

Agile Product Lifecycle Management for Process

Configuration Guide

Release 5.2.1

Part No. E13905-01

September 2008

Copyrights and Trademarks

Copyright © 1995, 2008, Oracle Corporation and/or its affiliates. All rights reserved.

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are “commercial computer software” or “commercial technical data” pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Oracle and Agile are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Java and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. or other countries.

Microsoft, Windows, Excel, and Microsoft Internet Explorer are trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries.

September 2008

DOCUMENT CONTROL

Change Record

Date	Author	Version	Change Reference
Sept-07	Agile/Oracle	1.0	Initial release of manual
Feb-08	Oracle	2.0	Second release, Part No. E11041-01
Sept-08	Oracle	3.0	Third release, Part No. E13905-01

CONTENTS

About This Manual

Agile Product Lifecycle Management for Process Documentation	vii
Audience	vii
Variability of Installations	vii
Where to Find Information	viii
Readme	viii
Agile Training	viii
Document Conventions	ix
Overview	1-1
System Configuration Files	1-3
Environment Variable Settings	1-4
Overview	1-4
Environment Variable Configuration File Format	1-4
Environment Configuration Settings	1-6
Overview	1-6
Configuration File Format	1-6
Core	1-6
SiteInfo	1-6
AppURLs	1-6
AppSettings	1-6
FormsAuthenticationHelper	1-7
Portal	1-7
Prodika / Services	1-8
Prodika / ConnectionPools	1-10
Prodika / Managers / EnvironmentManager	1-10
ProdikaSettings	1-11
ProdikaSettings / Attachments	1-11
ProdikaSettings / ProdikaModules	1-11
WebCommon	1-11
SideNav	1-11
Application Settings	1-12
RemotingContainer Section	1-14
DexConfig	1-15
Background	1-15
Solution	1-15
Deployment Considerations	1-16
Customer Configuration Settings	1-17
Overview	1-17
Configuration File Format	1-17
Core	1-17
FormsAuthenticationHelper	1-17
SingleSignOn	1-17
Portal	1-17
Prodika / Services	1-18
Prodika / Managers	1-18
ProdikaSettings / SingleSignOn	1-18
Deployment Settings	1-19
DWB	1-19
Extended Attributes	1-19
DynamicScriptMethods	19

GSM	1-20
GSMConfig	1-20
GSMSettings	1-21
frmPrint2	1-21
CustomUOMLists	1-22
Xeno.Data.GSMWeb.MiscSettings	1-22
ProdikaSettings / ConceptAwareSpecs	1-22
ProdikaSettings / GlobalSuccession	1-22
GSMLibConfig / RequiredPropertiesForValidSpecs / Validator	1-22
GSMLibConfig / TestingProtocolAPConfig	1-23
AssociatedSpecsConfiguration	1-23
NPD	1-23
AppSettings	1-23
Analytics	1-24
ActivityTypeUIMapper	1-24
PQS	1-25
AppSettings	1-25
SCRM	1-25
SCRMConfig	1-25
Supplier Portal	1-26
SupplierPortalConfig	1-26
WFA	1-26
AppSettings	1-26
EQTWorkflowMappingConfig	1-26
PMA (UGM)	1-26
SuiteApplications	1-26
Workflow	1-27
LinearWorkflowResolverConfig / TransientWorkflowPKIDs	1-27
FeatureConfig	1-27
ExtendedAttributesSettings	1-28
Validation Rules Configuration	1-29
Overview	1-29
Configuration File Format	1-29
ValidatorFactories	1-29
ValidationRules	1-29
Rules	1-29
Conditions	1-30
The <if> Element	1-30
Validators	1-30
ReflectiveRegexValidator	1-30
ReflectiveRequiredValidator	1-31
ReflectiveRangeValidator	1-31
ReflectiveDateValidator	1-31
SpecWorkflowSystemActionValidator	1-32
CollectionObjectRequiredValidator	1-33
ReflectiveItemValueComparisonValidator	1-33
MultipleClauseReflectiveRequiredValidatorWrapper	1-33
TIPInitiatedValidator	1-34
Content Synchronization and Syndication Settings	1-34
Overview	1-34
Internal Syndication	1-34
External Syndication	1-34
Configuration Sections	1-35
ConfigSections	1-35
<NamedParams> Section	1-35
<CSSConfigurations> Section	1-35
PublicationPathConfigs	1-39
Exchange	1-40
TargetMarket 40	
TradingPartner	1-40

Validator	1-40
NamedWorkflowStatuses	1-40
Syndicating Custom Sections	1-41
Overview	1-41
Configuration Sections	1-41
Config Sections	1-41
<ExportExtensibilityHandlers> Section	1-41
<SpecType#> Section	1-41
Example configuration	1-41
Search Properties Configuration Settings	1-43
Overview	1-43
Configuration Sections	1-43

ABOUT THIS MANUAL

Agile Product Lifecycle Management for Process Documentation

The Agile Product Lifecycle Management (PLM) for Process documentation set includes user guides, an administrator's guide, and release notes, all in Adobe® Acrobat™ PDF format. The Oracle Documentation Web site contains the latest versions of the Agile PLM for Process PDF files. You can view or download these manuals from the Web site, or you can ask your administrator if there is an Agile PLM for Process Documentation folder available on your network from which you can access the documentation (PDF) files. Visit the Oracle documentation Web site at:

<http://www.oracle.com/technology/documentation/index.html>

Note The minimum software requirement for reading the PDF files is Adobe Reader™ version 6.0. You can download this free program from www.adobe.com.

If you need additional assistance or information, please contact support@agile.com or phone (408) 284-3900 for assistance.

Before calling Agile Support about a problem with an Agile PLM for Process manual, please have ready the full part number, which is located on the cover.

Audience

This user guide is intended for end users who are responsible for creating and managing information in Agile Product Lifecycle Management for Process. Information about administering the system resides in the *Agile Product Lifecycle Management for Process Administrator User Guide*.

Variability of Installations

Descriptions and illustrations of the Agile PLM for Process user interface included in this manual may not match your installation. The user interface of Agile PLM for Process applications and the features included can vary greatly depending on such variables as:

- ❑ Which applications your organization has purchased and installed
- ❑ Configuration settings that may turn features off or on
- ❑ Customization specific to your organization
- ❑ Security settings as they apply to the system and your user account

Where to Find Information

Consult the table below to find specific information from the relevant Agile PLM for Process information source.

Table 1: Agile PLM for Process documentation topics, by source

Information type	Config uration Guide	Admin. User Guide	Readme file	Agile training	Agile Help Desk	Agile sales rep
Administering Agile PLM for Process		●		●		
Cache management		●				
Compliance	●					
Configuring Agile PLM for Process	●					
Core data management		●				
Creating specifications	●					
Custom data management		●				
Custom sections	●	●		●		
Extended attributes	●	●		●		
Feature requests					●	●
Group management		●				
Installing Agile PLM for Process				●		●
Known issues			●			
Last-minute changes			●			
Managing specifications				●		
New in this release			●	●		●
Printing				●		
Resolved issues			●			
System-based roles		●				
Technical support					●	
User management		●				
Using the GSM application	●					
Workflow management		●				

Readme


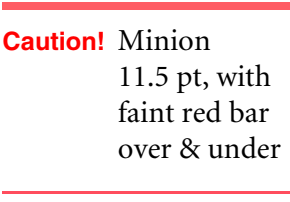

Any last-minute information about Agile PLM for Process can be found in the Readme file on the Oracle documentation Web site (<http://www.oracle.com/technology/documentation/index.html>).

Agile Training

Agile offers end user, administrator, developer, and implementation training courses. For more information, contact your Agile project manager or sales representative.

Document Conventions

The following formatting elements appear in Agile PLM for Process documentation.

Element	Meaning
Helvetica Condensed, 9 pt. bold type	A user interface (UI) element that a procedure is instructing you to click, select, or type into. For example, buttons or text entry fields.
9 pt. monospace font	Code samples
10 pt. monospace font	File names or directory names
<i>Blue italic font</i>	The linked portion of a cross-reference. Click it to go to the referenced heading, table, or figure.
Minion Typeface, Title Case	A named UI element that a procedure is describing but not instructing you to click, select, or type into.
 Note Minion 11.5 pt, with faint blue bar over & under	Alerts you to supplemental information.
 Caution! Minion 11.5 pt, with faint red bar over & under	Alerts you to possible data loss, breaches of security, or other more serious problems.
 Important Minion 11.5 pt, with thick red bar over & under	Alerts you to supplementary information that is essential to the completion of a task.

Configuring Agile Product Lifecycle Management for Process 5.2.1

This guide discusses basic configuration information regarding Agile Product Lifecycle Management for Process. Topics in this manual include:

- ❑ *Environment Variable Settings*
 - ❑ *Environment Configuration Settings*
 - ❑ *DexConfig*
 - ❑ *Customer Configuration Settings*
 - ❑ *Validation Rules Configuration*
 - ❑ *Content Synchronization and Syndication Settings*
 - ❑ *Syndicating Custom Sections*
 - ❑ *Search Properties Configuration Settings*
-

Overview

Using configuration files, you can limit or extend the behavior of your Agile installation. Further, Agile installations need a certain level of configuration to describe the location of the database, the reporting services, and some topology information to allow smooth interoperability with other subsystems within the Agile suite.

Agile configuration files are text-based XML files. Configuration settings can stand alone, or they can be organized within a nested set of XML elements.

