



Agile Product Lifecycle Management

Product Interchange Administrator Guide

v9.2.2.6

Part No.E14192-01

January 2009

Copyright and Trademarks

Copyright © 1995, 2009, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. 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, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third party content, products and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third party content, products or services.

Preface

The Agile PLM documentation set includes Adobe® Acrobat PDF files. The [Oracle Technology Network \(OTN\) Web site](http://www.oracle.com/technology/documentation/agile.html) <http://www.oracle.com/technology/documentation/agile.html> contains the latest versions of the Agile PLM PDF files. You can view or download these manuals from the Web site, or you can ask your Agile administrator if there is an Agile PLM Documentation folder available on your network from which you can access the Agile PLM documentation (PDF) files.

Note To read the PDF files, you must use the free Adobe Acrobat Reader version 7.0 or later. This program can be downloaded from the [Adobe Web site](http://www.adobe.com) <http://www.adobe.com>.

The [Oracle Technology Network \(OTN\) Web site](http://www.oracle.com/technology/documentation/agile.html) <http://www.oracle.com/technology/documentation/agile.html> can be accessed through Help > Manuals in both Agile Web Client and Agile Java Client. If you need additional assistance or information, please contact [support](http://www.oracle.com/agile/support.html) <http://www.oracle.com/agile/support.html> (<http://www.oracle.com/agile/support.html>) for assistance.

Note Before calling Oracle Support about a problem with an Agile PLM manual, please have the full part number, which is located on the title page.

TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, 7 days a week. For TTY support, call 800.446.2398. Outside the United States, call +1.407.458.2479.

Readme

Any last-minute information about Agile PLM can be found in the Readme file on the [Oracle Technology Network \(OTN\) Web site](http://www.oracle.com/technology/documentation/agile.html) <http://www.oracle.com/technology/documentation/agile.html>

Agile Training Aids

Go to the [Oracle University Web page](http://www.oracle.com/education/chooser/selectcountry_new.html) http://www.oracle.com/education/chooser/selectcountry_new.html for more information on Agile Training offerings.

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle

does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

CONTENTS

Copyright and Trademarks	ii
Introduction	1
Functional Overview	1
Starting Agile Product Interchange.....	1
Product Interchange Process	2
Administration	5
Administration Tab.....	5
Users	5
Creating New Users	5
Searching For Existing Users	6
Updating User Information	6
Defining Roles	7
Mapping Roles	7
Organizations	8
Creating Organizations.....	8
Updating Organization Information.....	8
Party Relationships	9
Creating Party Relationships	9
Updating Party Relationship Information	10
User Types	10
Creating User Types.....	10
Security	13
Security Tab	13
Permissions	13
Creating New Permissions	13
Searching For Permissions	14
Updating Permission Information	14
Roles	15
Creating A New Role.....	15
Assigning Permissions to a New Role.....	15
Customizing Product Interchange	16

Parser Profile Administration.....	19
Key Concepts	19
Configuring Parser Profiles	20
Custom Part Attribute Mapping	21
Setting Recognition Lines and Testing.....	21
About Parser Profiles	22
Advanced Options	23
Module Names	24
Line Patterns	30
RecognitionLine.....	30
RecognitionRegExpLine	31
SkipLine.....	31
SkipRegExpLine	31
SkipRegExpLineStart	31
SkipLineStart	31
SkipLineEnd	31
SkipRegExpLineEnd.....	31
FirstLineToParse	31
LastLineToParse	32
TLAPartNumberLine.....	32
TLARevisionLine	32
TLATypeLine	32
TLADescriptionLine	32
TLAUOMLine.....	32
Component Management.....	33
Component Management Tab.....	33
Component Management Profiles	33
Data Source Profile	34
Creating a New Data Source Profile.....	34
Updating an Existing Data Source Profile	35
Selection of Data Source Fields for MPN Search Display.....	35
Mapping Data Source Attributes	35
Mapping Standard (Non-Compliance) Fields	35
Mapping of Partminer (Non-Standard) Attributes	36
Mapping of Total Part Plus (Non-Standard) Attributes	36
Weight_ppm	37
Standard Alias Search.....	37
Performing Manufacturer Alias Search.....	38
Adding A New Manufacturer/Manufacturer Alias.....	38

