



# USING DYNAMIC PROPERTY MANAGER ADMINISTRATION CONSOLE

---

(a.k.a. "DynaProp Admin Console")

July 26, 2023~~March 16, 2023~~~~January 19, 2018~~~~July 25, 2017~~

Describes how to setup, configure, and use the Dynamic Property Manager Administration Console which is part of the Ford Java Frameworks (FJF).

For more information or feedback, contact the Java Center of Excellence.

\*For current version, see: <https://www.tc2.ford.com/ts/JCOE/default.aspx>

JAVA CENTER OF EXCELLENCE

### **Proprietary Notice**

This document and its contents are proprietary and comprise legally protected subject matter belonging to Ford Motor Company and are loaned on the basis of a confidential relationship. All use, reproduction, and disclosure are strictly controlled and prohibited without the prior written consent of Ford Motor Company.

## Revision History

This page documents general changes/modifications made to the document.

Version	Update Date	Change Description	Author
0.9	08/16/2004	Initial Draft.	aordaz
1.0	09/24/2004	Updates from user review.	wnielsen
1.1	09/28/2004	Additional formatting and content changes from user review.	wnielsen
1.2	04/04/2005	Minor updates to include support for DB2 and improve detailed descriptions of processes.	wnielsen
1.3	07/27/2005	Updated to reference updates that allow EAA to be plugged-in as the security authorization implementation instead of only referencing the <code>SimpleAuthorization</code> that is packaged with the framework.	wnielsen
1.4	10/09/2006	Minor revision to clarify how the ITCore and DynaPropAdmin frameworks have been split out.	wnielsen
1.5	2/21/2007	Updated for RAD7 & WAS6.1	jburnard
1.6	1/25/2008	Updated document for SQL Server	amehta5
1.7	4/15/2009	Updated document with the APS plug-in and setup details.	Vbhaska6
2.0	10/9/13	Reformatted document to newer format and re-wrote user guide sections to adopt the guide for the release of JSF.	dpainte7
2.1	11/9/13	Added sections describing how to use the password management module that is being released.	dpainte7
2.2	3/26/14	Added & updated sections for password management- new functionality supported in release (notifications, new credential group types) as well as changes to how the actual DynaProp credentials will be updated.	dpainte7
2.3	1/9/2014	Added section on how to receive notifications using Event Handler framework	oshvarts
2.4	12/7/2015	Added the sections for handling AutoSys Persistent Cookie renewal in DynaProp	Lzhang20
2.7	1/8/2016	Added details regarding functionality and behavior of the new Credential Type "SQL Server AlwaysOn Failover Cluster"	thall6
2.8	2/2/2016	Added sections for Health Check cookies.	tnewma27
2.9	7/18/2016	Added a note explaining that DynaProp cannot manage Health Check cookies for applications that share a single DynaProp database between multiple applications.	tnewma27
2.10	7/25/2016	Added section explaining how to add notification recipients that will be used to send notifications for all Credential Groups.	
<a href="#">2.15</a>	<a href="#">1/19/2017</a>	<a href="#">Explained DynaPropEnterPasswordsForbidden role and how to block users from hand-entering passwords</a>	<a href="#">oshvarts</a>

<a href="#">2.31</a>	<a href="#">3/16/2023</a>	<a href="#">Updated Dynaprop Admin Console to work without aps_config.xml file if ApsAPI namespace is loaded to DB</a>	<a href="#">ppratik4</a>
----------------------	---------------------------	--	--------------------------

## Contents

1. Introduction .....	11
1.1 Assumptions .....	11
1.2 Constraints / Minimum Requirements.....	12
2. Security Role Setup.....	13
2.1 Setup for Default SimpleAuthorization Implementation.....	13
2.2 Setup for EAA Authorization Implementation.....	16
2.2.1 One-Time Configuration / Setup .....	17
2.2.2 On-Going Configuration as Personnel Changes: .....	18
2.3 Setup for APS Authorization Implementation.....	19
2.3.1 One-Time Configuration / Setup .....	19
2.3.2 On-Going Configuration as Personnel Changes: .....	21
2.3.3 Using Dynaprop Admin Console without aps_config.xml file:.....	21
2.4 Troubleshooting.....	22
3. Configuring Database Credentials.....	23
3.1 Set Credentials.....	23
3.2 Maintaining DynaProp Credentials .....	24
3.3 Troubleshooting.....	24
3.3 Update period for Credential .....	25
4. Using the DynaProp Admin Console .....	26
4.1 Namespace Administration.....	26
4.1.1 How-To Find Namespaces .....	26
4.1.2 How-To Create a Namespace.....	26
4.1.3 How-To View / Edit a Namespace Tree Hierarchy (Property Groups and Properties) .....	27
4.1.4 How-To Edit Namespace Attributes .....	27
4.1.5 How-To Delete a Namespace.....	27
4.1.6 How-To Import a Namespace .....	27
4.1.7 How-To Export a Namespace.....	28
4.2 Administering Properties and Property Groups.....	28
4.2.1 How-To Create a Property Group: .....	29
4.2.2 How-To Edit Property Group Attributes: .....	29
4.2.3 How-To Delete a Property Group: .....	29

4.2.4 How-To Create a Property .....	29
4.2.5 How-To Edit a Property.....	30
4.2.6 How-To Delete a Property .....	30
4.2.7 How-To Copy a Property Group .....	30
4.2.7 How-To Copy a Property Group .....	30
4.3 Password Administration with Password Management Module.....	30
4.3.1 Types of Credential Groups Supported in Password Management .....	32
4.3.1.1 Oracle .....	32
4.3.1.2 SQLServer .....	32
4.3.1.3 SQL Server AlwaysOn Failover Cluster .....	33
4.3.1.4.1 RACF Credentials.....	33
4.3.1.4.2 RACF Passphrase.....	33
4.3.1.5 UNIX .....	33
4.3.1.6 AutoSys Persistent Cookies .....	33
4.3.1.7 Health Check Cookies .....	34
4.3.1.8 Other .....	34
4.3.2 Password Management Notifications .....	35
Step 1: Configure the Global Notification Settings .....	35
Step 2: Configure Notifications for a Credential Group .....	36
Step 2b: Configure Notifications for All Credential Groups .....	36
Step 2b: Configure Notifications for a Single Credential Group .....	37
Step 3: Begin Receiving Password Management Notifications.....	39
4.3.3 Password Management How-Tos.....	39
4.3.3.1 How-To Add a Credential Group: .....	39
4.3.3.2 How-To Edit a Credential Group: .....	40
4.3.3.3 How-To Delete a Credential Group: .....	40
4.3.3.4 How-To Add a Credential to a Credential Group: .....	40
4.3.3.5 How-To Load a Credential From a Namespace: .....	41
4.3.3.6 How-To Edit a Credential: .....	41
4.3.3.7 How-To Update a Credential's Password: .....	42
4.3.3.8 How-To Auto-generate a Credential's Password and Update It: .....	42
4.3.3.9 How-To Activate a Credential: .....	42
4.3.3.10 How-To Test a Credential for Connectivity: .....	43

4.3.3.11 How-To View a Credential's Decrypted Password:.....	43
4.3.3.12 How-To Delete a Credential:.....	43
4.3.3.13 How-To Export a Password File: .....	44
4.3.3.14 How-To Import a Password File:.....	44
4.3.3.15 How-To Manage AutoSys Persistent Cookies:.....	44
4.3.3.16 How-To Manage Health Check Cookies .....	49
4.3.4 Configuring DynaProp Credentials with Password Management .....	53
5. Quick Start Example.....	55
6. References and Resources.....	57
6.1 Using Property Manager .....	57
6.2 How-To Configure J2EE Security .....	57
6.3 FJF APS Security Client Framework .....	57
7. Appendix A – Property Value Encryption.....	58
7.1 Property Value Encryption Rules .....	58
7.2 Add New Property .....	58
7.3 Edit an Existing Property .....	58
8. Appendix B – Change Notifications .....	60
8.1 Change notifications overview .....	60
8.2 Enabling change notifications .....	60
8.3 Sample (default) configuration file.....	60
8.4 Customizing defaults.....	61
1. Introduction.....	5
1.1 Assumptions.....	5
1.2 Constraints / Minimum Requirements.....	6
2. Security Role Setup.....	7
2.1 Setup for Default SimpleAuthorization Implementation.....	7
2.2 Setup for EAA Authorization Implementation.....	9
2.2.1 One Time Configuration / Setup.....	10
2.2.2 On-Going Configuration as Personnel Changes:.....	11
2.3 Setup for APS Authorization Implementation.....	12
2.3.1 One-Time Configuration / Setup.....	12
2.3.2 On-Going Configuration as Personnel Changes:.....	14
2.4 Troubleshooting.....	14

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

3. Configuring Database Credentials.....	15
3.1 Set Credentials.....	15
3.2 Maintaining DynaProp Credentials.....	16
3.3 Troubleshooting.....	16
3.3 Update period for Credential.....	17
4. Using the DynaProp Admin Console.....	18
4.1 Namespace Administration.....	18
4.1.1 How To Find Namespaces.....	18
4.1.2 How To Create a Namespace.....	18
4.1.3 How To View / Edit a Namespace Tree Hierarchy (Property Groups and Properties).....	19
4.1.4 How To Edit Namespace Attributes.....	19
4.1.5 How To Delete a Namespace.....	19
4.1.6 How To Import a Namespace.....	19
4.1.7 How To Export a Namespace.....	20
4.2 Administering Properties and Property Groups.....	20
4.2.1 How To Create a Property Group.....	21
4.2.2 How To Edit Property Group Attributes.....	21
4.2.3 How To Delete a Property Group.....	21
4.2.4 How To Create a Property.....	21
4.2.5 How To Edit a Property.....	22
4.2.6 How To Delete a Property.....	22
4.2.7 How To Copy a Property Group.....	22
4.2.7 How To Copy a Property Group.....	22
4.3 Password Administration with Password Management Module.....	22
4.3.1 Types of Credential Groups Supported in Password Management.....	24
4.3.1.1 Oracle.....	24
4.3.1.2 SQLServer.....	24
4.3.1.3 SQL Server AlwaysOn Failover Cluster.....	25
4.3.1.4 RACF.....	25
4.3.1.5 UNIX.....	25
4.3.1.6 AutoSys Persistent Cookies.....	25

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted** ... [1]

**Formatted** ... [2]

**Formatted** ... [3]

**Formatted** ... [4]

**Formatted** ... [5]

**Formatted** ... [6]

**Formatted** ... [7]

**Formatted** ... [8]

**Formatted** ... [9]

**Formatted** ... [10]

**Formatted** ... [11]

**Formatted** ... [12]



4.3.1.7 Health Check Cookies.....	26
4.3.1.8 Other.....	26
4.3.2 Password Management Notifications.....	26
Step 1: Configure the Global Notification Settings.....	27
Step 2: Configure Notifications for a Credential Group.....	27
Step 3: Begin Receiving Password Management Notifications.....	29
4.3.3 Password Management How Tos.....	29
4.3.3.1 How To Add a Credential Group:.....	29
4.3.3.2 How To Edit a Credential Group:.....	30
4.3.3.3 How To Delete a Credential Group:.....	30
4.3.3.4 How To Add a Credential to a Credential Group:.....	30
4.3.3.5 How To Load a Credential From a Namespace:.....	31
4.3.3.6 How To Edit a Credential:.....	31
4.3.3.7 How To Update a Credential's Password:.....	32
4.3.3.8 How To Auto-generate a Credential's Password and Update It:.....	32
4.3.3.9 How To Activate a Credential:.....	32
4.3.3.10 How To Test a Credential for Connectivity:.....	33
4.3.3.11 How To View a Credential's Decrypted Password:.....	33
4.3.3.12 How To Delete a Credential:.....	33
4.3.3.13 How To Export a Password File:.....	34
4.3.3.14 How To Import a Password File:.....	34
4.3.3.15 How To Manage AutoSys Persistent Cookies:.....	34
4.3.3.16 How To Manage Health Check Cookies.....	39
4.3.4 Configuring DynaProp Credentials with Password Management.....	43
5. Quick Start Example.....	45
6. References and Resources.....	47
6.1 Using Property Manager.....	47
6.2 How To Configure J2EE Security.....	47
6.3 FJF APS Security Client Framework.....	47
7. Appendix A - Property Value Encryption.....	48
7.1 Property Value Encryption Rules.....	48
7.2 Add New Property.....	48

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted** ... [13]

**Formatted** ... [14]

**Formatted** ... [15]

**Formatted** ... [16]

**Formatted** ... [17]

**Formatted** ... [18]

**Formatted** ... [19]

**Formatted** ... [20]

**Formatted** ... [21]

**Formatted** ... [22]

**Formatted** ... [23]

**Formatted** ... [24]

**Formatted** ... [25]

**Formatted** ... [26]

**Formatted** ... [27]

7.3 Edit an Existing Property.....	48
8. Appendix B Change Notifications.....	50
8.1 Change notifications overview.....	50
8.2 Enabling change notifications.....	50
8.3 Sample (default) configuration file.....	50
8.4 Customizing defaults.....	51

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Verdana, Check spelling and grammar

## 1. Introduction

The Dynamic Property Manager Admin Console (a.k.a. DynaProp Admin Console) allows an application team to securely and dynamically administer application properties that are stored in an Oracle, DB2 or SQL Server database through a web interface. These properties are accessible during application runtime through the Property Manager service. Furthermore, the DynaProp Admin Console allows an application team to organize application properties as hierarchical property groups in a database table. At a high level, the DynaProp Admin Console provides the following functionality:

- Securely update Dynamic Property Manager database credentials (user ID and password).
- Import/Export of namespaces from/to an XML file that conforms to the `PropertyGroup.dtd` format.
- Provides the Create/Delete/Update namespace functions to manage namespaces within the database.
- Allows applications to define hierarchical properties and property groups.
- Encrypts property values using either symmetric or asymmetric encryption.
- Facilitates management of application database passwords by providing a secure location to store and passwords and providing functionality to update them with ease.
- Facilitates management of AutoSys Persistent Cookies.