The Agile configuration files are located at:

[X]: \\%Prodika_Home%\Config

As of version 5.2.1, there are three subfolders and a few files under this location, as shown below.

```
[X]: \%Prodika_Home%\Config\Core
[X]: \%Prodika_Home%\Config\Custom
[X]: \%Prodika_Home%\Config\Extensions
[X]: \%Prodika_Home%\Config\environmentvariables.config
[X]: \%Prodika_Home%\Config\DeployedConfig.config
```

Several configuration files can be modified to affect application behavior or to model the application landscape, as listed in table 1-1 below.

Table 1-1: Agile configuration files, described

Configuration File Name	File Location	Description
environmentvariables.config	%PRODIKA_HOME%\Config	Defines variable settings for physical layout of servers and applications
EnvironmentSettings.config	%PRODIKA_HOME%\Config\Custom	Defines more advanced physical environment settings such as email addresses and service settings
CustomerSettings.config	%PRODIKA_HOME%\Config\Custom	Defines variables specific to the implementation requirements and provides the ability to override specific application feature configuration settings

As of version 5.2.1, the `RemotingContainer.exe.config` typically need not be modified.

The files under `[X]: \%Prodika_Home%\Config\Core` typically should not be modified as part of the deployment.

The files under `[X]: \%Prodika_Home%\Config\Extensions` can be modified as to extend the behavior of the product suite.

System Configuration Files

The following configuration files are system specific and typically should not need to be modified. They are necessary to the proper functioning of the system.

Table 1-2: System configuration files

Configuration File Name	Description
AssociatedSpec.config	Configures the “Associated Specifications” feature
BaseFeatureConfig.config	Core feature configuration
CoreAppSettings.config	Contains primary application configuration settings
DataLibConfig.xml	Data access layer configuration
DeployedConfig.config	Defines the Feature Configuration pipeline
EACConfig.config	Creates the mapping of extended attributes to CSS syndication overrides
EQTModel.xsd	Defines the internal schema for Entity Query Toolkit (EQT)
EqtUIModelDefinitions.xml	Defines the Entity Query Toolkit (EQT) display models
exportExtensions.xml	Contains settings needed to add custom section data to the specification syndication
ExtensionToMimeTypeMappings.xml	Specifies MIME Type mappings
LinearWorkflowResolverConfig.xml	Configures the workflow resolution engine
LinkedObjectSaveConfig.xml	Defines the permissible linked object relationships for the data access layer
MenuConfig.xml	Defines the navigation menu rendering subsystem
MethodAccessPermission.xml	Defines security around core platform features
NutrientCalculationConfig.xml	Specifies default nutrient calculations.
ObjectURLMapping.config	Maps business object types to the applications responsible for managing these object types
XenoDoc.xml	Core file management configuration
eQuestionnaire.xml	Core eQuestionnaire (EQ) settings
gsmLibConfig.xml	Core Global Specification Management (GSM) settings
securityPolicy.xml	Specifies handlers and settings for the permission management engine
stylesMap.xml	Used to override stylesheets for specific cultural preferences

These system files have been encrypted as part of the installation to prevent accidental changes.

Environment Variable Settings

Overview

The `environmentvariables` configuration file contains the aspects of the product suite configuration that are related to the topology of the product suite. Often customers deploy the product suite on multiple servers with each server hosting one or more applications from the product suite. You can specify the location of these applications and the port numbers of the various services in this configuration file.

A specified set of variables in this file are system specific, and the system cannot function as expected if they are missing. Additionally, you can add custom variables to this file for ease of re-use.

See table 1-3 on page 4 for the list of system-specific variables.

Environment Variable Configuration File Format

Table 1-3: Environment variable configuration settings, described

Configuration Setting	Description
Prodika.Server1.URL	This is a variable defined by the Installation Admin indicating the name of the server where the product suite is deployed. More Server settings can be added to the <code>environmentvariables</code> config (for example, <code>Prodika.Server2.URL</code>) as needed and referenced in the settings below
Prodika.SCHEME	HTTP Scheme for the Web applications (can be <code>http</code> or <code>https</code>)
Prodika.DWB.URL	Location of the DWB Web application
Prodika.GSM.URL	Location of the GSM Web application
Prodika.GSMInterApp.URL	Location of the GSM Web application for inter-application communication
Prodika.SCRM.URL	Location of the SCRM Web application
Prodika.NPD.URL	Location of the NPD Web application
Prodika.Portal.URL	Location of the Portal Web application
Prodika.PQS.URL	Location of the PQS Web application
Prodika.REG.URL	Location of the Reg Web application
Prodika.DRLAttachment.URL	Location of the DRL Web application that is deployed for file transfer
Prodika.DRLService.URL	Location of the DRL Web service that is deployed for file transfer
Prodika.DRL.URL	Location of the DRL Web application
Prodika.EQ.URL	Location of the EQ Web application
Prodika.CSSPortal.URL	Location of the CSS Portal Web application
Prodika.Reporting.URL	Location of the Reporting Web application
Prodika.WFA.URL	Location of the Workflow Admin Web application
Prodika.UGM.URL	Location of the User Group Management Web application

Table 1-3: Environment variable configuration settings, described (continued)

Configuration Setting	Description
Prodika.Print.URL	Location of the Print application
Prodika.SPA.URL	Location of the Supplier Portal Admin Web application
Prodika.SupplierPortal.URL	Location of the Supplier Portal Web application (typically used for external vendors)
Prodika.WebDAV.URL	Location of the WebDAV virtual folder
Prodika.AuthenticationBridge.Port	Port number for the authentication bridge
Prodika.Reporting.Port	Port number for the Report Generation service
Prodika.AuthorizationBridge.Port	Port number for the Authorization service
Prodika.TaxonomyDenormalization.Port	Port number for the Taxonomy Denormalization service
Prodika.OLSDenormalization.Port	Port number for the Object Level Security Denormalization service
Prodika.Syndication.Port	Port number for the Data Syndication service
Prodika.DataImport.Port	Port number for the Data Exchange service
Prodika.DRLService.SysUser	User Id for the system user with access to the DRLService Web application
Prodika.DRLService.SysPassword	Password for the system user for the DRLservice
Prodika.RemotingContainer.SysUser	User Id for the system user for the Remoting Container
Prodika.RemotingContainer.SysPassword	Password for the system user for the Remoting Container

In a load-balanced environment, domain and subdomain of the URL will be the externally facing domain and subdomain respectively.

As described above, you can add custom variables in the beginning of the `environmentvariables.config` file and can then reference them in the settings that follow it. A default variable called `Prodika.Server1.URL` is available in the configuration file. This can be expanded upon by introducing `Prodika.Server2.URL` and so on depending on the number of servers in the deployment landscape. The custom variables can then be consumed by system-specific variables that follow (for example, `Prodika.DRL.URL`).

Further, the custom variables introduced can reference other Windows environment variables. This provides a way for some settings—such as passwords and usernames and possibly database connection strings—to be available to be viewed by only an administrator or a highly privileged user.

Environment Configuration Settings

Overview

The `EnvironmentSettings` configuration file contains those aspects of the product suite's configuration that are related to the environment. Though similar in spirit, the `EnvironmentSettings` configuration file contains more complex configuration settings than those in the `environmentvariables` configuration file, as described in detail below.

As opposed to the `CustomerSettings.config` file, the `EnvironmentSettings.config` will be the same for a specific version of the software on a set of servers in one specific environment (for example, Production). The `CustomerSettings.config`, however, would almost always be the same for a specific version of the software across all environments of that version (for example, production, QA, dev, and so on).

It is possible that there could be some circumstances that could necessitate different settings across servers that could impact these assumptions.

Configuration File Format

The elements in the file are outlined below, with details following in a table.

Core

Used By: All applications

SiteInfo

This tag specifies the scheme under which this application will operate (http or https)

AppURLs

This section has been relocated to the `environmentvariables.config` file. See [Environment Variable Settings](#) on page 4.

AppSettings

This key indicates the location of `HomeURL` (portal) and the `SiteMap` (do not modify) of the Agile installation.

FormsAuthenticationHelper

This configuration node contains two elements: SingleSignOn and Portal.

The SingleSignOn element can be enabled for systems using a Single Sign On (SSO) solution that conforms to the Agile SSO interface.

```
<SingleSignOn xmlMergeKey="SingleSignOn"
paramName="externalid"
columnName="username"
enabled="true" />
```

Enable this setting by setting enabled to “true.”

For systems using an LDAP solution, this SSO setting is not appropriate. The “AuthenticationStrategy,” which is described later on, is used for enabling LDAP-based authentication.

Table 1-4: FormsAutheneticationHelper: key names, expected values, and descriptions

Key Name	Expected Value	Description
paramName	Name of header key	The header key set by the SSO solution once it authenticates the user in the incoming request
columnName	Name of the column in the user table	The name of the column in the Users table that contains the value to be compared to the value of the header key in the Request
enabled	true / false	Indicates if SSO is enabled.

Portal

If SSO is not used but Agile or if LDAP authentication is used, this element indicates the location of the Agile portal for the authentication to be centralized. The attribute loginUrl is used to indicate the location of the Agile portal application. If the portal is enabled, applications redirect to the loginUrl for authentication. Enabling the portal is covered in the CustomerSettings.config file.

Prodika / Services

This element configures the services used by Agile. Many Agile services rely on their environment to enable certain behavior.

