**, enter `C`.

- For the exporting parameters, specify the following information:
    - In **Parameter**, enter `MESSAGE_IV1` for the first parameter. Enter `MESSAGE_EV1` for the second parameter.
    - In **Type**, select Exporting.
    - Select the **Pass by value** check box.
    - In **Typing Method**, select Type.
    - In **Associated type**, enter `STRING`.
- 14 Click **Save**, then click **Back**.
- 15 Click **Exception**.
- 16 In Exception, enter `nextlabs_check`. Enter a description.
- 17 Click **Save**, then click **Back**.
- 18 Double-click the `NEXTLABS_POLICY_CHECK_OUTPUT` method, which you just created, and enter the following code:
- ```
INCLUDE /NXLECC/CI_MAT_NXL_PLC_CHK2.
```
- 19 Save and activate the method.
- 20 Save and activate the BADI implementation.



# B

## Implementation Reference for EasyDMS

Table B-1 provides the pertinent information for defining the required enhancement implementations for the NextLabs Entitlement Pack for SAP EasyDMS. For step-by-step instructions for creating a BADI enhancement implementation and an implicit enhancement implementation, see [BADI Enhancements](#) on page 291 and [Implicit Enhancements](#) on page 293, respectively.

Table B-1: Implementation Reference for EasyDMS

Feature	Type	T-Code	Function Module/BADI Name	Method	Location	Code
View Filter Browse	Implicit Enh.	SE37	EASYDMS_DOCUMENT_GET LATEST2		End of function module	INCLUDE /NEXTLABS/ TRANS_PROCESS_EDMS6.
View Filter Search	Implicit Enh.	SE37	BAPI_DOCUMENT_GETLIST2		End of function module	INCLUDE /NEXTLABS/ TRANS_PROCESS_EDMS8.
Document Cut and Paste	Implicit Enh.	SE37	BAPI_DOCUMENT_CHANGE2		Start of function module	INCLUDE /NEXTLABS/ TRANS_PROCESS_EDMS5.
Folder Copy, Export, and Print	BADI	SE18	EASYDMS MAIN01	FOLDERCOPY		INCLUDE /NEXTLABS/ TRANS_PROCESS_EDMS4.
				FOLDEREXPORT		INCLUDE /NEXTLABS/ TRANS_PROCESS_EDMS3.
				PRINT		INCLUDE /NEXTLABS/ TRANS_PROCESS_EDMS2.
Documents	BADI	SE18	DOCUMENT_MAIN01	BEFORE_READ_DATA		INCLUDE /NEXTLABS/ TRANS_PROCESS_EDMS1.
				AFTER_READ_DATA		INCLUDE /NEXTLABS/ TRANS_PROCESS_EDMS1.



To use the NextLabs Entitlement Pack for SAP PLM, you must configure the BADI enhancement implementation.

### Procedure

- 1 In the SAP interface, enter transaction `SE19`. The *BADI Builder* appears.
- 2 Create a new BADI implementation with this value:  
`/PLMB/ES_SPI`. Click **Create Impl.**
- 3 In Create Enhancement Implementation, enter a name and description for the enhancement implementation. Click OK.
- 4 Create a BADI implementation by specifying the following values:
  - In BAdI Implementation, enter an implementation name.
  - In Implementation Class, enter a class name.
  - In BAdI Definition, select `/PLMB/EX_SPI_APPL_ACCESS`.
- 5 Click on the class name, click Edit, and enter the following attributes:

Table C-1: Attributes for the Entitlement Pack for PLM

Attribute	Level	Visibility	Read-only	Typing	Associated Type
MESSAGE	Instance Attribute	Public		Type	BAPIRET2_T
MESSAGE_C	Instance Attribute	Public		Type	BAPIRET2_T

- 6 Activate the class.
- 7 In the `/PLMB/IF_EX_SPI_APPL_ACCESS-BEFORE_RETRIEVE` method, insert the following code: `INCLUDE /NEXTLABS/TRANS_PRCSS_PLM_RETV`
- 8 In the `/PLMB/IF_EX_SPI_APPL_ACCESS-BEFORE_ACTION` method, insert the following code: `INCLUDE /NEXTLABS/TRANS_PRCSS_PLM_ACTN`
- 9 In the `/PLMB/IF_EX_SPI_APPL_ACCESS-BEFORE_DELETE` method, insert the following code: `INCLUDE /NEXTLABS/TRANS_PRCSS_PLM_DELT`
- 10 Activate the implementation.



This section describes the enhancement implementations for the Entitlement Pack for SAP BW.