### 1.1 Assumptions

It is assumed that application teams utilizing the DynaProp Admin Console performed the following tasks:

1. Read the "~~6.1 Using Property Manager~~*6.1 Using Property Manager*" document and have a sound understanding of the Property Manager service.
2. Configured the Dynamic Property Service as described in the "~~6.1 Using Property Manager~~*6.1 Using Property Manager*" document.
3. Acquired Oracle, DB2 or SQL Server database storage space and have followed the instructions of setting up the `DYNAPROP` database as described in the "~~6.1 Using Property Manager~~*6.1 Using Property Manager*" document. It should be noted that database space typically has a cost associated with it. If the application doesn't already have database dependencies, this could be additional cost to the application. It is anticipated that the space requirements for the `DYNAPROP` table should very minimal due to the nature of managing properties.
4. The application team has received concurrence from the Application DBA that JDBC Prepared Statements are an acceptable way to access the `DYNAPROP` table. The current version of DynaProp includes SQL in java files and does not currently support Stored Procedures.
5. The standard Ford WAS deployment framework is being utilized for deployment of the application and `DynaPropAdmin`. There are steps in the deployment framework that set-up underlying FJF configuration required by `DynaPropAdmin`.

Formatted: Font: Italic

Formatted: Font: Italic

Formatted: Font: Italic

6. Due to issues with JNDI lookups introduced with WAS clustering, the DynaProp credentials management requires the fix introduced in `dynapropadmin-v1.1` in order to work in a multi-node shared environment (usually QA and PROD).

## 1.2 Constraints / Minimum Requirements

The following constraints apply to the DynaProp Admin Console.

1. DynaProp Admin Console is targeted/tested for the Ford WAS 8.0 shared environment.
2. DynaProp Admin Console's target browser is Microsoft's Internet Explorer Version 8.
3. Oracle v8.1.0+, DB2 v7.1.1+ or SQL Server 2005+ is the required DBMS.
4. Support for Java Prepared Statements.
5. The current version of the Dynamic Property Manager Admin Console only supports an English user interface. Though there are many internationalization features already implemented, they have not been tested with other languages.
6. Not tested with Unicode characters.
7. Administrating application properties through the DynaProp Admin Console is specific ONLY to the environment it is running in.

**NOTE:**

- DynaProp Admin Console's Desktop and DEV environments should only be used to administer application properties contained in the DEV database.
- DynaProp Admin Console's QA environment should only be used to administer application properties contained in the QA database.
- DynaProp Admin Console's PROD environment should only be used to administer application properties contained in the PROD database.

## 2. Security Role Setup

The DynaProp Admin Console has been functionally divided into two areas, namely: managing passwords and configure Dynamic Property Manager database credentials; and, administering namespaces and properties. As a result, DynaProp Admin Console has provided five security roles to enable granular provisioning of the features and data accessible in the DynaProp:

- `DynaPropUnrestricted` – Super user security role able to configure database credentials, use the password management module, use the audit information module, and administer namespaces.
- `DynaPropCredentialsMgmt` – Security role ONLY able to configure database credentials and use password management. NOT able to administer namespaces.

**NOTE:** Due to size limitations, this role is defined as `DynaPropCredentials` in EAA.

- `DynaPropAdminConsole` – Security role ONLY able to administer namespaces. NOT able to use password management or update DynaProp credentials.
- `DynaPropReadOnly` – Security role ONLY able to read data. Cannot edit anything. Includes `ReadOnly` for namespaces and password management.
- `DynaPropPasswordRecovery` – Security role that provides only one additional function- the ability to view decrypted passwords for credentials in password management. Note that a user must have either the `DynaPropUnrestricted` or `DynaPropCredentialsMgmt` roles to use the feature this role provides, and ONLY this role provides the ability to view a decrypted password. This is the only feature that `DynaPropUnrestricted` does not automatically provide.
- `DynaPropAuditAdminConsole` – Security role ONLY able to view and manage audit information module.
- [`DynaPropEnterPasswordsForbidden`](#) – Special security role that forbids, rather than allows, something. Personnel granted this security role will not be allowed to enter passwords, they will only be permitted to Auto-generate new ones.

### 2.1 Setup for Default SimpleAuthorization Implementation

- A sample `simple_auth_groups.xml` file is distributed with `DynaPropAdminWeb`.
- The DynaProp Admin Console loads `simple_auth_groups.xml` at application start-up using Property Manager.
- The `simple_auth_groups.xml` root property group MUST be called `SimpleAuthorizationGroup`.

It should be understood that property group names *cdsid1*, *cdsid2*, and *cdsid3* used in [Figure 1](#)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE property-group PUBLIC "PropertyGroup.dtd" "PropertyGroup.dtd">

<!--***** -->
<!-- Property Group defining user-id / resources / privileges -->
<!-- that will be controlled through the SimpleAuthorizationProvider -->
<!-- Note: All user-ids are case sensitive and MUST be defined as -->
<!-- lower-case. -->
<!--***** -->
<property-group name="SimpleAuthorizationGroup">

  <property-group name="cdsid1">
    <property name="DynaPropUnrestricted">execute</property>
  </property-group>

  <property-group name="cdsid2">
    <property name="DynaPropCredentialsMgmt">execute</property>
  </property-group>

  <property-group name="cdsid3">
    <property name="DynaPropAdminConsole">execute</property>
  </property-group>

  <property-group name="cdsid4">
    <property name="DynaPropReadOnly">execute</property>
  </property-group>
</property-group>
```

Figure 1 – Example simple\_auth\_groups.xml file

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE property-group PUBLIC "PropertyGroup.dtd" "PropertyGroup.dtd">

<!--***** -->
<!-- Property Group defining user-id / resources / privileges -->
<!-- that will be controlled through the SimpleAuthorizationProvider -->
<!-- Note: All user-ids are case sensitive and MUST be defined as -->
<!-- lower-case. -->
<!--***** -->
<property-group name="SimpleAuthorizationGroup">
  <property-group name="cdsid1">
    <property name="DynaPropUnrestricted">execute</property>
  </property-group>

  <property-group name="cdsid2">
    <property name="DynaPropCredentialsMgmt">execute</property>
  </property-group>

  <property-group name="cdsid3">
    <property name="DynaPropAdminConsole">execute</property>
  </property-group>

  <property-group name="cdsid4">
    <property name="DynaPropReadOnly">execute</property>
  </property-group>
</property-group>

```

Figure 1 — Example simple\_auth\_groups.xml file

- were used as examples. It is up to application teams to determine which user's CDSIDs should have appropriate DynaProp Admin Console access.
- CDSIDs can be assigned more than one role, but, if the broadest scoped role will take precedence. Ex. If a user were given DynaPropCredentialsMgmt and DynaPropReadOnly, the user would have the ability to use password management and read namespaces. However, if a user were given DynaPropUnrestricted and DynaPropReadOnly, DynaPropUnrestricted is the broadest role, so that will be used. Further, if duplicate cdsid definitions are defined in the simple\_auth\_groups.xml file, only the last cdsid entry read by the XML parser will be assigned to the user.
- DynaProp Admin Console's current version does **NOT** have permission privileges implemented. Nevertheless, the properties in simple\_auth\_groups.xml should specify **execute** as their value.
- DynaProp Admin Console's current version has only been tested with its default security provider and with the APS security provider.

- Application teams should place the simple\_auth\_groups.xml file in their <<AppEAR>>/resources/properties/<<environment>>/fjf/ folder.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE property-group PUBLIC "PropertyGroup.dtd" "PropertyGroup.dtd">

<!--***** -->
<!-- Property Group defining user-id / resources / privileges -->
<!-- that will be controlled through the SimpleAuthorizationProvider -->
<!-- Note: All user-ids are case sensitive and MUST be defined as -->
<!-- lower-case. -->
<!--***** -->
<property-group name="SimpleAuthorizationGroup">

  <property-group name="cdsid1">
    <property name="DynaPropUnrestricted">execute</property>
  </property-group>

  <property-group name="cdsid2">
    <property name="DynaPropCredentialsMgmt">execute</property>
  </property-group>

  <property-group name="cdsid3">
    <property name="DynaPropAdminConsole">execute</property>
  </property-group>

  <property-group name="cdsid4">
    <property name="DynaPropReadOnly">execute</property>
  </property-group>
</property-group>
```

Figure 1 – Example simple\_auth\_groups.xml file

## 2.2 Setup for EAA Authorization Implementation

**Note: EAA is in declining mode and all new applications should use APS authorization.**

The purpose of this section is to outline how to configure the Dynamic Property Manager framework to work with an existing EAA setup. It is beyond the scope of this paper to detail how to setup EAA itself; only what is needed for the Dynamic Property Manager Admin Console to work properly. For information on EAA, see the 6.3 FJF APS Security Client Framework page on the Ford JCOE website.

As described in the "6.1 Using Property Manager" document, the fjf-security-plugins-config.xml file specifies what authorization implementation DynaProp uses. The file needs



to specify the EAA authorization provider in order for the Dynamic Property Manager Admin Console to use it as shown:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE property-group SYSTEM "PropertyGroup.dtd">
<property-group name="AuthorizationProviderFactory">
  <property name="DYNAPROP_FW">
    com.ford.it.security.plugins.eaa.EaaAuthorizationProviderFactory
  </property>
</property-group>
```

### 2.2.1 One-Time Configuration / Setup

Using the EAA Security Admin Console, the following one-time setup steps configure EAA with the roles and groups that the Dynamic Property Manager Admin Console requires:

1. In *"Application Administrator"* screen under *"Resources"*, create the following EAA Application Resources:
  - 1.1. DynaPropUnrestricted
  - 1.2. DynaPropCredentialsMgmt
  - 1.3. DynaPropAdminConsole
  - 1.4. DynaPropReadOnly
  - 1.5. DynaPropPasswordRecovery
2. In *"Application Administrator"* screen under *"Resources"*, create the following EAA Application Roles:
  - 2.1. DynaPropUnrestricted
  - 2.2. DynaPropCredentialsMgmt
  - 2.3. DynaPropAdminConsole
  - 2.4. DynaPropReadOnly
  - 2.5. DynaPropPasswordRecovery
3. In *"Application Administrator"* screen under *"Resources"*, create the following EAA Access Policies:

Resource:	DynaPropUnrestricted
Role:	DynaPropUnrestricted
Privilege:	READ

Resource:	DynaPropCredentialsMgmt
Role:	DynaPropCredentialsMgmt
Privilege:	READ

Resource:	DynaPropAdminConsole
Role:	DynaPropAdminConsole
Privilege:	READ

Resource:	DynaPropReadOnly
Role:	DynaPropReadOnly
Privilege:	READ

Resource:	DynaPropPasswordRecovery
Role:	DynaPropPasswordRecovery
Privilege:	READ

4. In *"Application Administrator"* screen under *"Groups"*, create the following EAA Groups:
  - 4.1. DynaPropUnrestricted
  - 4.2. DynaPropCredentialsMgmt
  - 4.3. DynaPropAdminConsole
  - 4.4. DynaPropReadOnly
  - 4.5. DynaPropPasswordRecovery
5. In *"Application Administrator"* screen under *"Groups"*, assign the newly created Roles to Groups:
  - 5.1. Assign the DynaPropUnrestricted role to the DynaPropUnrestricted group.
  - 5.2. Assign the DynaPropCredentialsMgmt role to the DynaPropCredentialsMgmt group.
  - 5.3. Assign the DynaPropAdminConsole role to the DynaPropAdminConsole group.
  - 5.4. Assign the DynaPropReadOnly role to the DynaPropReadOnly group.
  - 5.5. Assign the DynaPropPasswordRecovery to the DynaPropPasswordRecovery group.