Table 1-5: Prodika services: key names and descriptions, by service name

Service name	Key Name	Description
AuthenticationService	UseTrustBridge	Specifies if the Authentication Service needs to use the trust bridge, which is hosted in the Remoting Container. It is generally set to true unless only one application is used. Setting this to “false” may deteriorate usability, as many logical use cases are implemented with a combination of multiple applications.
	AuthenticationStrategy	Agile allows a set of Authentication Strategies. They are <code>id= "LDAP"</code> , <code>id= "Prodika"</code> or <code>id= "ProdikaAndLdap"</code> . Setting the id to “Prodika” uses the Prodika User Management System to authenticate the users. Setting id to “LDAP” forces the use of an LDAP system specified by the <code>ldapServer</code> attribute. If the “ProdikaAndLdap” option is picked, another option, “allMustPass,” may also be set to “true” or “false.” This defines whether the user’s credentials must be accepted by all strategies (Prodika and LDAP) or by just one.
AuthenticationBridgeService	REMOTE_URL	Specifies the location of the AuthenticationBridgeService
EANnetEmailService	EMAIL_DOMAIN_FILTER	Specifies the domains to which emails can be sent. Used for security and privacy purposes so that a user cannot mistakenly send an email to an insecure domain name. Typically is left blank in a production environment but is specified in QA or Maintenance environments, to prevent emailing to outside parties.
	BCC_AUDIT_EMAIL_ADDRESS	Specifies an email address to which every email sent out by the system is BCC-ed for auditing purposes. May be left blank.
	SMTP_SERVER	Location of the SMTP server that the system will use for sending emails.

Table 1-5: Prodika services: key names and descriptions, by service name (continued)

EmailService	EMAIL_DOMAIN_FILTER	Specifies the domains to which emails can be sent. Used for security and privacy purposes so that a user cannot mistakenly send an email to an insecure domain name. Typically is left blank in a production environment but is specified in QA or maintenance environments, to prevent emailing to outside parties.
	BCC_AUDIT_EMAIL_ADDRESS	Specifies an email address to which every email sent out by the system is BCC-ed for auditing purposes. May be left blank.
	SMTP_SERVER	Location of the SMTP server that the system will use for sending emails.
SyndicationService	ErrorNotifyFromAddress	The FROM address used for error notification
	ErrorNotifyToAddress	The TO address used to send the error notification
AuthorizationTokenManagementService	AuthorizationBridgeUrl	Location of the <code>AuthorizationBridge</code> service. Could be any unique location that is accessible by the <code>RemotingContainer</code> .
DataExchangeService	DexConfiguration	Specifies the configuration of the <code>DataExchangeServer</code> . See DexConfig on page 15
ReportProcessingService	MAX_REPORT_GENERATOR_THREADS	Maximum number of threads for report generation
	SLEEP_INTERVAL_IN_SECONDS	Sleep interval for polling
	SQL_TIMEOUT_IN_SECONDS	SQL command timeout
	EXPIRATION_IN_DAYS	Expiration of report request
	GRACE_PERIOD_IN_DAYS	Grace period after report expiration

Prodika / ConnectionPools

This element specifies the Prodika and Reporting Connection pools to be used by the applications. The key-value pairs are specified in table 1-6, below.

Table 1-6: Prodika connection pools

Key Name	Required	Expected Value	Description
MAX	Yes	> 49	Depending on the scenario and the number of services active at any point, the connection pool size can vary. A number greater than 49 is considered safe, but this value will depend on utilization patterns.
CONNECT_STRING	Yes	(format is data provider specific)	This entry specifies the location of the database and its credentials. The appropriate connection string should be defined and provided by a database administrator.

The above settings are for the Agile database. The ReportDB element, which is a copy of the ProdikaDB element, specifies the connection pool properties for the reporting application. Specifying it separately allows reporting to be run on a copy of the Agile main database to avoid performance degradation of the main application.

Prodika / Managers / EnvironmentManager

The **<Managers>** element contains definitions and configurations for the Agile platform.

The following table summarizes the various settings in this node. The EnvironmentManager node is detailed in table 1-7 below.

Table 1-7: EnvironmentManager node: key names and descriptions

Key Name	Description
GSM_ATTACHMENT_PATH	Location of the attachments folder used by the GSM application. {being deprecated}
XDOCUMENTS_HOME	Location of the XDocuments folder used by most applications
PDFWorkArea	When using WebDAV for document collaboration in NPD, set this as the area exposed for WebDAV for PDFs. This value requires a trailing '\ ' or '/'
PDFCommentRepository	When using WebDAV for document collaboration in NPD, set this as the location for WebDAV PDF comments.
PDFWorkAreaUNC	The physical location that the WebDAV virtual folder (above) maps to. This value requires a trailing '\ ' or '/'
PDFCommentRepositoryUNC	The physical location that the WebDAV PDFComments virtual directory maps to. This value requires a trailing '\ ' or '/'
OfficeDocWorkArea	When using WebDAV, set this as the location for Microsoft Office Documents. This value requires a trailing '\ ' or '/'
OfficeDocWorkAreaUNC	The physical location that the WebDAV OfficeDocWorkArea virtual directory maps to. This value requires a trailing '\ ' or '/'
PasswordExpiration	The number of days for user passwords to expire. This setting is only picked up if password expiration is enabled in the customer settings.

ProdikaSettings

The ProdikaSettings element extends the Prodika element with some additional settings.

ProdikaSettings / Attachments

Table 1-8: ProdikaSettings / Attachments: key names, expected values, and descriptions

Key Name	Expected Value	Description
UNC	Location of folder	Deprecated
company	Location of folder	Deprecated
facility	Location of folder	Deprecated
sac	Location of folder	Deprecated

ProdikaSettings / ProdikaModules

This node maps keys of application name to the application virtual folder names of various Agile modules. (Do not modify without consulting Agile.)

WebCommon

Table 1-9: WebCommon: key names and descriptions

Key Name	Description
ActionMenuImageOnClickJScript	Location of the Agile portal for forwarding application on certain JavaScript calls in the application. The URL to the Agile portal should be modified, but nothing else should typically be changed.
LogoutEnabled	Do not modify without consulting Agile. This value should typically be set to False.

SideNav

This specifies the location of the Agile portal to forward the application to when a user clicks the Agile icon on the left navigation panel of any application. It also specifies the images to be used for the link back to the portal, as well as the site map.

Application Settings

This section describes the various configuration keys utilized by the Agile application. Table 1-10 below briefly denotes the location of the configuration key, then follows it with the description.

Table 1-10: Application settings, keynames and descriptions, by category

Key Name	Description
DRL / DRLSettings	
	See Environment Variable Settings on page 4
DWB / DWBSettings	
HomeURL	Location of the Agile portal See Environment Variable Settings on page 4
SiteMapURL	# - This should not be modified.
EQ / EQParam	
EQUrl	Location of the Supplier Portal application. This URL is used in constructing emails to suppliers when an eQuestionnaire (eQ) is submitted. Environment Variable Settings on page 4
EQQueryParam	Parameter used to specify the eQ in the URL. This should not be changed without prior consultation.
EMAIL_FROM_ADDRESS	The email address used to specify the FROM line of the email sent to a supplier from eQ
EQ / EQSettings	
GSMApplicationPath	Location of the GSM application. See Environment Variable Settings on page 4
HomeURL	Location of the Agile portal. See Environment Variable Settings on page 4
SiteMapURL	Do not modify
GSM / Xeno.Data.GSMWeb.MiscSettings	
WorkflowUserPopupURL	System – do not modify
ActionMenuImageOnClickJScript	JavaScript to point to portal. Do not modify this value. Instead, directly change the value of VAR:Prodika.Portal.URL. See Environment Variable Settings on page 4
PLM/AppSettings/add	
LogDirectory	Value can be changed to a custom directory, but this setting would typically not require modification.
InstrumentationLogFile	Uncomment the line to add instrumentation. If the InstrumentationLogFile is not set, no instrumentation logging will take place. The value indicates the file name to use.
NPD / AppSettings	
WorkflowEmailAddress	The FROM address in workflow emails sent by NPD
ProdikaReporting / AppSettings	
AdditionalReport.GroupLocation	URL to the Customized Reporting groups. See Environment Variable Settings on page 4
ProdikaReporting / ReportConfig / ServiceConfig / AuthenticationCredentials	
username	User name for the MS SQL Reporting Server that can execute reports uploaded into the Reporting Server.
password	Password for the user specified above
domain	Domain name for the user specified above (empty if the account is not a domain account)

Table 1-10: Application settings, keynames and descriptions, by category (continued)

ProdikaReporting / ReportConfig / ServiceConfig / ReportingService	
serviceLocation	Location of the MS SQL Reporting Service Web service
Reg / AppSettings	
ExtendedAttributes.Calculated.Scripting.HelpURL	If Calculated Extended Attributes help content is available, specify the URL to the help documentation.
SCRM/Xeno.Data.SCRMWeb.MiscSettings	
ActionMenuImageOnClickJScript	Location of the Agile portal to be available to SCRM JavaScript. . Should not be modified here. Modification is controlled by changing the environment variable VAR:Prodika.Portal.URL See Environment Variable Settings on page 4
SupplierPortalAdmin	
HomeURL	Location of the Agile portal
SiteMapURL	# - do not modify
SupplierPortal	
HomeURL	Location of the Agile portal
SiteMapURL	# - do not modify
RefLibURL	Document Reference Library URL. Specify the URL to the Document Reference Library.
WFA/AppSettings	
AllowCreate	Specifies if workflows can be created in this environment. For example, in a live production system, workflow creation should be disabled, but it should be enabled in a preproduction system. This functionality is also dependent on how DataExchangeServer is configured. See DexConfig on page 15.
AllowEdit	Specifies if workflows can be edited in this environment. This functionality is also dependent on how DataExchangeServer is configured. See DexConfig on page 15.
AllowCopy	Specifies if workflows can be cloned in this environment. This functionality is also dependent on how DataExchangeServer is configured. See DexConfig on page 15.
AllowImport	Specifies if workflows can be imported into this environment. In live production systems, workflows are typically imported from a preproduction environment. This functionality is also dependent on how DataExchangeServer is configured. See DexConfig on page 15.
PMA/AppSettings	
AllowUserCreate	Specifies if users can be created in this environment. In a live production system, users might be not created, whereas in a preproduction environment, they might be. This functionality is also dependent on how DataExchangeServer is configured. See DexConfig on page 15.
AllowUserEdit	Specifies if a user can be edited in this environment. This functionality is also dependent on how DataExchangeServer is configured. See DexConfig on page 15.
AllowUserImport	Specifies if a user can be imported into this environment. This functionality is also dependent on how DataExchangeServer is configured. See DexConfig on page 15.
AllowUserExport	Specifies if a user can be exported from this environment. This functionality is also dependent on how DataExchangeServer is configured. See DexConfig on page 15.