Merging Existing Manufacturers and Aliases	38
Part Search	39
Standard Alias Bulk Load	39
DataLoad Scripts	40
Guidelines to unpack and run this script	40
MFR and MPN Alias Table Format.....	41
MFR and MPN Alias Loader.....	42
Examples.....	42
Commodity Codes	43
MFR Code Loader	43
MFR and MPN Alias Getter	44
JDBC Data Source Configuration.....	45
Column Mapping	45
SQL for JDBC Data Source Profile for MPN Search against Agile 90	46
Updating the Compare AML's Data Source Configuration	46
Using Agile PC as a Data Source.....	47
Agile Import Preferences	49
Agile PC Import Profile	49
Selecting Import Preferences	49
Parsing and Validation Options	49
Business Rule Options	50
PG&C Integration.....	53
PG&C Integration	53
Import Configuration	53
PG&C Privileges Required for Declaration Import.....	53
PG&C Configurations Required For Declaration Import.....	54
PI Configurations For Declarations Import	54
Regular Expressions.....	57
Introduction	57
Regular Expression Syntax	58
Literals and Meta-characters	58
Build a Regular Expression	61
Testing a Regular Expression	62
Order of Precedence	62
Ordinary Characters	62

Single-Character Regular Expression Patterns.....	63
Special Characters	63
Non-Printable Characters	64
Character Matching	64
Bracket Expressions.....	65
Quantifiers	66
Anchors	67
Alternation and Grouping	69
Back References	70
Character Classes	71
EXAMPLES	72

Introduction

This chapter includes the following:

▪ Functional Overview	1
▪ Starting Agile Product Interchange	1
▪ Product Interchange Process	2

Functional Overview

The Agile Product Interchange solution enables users to rapidly identify any structural or parts related issues in BOMs/AMLs and resolve these issues in a systematic fashion.

The Product Translation function performs automated conversion of a wide range of text-based BOM/AML formats into user-defined format. The supported input formats include:

- MS-Excel, MS-Word, Text, HTML, PDX formats.
- Single/Multiple Levels, Single/Multiple Files.
- Delimited/fixed width formats, Complex rows/columns, Multi-line fields, Wrap-around fields.

It also performs comprehensive 'rules-driven' validation of the BOM/AML structure and provides an interactive framework for correcting errors.

The Product Cleansing function performs manufacturer name aliasing by converting manufacturer name variants into standard 'corporate' manufacturer names. It also performs part number aliasing by mapping 'dirty' incoming part numbers (manufacturer or internal) to parts numbers previously 'cleaned' in the organization. Over a period of time this knowledge base of manufacturer name aliases and part number aliases leads to more accurate parts information through out the PLM processes.

The Product Validation function enables users to confirm the accuracy of part information by matching against reference databases or parts catalogs. This also enables users to find alternate replacement parts where issues are detected.

The Product Enrichment function provides a framework for adding a breadth of attributes to the individual part / item records to make the product information more complete and usable throughout the product lifecycle. The additional product attributes can be sourced from one or more reference sources.

Starting Agile Product Interchange

To start Agile Product Interchange:

1. Start your browser.
2. Click the Agile Product Interchange bookmark, if one exists, or type the URL of the server

where Agile Product Interchange is installed.

3. Enter your user name. Your user name is not case-sensitive.

Important User Name and Password in Product Interchange must match the User Name and Password in Agile PLM.

4. Enter your password. Your password *is* case-sensitive.

You can change your password at any time by clicking the Profile link after you have logged into Agile Product Interchange.

5. Click Login button.

If you make a mistake, click Clear and retype your user name and password. The login process is complete.

6. Click Component Management tab.

You are now ready to use Agile Product Interchange.

Product Interchange Process

The broad capabilities offered by Agile Product Interchange for processing BOMs and AMLs can be executed by following a standard series of steps outlined in the following sections. Based on the business use cases, some of these steps may be skipped altogether or performed multiple times to achieve the desired goals.

The standard steps are as follows:

Submitting Input Files

Select the files to be processed and submit them into Agile Product Interchange.

Performing Bulk Edits

Once the files are submitted, the format translation is performed automatically and you are presented with a screen to edit specific fields in the input file as necessary.

Performing Structural Validation

Select from a pre-defined set of rules that can be used to validate the structure and the integrity of the input file. Once the validation has been performed, you are presented with a screen to interactively correct any errors found during the validation process.

Performing Part Search

Part number specific errors can be corrected by performing an online search for part numbers in one or more reference data sources connected to Agile Product Interchange

Resolving Manufacturer Names

Manufacturer name variants that are found in the input file are automatically aliased to standard manufacturer names. When entirely new variants are detected, you are provided an interface to create new aliases for automated re-use in the future.