### 2.2.2 On-Going Configuration as Personnel Changes:

1. In the *"Group Administrator"* screen, add users to DynaPropUnrestricted group:
  - 1.1. Add the CDS id's of the users that should have unrestricted access to the DynaProp Admin Console (i.e., be able to set the credentials and/or configure properties in the DynaProp database).

Special care should be taken when assigning users to this group in the Production environment as it may raise 'segregation of duties' concerns for the application SCRP.
2. In the *"Group Administrator"* screen, add users to DynaPropCredentialsMgmt group:
  - 2.1. Add the CDS id's of the users that should only be able to set the credentials for the application.

3. In the "Group Administrator" screen, add users to DynaPropAdminConsole group:
  - 3.1. Add the CDS id's of the users that should only be able to configure properties in the DynaProp database.
4. In the "Group Administrator" screen, add users to DynaPropReadOnly group:
  - 4.1. Add the CDS id's of the users that should only be able to read data from namespaces using the console but should not edit anything.
5. In the "Group Administrator" screen, add users to DynaPropPasswordRecovery group:
  - 5.1. Add the CDS id's of the users that should be able to view a decrypted password for a credential in password management. For security reasons, BE CAUTIOUS WHEN DISTRIBUTING THIS ROLE.

## 2.3 Setup for APS Authorization Implementation

The purpose of this section is to outline how to configure the Dynamic Property Manager framework to work with an existing APS setup. It is beyond the scope of this paper to detail how to setup APS itself; only what is needed for the Dynamic Property Manager Admin Console to work properly. For information on APS, see the developer's index on the JCOE website and look for APS.

Prior to using the APS authorization, ensure that a PropertyGroup named [LDAPLookupConfig.FDS.Credentials] is loaded. APS plug-in uses this property to access user information from LDAP. You can get more information about this property from iTCore's LDAPLookupFactory. As described in the "6.1 Using Property Manager" document, the `fjf-security-plugins-config.xml` file specifies what authorization implementation DynaProp uses. The file needs to specify the APS authorization provider in order for the Dynamic Property Manager Admin Console to use it as shown:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE property-group SYSTEM "PropertyGroup.dtd">
<property-group name="AuthorizationProviderFactory">
  <property name="DYNAPROP_FW">
    com.ford.it.security.plugins.aps.APSAuthorizationProviderFactory</property>
  <property-group name="APSSecurityPluginConfig">
    <property name="class">
      com.ford.it.security.plugins.aps.APSAuthorizationProviderFactory
    </property>
    <property name="appName">APPNAME</property>
    <property name="apsConfigFile">aps_config.xml</property>
  </property-group>
</property-group>
```

### 2.3.1 One-Time Configuration / Setup

Using the [APS Security Admin Console](#), the following one-time setup steps configure APS with the roles and groups that the Dynamic Property Manager Admin Console requires:

1. In "Application Administrator" screen under "Resources", create the following APS Application Resources:
  - 1.1. DynaPropUnrestricted

- 1.2. DynaPropCredentialsMgmt
- 1.3. DynaPropAdminConsole
- 1.4. DynaPropReadOnly
- 1.5. DynaPropPasswordRecovery
2. Using the [security provisioning service](#), a "Role Administrator" should create the following APS Application Roles:
  - 2.1. DynaPropUnrestricted
  - 2.2. DynaPropCredentialsMgmt
  - 2.3. DynaPropAdminConsole
  - 2.4. DynaPropReadOnly
  - 2.5. DynaPropPasswordRecovery
3. In "Application Administrator" screen under "Resources", Define the following resource access policies:

Resource:	DynaPropUnrestricted
Role:	At least a member of DynaPropUnrestricted
Privilege:	EXECUTE

Resource:	DynaPropCredentialsMgmt
Role:	At least a member of DynaPropUnrestricted or DynaPropCredentialsMgmt
Privilege:	EXECUTE

Resource:	DynaPropAdminConsole
Role:	At least a member of DynaPropUnrestricted or DynaPropAdminConsole
Privilege:	EXECUTE

Resource:	DynaPropReadOnly
Role:	At least a member of DynaPropReadOnly
Privilege:	EXECUTE

Resource:	DynaPropPasswordRecovery
Role:	At least a member of PasswordRecovery- to use this, user must also be a member of Unrestricted or CredentialsMgmt
Privilege:	EXECUTE

### 2.3.2 On-Going Configuration as Personnel Changes:

1. In the "Group Administrator" screen, add users to DynaPropUnrestricted group:
  - 1.1. Add the CDS id's of the users that should have unrestricted access to the DynaProp Admin Console (i.e., be able to set the credentials and/or configure properties in the DynaProp database).

Special care should be taken when assigning users to this group in the Production environment as it may raise 'segregation of duties' concerns for the application SCRP.
2. In the "Group Administrator" screen, add users to DynaPropCredentialsMgmt group:
  - 2.1. Add the CDS id's of the users that should only be able to set the credentials for the application.
3. In the "Group Administrator" screen, add users to DynaPropAdminConsole group:
  - 3.1. Add the CDS id's of the users that should only be able to configure properties in the DynaProp database.
4. In the "Group Administrator" screen, add users to DynaPropReadOnly group:
  - 4.1. Add the CDS id's of the users that should only be able to read data from namespaces using the console but should not edit anything.
5. In the "Group Administrator" screen, add users to DynaPropPasswordRecovery group:
  - 5.1. Add the CDS id's of the users that should be able to view a decrypted password for a credential in password management. For security reasons, BE CAUTIOUS WHEN DISTRIBUTING THIS ROLE.

### 2.3.3 Using Dynaprop Admin Console without aps\_config.xml file:

1. Import the existing aps\_config.xml file into your Dynaprop, using the **Import Namespace** feature
  - a. This will show the property values which are being loaded. Ensure that for any secrets, you set **Encrypted** as true
2. In fjf-security-plugins-config.xml, update the **apsConfigFile property to be blank**, removing this will cause errors. For example -

**Before**

```
<property name="apsConfigFile">aps_config.xml</property>
```

**Formatted:** Outline numbered + Level: 3 + Numbering Style: 1, 2, 3, ... + Start at: 3 + Alignment: Left + Aligned at: 0.25" + Indent at: 0.75"

**Formatted:** Font: Bold

#### **After**

<property name="apsConfigFile"></property>

3. Ensure that the Dynaprop Admin Console loads up

4. Remove the aps\_config.xml file and verify it's functionality

5.2.

**Formatted:** Font: Bold

**Formatted:** List Paragraph, Space Before: 0 pt, Numbered + Level: 1 + Numbering Style: 1, 2, 3, ... + Start at: 1 + Alignment: Left + Aligned at: 0.25" + Indent at: 0.5"

**Formatted:** Indent: Left: 0.3", No bullets or numbering, Tab stops: Not at 0.81"

## 2.4 Troubleshooting

Correct configuration should be tested through the following url: `http://<<environment specific domain name>>/DynaPropAdminWeb/`. If configured incorrectly you should see the "Customer Error 403: access forbidden message". Otherwise you should get the *Dynamic Property Manager Manage Namespaces* view, if you have the correct role to manage namespaces (any role but CredentialsMgmt).

## 3. Configuring Database Credentials

### 3.1 Set Credentials

Before administrating a set of namespaces, the *Dynamic Property Manager Database Credentials* must be configured. This feature of the Admin Console is only available as an option if security is properly setup for the logged in user. If security roles are improperly set up, the user will not see or be able to access the page from which credentials can be changed.

The *Dynamic Property Manager Database Credentials* view defines the credentials needed for the console to connect to the DYNAPROP database. The location of the credentials file is defined via the DYNAPROP\_FILENAME system property. This system property is automatically defined by the WAS deployment framework when the following line is included in the

application.py file:

```
set USE_DYNAPROP="TRUE"
```

Depending on target environment, database credentials configuration should be performed by following the next steps:

1. Ensure that the FJFConfig framework is deployed to the target environment as the database credentials page requires its FJFAsymmetric encryption functionality. For the desktop environment, ensure that the desktop version of FJFConfig was downloaded with the Desktop Server Configurator as described in the "[6.1 Using Property Manager](#)" document.
2. To configure *Dynamic Property Manager Database Credentials* and create a credentials file with an asymmetrically encrypted password:
  - Type the environment specific URL into their Web browser's Address bar after the server with the deployed DynaPropAdmin.ear project is running:  
http://<<environment specific domain name>>/DynaPropAdminWeb/
  - After loading the application in the browser, if there is no connectivity, the application should redirect to the *Server Status* page, which will display an error indicating that DynaProp Admin Console cannot connect to the database (the credentials have not been configured yet).
  - **NOTE:** To set the database credentials that DynaProp will use to connect, the user performing this operation must have the permission DynaPropCredentialsMgmt.
  - To set the database credentials for DynaProp, look to the main menu bar. Mouse over the *DynaProp Configuration* tab, and click on the *Update DynaProp Database Credentials* link. **Note:** when the logged in user only has permissions to setup the database credentials (the DynaPropCredentialsMgmt role and no other) they will automatically be routed to the *Edit Credential Group* page to set the credentials, and bypass the *Manage Passwords* screen, as they do not have access to manage password data when there is no database connection.

Commented [DsP(1): application.jacl-> application.py

Formatted: Font: Italic

- The *Update DynaProp Database Credentials* link will take you to the auto-generated 'DynaProp Cfg' credential group in Password Management. (Please see the '**4.3.4 Configuring DynaProp Credentials with Password Management**' section of the Password Management section of this document for more detailed information and directions).
- **IMPORTANT:** To configure DynaProp, once on the page above, Add a credential to this group by clicking the 'Add Credential' button. Then enter User ID, Password and confirmation of the Password that is needed to connect to the previously setup DYNAPROP database table. Note, as stated in the "*6.1 Using Property Manager6.1 Using Property Manager*" document, this user name and password should be different than what the application uses to access their app-specific tables.
- **After this step, Restart the JVM to invalidate any cached credentials and refresh the system.**

Formatted: Font: Italic

## 3.2 Maintaining DynaProp Credentials

1. Once the credentials have been set and DynaProp has been restarted, the system should be configured and ready to use.
2. Maintaining the DynaProp password consists of remembering to update the password to avoid password expiration. Password management makes that very easy to do—simply go to Password Management and edit the 'DynaProp Cfg' group, and update the password for the credential that exists there. See the password management section for more information.
3. If, for some reason, the password expires, you will have to set up the credentials for DynaProp as if you were setting them up the first time.
  - I.e. manually change the password in the database,
  - And then go to DynaProp and click the '*Update DynaProp Database Credentials*' Link,
  - And add a new credential.
4. **NOTE:** If for some reason the password does expire, or DynaProp cannot access the database for some other reason, all functionality will be disabled and the page will redirect to 'Server Status', which will display a message describing the issue and why DynaProp cannot access the database.
5. **NOTE:** See section **4.3.4** for more information on using Password Management to update DynaProp credentials.

## 3.3 Troubleshooting

To troubleshoot the *Dynamic Property Manager Database Credentials*:

- Mouse over the *DynaProp Configuration* tab, and click on the *Server Status* link.
  - Read the status message on the *Dynamic Property Manager Database Credentials Status* table, which will indicate database connectivity success or failure and potential errors (if any). These status messages are for a technical / systems user audience and are not intended for Business customers!



- Note that an *Access forbidden* page will be displayed when an unauthorized user attempts to access a page. The `DynaPropAdminConsole` security role is not allowed to access the *Update Database Credentials* page, so keep that in mind.

### 3.3 Update period for Credential

If *highly dynamic flag* is set and **update period** (timeout period) for a Credentials namespace set greater than null then credentials are updated from database every time as request comes after update-period interval. If *highly dynamic flag* is set and update period = 0 then Credentials are updated with frequency of preset interval (timeout period) equal 10 minutes.

## 4. Using the DynaProp Admin Console

### 4.1 Namespace Administration

After configuring the database credentials, application teams can begin administering namespaces by simply loading the application. If the user has the correct privileges, the *Manage Namespaces* page will be the home page for the user. The home link will route back to this page, and you can also get to it by mousing over the *Namespace* tab on the main menu bar and clicking *Manage Namespaces* as well. Specific actions that can be performed on a namespace from the *Dynamic Property Manager Administration Console* are listed below.

#### 4.1.1 How-To Find Namespaces

- If namespaces are in your system, the Manage Namespaces page will display a table of namespaces and some corresponding data for each namespace (such as last update user & time, highly dynamic, etc.) Each of these columns is searchable by typing in the filter box above the column or selecting a value in the box. **Note** that if the filter box is just a blank box, it operates by a 'contains' match to whatever you type on the values in the column.

#### 4.1.2 How-To Create a Namespace

- Click on the *Create Namespace* button on the *Manage Namespaces* page, or, mouse over the *Namespace* menu tab and click the link called *Create Namespace*.
- Type a namespace name in the *Namespace Name* text box. This namespace name is the unique identifier for properties stored in the database and is needed when the application run-time code loads the properties from the database. The namespace name must be unique. The other possible options on this screen are:
  - The *Search Parent Group* option: When asking for a property, search parent group instructs the property manager to search the groups up the namespace hierarchy for a property, if no property of that name is specified in the current group. Used for specifying a default value for a property, and allowing that value to be overwritten at a lower level.
  - The *Highly Dynamic Flag* option: Instructs Property Manager to compare the version of a namespace loaded in memory with the version stored in the database each time a property within that namespace is read by the application via the `PropertyManager` singleton (this check does **not** occur if the `PropertyGroup` is stored in a local variable for example). If the versions are different, `PropertyManager` automatically loads (see *below notes about "update period"*) the new version of the namespace into memory, overriding all of its contents with the updated information. Highly dynamic namespaces should only contain a limited amount of information and should not have properties that require programming changes if the values are modified (see the ["6.1 Using Property Manager"](#) document for more information on proper use of Highly Dynamic namespaces).
  - In addition to *"Highly Dynamic Flag"*, Property Manager can handle *"update-period"* attribute in the highly dynamic namespace. *This attribute is only*

Formatted: Font: Italic

available if the namespace is highly dynamic. If update-period is set to value > 0, Property Manager updates the namespace from the database (as described in previous item above) *only if time interval from previous request is greater than update-period value*. Update period value = 0 is simply treated as *Highly Dynamic Flag- always update*.

- **Note:** To set update period value=0 select "Always" from the drop down ("Always" means that this value results in the behavior equivalent to previous version of *Highly Dynamic Flag* - *if it is set the property is updated from the database on each request*).
- For *Highly dynamic credentials* managed by a Credential Manager (in addition to Property Manager) *Highly Dynamic Flag* algorithm behaves differently:  
If a credential namespace set as *highly dynamic with update period = 0* (see above) then this credential namespace is updated based on internal Credential Manager value = 10 minutes. If updates period for the credential namespace set > 0 then Credential Manager uses this update period for this particular credential. Such behavior provides for possibility for individual control of update period for the different Highly Dynamic Credentials as well as compatibility with older code (*if update period = 0*).
- Click on the *Save Namespace* button to submit the form and create the namespace. The user will be redirected back to the *Manage Namespaces* page.

#### 4.1.3 How-To View / Edit a Namespace Tree Hierarchy (Property Groups and Properties)

- From the *Manage Namespaces* page, click on the namespace name link of your choice to manage that namespace and its contents. **Note:** The details of property group and property administration (editing the namespace tree) are covered in the sections 4.2 and 4.3.

#### 4.1.4 How-To Edit Namespace Attributes

- From the *Manage Namespaces* page, click on the *Edit* link under the actions column that corresponds to the namespace that needs to be changed.
- Edit the namespace name, *Search Parent Group* flag, and/or *Highly Dynamic Flag*.
- Beneath those fields, the page will also show the last user to update this namespace and the time of that update.
- Click on the *Save Namespace* button to modify the namespace.

#### 4.1.5 How-To Delete a Namespace

- From the *Manage Namespaces* page, under the actions column, click on the *More* link to display additional options, and then click *Delete Namespace*. A dialog will appear asking for confirmation from the user, because this is a permanent action. Click *Yes* to permanently delete the namespace.

#### 4.1.6 How-To Import a Namespace

- From the *Manage Namespaces* page, click on the *Import Namespace* button. Alternatively, mouse over the *Namespace* tab on the main menu bar and click the *Import Namespace* link.
- From the *Import Namespace* page, click on the *Choose Namespace to Import* button and navigate to the namespace file source.

- From the *Choose file* pop-up window, click on the *Open* button after selecting the namespace XML file.
- From the *Import a Namespace* View, click on the *Upload* button to process the namespace chosen or click on the blue *Cancel* button to remove the file. Click on the gray cancel button to return to the *Manage Namespaces* page.

**IMPORTANT NOTE:**

Be careful when trying to import encrypted values from another environment! The encryption keys will be different between environments so you will need to manually re-enter your sensitive data as clear-text and have the admin console re-encrypt the sensitive data in the new environment.

#### 4.1.7 How-To Export a Namespace

- From the *Manage Namespaces* View, for the specific namespace you wish to export, click on the *More* link to display additional options, and then click *Export Namespace*. This launches a file download for your browser.
- From the *File Download* pop-up window, click on the *Save* button or the *OK* button.
- (Internet Explorer 8) From the *Save As* pop-up window, navigate to the where on the local disk you wish to export the file and click on the *Save* button.

## 4.2 Administering Properties and Property Groups

Once a namespace has been created, application teams are able to administer its property groups and properties. The namespace itself acts as a root property group, to which either individual properties or additional property groups can be added. This forms a tree-like structure. Property groups and properties are administered through the *Edit Namespace Property Group and Properties* page.

On this page, click the arrows next to the namespace or property group to expand that group and show its contents on screen. The table will show the type of row (Namespace, Property Group, or Property) and also show the data that a property contains. On the far right is a list of contextualized actions for each data type. For instance, a namespace cannot be edited, but a property group or property may be added to it. A list of these actions and contexts follows below:

**IMPORTANT NOTE:**

Modified values on the *Edit Namespace Property Group and Properties* page are NOT committed to the database until the *Save* button is clicked on the main page. The browser will alert you if it believes you are navigating away from the page without saving.

#### 4.2.1 How-To Create a Property Group:

- **Context:** This action item is only available from property groups and the root namespace. (A property group may be added to a namespace or another property group, but not to a property)
- Click the *Actions* link to show a list of available actions for the type of row that you have selected. Then click *Add Property Group*.
- The *Add Property Group* dialog will appear. Type a property group name of your choice in the group name text box.
- From the *Add Property Group* dialog, click on the *Add Property Group* button to add the property group to the namespace. Note that this is not persisted to the database until the *Save Namespace* button is clicked on the actual page.

#### 4.2.2 How-To Edit Property Group Attributes:

The only attribute you can modify is the name of the Property Group.

- **Context:** This action item is only available from a selected Property Group.
- Click the *Actions* link to show a list of available actions for the type of row that you have selected. Then click *Edit Property Group*.
- The *Edit Property Group* dialog will appear. Type a new property group name of your choice in the group name text box.
- From the *Edit Property Group* dialog, click on the *Update Property* button to save the changes to the property group. Note that this is not persisted to the database until the *Save Namespace* button is clicked on the actual page.

#### 4.2.3 How-To Delete a Property Group:

- **Context:** This action item is available for property groups or properties.
- Click the *Actions* link to show a list of available actions for the type of row that you have selected. Then click *Delete Property or Group*.
- The *Delete Property or Group* confirmation dialog will appear. Click the *yes* button to delete the property or group.
- **Note:** Deleting a property group will delete all nested items inside of it, including other property groups. This action will not be persisted to the database until the *Save Namespace* button is clicked on the actual page.

#### 4.2.4 How-To Create a Property

- **Context:** This action item is available for the root namespace and property groups, but not to other properties.
- Click the *Actions* link to show a list of available actions for the type of row that you have selected. Then click *Add Property*.
- The *Add Property* dialog will appear. Fill in the form with a property name and data.  