Table 1-10: Application settings, keynames and descriptions, by category (continued)

AllowGroupCreate	Specifies if a group can be created in this environment. This functionality is also dependent on how DataExchangeServer is configured. See DexConfig on page 15.
AllowGroupEdit	Specifies if a group can be edited in this environment. This functionality is also dependent on how DataExchangeServer is configured. See DexConfig on page 15.
AllowGroupImport	Specifies if a group can be imported into this environment. This functionality is also dependent on how DataExchangeServer is configured. See DexConfig on page 15.
AllowGroupExport	Specifies if a group can be exported from this environment. This functionality is also dependent on how DataExchangeServer is configured. See DexConfig on page 15.

RemotingContainer Section

Table 1-11: Remoting Container/config info

Key name	Description
UserID	System UserID to be used to log into Remoting Container (see Environment Variable Settings on page 4)
Password	System password for logging into the system from Remoting Container (see Environment Variable Settings on page 4)

Table 1-12: Remoting Container Remote Services

Service name	Description
AuthenticationBridgeService	Needed for Single Sign On
ReportProcessingService	Needed for report generation
AuthorizationBridgeService	Required for DRL file authorization access check
TaxonomyDenormService	Required for taxonomy denormalization for search
OLSDenormService	Required for Object Level Security service
SyndicationService	Required for data syndication
DataImportProcessor	Required for processing Data Exchange Requests

The IsActive flag needs to be set on the services for them to be active.

Most of these values are controlled by environment variables. (See [Environment Variable Settings](#) on page 4). Each of the services (RemotingContainer/RemoteServices/Service) can be turned on or off by setting the isActive attribute to “true” or “false.”

DexConfig

Background

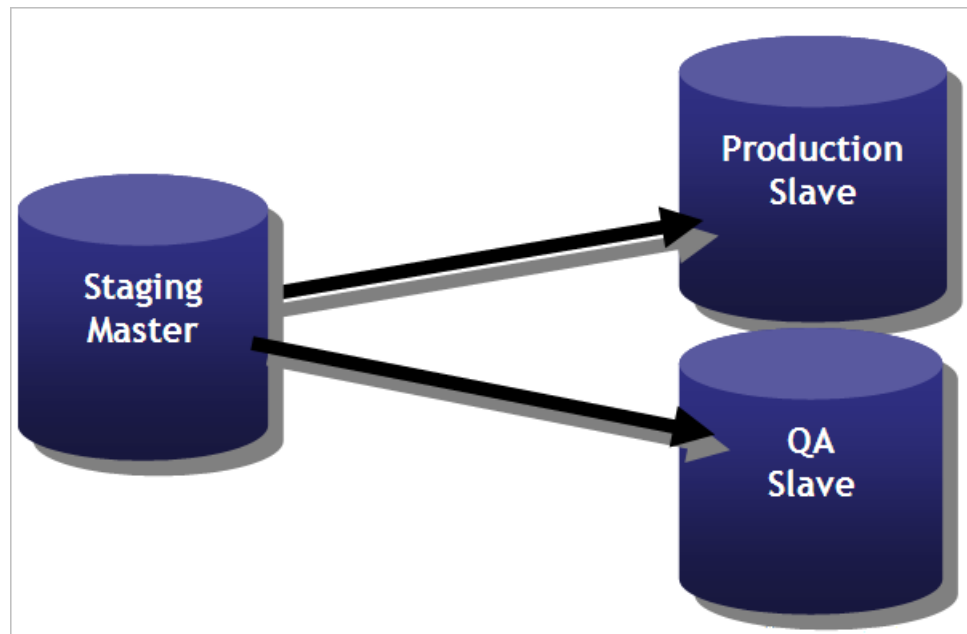
In a typical landscape, there are multiple environments where data is moved into finally culminating at the Production server. Traditionally, this data has either been moved around using SQL scripts or in rare scenarios has been re-entered when moving between systems. Re-entering the data in multiple systems is not only a cumbersome process, but also can be error prone. Furthermore, if some of the data is created locally to a system while the rest is imported using SQL scripts, then data consistency is lost because the Identifiers are recreated on every data creation effort. For example, if two Business Units with the same title, “BU1,” are created in two different environments, their identifiers would be disparate. Thus, if a workflow referencing “BU1” has to be migrated from the first to the second environment, the identifiers would not match up.

Solution

To address the problem of having to create data using SQL scripts, we had introduced a data administration application a few releases back. To address the problem of data migration, we introduced the Data Exchange Service and the concept of a master and a slave environment.

The master environment is typically where the data is created. It is typically the only origination point for data for a particular version of the software. All admin data is generated on the master environment and is then migrated into multiple slave environments. A typical data exchange topology looks like that shown in figure 1-1 below.

Figure 1-1: Typical data exchange topology



The Staging Master environment is where the data is created and then gets migrated to the QA Slave first. User acceptance testing is performed on the QA Slave environment during which admin data might be required to be updated based on issues found. These changes to admin data will be performed in the Staging Master and remigrated to the QA Slave environment until the user acceptance testing is complete. Once the data is considered “golden,” a data export packet would be created on the Staging Master for Production Slave and data migrated into the Production environment.

By following the process outlined above, you can keep data in sync and preserve data identifiers.

Deployment Considerations

While deploying a Master and Slave set of servers, it is important to note that each system has its own database and its own installation of the software. They are truly disparate systems. In the `EnvironmentSettings.config` file, a section named **DataExchangeConfig** is available. This section in the configuration file helps determine if a particular system is a Master or a Slave. The configuration node (shown below) has two child nodes named `TargetSystems` and `SourceSystems`. When a target system is defined, the current system is declared to be a Master. In other words, the current system (“Staging Master” in the example below) has target systems (“QA Slave” and “Production Slave”) that it can export data to.

```
<DataExchangeConfig System="Staging Master"
NotifierEmail="#REPLACE_ME@DOMAIN.COM">
  <TargetSystems>
    <System>QA Slave</System>
    <System>Production Slave</System>
  </TargetSystems>
</DataExchangeConfig>
```

A typical Slave environment will have its configuration set as below. The example below is particularly for the production environment.

```
<DataExchangeConfig System="Production Slave"
NotifierEmail="#REPLACE_ME@DOMAIN.COM">
  <SourceSystems>
    <System>Staging Master</System>
  </SourceSystems>
</DataExchangeConfig>
```

A slave environment can have only one source system, whereas a master environment can have multiple target systems.

Customer Configuration Settings

Overview

The Customer Settings configuration file contains customer-specific and core system configuration parameters.

The file is located at:

[X]: \\%Prodika_Home%\Config\Custom\CustomerSettings.config

This file is typically edited in conjunction with `EnvironmentSettings.config`.

Configuration File Format

The elements in the file are outlined below with details following in tables.

Core

Used By: All applications

FormsAuthenticationHelper

This configuration node is used to augment the configuration parameters in the `EnvironmentSettings.config` Core/`FormsAuthenticationHelper` node.

SingleSignOn

The attribute `accessDeniedURL` is used to specify the URL to the system “access denied” message page. This should not be changed.

Portal

The attribute `enabled` is used to specify whether or not the Agile portal is enabled for site login and navigation.

Prodika / Services

This element is part of the configuration pipeline for the services used by Agile. In the Customer Settings file it is used to override or extend service configuration parameters.

Table 1-13: Prodika / services: key name and description, by service name

Service name	Key Name	Description
ExtendedAttributeSectionBuilderService	MaximumNumberOfRowsThroughCopy	Specifies the allowable number of rows that can be copied in the Extended Attribute Template Editor (in the Data Admin Application).
TaxonomyService	NamespaceAliasMappings	Specifies a Taxonomy Alias handler, which should not typically be changed. Further specifies one or more <AliasMapping> nodes, used when GSM specification taxonomy applies across multiple specification types.

Prodika / Managers

The <**Managers**> element contains definitions and configurations for the Agile platform.

The following table summarizes the various sections and settings in this node.

Table 1-14: Prodika / Managers: key name and description, by manager name

Manager name	Key Name	Description
TranslationManager	CUSTOMER_CONTEXT	Used to specify a Customer Context (Name) that corresponds to a Customer Proprietary Translation Override table in the underlying application database. This typically should not be changed.
EnvironmentManager	MinPasswordSize	The minimum password length (used during UGM user validation).
EnvironmentManager	MaxPasswordSize	The maximum password length (used during UGM user validation).