Topics:

- [Types of Access Control](#)
- [Implementations](#)

---

## Types of Access Control

The enhancements that you define for the Entitlement Pack for SAP BW depend on the type or types of access control you want to implement. The Entitlement Pack for BW provides the following ways to implement access control on classified BW objects:

- Filter BW objects to display to a Business Explorer (BEx) Analyzer user only the objects or data that the user is authorized to access. For example, if a user is not authorized to access a particular InfoArea, DataSource Object (DSO), or InfoCube, BEx Analyzer does not display those objects.
- Display all InfoProvider objects to a BEx Analyzer user, but control what objects the user can open or use in a report. For example, when a user selects a classified DSO or InfoCube, Dynamic Authorization Management for SAP allows authorized users to access the object and blocks unauthorized users from using the object.
- Display a report only if a BEx user is authorized to view the classified data returned by a query.

**Note:** In this release, access control is supported for these BW objects: InfoArea, InfoProvider (InfoCubes and DSOs only), and InfoObject. In addition, BEx Analyzer is the only BEx tool supported.

You can implement the different types of access control. Generally, however, it is best practice to define the enhancement implementations that support *either* type (filter or display) of access control. Implementing both can result in access control policies that are incompatible or that produce unexpected behavior. For example, logically, policies governing InfoProvider objects either display all objects and restrict access when an object is selected, or those policies filter what objects the user sees in the first place.

For more information about how Dynamic Authorization Management runs policy checks for the different BW objects, see the section [Entitlement Pack for BW](#) on page 185 in [What Can Dynamic](#)

[Authorization Management Do?](#) on page 180 For examples of designing policies, see [Designing Access Control Policies for SAP BW](#) on page 221.

[Table D-1](#) describes the enhancement options for the Entitlement Pack for BW. Configure only the enhancements required for your access control strategy. Implementation details are provided in the next section.

*Table D-1: Enhancement Options for the Entitlement Pack for BW*

Enhancement for ...	Description	Intercepted Function
InfoObject Data Filter	Unauthorized data is excluded from the report. Configure either this enhancement or the next enhancement, InfoObject Data Display.	ANALYZER_FILTER
InfoObject Data Display	Display InfoObject data in a report if the query returns only data that the user is authorized to see. If the query results include any unauthorized data, the report displays nothing. Configure either this enhancement or the previous enhancement, InfoObject Data Filter.	ANALYZER_DISPLAY
InfoProvider Object Display	Display all InfoProvider objects (InfoCubes and DSOs) and apply access control when the user selects an object. Typically, you configure this enhancement or the next enhancement, InfoProvider and InfoArea Object Filter.	ANALYZER_DISPLAY
InfoProvider and InfoArea Object Filter	Filter the InfoArea and InfoProvider objects (InfoCubes and DSOs) to display on the objects that a user is authorized to access. Typically, you configure either this enhancement or the previous enhancement, InfoProvider Object Display.	ANALYZER_FILTER

## Implementations

[Table D-2](#) provides the pertinent information for configuring the enhancement implementations for BEx Analyzer. For step-by-step instructions for creating an implicit enhancement implementation, see [Implicit Enhancements](#) on page 293.

**Note:** You do not configure every enhancement implementation. Decide on an access control strategy and configure the corresponding enhancement implementations.

*Table D-2: Configuration options for BEx Analyzer*

Enhancement	Type	T-Code	Object type	Method	Location	Code
<a href="#">InfoObject Data Filter</a> on page 306 (2 implementations)	Implicit Enh.	SE24	CL_RSR_REQUEST	READ_DATA_INTERNAL	End of method	INCLUDE /NXLBW/ TRANS_PROCESS_EXCEL_2.
	Implicit Enh.	SE24	CL_RSMD_RS	IF_RSMD_RS-READ_DATA	End of method	For SAP BW 731 and earlier: INCLUDE /NXLBW/ TRANS_PROCESS_EXCEL_3. For SAP BW 740 and later: INCLUDE /NXLBW/ TRANS_PROCESS_EXCEL_3N.
<a href="#">InfoObject Data Display</a> on page 306	Implicit Enh.	SE24	CL_RSR_REQUEST	READ_DATA_INTERNAL	End of method	INCLUDE /NXLBW/ TRANS_PROCESS_EXCEL_5.