**Note:** If using encryption, follow the rules found in the [Appendix A --Property Value Encryption](#) section.
- From the *Add Property* dialog, click on the *Add Property* button to add the property to the namespace. Note that this is not persisted to the database until the *Save Namespace* button is clicked on the page.

Formatted: Font: Bold, Italic

Formatted: Font: Bold, Italic

Formatted: Font: Bold, Italic

#### 4.2.5 How-To Edit a Property

- **Context:** This action item is available for properties only.
- Click the *Actions* link to show a list of available actions for the type of row that you have selected. Then click *Edit Property*.
- The *Edit Property* dialog will appear. Fill in the form with a property name and data.  
**Note:** If using encryption, follow the rules found in the [Appendix A – Property Value Encryption](#) section.
- From the *Edit Property* dialog, click on the *Update Property* button to submit your changes. Note that this is not persisted to the database until the *Save Namespace* button is clicked on the page

#### 4.2.6 How-To Delete a Property

- Same as deleting a property group – see section 4.2.3. Context must be a property.

#### 4.2.7 How-To Copy a Property Group

- **Context:** This action item is available for Property Groups only, and not namespaces.
- Click the *Actions* link to show a list of available actions for the type of row that you have selected. Then click *Copy Property Group*.
- The *Copy Property Group* dialog will appear. Fill in the form with the name that you want to call the copy, and any comments for the copied group.
- Click the *Add Property Group* button to add a copy of the property group to the location as the original group, with the settings selected.  
**Note:** This operation will also copy all nested properties and property groups within the property group that is being copied.

#### 4.2.7 How-To Copy a Property Group

- **Context:** This action item is available for any non-namespace property or group.
- Click the *Actions* link to show a list of available actions for the type of row that you have selected. Then click *Move Property Group*.
- The *Move Property or Group* dialog will appear. The item that will be moved is shown in bold on a tree-representation of the structure of the namespace. Click another property group in the tree structure to identify it as a new parent of the item shown in bold.
- Click the *Move Property or Group* button to move the bold property or group to the new desired parent, which is highlighted.

### 4.3 Password Administration with Password Management Module

The Password Management module of DynaProp Admin Console allows a user to manage different kinds of ids and passwords (known collectively as '**credentials**'), and automatically update those passwords in the database. Password management can go

**Formatted:** Font: Bold, Italic

**Formatted:** Font: Bold, Italic

**Formatted:** Font: Bold, Italic

to the '**source system**' (ex. Oracle) and **change the password in that source system automatically**, and then, if that operation succeeds, it stores the updated data in the password management module.

This ties into DynaProp in the following way: When a user creates a credential group they are forced to select a DynaProp namespace that the group will reference. Then, the user can place the new credential in the DynaProp namespace that is specified, which is called '**Activating**' a credential. At this point the application referencing that namespace to retrieve the credential will now be pointing at the new credential. See the below diagram for an illustrated view of how it fits together.

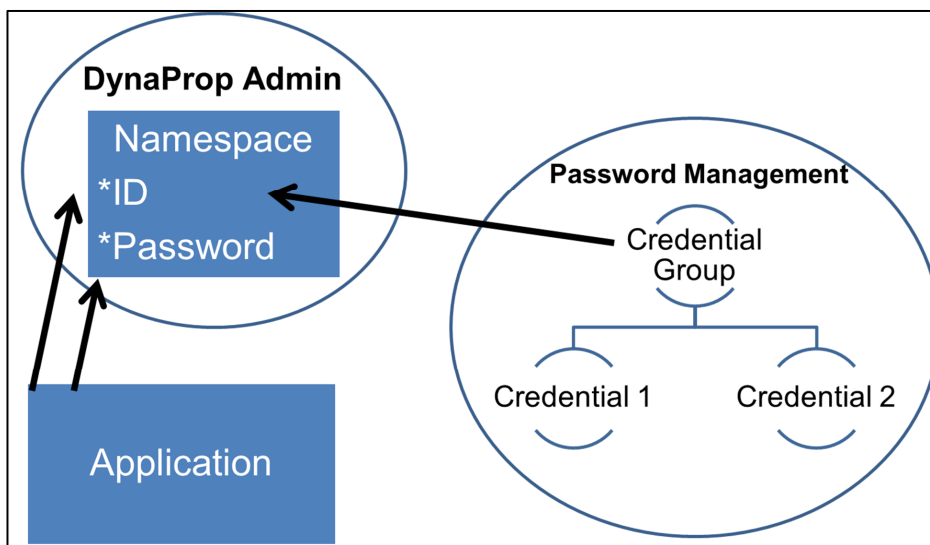
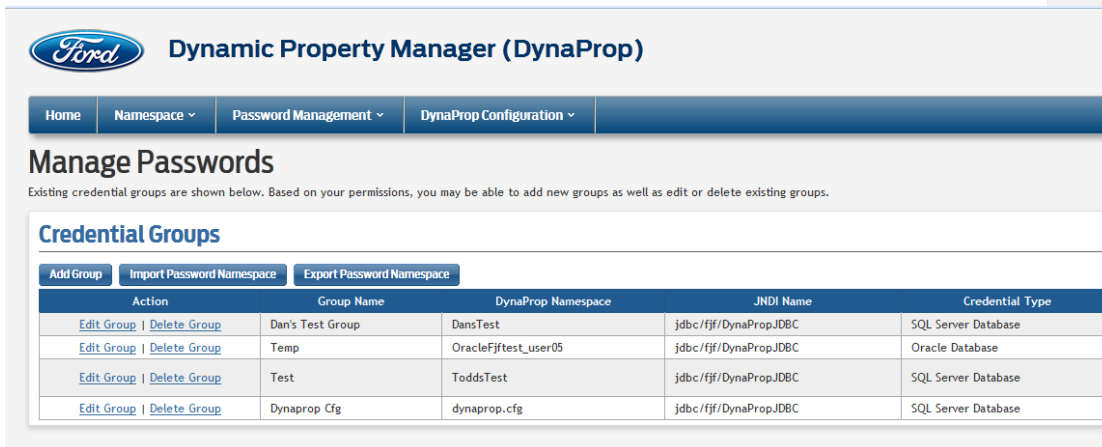


Figure 2 – Password Administration with Password Management Module

The Password Management module is not enabled by default, to view and use it; a user must have the correct security privileges. All password management functionality is accessed from a page called *Manage Passwords*, which is accessible from the Password Management menu item on the main menu bar.

In Password Management, credentials are organized into “Credential Groups”. Grouping related credentials facilitates the JCOE pattern of a dual-id approach, and allows several credentials to be stored for any given database. These can then be rotated in and out of use. Because of this structure, individual credentials can only be added to existing credential groups. The *Manage Passwords* page lists all credential groups and allows a user to add or edit credential groups (editing a credential group includes adding credentials to it), as well as to view the credentials in a credential group in a column on the far right. (Not visible in the figure below).



**Dynamic Property Manager (DynaProp)**

Home | Namespace | Password Management | DynaProp Configuration

## Manage Passwords

Existing credential groups are shown below. Based on your permissions, you may be able to add new groups as well as edit or delete existing groups.

### Credential Groups

Add Group | Import Password Namespace | Export Password Namespace

Action	Group Name	DynaProp Namespace	JNDI Name	Credential Type
<a href="#">Edit Group</a>   <a href="#">Delete Group</a>	Dan's Test Group	DansTest	jdbc/fjf/DynaProp.JDBC	SQL Server Database
<a href="#">Edit Group</a>   <a href="#">Delete Group</a>	Temp	OracleFjfTest_user05	jdbc/fjf/DynaProp.JDBC	Oracle Database
<a href="#">Edit Group</a>   <a href="#">Delete Group</a>	Test	ToddsTest	jdbc/fjf/DynaProp.JDBC	SQL Server Database
<a href="#">Edit Group</a>   <a href="#">Delete Group</a>	DynaProp Cfg	dynaProp.cfg	jdbc/fjf/DynaProp.JDBC	SQL Server Database

Figure 3 – Left Side of the Manage Passwords Page Showing Controls and Features

### 4.3.1 Types of Credential Groups Supported in Password Management

There are several different types of credentials that are compatible with Password Management. The following section describes these types and any important details about a particular type.

#### 4.3.1.1 Oracle

The 'Oracle' credential group supports managing credentials for an Oracle database. A JNDI name is required information to access the database. See the 'How-To' section for further details.

#### 4.3.1.2 SQLServer

The 'SQLServer' credential group supports managing credentials for a SQLServer database. A JNDI name is required information to access the database. See the 'How-To' section for further details.



#### 4.3.1.3 SQL Server AlwaysOn Failover Cluster

The 'SQL Server AlwaysOn Failover Cluster' credential group supports managing credentials for a SQL Server database that is configured in an AlwaysOn Failover Cluster. This type AlwaysOn solution allows for high availability of a database resource to an application through the use of primary and secondary server(s). Using this type in Password Management supports managing the password update on all the servers that make up the AlwaysOn Failover Cluster, as opposed to simply using the SQL Server type which will only manage the password on the primary server. A JNDI name that is configured to use an AlwaysOn listener is required information to access the database. See the 'How-To' section for further details.

#### 4.3.1.4.1 RACF Credentials

The 'RACF' credential group supports managing credentials that are 'RACF' credentials stored and maintained on the Company's mainframe systems. From an application's perspective, this RACF credential could be used in several different ways, i.e. as a DB2 id, or as an FTP id. To be able to connect to a RACF system, you will need to specify the hostname that your RACF credential connects to (ex. syc1.dearborn.ford.com).

#### 4.3.1.4.2 RACF Passphrase

The 'RACF Passphrase' credential group supports managing credentials that are 'RACF' credentials stored and maintained on the Company's mainframe systems. These credentials support longer fields. From an application's perspective, this RACF credential could be used in several different ways, i.e. as a DB2 id, or as an FTP id. To be able to connect to a RACF system, you will need to specify the hostname that your RACF credential connects to (ex. syc1.dearborn.ford.com).

#### 4.3.1.5 UNIX

The 'UNIX' credential group supports managing UNIX account credentials on UNIX servers. A common use case for this credential group might be an FTP id. To be able to connect to a UNIX server, you will need to specify the server's host name to connect to it (ex. fcws123.dearborn.ford.com)

#### 4.3.1.6 AutoSys Persistent Cookies

A lot of application teams use AutoSys to trigger a batch job hosted in their Java App servers. AutoSys uses a persistent cookie to get authenticated by the application team's App Server to trigger the job to run. The persistent cookie is only valid for one year and thus it has to be renewed yearly.

There are two AutoSys servers (fcas2061.md2.ford.com and eccas2061.md2.ford.com) that trigger the batch job, and thus for each of these

Formatted: Indent: Left: 0"

AutoSys servers, there should be a corresponding persistent cookie. Under the current AutoSys Gateway and Java WebSphere contract, the cookie value will have to be stored in a file named fcas2061.cookie for server fcas2061 and eccas2061.cookie for server eccas2061. Previously, these two files had to be deployed via a standard Java build and deployment process so that the AutoSys batch job can be triggered and then work. At the renewal time, in the previous process, the application team would have to get the new cookie values, store the values in these two .cookie files and then do a build and deployment; even though only these two files are changed, a new build and deployment has to occur to guarantee the batch job can be triggered by AutoSys and then work.

Starting from DynaProp version 2.7, the team can simplify this process and eliminate the new build and deployment by managing the AutoSys Persistent Cookies in the DynaProp. See the "How-To" section for the steps to set up the AutoSys Persistent Cookies in the DynaProp.

#### 4.3.1.7 Health Check Cookies

The WebSphere Application Monitoring Framework allows applications to define a Health Check Servlet that is queried on all servers by the Monitoring Framework's servlet. The Monitoring Framework's servlet then reports status back from all of the application servers. The Monitoring Framework uses WSL/WSL-X Persistent Cookies to access the application's servlet. The WSL/WSL-X Persistent Cookies are read from a HealthCheck.properties file deployed with the application. HealthCheck.properties contains a timeout setting and a relative path to the application's monitoring servlet. See the "How-To" section for further details.

#### 4.3.1.8 Other

The 'Other' credential group is different than all the others. This group would be used when it is desirable to store some passwords in Password Management for systems that Password Management does not support. A good example of a system of this kind would be Datastage. By using the 'Other' type, credentials of all kinds can be stored in Password Management and therefore gain all the benefits of using Password Management (ease of password update, integration with DynaProp, security, notification support, all passwords in one place). The only difference is this: **IMPORTANT:** You will still have to update the credential on the source system manually. So, using Datastage as an example, the credential would have to be updated in Datastage first, and then updated in Password Management, to keep the two in sync, as Password Management does not know how to talk to Datastage. In summary, using the 'Other' group type provides a consistent, convenient place to store passwords that are not supported, enabling application teams to keep all types of application-necessary passwords stored in one convenient place. It also provides integration with DynaProp, as 'Other' group type credentials still must have a namespace and can be activated, etc. There is just no real source system.

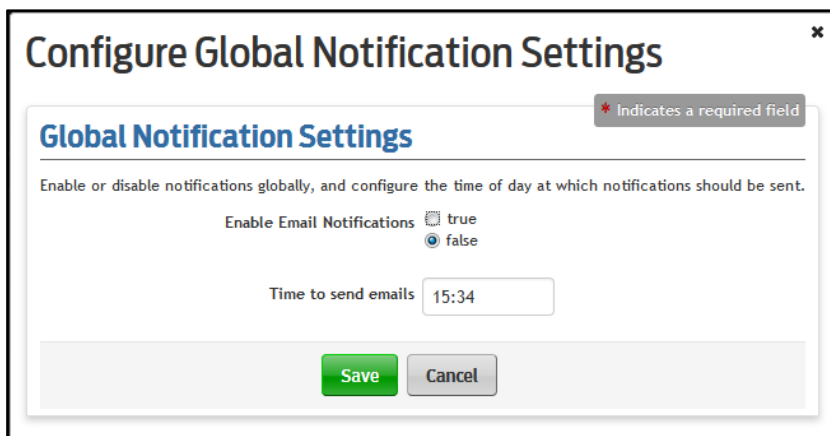
### 4.3.2 Password Management Notifications

Password Management, in release v2.2, provides e-mail notifications for expiring passwords out of the box. Notifications are designed to lighten the load for users who have to remember many different passwords and when they expire. The basic idea is that on a per credential group basis, notifications can be set up to alert a given set of users (over email) when a password is going to expire. Notifications will be sent to only the emails provided in the configuration, and they will be sent daily starting on a configurable number of days before the password expires. The following section describes how to enable and use notifications to alert you when a password is about to expire.

#### Step 1: Configure the Global Notification Settings

The first step is to configure the global notification settings. Notifications must be enabled on a *per-application* basis before any will be sent from a given credential group. This is to give flexibility in notification configuration- if notifications are no longer desired or if an unexpected error occurs, they can be disabled globally with just a few clicks.

To configure these settings, go to the *Manage Passwords* page, and click the *Manage Notification Settings* button, which is right above the *Credential Groups* table. The dialog pictured below will open on the screen:



**Configure Global Notification Settings** ✕

**Global Notification Settings** \* Indicates a required field

Enable or disable notifications globally, and configure the time of day at which notifications should be sent.

Enable Email Notifications ☐ true ☒ false

Time to send emails

**Save** **Cancel**

Figure 4 – Configure Global Email Notifications

The first field, Enable Email Notifications, enables or disables notifications for **ALL** credential groups.

The second field is the time of day at which the emails will be sent. Emails are sent once per day at a configurable time. It is recommended that the user avoid sending


emails at times that could be affected by Daylight Savings shifts (i.e. avoid 1am-2am time range for notifications).

Step 2: Configure Notifications for a Credential Group

Once notifications have been enabled globally, they must be configured for all credential groups or on a per-credential group basis. It is possible to have notifications configured for all credential groups and on a per-credential group basis. This is useful for sending notifications if the individual primarily responsible for updating credentials is unavailable.

Step 2b: Configure Notifications for All Credential Groups

To configure notifications for all credential groups, select "Notification Recipients" on the Manage Passwords screen.

 **Dynamic Property Manager (DynaProp)**

Home

Namespace

Password Management

DynaProp Configuration

Manage Passwords

Existing credential groups are shown below. Based on your permissions, you may be able to add new groups as well as edit or delete existing groups.

Add Group

Import Passwords File

Export Passwords File

Password Notification Settings

Notification Recipients

Health Check Settings

Action	Group Name	DynaProp Namespace	Host/NDI Name	Credential Type	Credentials
<a href="#">Delete Group</a>	AAAA Testing Refresh	TestingRefresh	wgc1w7bf7kyq1	Unsupported Credentials	Test Period in ID , expires in -391 days
<a href="#">Edit Group</a>   <a href="#">Delete Group</a>	AAAA Testing Refresh 2	TestingRefresh2		Other Credentials	aadf , expires in 72 days (Active)
<a href="#">Edit Group</a>   <a href="#">Delete Group</a>	agagaag	CredentialTestNamespace		Other Credentials	EncodedCharacters , expires in -453 days (Active)
<a href="#">Edit Group</a>   <a href="#">Delete Group</a>	autosys	autosys	eccaa2061.md2.ford.com	AutoSys Persistent Cookie	autosys , expires in 0 days (Active)
					dpainte7 , expires in -415 days test , expires in -448 days

The Configure Notification Recipients dialog is displayed. This dialog contains several fields:

The first field, *Enable Notifications*, specifies whether or not notifications will be sent for all credential groups.