ProdikaSettings / SingleSignOn

The key PasswordChangeEnabled allows the user to change their Agile password under User Preferences when set to “true.”

Deployment Settings

Enables or disables specific applications and functionality from the deployed instance.

Table 1-15: Deployment settings: key names, described

Key Name	Description
ALL	Enables all Agile PLM applications, unless otherwise overridden in this section
DEMO	Should be set to “disabled”
CACS	Enables or disables the Computer Aided Compliance Screening (CACS) application
ExtendedAttributes	Enables or disables the Extended Attributes feature
CSS	Enables or disables the Content Synchronization and Syndication (CSS) application

The deployment settings will enable or disable the above features and applications contextually.

DWB

This section contains one node, DWBSettings, which can override or extend DWB Settings higher in the configuration pipeline. The typical keys are listed below.

Table 1-16: DWBSettings: key names, described

Key Name	Description
BusinessUnitNamespace	Specifies the appropriate Business Unit taxonomy namespace to use in DWB. Typically “GSMBusinessUnit” and should not be changed.
DWB.DefaultUOM.ISOCode	Specifies the default UOM to use in the DWB Formulation editor.
DWB.Security.Enabled	Enable or disable DWB security features with regard to read and write access to specifications.

Extended Attributes

DynamicScriptMethods

This is an extensibility point by which you can extend the behavior of calculated extended attributes and what they can resolve to.

The key denotes a unique namespace for a Dynamic Script Method. Custom implementations need to implement the `Xeno.Prodika.ExtendedAttributes.Calculation.IDynamicScriptMethod` interface. The full Object URL to that implementation should be specified as the value.

For example:

```
<add key="Mock"
value="Class:Xeno.Prodika.ExtendedAttributes.Calculation.Mock
DynamicScriptMethod,ProdikaLib"/>
```

GSM

GSMConfig

This section contains core settings for the GSM application as key-value pairs.

Table 1-17: GSMConfig: key names, described

Key Name	Description
DisplayNutrientItemServingSize	Do not modify
DisableComplianceRollupSpecTypeList	Do not modify
InactiveSpecTypes	Do not modify
Allow3rdPartyControlledCopy	Do not modify

GSMSettings

Table 1-18: GSMSettings: key names, described

Key Name	Description
RoundingRulesProcessor	System – do not modify
DisplayFormulaComparison	true – do not modify
ShowReferenceAmountClassification	Shows the reference amount classification on the nutrient profile
AddHTCBehaviorToContextHelp	Set to “true” or “false” based on the behavior that needs to be added.
GSM.<SpecTypeName>.SpecSummary.ShortName.MaxLength	<p>Specify the max Short Name length for a given Specification Type where <SpecTypeName> can be one of the following values:</p> <ul style="list-style-type: none"> • TradeUnitSpec • ProcessSheetSpec • MenuItemSpec • NutrientProfile • ProductSpec • IngredientSpec • PackagingSpec • EquipmentSpec • FinishedPackagingSpec (Printed packaging spec) • DeliveredPackingSpec • PackingConfigSpec • LabelingSpec • MasterSpec <hr/> <p>Note Care must be taken that the ShortName is not larger than the actual database column in which it needs to be persisted.</p> <hr/>
GSM.ProcessSheetSpec.DefaultUOM.ISOCode	Specifies the default UOM for process specifications
GSM.EQT.SpecSummary.IngredientSpec.ShortName.Enabled	If enabled, includes specification short name in the key field search criterial drop-down list and also displays the Short Name column in the search results table

frmPrint2

This is an extensibility point by which you can specify the location of custom print formatting transformations in the configuration pipeline.

The configExtends value specifies a file location and XPath query to the custom configuration data.

For example, if custom print settings were located in the file

[X]\%PRODIKA_HOME%\Config\Extensions\PrintSettings.config, and the configuration data in that file was in the XML XPath PrintSettings/GSM/FormattingObjectMap, then the configuration node would appear as follows:

```
<frmPrint2
configExtends="file:%PRODIKA_HOME%\Config\Extensions\
PrintSettings.config,PrintSettings/GSM/FormattingObjectsMap"
/>
```

CustomUOMLists

This is an extensibility point by which you can specify the custom UOM Lists to be leveraged in specific locations of the Agile application. The core UOMs are listed in %PRODIKA_HOME%\config\Extensions\CustomUOMLists.config. That location can also act as a template for additional items.

The configExtends value specifies a file location and XPath query to the custom configuration data.

Xeno.Data.GSMWeb.MiscSettings

Table 1-19: Xeno.Data.GSMWeb.MiscSettings: key names, described

Key Name	Description
WorkflowUserPopupURL	This should not be modified.
WorkflowUserPopupWindowParams	This should not be modified.
ProcessSheetAddFacilityDefaultBUupid	Specifies the default Business Unit to assign to a new facility when added from a process specification on the GSM user interface. The value must refer to the primary key for a row in the commonBusinessUnit table.
MasterSpecAppliesToSpecTypes	A comma-delimited list of specification type identifiers to which master specifications can be applied. This typically should not be changed.

ProdikaSettings / ConceptAwareSpecs

This key-value list is used to enable or disable the Concept feature for specification types. The key is a specification type identifier, and the value is either “true” or “false”.

ProdikaSettings / GlobalSuccession

This key-value list is used to enable or disable the Global Succession feature for specification types, provided Global Succession is supported out-of-the-box for a given specification type. The key is a specification type identifier and the value is either “true” or “false”.

GSMLibConfig / RequiredPropertiesForValidSpecs / Validator

Caution! This is a system setting – do not modify.

GSMLibConfig / TestingProtocolAPConfig

Table 1-20: GSMLibConfig / TestingProtocolAPConfig: key names, described

Key Name	Description
MinScore	The allowed systemwide min score for testing protocol criteria
MaxScore	The allowed systemwide max score for testing protocol criteria

AssociatedSpecsConfiguration

This configuration section specifies the EQT search views to be used when searching to add associated specifications in GSM. The Key attribute is a GSM specification type identifier that refers to the specification to which associated specifications are being added. The Value specifies the EQT Search View to use for that specification type.

For example:

```
<add key="1004" value="SearchableView:Config:ProdikaSettings/
SearchableMultiSelectViews,IngredientSpecAssociatedSpecView"
/>
```

Caution! This is a system setting – do not modify.

NPD*AppSettings*

This section contains core settings for the NPD application as key-value pairs.

Table 1-21: AppSettings: key names, described

Key Name	Description
ReadAccessDeniedHtml	System – do not modify
FinancialAccessDeniedHtml	System – do not modify
PortfolioConcurrentLoadLimit	Can be lowered if concurrent testing of NPD project portfolios results in a performance issue in your topology
PortfolioSizeLimitWithSecurity	Limits the number of portfolio returns for roles for which security is checked, to protect application server performance
PortfolioSizeLimitNoSecurity	Limits the number of portfolio returns for roles for which security is NOT checked, to protect application server performance
ISPSupport	Off – do not modify {This feature is being deprecated}
IdeaSupport	On – do not modify (This is the supported ISP)
PackageCopySupport	Off – do not modify
POPSupport	On – do not modify
ConsumerTechMatrixSupport	Off – do not modify

Table 1-21: AppSettings: key names, described (continued)

ViewPrivateTemplateSupport	Off – do not modify
DefaultStageMaster	System – do not modify
CommonCurrencyISOCode	System – do not modify
DefaultStageSecondReviewAccelerationFactorPercentage	Changes the default review acceleration percentage

Analytics

This section contains a series of Analytic nodes that define custom analytics for NPD. The Analytic nodes are defined below. Using analytics requires third-party charting software.

Table 1-22: Analytics for NPD: attribute/element names, described

Attribute / Element Name	Description
id	A unique identifier for the Analytic.
chartType	Specifies the type of analytic chart to render. Valid options include “BarChart” and “PieChart.”
title	Specifies the unique title for the Analytic. This can correspond to a custom translation, which will be rendered on the chart if available.
aggregationField	Specifies the scope of aggregation for the metric. Valid values include “Stage,” “Project,” and “ProjectType.”

Each Analytic node contains a ProjectMetrics node. The ProjectMetrics node contains one or more ProjectMetric children. A ProjectMetric has one attribute, projectMetricId. The projectMetricId attribute specifies the NPD Metric identifier that the analytic corresponds to.

For example:

```
<Analytic id="DiPie" chartType="PieChart"
title="lblDiByProjectType" aggregationField="ProjectType">
<ProjectMetrics>
<ProjectMetric projectMetricId="FY1_DI" />
</ProjectMetrics>
</Analytic>
```

ActivityTypeUIMapper

This section is used to relate specific NPD activity types to corresponding user interface controls. It should not be modified.

PQS

AppSettings

This section contains core settings for the PQS application as key-value pairs.

The formula comparison limit can be modified to an appropriate value. Do not modify without consulting the services team.

Table 1-23: PQS AppSettings: key names, described

Key Name	Description
UniversalTestingProtocols	When set to “true,” allows scorecards to be created on any testing protocol in the system regardless of the spec associated with the scorecard.
AllowScorecardToggle	When set to “true,” allows the scorecard view to be toggled between an admin and non-admin view.
InactiveScorecardTypes	System – Do not modify
ReportingPopupFilteredDataSource_PQSScorecardQualificationType	Should not be modified.
ReportingPopupFilteredDataSource_PQSScorecardTypesMS	Should not be modified.
ConfiguredRatingStrategyResolver	Should not be modified.
NonPQSAdminQualificationLookupFilter	Should not be modified.