*Table D-2: Configuration options for BEx Analyzer (Continued)*

Enhancement	Type	T-Code	Object type	Method	Location	Code
<a href="#">InfoProvider Object Display</a> on page 306	Implicit Enh.	SE24	CL_RSR_XLS_VIEW	ASSIGN_QUERY_TO_DP	End of method	INCLUDE /NXLBW/ TRANS_PROCESS_EXCEL.
<a href="#">InfoProvider and InfoArea Object Filter</a> on page 306	Implicit Enh.	SE24	CL_CORE_RSOBJ _BASE	GET_OBJECT_MAIN	End of method	INCLUDE /NXLBW/ TRANS_PROCESS_EXCEL_4.





## E

# Implementation Reference for PBSC

[Table E-1](#) lists the required enhancement implementations for systems with Policy Based Security Classifications (PBSC). For step-by-step instructions for creating enhancement implementations, see [Implementations](#) on page 291.

*Table E-1: Required Enhancement Implementations for Systems with PBSC*

T-Code to Intercept	Implementation Type	T-Code	Enhancement Spot/ BADI Name	Enhancement Option/Method	Code
MM01	BADI	SE18	Enhancement Spot: ES_SAPLMGMU	\PR:SAPLMGMU\EX:LMGMUF01_13	INCLUDE /NEXTLABS/ PBSC_MATERIAL. INCLUDE /NEXTLABS/ PBSC_ECC.
CV01N	BADI	SE18	BADI Name: DOCUMENT_MAIN01	AFTER_SAVE	INCLUDE /NEXTLABS/ PBSC_DOCUMENT. INCLUDE /NEXTLABS/ PBSC_ECC. INCLUDE /NEXTLABS/ PBSC_PLM. INCLUDE /NEXTLABS/ PBSC_EASYDMS.



## F

# Implementation Reference for DFPS

This section provides the information for defining the Defense Forces and Public Security (DFPS) related implementations and designing DFPS policies.

Topics:

- [Implementations for DFPS](#)
- [Example DFPS Policies](#)

## Implementations for DFPS

There are four types of enhancement implementations:

- BADI Enhancements
- Explicit Enhancements
- Implicit Enhancements
- User Exit

The step-by-step procedure is provided once for each type of implementation. For more information, see [Implementations](#). Only the pertinent details are provided for each Implementation ID, for example, the transaction to run, the program or function module to modify, the enhancement point, and the code to insert.

For information about required transactions and implementations for DFPS, see [Table F.1](#) and [Table F.2](#).

*Table F.1: Supported Transactions*

Transaction Code	Description	Implementation ID
/ISDFPS/CHANGE_SLOC	Split/Merge Storage Locations	<a href="#">ISDFPS_01_CI</a>
/ISDFPS/DISP_INITSUP	Display Initial and Subs. Supply	<a href="#">ISDFPS_01_CI</a> <a href="#">ISDFPS_DISP_INITSUP_01</a>
/ISDFPS/DISP_MAT_SIT	Display Material Situation	<a href="#">ISDFPS_01_CI</a>
/ISDFPS/EPA_HU	Processing of HUs from EPA	<a href="#">ISDFPS_EPA_HU_01</a>
/ISDFPS/LMSTB1	Status Board (Change Mode)	<a href="#">ISDFPS_LMSTBX</a>
/ISDFPS/LMSTB2	Status Board (Display Mode)	<a href="#">ISDFPS_LMSTBX</a>
/ISDFPS/LN03	Display Warehouse Structure	<a href="#">LN03_01</a>
/ISDFPS/LSP1	Personnel Categories Planning	<a href="#">ISDFPS_01_CI</a>
/ISDFPS/LSP2	Logistical Mission Support	<a href="#">ISDFPS_01_CI</a>
/ISDFPS/MAT_ASSIGN	Material Assignment	<a href="#">ISDFPS_01_CI</a>