Selecting Part Number Aliases

All the part number aliases that are currently available in the Agile Product Interchange knowledge base are displayed. You have the option to select specific aliases to be included in the file being processed.

Performing AML Adds/Deletes

After Manufacturer Name and Part Number aliasing has been completed, you are given the option to add or delete entries to the AML being processed. This can be done by either manually entering new part records from the user interface or by part number matching against reference ERP/PDM systems connected into Agile Product Interchange.

Validating Manufacturer Parts

All the part numbers in the file are validated in bulk against a reference source like a parts catalog.

Finding Alternate Parts

You can also search for alternate parts to replace parts with issues or to add new sources of equivalent parts.

Assigning Commodity Codes

Once all the parts information has been cleaned and validated, you can assign Commodity Codes to individual parts or part groups.

Exporting Processed Files

The processed file can be exported to Agile PC along with the automated creation of relevant change order (ECO/MCO). The processed file can also be exported as a PDX package or in MS-Excel format to allow easy import into downstream systems.

Administration

This chapter includes the following:

▪ Administration Tab	5
▪ Users	5
▪ Organizations.....	8
▪ Party Relationships.....	9
▪ User Types	10

Administration Tab

In the Administration tab, you can perform actions on Users, Organizations, Party Relationships, and User Types by selecting the applicable link. It is important to understand the relationships between some of these key objects within Product Interchange.

Users

Users are created and assigned Roles and Privileges allowing them access to the tabs, functions, and processes within Product Interchange.

- Users are connected to Organizations.
- Users are assigned Roles or User types that allow them to perform actions within the Product Interchange.
- These Roles or User types consist of one or more Permissions.

After logging into Product Interchange and accessing the Administration tab, you are presented with the User Search screen by default.

Creating New Users

To create a new user:

1. In the Admin tab, click Create New. The Create User page appears.
2. Select the user's organization from the Organization Name drop-down list.
3. Select the user type from the User Type drop-down list.
4. Type the following required information in their respective fields (if you leave any of these fields blank, the user is not added to the system):
 - First Name and Last Name - The new user's full name.
 - Address - The new user's address.

- Email - The new user's e-mail address.
- User Name - The new user's login ID/user name, which is not case-sensitive.
- Password - The new user's password, which is case-sensitive) and Confirm Password.

5. Click Create.

After the new user is created, you must assign roles to define the user's actions in the system. For more details, see [Mapping Roles](#) (on page 7).

Searching For Existing Users

To search for an existing user:

1. Select the search parameter from the drop-down list. The available fields are as follows:

- User ID (User identification number)
- First Name
- Last Name
- User Name (User's Product Interchange login ID/username)
- Organization Name

2. Enter the search characters in the Search field.

Product Interchange uses two wildcards:

- The asterisk (*) - The asterisk matches with everything.
- The percent sign (%) - The percent sign finds matches with any other characters. For example, %r% finds all results that contain the letter "r".

3. Click Search to display the search results.

You can modify the information for the displayed users. For more information, see [Updating User Information](#) (on page 6).

Updating User Information

You can modify any fields for existing users from the User Search Results page. The following is a general process for modifying User information.

To modify the information for an existing user:

1. Click on the User ID link of the user whose information you want to modify.

The Update User page appears.

2. The fields are activated, which allow you to either directly edit the field or make a selection from a list. Modify the necessary fields.

3. Click Update.

The changes are reflected during the user's next login.

4. Click the User ID to modify the user profile.

Defining Roles

Roles are used to define the User's actions in the system.

- Users must be assigned to at least one role in order to perform actions within Product Interchange.
- Users who perform similar functions can be grouped together into one role that defines the common requirements for access in the system.
- Users can be assigned to multiple roles, depending on the actions they needs to perform.

New users of Product Interchange may want to start with a basic set of roles.

Example role sets are shown below:

Administration

- dashboard_todoAllPackages.
- CreateProject.
- Administration.
- ParserAdmin.
- Security.

Users

- dashboard_todo.
- CreateProject.
- ProjectMember.
- CreateStdMfr.
- CreateRecipe.

Mapping Roles

To map roles to a User:

1. Click on the Map Roles link from the User Search Results page. The Map Roles to User page appears.
2. Select roles in the Available Roles list and use the right arrow to move roles to the Assigned Roles list.
3. Select the Update Roles for the User in Workflow Engine check box if the role is involved in the workflow.
4. When you are finished, click Map Roles.