SCRM

SCRMConfig

This section contains core settings for the SCRM application as key-value pairs.

Table 1-24: SCRMConfig: key names, described

Key Name	Description
ShowLegalAgreements	Off – do not modify
SupplierDocumentManagmentEnabled	On – do not modify without consulting services
FacilitySupplierDocumentManagmentEnabled	On – do not modify
ReferencesTabEnabled	Enables or disables the SCRM references tab
HidePublishToSupplierPortal	Specifies whether or not the Publish To Supplier Portal field is available within SCRM
EmailSupplierRep	The FROM email address for emails generated to suppliers from SCRM

Supplier Portal

SupplierPortalConfig

This section contains core settings for the Supplier Portal application as key-value pairs.

Table 1-25: SupplierPortalConfig: key names, described

Key Name	Description
EmailFromAddress	The FROM email address for emails generated to suppliers from Supplier Portal
BUFacilitySearchPattern	Specifies the Business Unit filter for SCRM supplier searches. Should be modified to represent a node in the SCRM business unit taxonomy
DRLEnabled	Enables or disables usage of the Document Reference Library from within Supplier Portal
AddNewContactsEnabled	Enables external providers to add new contacts to the Supplier Portal
EditContactsEnabled	Enables external providers to maintain contacts in the Supplier Portal
RemoveContactsEnabled	Enables external providers to remove contacts from the Supplier Portal

WFA

AppSettings

This section contains core settings for the WFA application as key-value pairs.

Table 1-26: WFA AppSettings: key names, described

Key Name	Description
MaxSearchCriteriaCount	This should not be modified.
TemplateSummaryControl	This should not be modified.
AllowEasyNewWorkflowStatusCreation	This should be set to “false” and not modified.
AllowExplicitNewWorkflowStatusCreation	This should be set to “true” and not modified.

EQTWorkflowMappingConfig

This section specifies the workflow process types that are available from the main search screen in the Workflow Admin Tool (WFA). These values should not be modified.

PMA (UGM)

SuiteApplications

The SuiteApplication values are system defaults and should not be modified.

Workflow

LinearWorkflowResolverConfig / TransientWorkflowPKIDs

This section specifies the system identifier for the default (“Transient”) workflow state. This setting should not be modified.

FeatureConfig

The FeatureConfig defines the available, configurable applications features that are enabled in the application instance. The FeatureConfig section can be used in two ways, as explained below.

First, it can be modified to specify a particular Feature Configuration pipeline, which is a predefined feature configuration for a particular industry or customer. This is done using the configExtends attribute of the **<FeatureConfig>** element.

For example, to enable the out-of-the-box settings for a consumer packaged goods application feature set, specify the following **<FeatureConfig>** element with the configExtends attribute:

```
<FeatureConfig  
configExtends="file:%CONFIG_HOME%\CPGFeatureConfig.config,  
FeatureConfig">
```

Second, the **<FeatureConfig>** section can be used to override **<FeatureConfig>** settings that are enabled or disabled earlier in the configuration pipeline. To do this, you must identify the key for the desired **<FeatureConfig>** setting and then add a key-value node inside the **<FeatureConfig>** section that specifies its value as either “true” or “false.”

For example, to enable the food item catalog (FIC) – DWB integration feature, add the following element to the **<FeatureConfig>** section:

```
<add key="REG.FIC.Integration.DWB.Enabled" value="true"/>
```

ExtendedAttributesSettings

This is an extensibility point by which you can specify the location of Extended Attribute Syndication Overrides in the configuration pipeline.

The configExtends value specifies a file location and XPath query to the custom configuration data. Under CustomerNutrientItems/NutrientItemConfig you can add custom nutrient items with the following properties.

Id	Translatable Label
translationCache	TranslationCache to lookup the nutrient translation from
Active	True/false

NutrientItem can have formulas as shown below.

Id	Translatable Label
Algebra	Can use formulas for calculation, for example, the formula {lblEnergyKJ}[/ 1.0] refers to another nutrient item indicated by {<nutrient item id>} and literal operations are demarcated by []. So in the above example the algebra represents a calculation of the value of nutrient item lblEnergyKJ divided by 1.

Formulas can have calculation methods and those are indicated by CalculationMethod nodes with the following properties.

Id	Translatable Label
name	Name of the calculation method These can have values that are either lblFSANZ, lblDefault or lblConvertFromKCal
active	Indicates whether the method is active or not.
default	Indicates if the method is the default for a particular formula

Validation Rules Configuration

Overview

The Agile application provides a framework to add custom validation rules to GSM Specifications on save and workflow events. Basic validations are declared in the XML configuration file shown below.

```
%PRODIKA_HOME%\config\Extensions\ValidationSettings.xml
```

Configuration File Format

The validation configuration file contains two key sections, **<ValidatorFactories>** and **<ValidationRules>**.

ValidatorFactories

The ValidatorFactories node registers the Object URLs for existing Validator components and assigns them a short name to be used in the **<ValidationRules>** section. All predefined validators are declared in the **<ValidatorFactories>** node in the file `\config\Core\validationRules.xml`. Custom validators and ValidatorFactories must be declared in the `ValidationSettings.xml` file. This configuration section should not change unless the customer has created a custom validator.

ValidationRules

This section contains the actual validation rules for each specification type. A set of validation rules are specific to a particular specification type.

The following example defines a rule for ingredient specifications (type “1004”). When this specification type is saved, the validation framework will check to ensure that the Specification has a Taxonomy classification assigned.

```
<rule type="1004">
  <condition event="save">
    <if type="ReflectiveRegExValidator"
      property="SpecSummary.TaxonomyNode.ParentNode.TaxonomyML.Name"
      expression="^[^\*]" report="true" />
  </condition>
```

Rules

The rule node (`/configuration/ValidationRules/rule`) is used to specify validation rules for a particular specification type. The `type` attribute is used to indicate the specification type (for example: 1004, 1009, 2121).

Rules are built by creating sets of **<conditions>** (described below). A rule can have multiple sets of conditions. Any single condition that returns “true” means that the specification has not passed the validation requirements and a “save” or “workflow” event will not be allowed.

Conditions

The condition node (`/configuration/ValidationRules/rule/condition`) is used to indicate which event should be evaluated: save or workflow.

This is specified using the event attribute of the **<condition>** element. When event= “save,” validations are evaluated on the save event. When event= “workflow,” validations are evaluated on a workflow transition to the next step.

The <if> Element

Conditions are built using a series of **<if>** XML elements. **<if>** elements can be nested, such that child **<if>** elements are evaluated when the parent **<if>** is true.

The **<if>** element contains several required attributes as well as optional attributes that are specific to the type of validator that the **<if>** is configured to use. Refer to the validators listed below for details.

Validators

Validators are registered with a unique name in the **<ValidatorFactories>** element at the beginning of the `validationRules.xml` configuration file. These registered names are then referenced as the type attributes of the **<if>** elements when configuring the validation conditions. Any additional configuration attributes required by the validator are specified as additional attributes of the **<if>** element.

ReflectiveRegexValidator

The `ReflectiveRegexValidator` uses regular expressions and reflection to determine if the provided value matches the value provided in the condition configuration. For usage, see figure 1-27, below.

Table 1-27: `ReflectiveRegexValidator`: attribute name, required, expected value, and description

Attribute Name	Required	Expected Value	Description
type	Yes	“ReflectiveRegexValidator”	The value of the type attribute is the named Validator that should be used in the evaluation.
property	Yes	property path	Full path to the property to be validated on the specification data object.
report	No	“true”, “false”	Reports a validation error. Set this to “false” to configure a guard condition.
required	No	“true”, “false” (defaults to “false”)	If the object at the specified path does not exist, the validator will throw an error message if this is set to “true.”
expression	Yes	regular expression	The regular expression to use in the validation

ReflectiveRequiredValidator

Checks that the bound specification data object property is not null or that the data object collection contains at least one item. For usage, see figure 1-28, below.

Table 1-28: ReflectiveRequiredValidator: attribute name, required, expected value, and description

Attribute Name	Required	Expected Value	Description
type	Yes	"ReflectiveRequiredValidator"	The value of the type attribute is the named Validator that should be used in the evaluation.
property	Yes	property path	Full path to the property to be validated on the specification data object.
report	No	"true", "false"	Reports a validation error. Set this to "false" to configure a guard condition.

ReflectiveRangeValidator

Validates that the bound specification data object property falls within a specific range. For usage, see figure 1-29, below.

Table 1-29: ReflectiveRangeValidator: attribute name, required, expected value, and description

Attribute Name	Required	Expected Value	Description
type	Yes	"ReflectiveRangeValidator"	The value of the type attribute is the named Validator that should be used in the evaluation.
property	Yes	property path	Full path to the property to be validated on the specification data object.
report	No	"true", "false"	Reports a validation error. Set this to "false" to configure a guard condition.
min	Yes	numeric	The minimum value for the range
max	Yes	numeric	The maximum value for the range
required	No	"true", "false"	Set to "true" to indicate that the property value should exist.

ReflectiveDateValidator

Validates that the bound specification data object property is greater or lesser than a specific date.

Usage

Table 1-30: ReflectiveDateValidator: attribute name, required, expected value, and description

Attribute Name	Required	Expected Value	Description
type	Yes	"ReflectiveDateValidator"	The value of the type attribute is the named Validator that should be used in the evaluation.
property	Yes	property path	Full path to the property to be validated on the specification data object.