Table F.1: Supported Transactions

Transaction Code	Description	Implementation ID
/ISDFPS/MAT_COMP	Authorized/Actual Comparison of MPO	ISDFPS_MAT_COMP_01
/ISDFPS/MB52	Material Stock List for MPO/MC	ISDFPS_MB52_01
/ISDFPS/MM_RL01	Create Return Delivery PReqs	ISDFPS_01_CI
/ISDFPS/MPO_COMP	Authorized/Actual Comparison of MPO	ISDFPS_MPO_COMP_01
/ISDFPS/PERS1	Personnel Categories Planning	ISDFPS_01_CI
/ISDFPS/PERS2	Personnel Categories Planning	ISDFPS_01_CI
/ISDFPS/RELOC1	Relocation Execution	ISDFPS_01_CI
/ISDFPS/STOCK_RETURN	Disband Force Element in Operation	ISDFPS_01_CI
/ISDFPS/TOEACC1	Accounting Organizational Basis	ISDFPS_01_CI
/ISDFPS/TOEACC2	Accounting Organizational Basis	ISDFPS_01_CI
/ISDFPS/TOELOG1	Logistics Organizational Basis	ISDFPS_01_CI
/ISDFPS/TOELOG2	Logistics Organizational Basis	ISDFPS_01_CI
/ISDFPS/TOEM1	Material Organizational Basis	ISDFPS_01_CI
/ISDFPS/TOEM2	Material Organizational Basis	ISDFPS_01_CI
/ISDFPS/TOEP1	Personnel Organizational Basis	ISDFPS_01_CI
/ISDFPS/TOEP2	Personnel Organizational Basis	ISDFPS_01_CI
/ISDFPS/WF_ACC	Structures Workbench	ISDFPS_01_CI
ADSUBCON	SUBCONTRACTING Monitor	ADSUBCON_01
BMBC	Batch Information Cockpit	BMBC_01
CO09	Availability Overview	CO09_01
CO27	Picking list	CO27_01
IE37	Change Vehicles	IE37_01
LS26	Warehouse stocks per material	LS26_01
LX02	Stock list	LX02_01
LX03	Bin Status Report	LX03_01
LX23	Stock comparison IM - WM	LX23_01
LX27	Stock levels by shelf life exp.date	LX27_01
MB51	Material Document List	MB51_01
MB52	List of Warehouse Stocks on Hand	MB52_01
MB54	Consignment Stocks	MB54_01
MB58	Consgmt and Ret. Packag. At Customer	MMBE_MBXX_01
MB5L	List of Stock Values: Balances	MB5L_01
MB5M	BBD/Prod. Date	MB5M_01
MBBS	Display valuated special stock	MBBS_01
MLB	Stocks at Subcontractor	MMBE_MBXX_01
MC44	INVCO: Analysis of Inventory Turnover	MCXX_01
MC45	INVCO: Analysis of Usage Values	MCXX_01
MC46	INVCO: Analysis of Slow-Moving Items	MCXX_01

Table F.1: Supported Transactions

Transaction Code	Description	Implementation ID
MC50	INVCO: Analysis of Dead Stock	<a href="#">MCXX_01</a>
MD02	MRP - Single-item, Multi-level	<a href="#">MDXX_01</a> <a href="#">MD02_03</a>
MD03	MRP-Individual Planning-Single Level	<a href="#">MDXX_01</a> <a href="#">MD02_03</a>
MD04	Display Stock/Requirements Situation	<a href="#">MDXX_01</a> <a href="#">MDXX_02</a> <a href="#">MD04_07</a>
MD05	Individual Display Of MRP List	<a href="#">MDXX_02</a>
MD06	Collective Display Of MRP List	<a href="#">MDXX_02</a>
MD07	Current Material Overview	<a href="#">MDXX_01</a> <a href="#">MDXX_02</a> <a href="#">MD04_07</a>
MM02	Change Material	<a href="#">MM0X_01</a>
MM03	Display Material	<a href="#">MM0X_01</a>
MMBE	Stock Overview	<a href="#">MMBE_MBXX_01</a>

Table F.2: Details for Implementations IDs

Implementation ID	Type	T-code	Program/Class/Include/Function Module/BADI Name	Form/Enhancement Point/ Method	Location	Code
ADSUBCON_01	Implicit Enh.	SE38	DI_SUBCON_1_I010	GET_STOCK_DATA	End of form	INCLUDE /NXLDFPS/ TRANS_PROCESS_ADSC.
BMBC_01	Implicit Enh.	SE38	RVBBINCO_F0SELECTION	SELECT_STOCK_RESULTS	Start of form	INCLUDE /NXLDFPS/ TRANS_PROCESS_BMBC.
CO09_01	Implicit Enh.	SE38	LATP4FRD	READ_MAT	Start of form	INCLUDE /NXLDFPS/ TRANS_PROCESS_CO09.
CO27_01	Implicit Enh.	SE38	PPIOMF01	FILL_INT_S000	End of form	INCLUDE /NXLDFPS/ TRANS_PROCESS_CO27.
IE37_01	Implicit Enh.	SE38	RIFLET20	SELECTION_L	End of form	INCLUDE /NXLDFPS/ TRANS_PROCESS_IE37.
ISDFPS_01_CI	Implicit Enh.	SE24	/ISDFPS/CL_FDP_STOCK_LIST	GET_LIST	End of method	INCLUDE /NXLDFPS/ TRANS_PROCESS_ISDFPS.
ISDFPS_DISP_INITSUP_01	Implicit Enh.	SE24	/ISDFPS/CL_INITIAL_SUPPLY	DISPLAY_OVERVIEW	Start of method	INCLUDE NXLDFPS/ TRANS_PROCESS_INITSUP.
ISDFPS_EPA_HU_01	Implicit Enh.	SE24	/ISDFPS/CL_EPA_PACK_HUM_DATA	CONSTRUCTOR	End of method	INCLUDE /NXLDFPS/ TRANS_PROCESS_EPA_HU.