To modify roles of the User:

1. Click the Back to User Search button to return to the user search to access additional user profiles.
2. Click on the Map Roles link of the user whose information is to be modified.

To delete Users:

1. Select the check box of the User to be deleted.
2. Click the Delete Checked button.

The User gets deleted.

Organizations

Organizations within Product Interchange serve multiple purposes:

- Identify and relate to the organization using Product Interchange.
- Identify and relate to the 3rd-party content providers whose data is being accessed by Product Interchange.
- Serve as the underlying technology behind how manufacturer aliases and manufacturer part aliases are managed within Product Interchange.

Note	Organizations need to be set up as new content sources are identified and leveraged in Product Interchange.
------	---

Creating Organizations

At least one Organization must be established for Product Interchange. Typically, there would be multiple organizations established - the corporate entity and a 3rd-party content provider.

To create the Organization:

1. In the Admin tab, click on the Organization link.
2. Click the Create New button.
3. Enter the Organization Name.
4. From the drop-down menu select the Organization Type.
Enter other information as desired.
5. Click the Create button.

The Organization is created successfully.

Updating Organization Information

To update the information, click on the Party ID link of the Organization. The following is a general process for modifying Organization information.

To update the information for Data Source Organization:

1. Click on the Party ID link of the organization you want to modify.
The Update Organization page appears.
2. Select from a drop-down list and enter details in the necessary fields.
The Supplier (for Import BOS) check box displays a list of suppliers present in Agile PLM. Every

Organization will have a unique supplier.

3. Check the Select Specification check box.
4. Click Update.

The Organization gets updated successfully.

To map specification:

1. A Configuration Specification link appears next to the Select Specification.
2. Click on the Configuration Specification link. The Select Specification for Partminer dialog opens.
3. From the Available Specification list on the left hand side, select a specification.
4. Click the right arrow. The selected specification gets listed on the Selected Specification list on the right hand side. It works even if specification is not selected.

Product Interchange will not add missing substances and substance groups for Selected Specification.

5. Click Save.

A message "Please remember to map the selected specifications to the appropriate data source fields by clicking on the 'Map Specification with Data Source Fields' pops-up.

6. Click OK. The specifications get mapped successfully.
7. Click on the Map Specification Data Source Field link.
8. In the Partminer DataSource Specs Map page, select a specification from the Selected Specification and a data source from the DataSource field.
9. Click Add Map. The specifications are mapped successfully.

To delete an Organization:

1. Select the associated check box of the Organization to be deleted.
2. Click the Delete Checked box.

The Organization gets deleted from the system.

Warning It is recommended that this permission be utilized with great caution.

Party Relationships

Party Relationships define the interaction between organizations or between Users and Organizations. The use of Party Relationships is optional depending on the use case defined for Product Interchange. The Party Relationship Tab displays the Party Relationships Search page by default.

Creating Party Relationships

To create a Party Relationship:

1. In the Admin tab, click on the Party Relationship link.
2. In the page that appears, click the Create New button.
3. Select the Subject Party from the drop-down menu. The potential subject parties will include the

lists of all users and organizations established in the system.

4. Select the relationship type from the “is” drop-down list. This list includes relationships such as Manufacturer Alias and Supply Chain Partner.

Note	Manufacturer Alias values are created in other means within Product Interchange, and that relationship would not be created here.
------	---

5. Select the Object Part from the drop-down list. This again will include the lists of all users and organizations established in the system.
6. Select the Sponsor Party of the Relationship from the drop-down list. This list will also include Users and Organizations.
7. Click on the Party ID link of the Organization to be modified.

Updating Party Relationship Information

You can modify Party Relationships from the Party Relationship Search page. The following is a general process for modifying user information.

To update the information for an existing user:

1. On the Update Party Relationship page, click on the Party Relationship ID link you want to modify.
2. Update or edit the necessary fields.
3. Click the Update button.
4. Click Party Relationship ID.

To delete Party Relationships:

1. Select the check box of the Party Relationship you want to delete.
2. Click the Delete Checked button.

The Party Relationship will be successfully deleted from the system.

User Types

User Types allow the Product Interchange system administrator to create “super roles” which, when assigned to individual users, assign both functional roles as well as party relationships. These user types are an extension of normal role creation. The User Type link on the Admin tab, displays the User Type Search Page by default.

Creating User Types

To create the User Type:

1. In the Admin tab, click on the User Type link.
2. Click the Create New button.
3. Enter the User Type.
4. Enter the Description.