Table 1-30: ReflectiveDateValidator: attribute name, required, expected value, and description (continued)

report	No	“true”, “false”	Reports a validation error. Set this to “false” to configure a guard condition.
propertyAlias	No	any string	If available, this value is used as the field name in the error message when validation fails. Otherwise, it defaults to the value of the property attribute.
value	Yes	“now” or a date	If the value is “Now,” the rule will be evaluated against the current date.
operator	Yes	“before”, “after” or “equals”	If the operator is “Before,” the system will evaluate if the property date is less than or equal to the value date. If the operator is “After,” the system will evaluate if the property date is greater than the value date.

SpecWorkflowSystemActionValidator

Inspects a specification or specification summary data object to see if it contains a SystemAction (also known as ExtendedBehavior). The validator returns “true” if the behavior is found.

Usage:

Table 1-31: SpecWorkflowSystemActionValidator: attribute name, required, expected value, and description

Attribute Name	Required	Expected Value	Description
type	Yes	“SpecWorkflowSystemActionValidator”	The value of the type attribute is the named Validator that should be used in the evaluation.
property	Yes	property path	Full path to the specification or specification summary that contains the workflow process to be evaluated.
report	No	“true”, “false”	Reports a validation error. Set this to “false” to configure a guard condition.
propertyAlias	No	any string	If available, this value is used as the field name in the error message when validation fails. Otherwise, it defaults to the value of the property attribute.
checkTargetState	No	“true”, “false” (defaults to “false”)	When true, the validator will use the current LinearTransition context to determine if the activity the specification is being transitioned to has the BehaviorID. When the value is “false,” the validator checks the current behavior of the current activity instead.
behaviorID	Yes	numeric	The numeric ID of the configured SystemAction that the validator should check for.

CollectionObjectRequiredValidator

Inspects a collection and verifies that an object exists in that collection. For example, validating that an extended attribute exists on a specification.

Table 1-32: CollectionObjectRequiredValidator: attribute name, required, expected value, and description

Attribute Name	Required	Expected Value	Description
collectionProperty	Yes	collection property path	Full path to the collection that the object exists in
objectKeyProperty	Yes	property path	Full path on the object in the collection to validate against
objectKey	Yes	any string	Value that the objectKey property should be equal to
report	No	“true”, “false”	Reports a validation error. Set this to “false” to configure a guard condition.

ReflectiveItemValueComparisonValidator

Tests two properties on an object and compares the values.

Table 1-33: ReflectiveItemValueComparisonValidator: attribute name, required, expected value, and description

Attribute Name	Required	Expected Value	Description
property	Yes	property path	First object to compare
comparedProperty	Yes	property path	Second object to compare
comparisonOperator	Yes	<, >, >=, <=, <>, =	Standard comparison operator symbols
report	No	“true”, “false”	Reports a validation error. Set this to “false” to configure a guard condition.
propertyUOM	No	property path	Full path to the UOM property of the first object
comparedPropertyUOM	No	property path	Full path to the UOM property of the second object

MultipleClauseReflectiveRequiredValidatorWrapper

Performs just like the ReflectiveRequiredValidator except it also verifies that the object is required in a collection object.

Table 1-34: MultipleClauseReflectiveRequiredValidatorWrapper: attribute name, required, expected value, and description

Attribute Name	Required	Expected Value	Description
type	Yes	“MultipleClauseReflectiveRequiredValidatorWrapper”	The value of the type attribute is the named Validator that should be used in the evaluation.
property	Yes	property path	Full path to the property to be validated on the specification data object
report	No	“true”, “false”	Reports a validation error. Set this to “false” to configure a guard condition.

TIPInitiatedValidator

Validates that a TIP has been created on a specification.

Table 1-35: ReflectiveItemValueComparisonValidator: attribute name, required, expected value, and description

Attribute Name	Required	Expected Value	Description
report	Yes	“true” , “false”	Reports a validation error. Set this to “false” to configure a guard condition.

Content Synchronization and Syndication Settings

Overview

The `CssLibConfig.xml` configuration file contains settings needed to run the Agile Content Synchronization and Syndication (CSS) application. Most of the configuration is provided out-of-the-box, but some key settings vary.

CSS is an integration application that requires engineering-level resources to configure for syndication.

There are two primary types of CSS syndication: internal and external.

Internal Syndication

The recommended architecture for doing internal syndication is for Agile to syndicate to a middleware product that can then be configured to send messages to one or many destination systems.

External Syndication

External syndication is used to publish specification data to a system outside of the customer firewall. The most common scenario is syndication to a third-party data exchange, such as 1Sync.

In an external syndication scenario, specific configurations to modify the message bundling, formatting, and transport are necessary.

Configuration Sections

ConfigSections

This section contains declarations of the configuration sections and configuration section handlers for the CSS configuration file. The **ConfigSections** node is core to the application configuration framework and should not be changed.

<NamedParams> Section

The **<NamedParams>** section is used to specify name-value pairs to be used elsewhere in the **CSSLibConfig** file or within the CSS application.

Key parameters:

Table 1-36: <NamedParams> section parameters, described

Parameter	Description
WorkflowFromEmailAddress	The “From” email used for workflow transition notifications
CssPortalActionItemsUrl	The URL used to build the link to Action Items in email notifications
ExcludeLinkTradeTypePkids	Specifies identifiers for trade specification types that should not be syndicated
SyndicationServiceUrl	This named parameter is used to configure the MessageTransporterFactory for a Web service enabled publication path. It represents the URL of the Web service endpoint that this CSS publication path will send messages to.

<CSSConfigurations> Section

The main portion of the configuration for CSS happens in this section. For each publication path, you specify how the messages will be created, organized, published, transported, and reconciled.

Each child node under **<CSSConfigurations>** represents a distinct CSS publication path. An individual publication path configuration will contain some or all of the following attributes.

Table 1-37: CSSConfigurations section attributes, described

Attribute	Description
CSS Config root node	<p>Custom. Must correspond to the CSS publication namespace in the database configuration.</p> <p>The internal name of the publication path</p>
Exchange	<p>Custom. Can be used to specify the GLN of the target system when the publication path is for external syndication.</p> <p>The value is a unique identifier for a specific target system.</p> <p>Other than the requirement that the value be unique, there are no constraints for this value for internal data syndication. It is used with the transaction ID as a key to manage transactions and is sent along with the syndication message.</p>
CssTransactionLoaderFactory	<p>The transaction loader determines which TIPs should be published. The <code>StandardCssTransactionLoaderFactory</code> creates an object that loads all TIPs in the appropriate <code>CSSPublicationNamespace</code> that are in the configured workflow status: <code>StagedForSyndication</code>. (See NamedWorkflowStatuses on page 40.)</p> <p>This configuration should not typically be changed for the standard internal or external CSS syndication.</p>
<MessageGenerators>	Contains a series of specific <MessageGenerator> nodes.
<MessageGenerator>	<p>Contains a value attribute and a <MessageGeneratorFactory> and <TipDataAdapter> node.</p> <p>This section maps specification types to a particular message generator configuration. There is one for each spec type that the publication path will support. This is a standard setting.</p> <p>The value attribute defines the four-digit specification type identifier that this MessageGenerator configuration is for.</p> <p>This configuration should not typically be changed for the standard internal or external CSS syndication.</p>
<MessageGeneratorFactory>	<p>Contains the path to the CSS <MessageGeneratorFactory> component and additional parameters, including the Export Model identifier (for example, <code>CSSProcessSpec</code>) and the <code>XSLTransformResolver</code>.</p> <p>This configuration should not typically be changed for the standard internal or external CSS syndication.</p>

Table 1-37: CSSConfigurations section attributes, described (continued)

<TipDataAdapter>	<p>The <TipDataAdapter> component that is used to expose the data for syndication</p> <p>This configuration should not typically be changed for the standard internal or external CSS syndication.</p>
<TipValidatorFactory>	<p>This is a standard setting for using the default validation configuration.</p>
<MessagePublisherFactory>	<p>The publisher factory generates the XML for the transactions that were loaded by the configured <code>CssTransactionLoader</code>. Additionally, the configuration parameters to this factory allow the formatting and transformation of the message to be customized</p> <p>There are two message publisher factories to choose from. External syndications use the <code>TransoraMessagePublisherFactory</code> because it publishes every GLN (provider) separately but puts multiple transactions in each message.</p> <p>Internal syndications use the <code>SingleTransactionPublisherFactory</code> since it publishes each transaction in a separate message. This makes sense when you are working with an internal network because message granularity becomes more important than message quantity.</p>
<MessageTransporterFactory>	<p>The <MessageTransporterFactory> determines how the syndicated TIP message gets transported to the receiver.</p> <p>For internal syndication, the two most common options are the <code>FileCopyMessageTransporterFactory</code> and the <code>SyndicationServiceTransporterFactory</code>.</p> <p>The <code>FileCopyMessageTransporterFactory</code> saves the message to a specified location on a local or mapped drive as a file with a specified extension.</p> <p>The <code>SyndicationServiceTransporterFactory</code> sends the TIP message as a payload of a Web service request.</p> <p>This is a standard setting for the Web service based transport. Note that the object path has additional parameters, described on the next line.</p>
MessageTransporterFactory object path parameters (for SyndicationServiceTransporterFactory)	<p>[CSS name] [Web service URL] (defined in the <NamedParams> configuration section) [target system user name] [target system password]</p> <p>These parameters will need to be modified to reflect the target system user name and password.</p>