Table F.2: Details for Implementations IDs

Implementation ID	Type	T-code	Program/Class/Include/Function Module/BADI Name	Form/Enhancement Point/ Method	Location	Code
ISDFPS_LMSTBX	Implicit Enh.	SE38	/ISDFPS/STB_SELECTION_LM_F01	SELECT_DATA	End of form	INCLUDE /NXLDFPS/ TRANS_PROCESS_LMSTB.
ISDFPS_MAT_COMP_01	Implicit Enh.	SE38	/ISDFPS/AUTH_INV_COMP_EXEC		End of include	INCLUDE /NXLDFPS/ TRANS_PROCESS_MAT_CMP.
ISDFPS_MB52_01	Implicit Enh.	SE38	RM07MLBS	LIST_OUTPUT	Start of form	INCLUDE /NXLDFPS/ TRANS_PROCESS_MB52_01.
ISDFPS_MPO_COMP_01	BADI	SE18	/ISDFPS/BADI_AUTH_INV	MODIFY_AUTH_INV_COMP_BADI		INCLUDE /NXLDFPS/ TRANS_PROCESS_MAT_COM.
LN03_01 (3 implementations)	Implicit Enh.	SE38	/ISDFPS/LWM_WHNRFO3	ADD_ATTRIBUTES	End of form	INCLUDE /NXLDFPS/ TRANS_PROCESS_LN03.
	Implicit Enh.	SE38	/ISDFPS/LWM_WHNRFO3	ADD_NODES_500	End of form	INCLUDE /NXLDFPS/ TRANS_PROCESS_LN03_02.
	Implicit Enh.	SE37	L_BIN_DISPLAY		Start of function module	INCLUDE /NXLDFPS/ TRANS_PROCESS_LN03_03.
LS26_01	Implicit Enh.	SE38	RLLS2600	LISTE_ERSTELLEN	End of form	INCLUDE /NXLDFPS/ TRANS_PROCESS_LS26.
LX02_01	Implicit Enh.	SE38	RLS10020	CALL_ALV	Start of form	INCLUDE /NXLDFPS/ TRANS_PROCESS_LX02.
LX03_01	Implicit Enh.	SE38	RLS10030	LISTVIEWER_AUFRUFEN	Start of form	INCLUDE /NXLDFPS/ TRANS_PROCESS_LX03.
LX23_01	Implicit Enh.	SE38	RLABGL00	FUELLEN_IT320	End of form	INCLUDE /NXLDFPS/ TRANS_PROCESS_LX23.
LX27_01	Implicit Enh.	SE38	RLS30010	AUSGABE_LISTE	Start of form	INCLUDE /NXLDFPS/ TRANS_PROCESS_LX27.
MB51_01	Implicit Enh.	SE38	RM07DOCS	PROCESS_LIST	End of form	INCLUDE /NXLDFPS/ TRANS_PROCESS_MB51.
MB52_01	Implicit Enh.	SE38	RM07MLBS	LIST_OUTPUT	Start of form	INCLUDE /NXLDFPS/ TRANS_PROCESS_MB52_01.
MB54_01	Implicit Enh.	SE38	RM07MKBS	LISTAUSGABE	Start of form	INCLUDE /NXLDFPS/ TRANS_PROCESS_MB54.
MB5L_01	Implicit Enh.	SE38	RM07MBST	SUMMEN_BILDEN	End of form	INCLUDE /NXLDFPS/ TRANS_PROCESS_MB5L.
MB5M_01	Implicit Enh.	SE38	RM07MMHD	DATEN_FILTERN	Start of form	INCLUDE /NXLDFPS/ TRANS_PROCESS_MB5M_02.