5. Click the Create button.
6. After the User Type is created, add roles to it by moving roles from the Available Roles pane to the Assigned Roles Pane.
7. Click Map Roles.
8. The Roles get mapped to the User type successfully.

Warning There is no functionality to delete User Types once created. By removing all roles from the User Type, the User Type is essentially deleted from the system.

This chapter includes the following:

▪ Security Tab.....	13
▪ Permissions	13
▪ Roles.....	15

Security Tab

The Security Tab allows Product Interchange System Administrators manage Roles and Permissions that allow users to perform actions in accordance with the use cases being deployed. From the Security tab, you can perform actions on Permissions and Roles by selecting the applicable link.

Permissions

Permissions would allow roles and the users assigned those roles the ability to access Product Interchange validation projects. While Product Interchange has a robust set of out-of-the-box permissions, there may be instances where additional Permissions may need to be created.

It is recommended that customers consult with Agile prior to deploying additional permissions. After logging into Product Interchange and accessing the Security tab, you are presented with the Permission Search screen by default.

Creating New Permissions

To create a new permission:

1. Click the Create New button. The Create Permissions page displays.
2. Select a value from the Business Process Name drop-down list. The valid entries include:
 - Administration
 - Agile Import Preferences
 - Parser Admin
 - Security
 - Seed Process
 - UI Setup
 - Validation Project
3. Select the permission type from the Permission Type drop-down list. The valid entries include:
 - COMMAND

- DATAACCESS
 - TAILOREDVIEW
 - URL
4. Enter the Permission Name, Permission Description, Display Name, which are required fields. Enter a value for Permission URL only for permissions specific to a URL permission type.
 5. Click the Create button.

Warning Agile strongly discourages the addition of new permissions without a thorough understanding of the Product Interchange product and its capabilities.

Searching For Permissions

To search for an existing permission:

1. Select the search field from the drop-down list. The available fields are as follows:
 - Permission ID
 - Permission Name
2. Enter the search characters in the Search field.

Product Interchange uses two wildcards:

 - The asterisk (*) - The asterisk matches with everything.
 - The percent sign (%) - The percent sign finds matches with any other characters. For example, %r% finds all results that contain the letter “r”.
3. Click Search to display the search results.

You can also update the information for the displayed users. For more information, see [Updating Permission Information](#) (on page 14).

Note The search parameter is not case-sensitive.

Updating Permission Information

You can update any fields for existing permissions from the Permission Search page. The following is a general process for modifying permission information.

To update the information of an existing user:

1. Click the Permission ID link of the permission whose information you want to modify from the Permission Search page.

The Update Permissions page appears.
2. The fields are activated. Modify the necessary fields. Either directly edit the field or make a selection from the list.
3. Click the Update button.

The permission changes are effective immediately. Click the Permission Id to modify the permission.

Warning It is recommended that this type of change be made while users are not actively engaged on the system.

Roles

Roles are used to define the user's actions in the system. Users who perform similar functions can be grouped together into one Role that defines the common requirements for access in the system.

Users must be assigned to at least one Role in order to perform actions within Product Interchange. A user can be assigned multiple roles, depending on the actions he needs to perform. Similarly, multiple Permissions may be added to a given Role in order to provide required functionality.

Creating A New Role

To create a new role:

1. Click the Create New button from the Role Search page.
2. Select the Role Type from the drop-down list. The role types include:
 - Dataaccess
 - URL
 - Workflow
3. Enter the Role Name and Role Description.
4. If the Role Type is URL, enter a valid URL for a screen within Product Interchange.
5. Enter the Display Name for the role.
6. Click the Create button.

Assigning Permissions to a New Role

To assign permissions to a new role:

1. To perform a search for the new role, click on the Map Permissions link.
The map Permissions to the Role Security /Administration page appears.
2. Select roles in the Available Roles list and use the right arrow to move roles from the Available Roles list to the Assigned Roles list.
3. Select the Update Roles for the User in Workflow Engine check box if the role is involved in the workflow.
4. The results of the search will display the new role just created.
5. Click on the Map Permissions link to assign permissions to this role.
6. Move over the desired Permissions from the Available Permissions pane to the Assigned Permissions pane.
7. When complete, click the Map Permissions button.

The results of this search will include the new role just created.

To delete Roles:

1. Select the check box of the Roles to be deleted from the Role Search page.
2. Click the Delete Checked button.

Note If a Role has been assigned Permissions, the Role cannot be deleted until the Permissions have been removed.