The *Notification Threshold* indicates the number of days left before the password expires that will determine when the notifications will begin to be sent. For instance, if the number 10 is entered, notifications will begin to be sent once per day starting 10 days before the password expires.

The *New Recipient* field allows email addresses to be added for users to be notified. Only Ford email addresses are accepted.

The *Notification Recipients* widget is a list of all email addresses currently receiving notifications for this credential group. Any number of email addresses can be selected and then removed by clicking the 'Remove Selected Emails' button.

After configuring notification recipients, select the "Save" button.

### Step 2b: Configure Notifications for a Single Credential Group

To set up notifications for a credential group, go to the *Edit Credential Group* page by clicking the *Edit Group* link next to the group to be edited on the *Manage Passwords* page.

Expand the group options section and expand the *Configure Notifications* section. The image below shows the configuration web page.

The screenshot displays the 'Configure Notifications' interface for a credential group. At the top, there's a 'New Group Options' tab. Below it, the 'Group Name' is set to 'EccServerPersistentCookie'. The 'Credential Type' dropdown is open, showing options like 'SQL Server Database', 'SQL Server Always on Follower Cluster', 'Oracle Database', 'RACF Credentials', 'UNIX Credentials', 'AutoSys Persistent Cookie' (highlighted), and 'Other Credentials'. The 'DynaProp Namespace' is set to 'EccAutosysCookie'. The 'Host Name' is 'eccas2061.md2.ford.com'. The 'Group Description' is 'This is the Eccas2061 Autosys Persistent Cookie'. The 'Configure Notifications' section is expanded, showing options to 'Enable Notifications' (Yes/No), a 'Notification Threshold (in days)' of 10, a 'New Recipient' field with an 'Add Email' button, a list of 'Notification Recipients' (currently 'gpmnsupp@ford.com'), a 'Remove Selected Emails' button, and a 'Test Notifications' button.

Figure 5 – Configure Notifications for a Credential Group

There are several configuration fields:

The first field, *Enable Notifications*, specifies whether or not notifications will be sent for this credential group.

The *Notification Threshold* indicates the number of days left before the password expires that will determine when the notifications will begin to be sent. For instance, if the number 10 is entered, notifications will begin to be sent once per day starting 10 days before the password expires.

The *New Recipient* field allows email addresses to be added for users to be notified. Only Ford email addresses are accepted.

The *Notification Recipients* widget is a list of all email addresses currently receiving notifications for this credential group. Any number of email addresses can be selected and then removed by clicking the 'Remove Selected Emails' button.

Finally, the *Test Notification* button is for the application team to test if its App server is registered to allow the Email ID ([adynapro@ford.com](mailto:adynapro@ford.com)) to send emails from the App server under the Ford SMTP new policy starting from July 2015. The user is encouraged to type his or her Ford email address in the *New Recipient* field and then click this button to test if he or she can receive the email from this ID. If not, he or she can register this ID at the SMTP Mail Relay Request Web Site ([http://www.tcs.ford.com/email/e\\_form.asp](http://www.tcs.ford.com/email/e_form.asp)) so that the DynaProp notification function works properly.

### Step 3: Begin Receiving Password Management Notifications

Notifications will now begin to be sent for any enabled credential groups, on a daily basis when the days threshold is reached, and at the time specified in the global configuration.

The emails will come from the address [adynapro@ford.com](mailto:adynapro@ford.com), DynaProp Admin (No Reply). Only one email will be sent per credential group, containing all of the credentials for that group that have expiration days left that pass the threshold. Do not reply to these emails as they are not monitored. If support for notifications is needed, post on the JCOE forum or file an Etracker against the DynaProp Admin Console. Notifications can also be globally disabled in the global configuration dialog in the event of an issue or problem.

### 4.3.3 Password Management How-Tos

The section below describes all the basic functions of password management in detail.

**Note that** to use many or all of these features, you must have the appropriate permissions (either `DynaPropUnrestricted` or `DynaPropCredentialsMgmt`). To view a decrypted password, you must have one of the previous roles plus the `DynaPropPasswordRecovery` role.

#### 4.3.3.1 How-To Add a Credential Group:

- **Context:** This action item is available from the *Manage Passwords* page, above the table.
- Click the *Add Group* Button above the table on the Manage Passwords page.
- The *Add New Credential Group* page will load. Fill in the form with the appropriate data:
  - **Group Name:** The name of the credential group. This is user-specified.
  - **JNDI Name:** The JNDI name of the database to update. This must be a valid name or the group will fail to be created.
  - **Description:** An optional description for this credential group.

- **Credential Type:** The database that you are working with. Currently only Oracle and SQLServer are supported.
- **DynaProp Namespace:** The DynaProp Namespace field is the name of the DynaProp namespace that holds your credential data (the application will point to this namespace to retrieve credentials at runtime). **This namespace must have a property named 'id' and a property named 'password'.**
  - **Important Note:** When you elect to manage a namespace's credential data (id and password properties) in Password Management, then the system will lock these properties for editing from the Edit Namespace Properties screen with a lock icon and notification. Therefore, these properties will only be available for access from Password Management.
  - **Secondary Note:** If you enter a namespace that does not exist, password management can create it for you and add the 'id' and 'password' properties automatically. Just enter a namespace name to access this functionality.
- When finished filling out this form, click the *Add Group* button to create the group.
- The page will redirect to the Edit Credential Group page, which allows credentials to be added to the credential group.

#### 4.3.3.2 How-To Edit a Credential Group:

- **Context:** This action item is available from the *Manage Passwords* page.
- Click the *Edit Group* link next to the group to be edited.
- The *Edit Credential Group* page will load. Click the group name button to expand the metadata form section.
- Fill in the form with any changed data that is needed.
  - **Note:** The *Group Name* field is not editable.
- Click the *Save* button to save changes to the Credential Group.
- Credentials can also be added from this page. See the below How-To for more information.

#### 4.3.3.3 How-To Delete a Credential Group:

Deleting a credential group will permanently remove it and all related credentials. Only use this option if you are sure the data is no longer needed.

- **Context:** This action item is available from the *Manage Passwords* page.
- Click the *Delete Group* link next to the group to be deleted.
- The *Delete Credential Group* confirmation dialog will load, prompting if you are sure you wish to take this action.
- Click the *Yes* button to permanently delete the Credential Group.

#### 4.3.3.4 How-To Add a Credential to a Credential Group:

- **Context:** This action item is available from the *Edit Credential Group* page.



- Above the Credentials Table, click the *Add Credential* button to open the dialog to add a credential. Fill in the form with the appropriate data:
  - **ID:** The user id for the database. Must be a valid id or the credential will fail to add.
  - **Password:** The password for the id above. Must be a valid password or the credential will fail to add.
  - **Confirm Password:** Re-enter the password. Confirms the password to ensure the right value is entered.
  - **Expiry Days:** The time-out period of this password (ex. 90 days, 365 days). This value must be less than 365.
- **Note:** A credential must be able to connect to the database to be added. If it cannot, an error will be displayed and the credential will not be added.

#### 4.3.3.5 How-To Load a Credential From a Namespace:

**Alternatively** to adding a credential, if your namespace already has properties 'id' and 'password' filled with valid values, then you can just load the credential data from the namespace directly.

- **Context:** This action item is available from the *Edit Credential Group* page, above the Credentials table.
- If a credential is available to be loaded, the *Load Credential* button will appear next to the *Add Credential* button.
  - **Note:** If a credential in the group is already active, the *Load Credential* button will be grayed out.
- Click the *Load Credential* button to load a credential from the namespace.
- The credential will appear in the table after it loads successfully. The credential can be edited to correct the expiry days value if needed (it defaults to 90 or 365). If there was an error, that will be displayed.
  - **Note:** The credential must be a valid, working credential to be loaded. Otherwise an error message will be displayed.

#### 4.3.3.6 How-To Edit a Credential:

Editing a credential allows the credential metadata to be changed after a credential has already been added.

- **Context:** This action item is available from the *Edit Credential Group* page, in the credential table.
- To edit a credential, click the *More* link next to the credential to edit.
- Click *Edit Credential*.
- Make any changes needed to the **Expiry Days**.
  - **Note:** Passwords are treated differently because they will be updated in the database as well, so they are not editable from this dialog.
  - **Note:** ID's are not editable either. A new credential will need to be added to change an ID.

- **Note:** Also, credentials cannot be made active from the edit dialog. There is a separate *Activate* link to do that. See below for more information on both updating passwords and activating credentials.

#### 4.3.3.7 How-To Update a Credential's Password:

Updating the password for a credential will update the password in the database and store the new password in Password Management. It will **NOT** load the new password into the DynaProp namespace for the application to use, **unless** the credential that you are updating is already 'Active.'

- **Context:** This action item is available from the *Edit Credential Group* page, in the credential table.
  - To update a password, click the *More* link next to the credential to update.
  - Click *Update Database Password*.
  - The update password dialog will appear. Type in a new password and retype it in the confirmation field.
  - Click the *Save* button to update the password.
- Note:** Some databases have very stringent requirements for passwords. If the password entered fails to meet those requirements, an error message will be displayed with the details.

#### 4.3.3.8 How-To Auto-generate a Credential's Password and Update It:

Updating a password can be done in two ways- either the above way of updating the password with a user-given value, or by auto-generating a secure value designed for the database type. This auto-generated value will then go through the same process as the user-defined updated password- it will be changed in the database and then stored in password management.

- **Context:** This action item is available from the *Edit Credential Group* page, in the credential table.
- To auto-generate a password, click the *More* link next to the credential to update.
- Click *Auto-generate Database Password*.
- A confirmation dialog will appear. Click the *Yes* button to auto-generate a new password and update the database with that value.
- [Best practice for higher security is to block the ability for users to enter their own passwords. Dynaprop will auto-generate passwords and synchronize them with databases, and prevents users from knowing the database passwords. Add DynaPropEnterPasswordsForbidden:execute role to all users to implement this.](#)

#### 4.3.3.9 How-To Activate a Credential:

Activating a credential will place that credential's data into the DynaProp namespace that is specified for the Credential Group the credential belongs to. Said another way, activating a credential will place that credential's 'id' and 'password' into the 'id' and 'password' properties on the DynaProp namespace that the Credential Group has

selected. This allows the application to access the new credential data at runtime, but only after a **SERVER RESTART** to clear the cache (if using Toplink).

- **Context:** This action item is available from the *Edit Credential Group* page, in the credential table.
- To activate a credential, click the *Activate* link next to the credential to update.
- A confirmation dialog will appear, asking the user to confirm that this credential should be activated.
- When DynaProp is finished loading, a notification message will appear informing the user of the results of the activation.

#### 4.3.3.10 How-To Test a Credential for Connectivity:

Testing a credential for connectivity will just run a test to the database using the credential that is specified. This can be useful to determine if the credential is still up to date or still works.

- **Context:** This action item is available from the *Edit Credential Group* page, in the credential table.
- To test a credential, click the *More* link next to the credential to update.
- Click the *Test Credential* link.
- When DynaProp is finished loading, a notification message will appear informing the user of the results of the test.

#### 4.3.3.11 How-To View a Credential's Decrypted Password:

Viewing the decrypted password of a credential is a highly secured action, and so DynaProp provides a separate security role for this feature. To use this feature, a user must have the *DynaPropPasswordRecovery* role.

- **Context:** This action item is available from the *Edit Credential Group* page, in the credential table.
- To view the decrypted password for a credential, click the *More* link next to the credential to update.
- Click the *View Decrypted Password* link.
- A dialog will appear displaying the password.

#### 4.3.3.12 How-To Delete a Credential:

Deleting a credential is a permanent action and cannot be undone. All of the credential's metadata will be lost, including the id and password; however, any data in the DynaProp namespace will not be deleted.

- **Context:** This action item is available from the *Edit Credential Group* page, in the credential table.
- To delete a credential, click the *More* link next to the credential to delete.
- Click the *Delete Credential* Link.

- A confirmation dialog will appear, asking the user to confirm that the credential should be deleted.
- Click the Yes button on the confirmation dialog to delete the credential.

#### 4.3.3.13 How-To Export a Password File:

Exporting a password file allows the user to make a backup of all of the credential groups and credentials that are currently stored in password management. This data can then be re-imported at a later time or on another system (although data validity on another system may be an issue).

- **Context:** This action item is available from two places: Either the Password Management menu-bar drop down item or a button above the table on the Manage Passwords page.
- To export a password namespace, click either the menu link or the *Export Password File* button.
- A browser-default file download dialog will appear, prompting the user to save or open the xml file.
- The user should take the desired action with the xml file.