Table F.2: Details for Implementations IDs

Implementation ID	Type	T-code	Program/Class/Include/Function Module/BADI Name	Form/Enhancement Point/ Method	Location	Code
MBBS_01	Implicit Enh.	SE38	RM07MBWS	DATENSELEKTION	End of form	INCLUDE /NXLDFPS/ TRANS_PROCESS_MBBS. (for Version - SAP_APPL - 605) OR INCLUDE /NXLDFPS/ TRANS_PROCESS_MBBS_01. (for Version - SAP_APPL - 617)
MCXX_01 (4 implementations)	Implicit Enh.	SE38	LMCBBF10	BESTAENDE_LESEN	End of form	INCLUDE /NXLDFPS/ TRANS_PROCESS_MCXX_01.
	Implicit Enh.	SE38	LMCGRF01	LAGERBESTAND_ERMITTELN	End of form	INCLUDE/NXLDFPS/ TRANS_PROCESS_MCXX_02.
	Implicit Enh.	SE38	RMMMBMUR	MENGE_OHNE_CHARGE	Start of form	/NXLDFPS/ TRANS_PROCESS_MCXX_03.
	Implicit Enh.	SE37	MCB_DETERMINE_STOCKTYPE		Start of function module	/NXLDFPS/ TRANS_PROCESS_MCXX_04.
MDXX_01	BADI	SE18	MD_CHANGE_MRP_DATA	CHANGE_MDPSX_MARD		INCLUDE /NXLDFPS/ TRANS_PROCESS_MDXX_01.
MD02_03	Implicit Enh.	SE38	MM61XDBR_MRP_AREA_PLANT_CHECK (Main Program - SAPMM61X)	MRP_AREA_PLANT_CHECK	End of form	INCLUDE /NXLDFPS/ TRANS_PROCESS_MD02_03.
MDXX_02	Implicit Enh.	SE38	LM61RF47	SORT_MDKEX_EINSTIEG	End of form	INCLUDE /NXLDFPS/ TRANS_PROCESS_MDXX_02.
MD04_07	Implicit Enh.	SE38	MM61XDBR_MRP_AREA_PLANT_CHECK (Main Program - SAPMM61R)	MRP_AREA_PLANT_CHECK	End of form	INCLUDE /NXLDFPS/ TRANS_PROCESS_MD04_07.
MM0X_01 (2 implementations)	Implicit Enh.	SE37	MARD_GENERIC_READ_MATNR_PLANT		End of function module	INCLUDE /NXLDFPS/ TRANS_PROCESS_MMXX_01.
	Implicit Enh.	SE38	LMGD1F2B (Main Program: SAPLMGD1)	INIT_BAUSTEIN	Start of form	INCLUDE /NXLDFPS/ TRANS_PROCESS_MMXX_02.
MMBE_MBXX_01	Implicit Enh.	SE38	RMMMBNMUR	MENGE_OHNE_CHARGE	Start of form	INCLUDE /NXLDFPS/ TRANS_PROCESS_MMBE_02.
MB51_01	Implicit Enh.	SE38	RM07DOCS	PROCESS_LIST	End of form	INCLUDE /NXLDFPS/ TRANS_PROCESS_MB51.

## Example DFPS Policies

This section lists few examples of DFPS policies.

### Example 1

This example policy ensures that the user is authorized to access stock levels in the specified MRP area.

The screenshot shows the configuration for a Document Policy titled "Document Policy" with the subtitle "MRP Area Access". The policy is configured with the following settings:

- Enforcement:** Deny
- Subject:**
  - User: Restricted Users (User Component)
  - Computer: (empty)
  - Application: (empty)
- Perform the Following:** Action: (empty)
- On Resources:** Target: Moved, Renamed or Copied
- Conditions:**
  - Connection Type: (empty)
  - Heartbeat: (empty)
  - Date/Time: Start: (empty), End: (empty)
  - Recurrence: Time: (empty), Day: (empty)
  - Condition Expression: `(resource.object.mrparea != null AND user.mrparea != resource.object.mrparea)`



## Example 2

This example policy ensures that only authorized users can see stock levels that are not assigned to an MRP area.

The image shows two panels from the SAP DFPS configuration interface:

- Document Policy:**
  - Title: Non-MRP Area Access
  - Enforcement: Deny
  - Subject:
    - User: not in Authorized Users (User Component)
    - Computer: +
    - Application: +
  - Perform the Following:
    - Action: Run (Action Component)
  - On Resources:
    - Target: in Allow Non MRP Area Access (Resource Component)
    - Moved, Renamed or Copied: +
- Object Component:**
  - Title: Allow\_Non\_MRP\_Area\_Access
  - Members:
    - SAP in SAP Component
  - With Properties:
    - Property Name: allow\_non\_mrp\_area\_access is yes
  - With Content:
    - Property Name: +

A red arrow points from the 'Allow Non MRP Area Access' resource component in the Document Policy panel to the 'Allow\_Non\_MRP\_Area\_Access' object component in the Object Component panel.



# G

## Read Tags: External Classifications

---

This section describes how to use the Read Tags feature to read the tags from documents that are classified by an external data classification system.