Customizing Product Interchange

Product Interchange allows system administrators to create new user look-and-feel features based upon the use case being deployed and the needs of the users performing use case functions. One feature that is particularly useful is the ability to create a new set of roles and permissions creating a custom tab to support a specific use case.

To create a new tab:

1. Create a permission of type URL, identifying the tab name as required.
2. Create a role of type URL, leveraging the permissionID of the permission created in step 1.
3. Assign specific permissions to the URL role, which provide the links available to that tab.
4. Assign the role to users as required.

See the following screen shots for examples.

The new tab being created will be called MikeTab.

The screenshot shows a web form titled "Update Role". At the top, it states "Fields with * are required. Maximum input size is 255 characters for all fields." The form contains the following fields and controls:

- * Role Type:** A dropdown menu with "URL" selected.
- * Role Name:** A text input field containing "MikeTab".
- * Role Description:** A text input field containing "MikeTab".
- Role URL:** A text input field containing "Portal.fma?tab=MikeTab&permissionId=112305".
- Display Name:** A text input field containing "MikeTab".

At the bottom of the form, there are three buttons: "Update", "Reset", and "Create New".

5. The Role URL references the permissionID for the permission MikeTab.

6. In this example, the new tab, MikeTab, will contain administrative functions for managing validation projects and component management profiles. The role MikeTab must be assigned to user(s).

The screenshot displays a web application interface for 'MikeTab'. At the top, a blue navigation bar contains the following links: Report Settings, MFR, All Projects, Agile Import Profiles, CM Profile, and Search. Below the navigation bar, the main content area is titled 'Default Report Settings'. It contains two settings with radio button options: 'Include Alternates in Reports:' with 'Yes' selected and 'No' unselected; and 'Apply Commodity Code Filter in Reports:' with 'Yes' selected and 'No' unselected. At the bottom of the settings area, there are two buttons: 'Reset' and 'Submit'. A mouse cursor is visible over the 'Reset' button.

Note The new tab is the only tab (role) that this particular user has access to.

Parser Profile Administration

This chapter includes the following:

▪ Key Concepts	19
▪ Configuring Parser Profiles	20
▪ About Parser Profiles	22
▪ Advanced Options	23
▪ Line Patterns	30

Key Concepts

Product Interchange performs automated conversion of a wide range of text-based BOM/AML formats into user-defined format. To do this, Product Interchange parses the BOM/AML file and recognizes, in an automated manner, each part attribute in the file, regardless of the file type or the format of the input BOM/AML. Currently, Product Interchange parser can read file formats including tab delimited and fixed width, and file types that include text, MS-Word documents, PDX, MS-Excel, and HTML.

Product Interchange parsing is based on the concept of a profile. A Parser Profile is a set of rules that tells the parsing engine on how to look for the part attributes in a given file. It includes:

- **Recognition Line:** Each profile is associated with a particular file format. This association is done by storing a unique string pattern in the profile that exists only in the files of the specific format. When the BOM/AML is submitted for parsing, Product Interchange will use the first matched pattern in the recognition line to select the profile for parsing.
- **Part Attribute Mapping:** Each profile contains information on extracting part attribute information from the specific file format. This will be discussed in more detail in later sections.
- **Advanced Options:** Product Interchange has the capability to parse a wide variety of file formats. This is done by additional post-processing functions, where required.

Product Interchange parsers significantly leverage regular expressions. Most BOM and AML files with low to moderately complex formats can be parsed without requiring regular expressions. To parse more complex formats, you have to gain an understanding of regular expressions. For more details see [Appendix - Regular Expressions](#) (see "Regular Expressions" on page 57) included in the Appendix.

In order to configure a Parser Profile, you have to identify a specific, repeating pattern in the file. This pattern will include the Part Number and all the attribute information associated with that part number(including attributes like part description, quantity, reference designators, etc) as well as associated manufacturers and manufacturer part numbers (if any), etc. Each instance of this repeating pattern is called a block. The Parser will look for information from the profile to understand how to extract part attribute information from the block.

Configuring Parser Profiles

The Parser Admin tab provides the Product Interchange system administrator with capabilities to create and manage parser profiles, which enables Product Interchange to recognize and process BOM and AML files in virtually any format. Product Interchange provides many pre-configured BOM and AML formats. In the process of receiving or creating BOM/AML formats, you may need test if a pre-configured profile exists, and if not, to create and test a new profile.

The Parser Admin tab displays the Parser Profile Search by default.

To create a new Parser Profile:

In the Parser Admin tab, the Parser Profile Search is displayed.