#### 4.3.3.14 How-To Import a Password File:

Importing a password file allows the user to overwrite all data currently stored in the password management module (all credential groups and credentials) and replace it with a new set of credentials and credential groups from an xml file that was previously exported.

- Context: This action item, like Export, is available from two places: Either the Password Management menu-bar drop down item or a button above the table on the Manage Passwords page.
- To import a password namespace, click either the menu link or the Import Password File button.
- The Import Password File page will load, allowing the user to select and upload a file.
- Select a file by clicking the Choose Namespace to Import button, and then choosing a file.
- After a file has been selected, upload it by clicking the 'Upload' button.
- A dialog will appear asking for confirmation of a password file import, because all previous password data will be removed.
- Click the 'Yes' button on the dialog to import the new password management file.

#### 4.3.3.15 How-To Manage AutoSys Persistent Cookies:

The following detailed steps illustrate how to use DynaProp to administer the AutoSys Persistent Cookie renewal.

### First Time Setup

If your team uses AutoSys to kick off the batch job, but your team doesn't use the DynaProp, you are encouraged to, firstly, set up the DynaProp by following the DynaProp *Quick Start Guide* <http://www.tc2.ford.com/ts/JCOE/Shared%20Documents/DynaProp/DynaProp%20Quick%20Start.pdf> .

If your team is currently using the AutoSys to kick off the batch job, and is also using the DynaProp to administer the configurations and credentials, you are encouraged to migrate your AutoSys Cookie's management to the DynaProp (start at Step 1 in the procedures below); if this is your team's first time setting up the AutoSys batch job, you are encouraged to set up the AutoSys Cookies directly in DynaProp (start at Step 2 in the procedures below); If your team doesn't use the AutoSys to kick off the batch job, ignore the procedures below.

**Note:** it is recommended that you start from Dev following the procedures below, once you are totally familiar with the steps, then move to QA and then Prod.

#### **Step 1: Remove the current eccas2061.cookie and fcas2061.cookie files**

Defunct the current eccas2061.cookie and fcas2061.cookie files from your AccuRev source control then make a build and deployment so that these two .cookie files are removed from your server's file system. This removes existing cookie files from deployment packages going forward, so future deployments do not include DynaProp-managed cookie files and do not confuse the AutoSys scheduler.

This step is only needed during the first time setup; future cookie renewals do not need a build and deployment.

#### **Step 2: Create DynaProp Group for the AutoSys Persistent Cookies**

The following 1) through 3) are procedures to create the DynaProp Group for the cookie for server fcas2061 and 4) is the procedure to create the DynaProp Group for the cookie for server eccas2061.

##### **1) Add Credential Group**

In the DynaProp, click on *Password Management* menu → *Manage Password* → *Add Group* (see Figure 1 for reference) → now a page shows up (see Figure 3 for reference).

There are several configuration fields:

**Group Name:** This is a name that can be defined freely by the user. You are suggested to give a name that is meaningful to all readers. In current Ford infrastructure, there are two AutoSys servers Fcas2061 and Eccas2061 that send the AutoSys job to the application team's App servers. Use a name like *FcasServerPersistentCookie* since we are setting up the cookie for fcas2061.

**DynaProp Namespace:** This should follow the standard Property Group naming convention. FCasServerCookie is a good name.

**Credential Type:** Since this is for the AutoSys Persistent Cookie, you will have to select "AutoSys Persistent Cookie."

**Host Name:** There are two values: eccas2061.md2.ford.com and fcas2061.md2.ford.com. Select fcas2061.md2.ford.com because we are doing the set up for fcas2061.

**Group Description:** This is an optional field, but you are encouraged to put something here so that future readers can understand what this group is.

- 2) Add the notification configuration (click on the "+" sign next to *Configure Notifications*)

**Note:** remember to turn on the global setting to receive the notifications; otherwise the notification will not function (see Figure 2 for reference).

Figure 3 and its explanations outline each field and its usage.

Once you set up the notification click on the *Add Group* button. You will see a confirmation window *Create Namespace?*, click on the *Yes* button, and then you will receive a successful message. Till now you have added the Credential Group to the DynaProp.

- 3) Add the Credentials  
Following the above step, under *Password Management* menu → *Create Credentials* → now a screen shows in the bottom for the current Credential

Group→ Click on Add Credential →an overlay shows (see [Figure 6](#) for reference).

Figure 6 – Add the AutoSys Persistent Cookie Value

There are several configuration fields:

**Cookie Name:** For the AutoSys Cookie management, this name can be defined freely.

**Cookie Value:** This is the real AutoSys Persistent Cookie value that you renewed from the WSL website, whose value is in an encoded format.

**Encryption Type:** Default is FJFAsymmetric

**Expiry Period:** The expiry period for an AutoSys Persistent Cookie is currently 365 at Ford

**Next Cookie Expiration date:** Enter the correct cookie expiration date; the system will start to send expiration notifications based on this date. For example, if you renewed the AutoSys Persistent Cookies on Dec 12, 2015 on the WSL website, then you set *Next Cookie Expiration Date* as Dec 12, 2016. If you had set *Notification Threshold (in days)* as 10 days for expiration

notification, then you will start to receive the expiration notifications from Dec 3, 2016.

**Note:** once you add the credential, you need to Activate the ID by clicking on *Activate* link (under *Action* heading) to make sure the credential is ready to take effect.

- 4) Create the DynaProp Group for the cookie for server ecccas2061.  
Repeat procedures 1) through 3) for server ecccas2061, but give a different name in fields such as Group Name and DynaProp Namespaces, etc.

**Step 3:** Restart all of the Application Servers so that the new Persistent Cookie values take effect when the next AutoSys job runs.

In the WebSphere stdout.log you will notice that some messages indicating that the cookie values are saved to your server from the DynaProp.

### **Renew the AutoSys Persistent Cookies following the First Time Set up**

If you have already set up the AutoSys Persistent Cookies in the DynaProp following the steps above, and you want to renew the cookies once they are approaching the expiration date, you can follow the procedures below.



**Step 1:** From *Password Management* menu → select the *Group Name* that you created earlier for the AutoSys Persistent Cookies and click on *Edit Group* → Click on *More*, now you will see an overlay as shown in the [Figure 7](#).

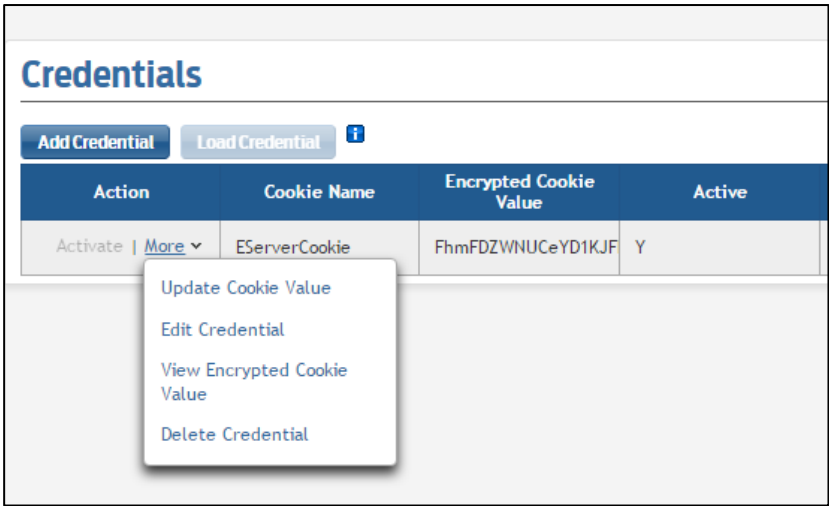


Figure 7 – Renew an AutoSys Persistent Cookie

To renew a Persistent Cookie, click on *Update Cookie Value* link, you will see a small overlay window. Add the new cookie value in the *Cookie Value* field and keep the default value ( FJFAsymmetric) for *Encryption Type* then click the *Save* button. You should receive a success message indicating the updated value is saved.

**Step 2:** Restart all of the Application Servers so that the new Persistent Cookie values take effect when the next AutoSys job runs.

4.3.3.16 How-To Manage Health Check Cookies

**WARNING:** *DynaProp should not be used to manage Health Check Cookies for applications that share a single DynaProp database between multiple applications because DynaProp does not have a way to differentiate between the different applications.*

**NOTE:** *It is recommended that these steps are tested in DEV and QA before they are attempted in PROD as they are somewhat more involved than the steps for a database password.*

**Step 1: Defunct the HealthCheck.properties File**

- 1. Defunct the HealthCheck.properties file in AccuRev.

2. Build the application.
3. Deploy the application.

These steps will ensure that the old HealthCheck.properties file is removed from the server's file system and will not conflict with the HealthCheck.properties file that is written by DynaProp.

### Step 2: Configure Global Health Check Settings in DynaProp

The path and timeout fields are common to all Health Check cookies and can be configured through **Password Management → Manage Passwords → Health Check Settings** to update these fields (see [Figure 8](#) below).

Figure 8 – Configure Common Health Check Settings

1. The Path is the path to your Health Check Servlet. For example, JAB's servlet is /JabUiWeb/HealthCheck.
2. The Timeout is the number of seconds before the Health Check will time out before it completes.

### Step 3: Configure Health Check Cookie Credential Groups in DynaProp

Each Health Check Cookie will need a Credential Group and Credential complete the following steps for each WSL/WSL-X Health Check Cookie that is used by the Health Check Framework.

1. Create a new Credential Group through **Password Management → Manage Passwords → Add Group** (see [Figure 9](#) below). The Group Name is

defined by the application team and should be a name that makes the cookie easy to identify.

The screenshot shows the 'Credential Group Details' form. The 'Group Name' field is 'WAS 8 DEV Env'. The 'DynaProp Namespace' field is 'WAS8DevEnv'. The 'Credential Type' dropdown menu is open, showing options like 'SQL Server Database', 'SQL Server AlwaysOn', 'Failover Cluster', 'Oracle Database', 'RACF Credentials', 'URRT Credentials', 'AutoSys Persistent Cookie', 'Health Check Cookie (WSL)', 'Health Check Cookie (WSL-X)', and 'Other Credentials'. The 'Group Description' field is 'WebSphere 8 Development Environment'. The 'Host Name' field is 'fcvaz781.fincig.ford.com'. There is a 'Configure Notifications' button and 'Add Group' and 'Cancel' buttons at the bottom.

Figure 9 – Add New Credential Group

2. The **DynaProp Namespace** is the Namespace that DynaProp will use to store the cookie credentials.
3. The Credential Type should either be **Health Check Cookie (WSL)** or **Health Check Cookie (WSL-X)**, depending on the type of cookie.
4. The **Group Description** can be used by the application team to further identify the cookie.
5. The **Host Name** should be the host name of the server associated with the Health Check cookie.
6. It is highly recommended that notifications are configured, so e-mail notifications are sent out when there are issues writing the HealthCheck.properties file on server restart and when credentials are about to expire (see Section 4.3.2).
7. Select **Add Group**.
8. Edit the Credential Group that was just created through **Password Management → Manage Passwords → Edit Group**.

9. Add a new Credential through **Add Credential** (see [Figure 10](#) below).

**Add New Credential** x

⚠ Credential group is not synced with namespace. Expand below form to view details.

**Credential Details** Indicates a required field

Enter information for a new credential below.

**\*Cookie Name**

**\*Cookie Value**

Encryption Type

**\*Expiry Period \***   
Must be between 1 and 365

**\*Next Cookie Expiration Date**  📅

**Save** **Cancel**

Figure 10 – Add New Credential

10. The **Cookie Name** is defined by the application team and should be a name that makes the cookie easy to identify.
11. The **Cookie Value** is the encoded WSL/WSL-X cookie provided by the WSL team.
12. The **Encryption Type** may be left as FJFAsymmetric, unless there is a specific reason to change it.
13. The **Expiry Period** is the length of time that the cookie is valid for. This value should be 365 as all WSL/WSL-X cookies at Ford have this expiry period.
14. The **Next Cookie Expiration Date** is the date that the current cookie will expire on.
15. Select **Save**.
16. Select **Activate** to activate the cookie that should be included in HealthCheck.properties for this Credential Group.
17. Repeat these steps for all of the application's Health Check Cookies

#### Step 4: Restart the Application Server

HealthCheck.properties is written to the disk by DynaProp on application startup, so the application server will have to be restarted for HealthCheck.properties to be written by DynaProp.

#### Step 5: Test Health Check

Test the Health Check to make sure it is working before applying these steps to higher environments.

### 4.3.4 Configuring DynaProp Credentials with Password Management

As of DynaProp Admin Console v2.2, due to the password capabilities that the Password Management provides, the functionality to set and update the actual DynaProp Admin Console credential has been moved so that is tightly integrated into Password Management.

Basically, this means that the *Update DynaProp Credentials* page no longer exists, and it has been replaced by an auto-loaded credential group called '**DynaProp Cfg**'. The **DynaProp Cfg** credential group is created on startup of DynaProp by the system, and can be edited by the user. If it is deleted, the system will recreate it for you. The link called '*Update DynaProp Credentials*' underneath the '*DynaProp Configuration*' menu heading links directly to this new credential group, where a credential can be added, activated, and the password can be updated. The **DynaProp Cfg** credential group can also be accessed like any other credential group, through the *Manage Passwords* page.

This is a big change from previous versions of DynaProp and how it was configured. The reasoning behind this change is as follows:

- Primarily, this allows the password to be changed in the database simultaneously as it is changed in DynaProp.
- Dual credentials can now be stored for the DynaProp Cfg group, and that allows rotation between valid credentials, supporting the dual ID approach.
- Viewing the decrypted password is now a possibility.
- Notifications are now supported for this credential group, like with any other.
- Finally, it provides a unified interface for dealing with passwords of any kind, even DynaProp configuration.

**Note:** Here are a few more things to know about the special DynaProp Cfg credential group:

- It doesn't have a real namespace associated with it like other credential groups- it is a special case.
- 'Activating' a credential in this group, instead of writing it to a DynaProp namespace, finalizes the password change and sends it to all nodes (if in a clustered environment).

- It will detect the database that is being used using the configuration file for DynaProp, and default to that credential group type.

For detailed instructions in configuring DynaProp's credential the first time you use it, see **section 3** of this document.

## 5. Quick Start Example

At this point the *Dynamic Property Manager Admin Console* should be configured, and you should have a working understanding of the application. This brief example is intended show the *Import Namespace from XML file* functionality. Application teams should save the contents of [Figure 11](#) into a file called `example.xml`.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE property-group SYSTEM "PropertyGroup.dtd">
<property-group name="example">
  <property-group name="Group1">
    <property-group name="Group1-SubgroupA">
      <property name="property1a">1a</property>
      <property name="property2a">2a</property>
    </property-group>
    <property-group name="Group1-SubgroupB">
      <property name="property1b">1b</property>
      <property name="property2b">2b</property>
    </property-group>
  </property-group>
  <property-group name="Group2">
  </property-group>
</property-group>
```