The Read Tags feature reads the external classifications (tags) applied to a document when you upload the document into SAP and then inserts those tags into the corresponding columns of the Security Classification Maintenance table. After you check-in and save the document, and based on policy, the Read Tags feature automatically inserts the tags into the corresponding columns of the Security Classification Maintenance table.

To use the Read Tags feature on the documents that are classified by an external data classification system, you must perform these configuration procedures for the Entitlement Manager for SAP.

### Procedure

- 1 Configure the RFC connection for the Read Tags feature.  
For more information, see [Configuring the RFC Connection for Read Tags](#).
- 2 Configure SAP data handling and connection settings for the Read Tags feature.  
For more information see [Configuring SAP Data Handling and Connection Settings for Read Tags](#).
- 3 Add the external classification (tag) values to the Security Classification Maintenance table. For more information, see [Adding New Classification Values](#).  
For example, specify the external classification component name characteristics:
  - Component name (for example, `EXPCONT1`)
  - Component data type (for example, `CHAR`)
  - Length of the field (for example, `50`)
  - Short description (for example, `Export Control 1`)

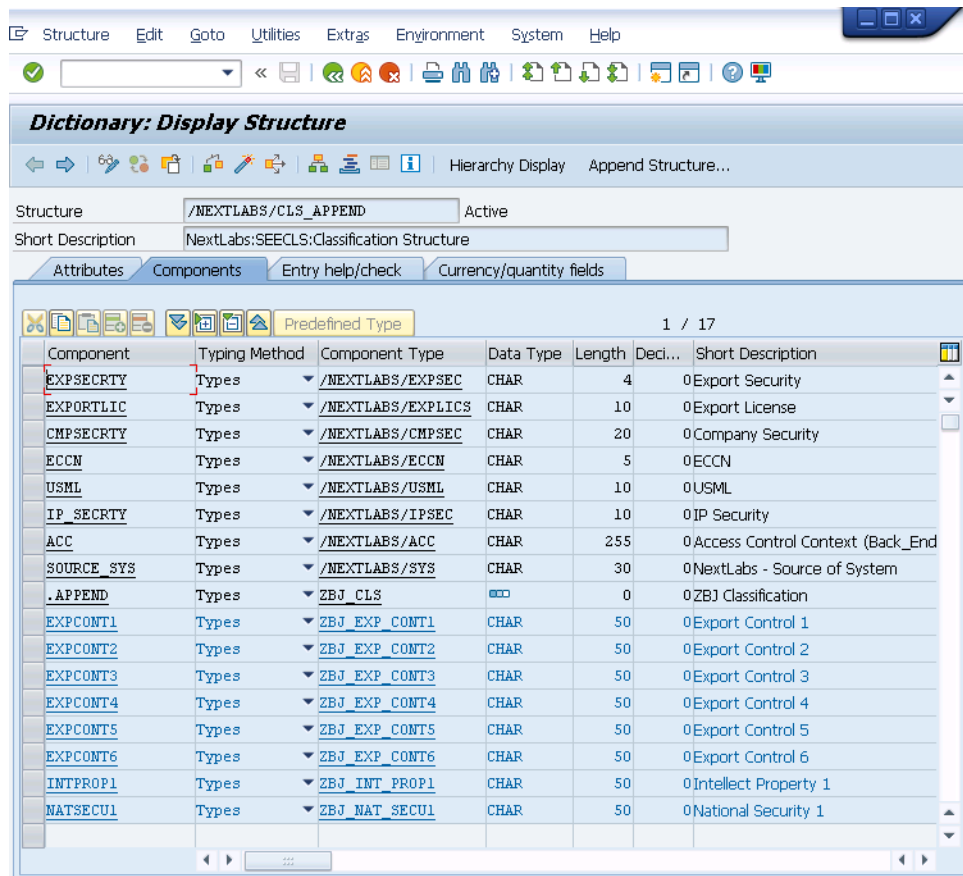


Figure G-1: Dictionary: Display Structure

- 4 Map the classification values configured in the Security Classification Maintenance table to the Policy Controller. Make sure that the External property name value matches the external classification value applied to the document. For more information, see [Mapping Security Fields \(SECM PG\)](#).