Table 1-37: CSSConfigurations section attributes, described (continued)

<NamePaths>	This configuration should not typically be changed for the standard internal or external CSS syndication.
<InfoProviderMap>	<p>Contains one or more <MapItem> nodes, which represent a key-value pair of attributes.</p> <p>Example:</p> <pre><MapItem key="GB" value="3155dc4be769-3a1e-4f89-98b4-2dd76124febb" /></pre> <p>The key corresponds to a particular CSS target market.</p> <p>The value represents the Agile PKID (a unique identifier) for the specific information provider(s) that the publication path is related to.</p>
<ResponseHandler>	This configuration should not typically be changed for the standard internal or external CSS syndication.
<ResponseAdapterSettings>	<p>There is currently only one ResponseAdapterFactory that can be configured for this setting: The CssResponseDataAdapterFactory.</p> <p>The ResponseAdapterFactory expects two parameters to be configured in its object URL string:</p> <ul style="list-style-type: none"> • Publication Namespace: This is the name of the current publication path, as defined in the configuration. • ResponseDataAdapter: The ResponseDataAdapter is the class responsible for loading a response and transforming it into a usable object by CSS. <p>Currently, there are three types of ResponseDataAdapters:</p> <ul style="list-style-type: none"> • CssServiceResponseDataAdapter: Used for the standard Web service response message • CssResponseXMLDataAdapter: Used for some types of external syndication responses. • CssResponseDatabaseMDNDataAdapter: A specialized class that is able to read responses from a third-party database
Specific Response Adapter Settings	
ResponsesFolderLocation	<p>The path to read responses from for reconciliation.</p> <p>This value should be changed to reflect the particular CSS environment.</p>
ResponsesArchiveFolderLocation	<p>The path to store inbound responses after processing.</p> <p>This value should be changed to reflect the particular CSS environment.</p>
ResponsesEnvelopeXSLFile	<p>Optionally, wrap the inbound response in another envelope by specifying an XSL transformation.</p> <p>This configuration should not typically be changed for the standard internal or external CSS syndication.</p>

Table 1-37: CSSConfigurations section attributes, described (continued)

ResponsesFullTransformXSLFile	<p>Optionally, transform the inbound response prior to processing.</p> <p>This configuration should not typically be changed for the standard internal or external CSS syndication.</p>
<Responses>	<p>This setting specifies which software component should process the incoming response.</p> <p>For the standard internal Web service enabled syndication, it is configured to retrieve and expose the Cross Reference values on the incoming response.</p> <p>This is a standard setting for Cross Reference-enabled integration.</p>
<CssReconcilerFactory>	<p>The <CSSReconcilerFactory> is the software unit that performs required actions based on the contents of the responses. Other than work-flowing the CSS TIP, the reconcilers do not have much else in common.</p> <p>The StandardTimedReconcilerFactory is used in situations in which the integration is asynchronous, so Agile cannot know when the responses will come in. This syndicate runs on a certain schedule and reconciles received responses and takes action to work-flow the TIPs that have not received responses in a timely manner. The StandardTimedReconcilerFactory works in conjunction with the StandardCssResponseFactory that is configured in the <Responses> XML node.</p> <p>The CrossReferenceReconcilerFactory is called both directly from the syndicate that is publishing the TIP and also from a separate syndicate. It can therefore be used in synchronous and asynchronous transactions. Additionally, it looks inside the responses for an ID that the spec can also be known by: a cross-reference. This cross-reference is usually an ID internal to and assigned by the target system. The CrossReferenceReconcilerFactory works in conjunction with the CrossReferenceResponseFactory, which is configured in the <Responses> node</p> <p>For the standard internal Web service enabled syndication, the configured component will set the cross-reference on the source specification. This is a standard setting for cross-reference-enabled internal integration.</p>
<ValidatorDelegates/>	This setting is not used for the standard internal or external CSS syndication.
<ConversionDictionary/>	This setting is not used for the standard internal or external CSS syndication.

PublicationPathConfigs

Configuration of the transformations and validation occurs here.

The reason that the PublicationPathConfigs are in a separate location from the rest of the configurations is that they are target market specific. You can have a different PublicationPathConfig for each target market per each configured publication path in the **<CssConfigurations>** section.

Each PublicationPathConfig contains the following sections:

Exchange

The exchange GLN needs to match the GLN configured in the **<CssConfigurations>** section for that publication path.

TargetMarket

This is typically the two-character country ID of the target market for this publication path.

TradingPartner

Usually this is configured as “default” for both the name and GLN. If there are multiple pub path configs (that is, different messages for different target markets), then each section should be configured independently.

Validator

If the TIP should be validated before it can be work-flowed or published, this is where the validator would be configured.

If the publication path is going to be used for more than one specification type, you can use MappedTipValidatorFactory to configure different validators for each specification type.

NamedWorkflowStatuses

This section contains a list of CSSWorkflowStatus PKIDs and names. The names are then referred to instead of the PKIDs from places like InitialNonGhostActivity column in the cssPublicationNamespace table. Additionally, the Transora Publisher uses the names to determine what step to workf-low the TIP to based on the response received after syndication.

The values in this section should not be changed unless the core CSS workflow status identifiers change.

Syndicating Custom Sections

Overview

The `exportExtensions.xml` file contains settings needed to add custom section data to the specification syndication. This file must be modified to specify the custom sections that should be syndicated for each specification type.

Configuration Sections

Config Sections

This section contains declarations of the configuration sections and configuration section handlers for the CSS configuration file. The `ConfigSections` node is core to the application configuration framework and should not be changed.

<ExportExtensibilityHandlers> Section

This section contains the configuration options for syndicating custom section data.

<SpecType#> Section

A **<SpecType#>** node is required for each specification type that is to be syndicated. For instance, to syndicate custom section data for an ingredient specification, a **<SpecType1004>** node must exist.

Each **<Spec Type#>** section contains `ExtensionItem` nodes that determine how the custom section data output should appear in the syndication. Two different handlers are available:

- 1 XmlNodeCreationExtensibilityHandler**—Creates a simple XML node used to wrap any other data. The name of the XML element created is indicated after the \$ symbol. Appending a pipe (|) symbol and the term “CloseTag” will create the closing XML tag.
- 2 CustomSectionExtensibilityHandler**—Used to produce the custom sections data. Only the custom section provided using a pipe-delimited list of custom section numbers will be syndicated.

Example configuration

...

```
<SpecType1004>
```

```
  <ExtensionItem  
    handler="Class:Xeno.Prodika.ExportExtension.XmlNodeCreationEx  
    tensibilityHandler,
```

```
    ExportExtension$ProdikaExtensibility" />
```

```
<ExtensionItem
  handler="Class:Xeno.Prodika.ExportExtension.XmlNodeCreationEx
  tensibilityHandler,

  ExportExtension$ProdikaExtensibilityItem" />

<ExtensionItem
  handler="Class:Xeno.Prodika.ExportExtension.XmlNodeCreationEx
  tensibilityHandler,

  ExportExtension$ProdikaCustomSections" />

<ExtensionItem
  handler="Class:CustomSectionXMLLib.handlers.CustomSectionExte
  nsibilityHandler,

  CustomSectionXMLLib$1000123|1000456|1000789" />

<ExtensionItem
  handler="Class:Xeno.Prodika.ExportExtension.XmlNodeCreationEx
  tensibilityHandler,ExportExtension$ProdikaCustomSections|
  CloseTag" />

<ExtensionItem
  handler="Class:Xeno.Prodika.ExportExtension.XmlNodeCreationEx
  tensibilityHandler,ExportExtension$ProdikaExtensibilityItem|
  CloseTag" />

<ExtensionItem
  handler="Class:Xeno.Prodika.ExportExtension.XmlNodeCreationEx
  tensibilityHandler,ExportExtension$ProdikaExtensibility|
  CloseTag" />

</SpecType1004>
```

...will result in the following XML being added to syndication:

```
<ProdikaExtensibility>
  <ProdikaExtensibilityItem>
    <ProdikaCustomSections>
      <CustomSection1000123>...custom section data... </CustomSection1000123>
      <CustomSection1000456>...custom section data... </CustomSection1000456>
      <CustomSection1000789>...custom section data... </CustomSection1000789>
    <ProdikaCustomSections>
  <ProdikaExtensibilityItem>
</ProdikaExtensibility>
```

...where **<CustomSection1000123>** is the name of the custom section, and
...custom section data... is the content of the custom section.

Search Properties Configuration Settings

Overview

The EQT Search Properties can be configured to show the desired order of search properties by modifying the file `EQTSearchablePropertyConfigs.xml`. The file sets the sequence of properties on the main search pages.

Configuration Sections

The configuration file contains the available options that can be re-sequenced on deployment. In a typical installation, these values can be modified to affect the sequencing of the search properties. Changing this file affects the entire system and all users' preferences.

To change the sequence, modify the value in **Bold** as shown below.

```
<Model modelName="Menu Item Specification">

    <SearchableProperty propertyId="SpecName" sequence="1"
    enabled="true" comment="" />
    <SearchableProperty propertyId="SpecStatus" sequence="2"
    enabled="true" comment="" />
    <SearchableProperty propertyId="SpecNumber" sequence="3"
    enabled="true" comment="" />
    <SearchableProperty propertyId="TaxonomyPlusChildren"
    sequence="4" enabled="true" comment="" />
    <SearchableProperty propertyId="Originator" sequence="5"
    enabled="true" comment="" />

</Model>
```

Do not modify any other values. Modifying anything else can cause the system to crash.