1. Click the Create New button.
2. Enter Parser Profile Name.
3. Click Save.

To define file attributes:

1. Click the Parser Admin tab.
2. Select the delimiter for the file.
 - For fixed width files, do not set the delimited tab.
 - For Excel files, set Delimiter to TAB
 - For files separated by other characters like '|', select Other.
3. Add the delimiting character in the Attribute Value text box.
4. Click the Add button.

To set other file attribute information:

1. If the part number is repeated in the file set PartNumber/Rev Unique to NO.
2. If the levels in the BOM are represented by character "." instead of a number, then set Level Type to COUNTCHAR.
3. If the file requires a manual input of the Top Level Assembly, then set Require TLA to YES.
By default, the profile assumes that the file is a BOM/AML.
4. Only if the file is BOM or an AML select File Type, and configure the appropriate value.

To configure part attribute mapping for tab delimited files:

The parsing engine needs to understand how to configure each part attribute from the file. For tab-delimited files, the fields in the profile will be mapped to the column numbers.

1. For each attribute, enter the column number where it exists in the file.
2. If the attribute does not repeat for a given part number, enter the second co-ordinate as 1.
3. If the attribute is the first column in the tab-delimited file, enter the second and the third co-ordinate as 0 and 1, respectively.

4. Click Save.

To configure part attribute mapping for fixed width files:

For fixed width format files, there are no tabs, or commas to delimit part attributes. These part attributes span within fixed width character columns.

For example, a Part Number may span from character column 10 through 41. In such a case, the profile Part Number will be set to 10-41.

Custom Part Attribute Mapping

Parser administration provides capabilities for the profile creator to capture custom part attributes.

To capture a custom part attribute:

1. In Add Additional Fields, type field name and their mapping value. Use the same notation for mapping values as used in the Map Fields form.
 - If the custom attribute is a part number attribute, then add the prefix BOM: to the attribute name.
 - If the custom attribute is a manufacturer part number attribute, then add the prefix AVL: to the attribute name.
2. Click Add.

To convert file types to text:

All files need to be converted to text before the core engine can begin parsing the file.

1. Select the file, set the sequence and click add.
 - To convert HTML files to text, select HtmlToTxtConverter and set Sequence to 2.
 - To convert PDX files to text, select PdxToTxtConverter and set Sequence to 2.
 - To convert DOC files to text, select DocToTxtConverter and set Sequence to 2.
2. Click Add.

The Advanced Options feature enables the profile creator to override the base behavior with advanced function and chain them using the sequence field.

Setting Recognition Lines and Testing

To set recognition and skip lines:

For the parser to automatically select the correct profile, it needs to associate a signature with the profile. This signature is a string pattern that should be the same for all files of the same format. Most often, this signature is the file header.

In addition, the parser needs to know which lines to ignore while parsing for part attributes. As a rule, the file header should always be ignored. In addition, there may be page headers (that also need to be ignored) to be configured.

1. Select Recognition Line from Line Patterns. Copy-paste the file header in the text box.
2. Click Add.
3. Select Skip Line from Line Patterns. Copy-paste the file header and other lines that need to be

ignored in the text box.

If there are multiple lines to be ignored, follow step 3, 4 for each line.

4. Click Add.

The Profile creator can use regular expressions to identify patterns that need to be skipped or used for profile recognition.

To test the recognition line:

1. In the Parser Admin tab, click on the Test Recognition link.

2. Click Browse.

Select the file for which the profile has been created.

3. Select Open.

4. Click Recognize the Parser.

Verify that the correct profile is returned.

If no profile is returned, it means that the recognition line in the profile was either not set, or was different than the one in the BOM/AML file.

To test the file parsing:

Once the recognition of the profile has been verified, ensure that the BOM/AML file gets correctly parsed. The Profile creator should verify the following things:

- Part attributes are correctly mapped.
- All part attributes are mapped.
- All junk lines have been ignored.

To test the Parser Profile:

1. In the Parser Admin tab, click on the Parse link.
2. Click Browse. Select the file for which the profile has been created.
3. Select Open. For Parser Profile, select the name of the profile that is to be tested.

Note	The profile name selected should correspond to the format in file selected in Step 3.
------	---

4. Click Parse. Check for errors during the parsing.
5. Click on the Text format link at the bottom of the screen. This file contains the normalized parsed data.

Open the file in Microsoft Excel. The part attributes are delimited by the character '|'. Check if the fields were parsed accurately.