Field Name	Property name	Cardinality	Source	External property name
ACC	ACC	Single	Classificatio...	
CHGDATE	CHANGE DATE	Single	Classificatio...	
CHGTIME	CHANGE TIME	Single	Classificatio...	
COMPSECRTY	COMPANY SECURITY	Multiple	Classificatio...	
CREDATE	CREATE DATE	Single	Classificatio...	
CRETIME	CREATE TIME	Single	Classificatio...	
CREUSER	CREATED BY	Single	Classificatio...	
ECCN	ECCN	Multiple	Classificatio...	
EXPCONT1	EXPORTCONTROL1	Multiple	Classificatio...	URN:BAILS:EXPORTCONTROL1:BUSINESSAUTHORIZATIONCATEGORY:IDENTIFIER
EXPCONT2	EXPORTCONTROL2	Multiple	Classificatio...	URN:BAILS:EXPORTCONTROL2:BUSINESSAUTHORIZATIONCATEGORY:IDENTIFIER
EXPCONT3	EXPORTCONTROL3	Multiple	Classificatio...	URN:BAILS:EXPORTCONTROL3:BUSINESSAUTHORIZATIONCATEGORY:IDENTIFIER
EXPCONT4	EXPORTCONTROL4	Multiple	Classificatio...	URN:BAILS:EXPORTCONTROL4:BUSINESSAUTHORIZATIONCATEGORY:IDENTIFIER
EXPCONT5	EXPORTCONTROL5	Multiple	Classificatio...	URN:BAILS:EXPORTCONTROL5:BUSINESSAUTHORIZATIONCATEGORY:IDENTIFIER
EXPCONT6	EXPORTCONTROL6	Multiple	Classificatio...	URN:BAILS:EXPORTCONTROL6:BUSINESSAUTHORIZATIONCATEGORY:IDENTIFIER
EXPSECRTY	EXPORT SECURITY	Multiple	Classificatio...	
INTPROPI	INTELLECTUALPRO...	Multiple	Classificatio...	URN:BAILS:INTELLECTUALPROPERTY:BUSINESSAUTHORIZATIONCATEGORY:IDENTIFIER
NATSECUI	NATIONALSECURIT...	Multiple	Classificatio...	URN:BAILS:NATIONALSECURITY:BUSINESSAUTHORIZATIONCATEGORY:IDENTIFIER
TCODE	TRANSACTION	Single	Transaction D...	

Figure G-2: Security Field Mapping Overview

- Build and activate a custom BADI for Dynamic User or Resource Attribute. The Table G-1 lists the required enhancement implementation for systems using the Read Tags External Classification feature.

Table G-1: Required Enhancement Implementation for Read Tags External Classification

Type	T-Code	BADI Name/Program	Method	Code
BADI	SE19	/NEXTLABS/ENH_DYNATT	GET_DYN_RESOURCE_ATT	INCLUDE /NXLECC/READ_TAGS_EXT_PROP.

For more information, see [BADI Implementation for Dynamic User or Resource Attribute](#).

- Implement and activate an access control policy. For more information, see [Example Policy: Access Control Based on Resource Attributes](#).

Example Policy: Custom Access Control

Use the Access Control policy to authenticate users who are accessing documents that are classified using a classification system that is outside of SAP.

To configure a Custom Access Control policy, shown in [Figure G-3](#), perform these general steps:

- a Create a Document policy.
- b Set the enforcement type to **Deny**.
- c Expand the Actions component panel and drag a **Run** action component into the policy.
- d In the On Resources section, specify the resource components:
  - **SAP SYSTEM**: specifies any resource in your SAP server, across all systems, clients, applications, functions and business objects.
  - **EXPORT Control 3**: specifies the custom classification tag applied to the document.
- e In the Obligations area, specify these values:
  - Name: **SAP User Message**
  - Language Code: *<language\_of\_your\_choice>*
  - Message Text: *<user\_defined\_message>*  
For example, "User is not authorized".
- f Drag the SAP resource components that you created in step 4, into the policy.
- g **Submit** and **Deploy** the policy.

## Document Policy

SAP Read Tags-Access Control

Application +

---

Perform the Following

Action -

**Run**  
Action Component

On Resources

Target -

in

SAP SYSTEM ED6

Resource Component

+

and

in

ExportControl3

Resource Component

Moved, Renamed or Copied:

+

Subpolicy

Subpolicy -

Subpolicy

Obligations

On Deny +

- Log
- Display User Alert
- Send Email
- Custom Obligation

Name - SAP User Message

Language Code - EN:English

MessageText Deny! This information is classified.

On Allow, Monitor +

- Log
- Display User Alert
- Send Email
- Custom Obligation

## Object Component

ExportControl3

[Members](#)

---

[With Properties](#)

Property Name

urn:bails:exportcontrol3:businessauthorizationcategory:identifier is TAA 2280-13

[With Content](#)

Property Name

Figure G-3: Example Policy: Access Control

1/10/17

NextLabs Dynamic Authorization Management for SAP • User's Guide

323