Figure 11 – `example.xml`

Import this file into DynaProp Admin Console, using the instructions in section 4.1.6.

On a successful import of the `example.xml` file, application teams should be able to view the following on the *Dynamic Property Manager Administration Console View* as displayed on [Figure 12](#).

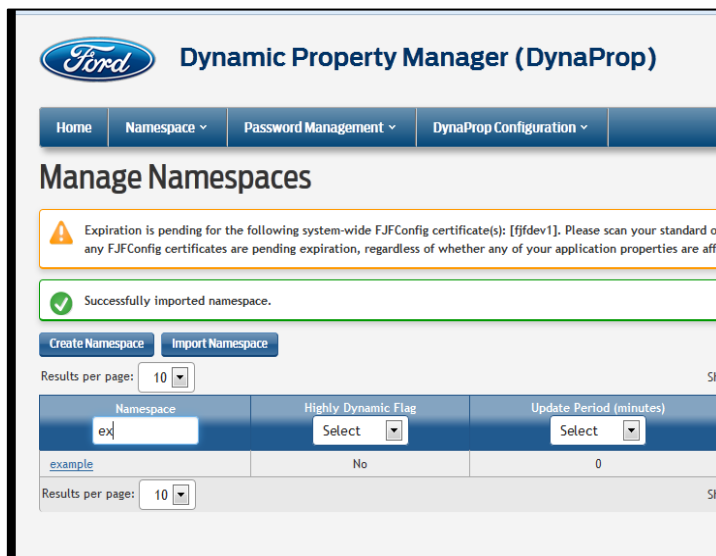


Figure 12 – Manage Namespaces View

Application teams should view the namespace hierarchy by click on the example namespace link. As a result, application teams should be able to view **Error! Reference source not found.**

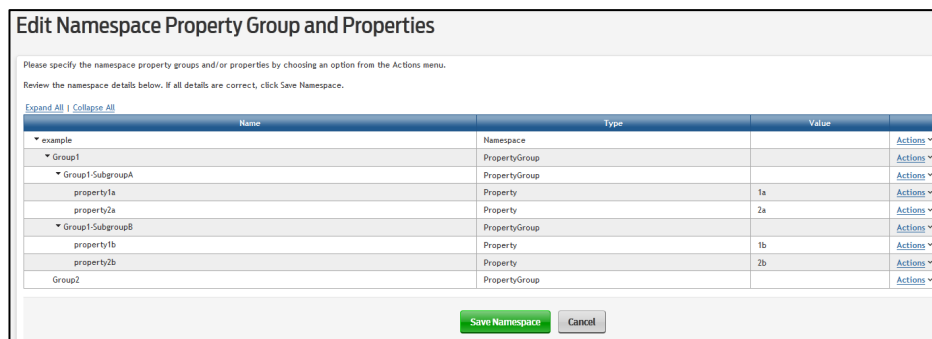


Figure 13 – Select a Property or PropertyGroup View



## 6. References and Resources

### 6.1 Using Property Manager

This document is packaged with the `dynapropadmin-vX.Y.Z.zip` framework as well as the `itcore-vA.B.C.zip` framework. In both cases, the document is located in the `"\docs\properties"` directory.

### 6.2 How-To Configure J2EE Security

<https://www.tc2.ford.com/ts/JCOE/Shared%20Documents/Security.aspx?PageView=Shared>

### 6.3 FJF APS Security Client Framework

<https://www.tc2.ford.com/ts/JCOE/Java%20Technologies%20Wiki/APS%20Plugin.aspx>

## 7. Appendix A – Property Value Encryption

### 7.1 Property Value Encryption Rules

In general there are two potential actions performed on a Property: Add a new Property or Edit an existing Property. Below are the guidelines that application teams should follow to be able to administer a Property.

### 7.2 Add New Property

To get to the *Add a Property* dialog, reference section 4.2 above in this document, and then specifically section 4.2.4.

When adding a property, there is a drop-down box that specifies the encryption 'action' to be taken upon the value of the 'Property Value' text box. There are three actions specified from this drop down box, as follows:

- *None*: Indicates that no encryption will be used for this property value.
- *Should be Encrypted with*: Indicates that the user would like the value in the property value box to be encrypted. Then, encryption type choices will appear allowing the user to specify a type of encryption.
- *Already Encrypted with*: Indicates that the user has input a property value that is already encrypted with an encryption type. Encryption type choices will also appear so that DynaProp can attempt to decrypt the value with a specified encryption type.

Encryption Types and their rules:

- A. "Symmetric"
  - a. Indicates Symmetrical encryption with one key.
- B. "FJFAsymmetric"
  - a. Indicates that the Property Value should be FJFAsymmetrically encrypted using the FJF encryption framework. As a result:
    1. *Certificate Name* dropdown should be the DynaProp Admin Console's current certificate name.

### 7.3 Edit an Existing Property

To get to the *Edit a Property* dialog, reference section 4.2 above in this document, and then specifically section 4.2.5.

Once in the *Modify a Property View*, the form's initial state is the following:

- *Encryption Action* dropdown will display either:
  - a. *None* if property is not currently encrypted, or,
  - b. *Already Encrypted With* and show the encryption types as well as the certification drop down, if applicable.

The encrypted property value can be changed, but only completely reset, including encryption settings! This means when a new value is typed into the Property Value text field, the encryption drop down must be changed to *Should be Encrypted with* and a type selected again.

**IMPORTANT NOTE:**

Take note that partial changes to an encryption value will result in an error due to the encryption engine not being able to decrypt the value again.

Otherwise, same encryption types and rules are applied as when adding a new Property.

## 8. Appendix B – Change Notifications

### 8.1 Change notifications overview

If desired, it is possible to get a notification of every change made by any user in the administrative console. The notification is sent using Event Handling framework, so it could be sent to any destination that Event supports: e-mail, pager, database, audit file, etc. The default is an HTML-formatted notification email.

### 8.2 Enabling change notifications

To enable change notifications, it is necessary to inject the event framework configuration file, eventframework-config.xml, into the EAR at build time, before deploying it to the server. If using custom destinations, or if customizing messages that are sent, the jar file containing the classes that implement these dependencies must also be injected into dynapropadmin.ear.

Upon startup, the framework will report whether it found the configuration file, and therefore enables notification. If the config file is not found, all notifications are disabled. Look for messages from EventFwInitializer class in the log.

### 8.3 Sample (default) configuration file

The below XML is a sample eventframework-config.xml file that sends each notification to a single user. Customizations to this file, to support other destinations for example, are allowed following the syntax described in user's guide to Event Handling framework. Dynaprop will fire a trigger named "DynapropChange"; direct it to events as destinations as appropriate.

```
<ford-event-handler xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="event.xsd">

  <!-- Triggers and event configuration -->
  <triggers>
    <trigger name="DynapropChange">
      <events>
        <event-ref>DynapropChangeEvent</event-ref>
      </events>
    </trigger>
  </triggers>

  <!-- events configuration -->
  <event-data>
    <event name="DynapropChangeEvent">
      <!-- throttle to no more than 100 per hour. -->
      <event-property name="limit" value="100" />
      <event-property name="timeThreshold" value="3600" />

      <destinations>
        <destination-ref name="Email" enabled="true" />
      </destinations>
      <implementation-class>com.ford.it.event.Event</implementation-class>
      <enabled value="true" />
    </event>
  </event-data>
</ford-event-handler>
```

```

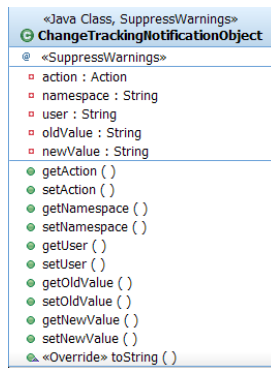
</event-data>

<!-- destination Definitions -->
<destinations>
  <destination name="Email">
    <destination-property name="recipient" type="string">oshvarts@ford.com</destination-property>
    <destination-property name="recipient" type="string">oshvarts@ford.com</destination-property>
    <destination-property name="sender" type="string">oshvarts@ford.com</destination-property>
    <destination-property name="subject" type="string">Dynaprop Change Notification</destination-
property>
    <implementation-class>com.ford.it.event.HtmlEmailDestination</implementation-class>
    <enabled value="true" />
  </destination>
</destinations>
</ford-event-handler>

```

## 8.4 Customizing defaults.

If using defaults, only changes to the configuration file are necessary. It's also possible to customize the framework to either send a differently formatted message, or send the message to a different destination. In either case, in order to do that, other than changing the config file the application must inject a jar file containing a class that would do the necessary formatting and processing of the message. This class (it would need to extend either com.ford.it.event.CustomDestination or one of the other existing Destination classes of Event FW) would need to consume a data object of type com.ford.it.dynaprop.outbound.email.ChangeTrackingNotificationObject and interrogate its fields (action, namespace, oldValue, newValue, user) to compose a message of their own desire.



**Page 8: [1] Formatted    Pratik, Patra (P.)    3/16/2023 7:23:00 PM**

Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Page 8: [2] Formatted    Pratik, Patra (P.)    3/16/2023 7:23:00 PM**

Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Page 8: [3] Formatted    Pratik, Patra (P.)    3/16/2023 7:23:00 PM**

Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Page 8: [4] Formatted    Pratik, Patra (P.)    3/16/2023 7:23:00 PM**

Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Page 8: [5] Formatted    Pratik, Patra (P.)    3/16/2023 7:23:00 PM**

Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Page 8: [6] Formatted    Pratik, Patra (P.)    3/16/2023 7:23:00 PM**

Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Page 8: [7] Formatted    Pratik, Patra (P.)    3/16/2023 7:23:00 PM**

Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Page 8: [8] Formatted    Pratik, Patra (P.)    3/16/2023 7:23:00 PM**

Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Page 8: [9] Formatted    Pratik, Patra (P.)    3/16/2023 7:23:00 PM**

Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Page 8: [10] Formatted    Pratik, Patra (P.)    3/16/2023 7:23:00 PM**

Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Page 8: [11] Formatted    Pratik, Patra (P.)    3/16/2023 7:23:00 PM**

Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Page 8: [12] Formatted    Pratik, Patra (P.)    3/16/2023 7:23:00 PM**

Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Page 9: [13] Formatted    Pratik, Patra (P.)    3/16/2023 7:23:00 PM**

Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Page 9: [14] Formatted    Pratik, Patra (P.)    3/16/2023 7:23:00 PM**

Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Page 9: [15] Formatted    Pratik, Patra (P.)    3/16/2023 7:23:00 PM**

Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Page 9: [16] Formatted    Pratik, Patra (P.)    3/16/2023 7:23:00 PM**

Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Page 9: [17] Formatted    Pratik, Patra (P.)    3/16/2023 7:23:00 PM**

Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Page 9: [18] Formatted    Pratik, Patra (P.)    3/16/2023 7:23:00 PM**

Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Page 9: [19] Formatted    Pratik, Patra (P.)    3/16/2023 7:23:00 PM**

Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Page 9: [20] Formatted    Pratik, Patra (P.)    3/16/2023 7:23:00 PM**

Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Page 9: [21] Formatted    Pratik, Patra (P.)    3/16/2023 7:23:00 PM**

Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Page 9: [22] Formatted    Pratik, Patra (P.)    3/16/2023 7:23:00 PM**

Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Page 9: [23] Formatted    Pratik, Patra (P.)    3/16/2023 7:23:00 PM**

Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Page 9: [24] Formatted    Pratik, Patra (P.)    3/16/2023 7:23:00 PM**

Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Page 9: [25] Formatted    Pratik, Patra (P.)    3/16/2023 7:23:00 PM**

Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Page 9: [26] Formatted    Pratik, Patra (P.)    3/16/2023 7:23:00 PM**

Default Paragraph Font, Font: Verdana, Check spelling and grammar

**Page 9: [27] Formatted    Pratik, Patra (P.)    3/16/2023 7:23:00 PM**

Default Paragraph Font, Font: Verdana, Check spelling and grammar