About Parser Profiles

Product Interchange does not apply version changes to Parser Profiles. It is strongly suggested that profiles be backed-up before any significant changes are applied to it. To create backup copies of the Parser Profile, Product Interchange provides the ability to export and import profiles using XML

files.

To export Parser Profile:

1. Click on the Export Profile link.
Select the Profile Name from the drop-down list.
2. Select the Export to XML File button.
Save the exported file.

To look-up existing Parser Profile:

1. In the Parser Admin tab, click on the Search Profiles link.
2. Enter the profile name to search for profiles. Use % for wildcard search.
3. Click the Search button.

To delete existing Parser Profile:

1. From the Parser Admin tab, click on the Search Profiles link.
2. Enter the profile name to search for profiles. Use % for wildcard search.
3. Click the Search button.
4. In the Profile Search Results, select check box of the profile that is to be deleted.
5. Click Delete Checked.

To edit Parser Profile:

1. In the Parser Admin tab, click on the Search Profiles link.
2. Enter profile name to search for profiles. Use % for wildcard search.
3. Click the Search button.
4. In the Profile Search Results, click on the Profile Name link.
5. Edit the profile changes on the profile detail screen.
6. Click the Save button.

Advanced Options

To associate Advanced Options with specific Parser Profiles, select the Advanced Option and input a Sequence ID during the creation of a new parser profile. The Sequence ID is the order in which the Advanced Option will be executed. This value must be two or greater.

Some of the Advanced Options perform a fixed task and do not require any additional configuration. Others require assignment of specific parameters, which identify which field needs to be processed, what is the required processing and what is the desired output.

Module Names

ExcelToTxtConverter

Converts an Excel input file to a tab-delimited text file, suitable for parsing. This option must be selected for parsing any excel based input file.

PDXToTxtConverter

Converts a PDX input file to a tab-delimited text file, suitable for parsing. This option must be selected for parsing any PDX based input file.

HTMLToTxtConverter

Converts a HTML input file to a tab-delimited text file, suitable for parsing. This option must be selected for parsing any HTML format input file.

DocToTxtConverter

Converts a Doc input file to a tab-delimited text file, suitable for parsing. This option must be selected for parsing any Doc format input file.

AvlFieldExtractor

Extracts information from one AVL column and splits this information into 2 or more AVL columns.

For example:

Attribute Name	Vendor Part #	Vendor Name
Attribute Value	1, (\S+)\s+	2, \S+\s+(.*)

The program will extract information from an AVL field, which has both Vendor Part Number and Vendor Name merged into one field. The extraction is done by a regular expression, which separates the portion of this field containing only the Vendor Part Number and marks it for populating the Vendor Part Number field in the output.

This is followed by the second regular expression, which separates the portion of this field containing only the Vendor Name and marks it for populating the Vendor Name field in the output.

The numbers 1 and 2 indicate the order of the parameter names.

BomFieldReplacer

Replaces specified characters in a specified field with a specified value.

For example:

Attribute Name	Ref Des 1
Attribute Value	# "

This program will replace the # in Ref Des 1 field with " and put it back into the Ref Des 1 field. | is

used as delimiter.

There are 2 special inputs: \s represents space, and \t represents tab.

This option takes only 2 parameters. The source and destination field are always identical. Multiple instances of BomFieldReplacer can be sequenced to replace multiple characters.

BomFieldExtractor

Extracts information from a range of text in one column and splits it into 2 or more columns.

For example:

Attribute Name	Attribute Value: (Sequence, Regular expression)
RefDes 1	1, .* (\\.*\\d+PIN[\\s*],JUMPER[S]\\.*).
BOM: RefDes Comment	2, .* \\.* \\d+PIN[\\s* ,] (JUMPER[S]\\.*).

The program will extract information from Ref Des 1 field, which has both Ref Des values and Ref Des Comments merged into one field. The extraction is done by a regular expression, which separates the portion of this field containing only the Ref Des value and marks it for populating the Ref Des 1 field in the output.

This is followed by the second regular expression, which separates the portion of this field containing only the Ref Des comment and marks it for populating the BOM:Ref Des Comment field in the output.

The numbers 1, 2 indicate the order of the parameter names.

BomFieldDiverter

Diverts value from a specified input field to one of 2 specified output fields based on matching a specified condition.

This option takes 3 inputs.

For example:

Description 2 Yes No (\\s\\D{1,2}\\d+_*\\d*) Ref Des 1 Description 2
--

If the input field Description 2 matches the regular expression, then it is diverted to Ref Des 1, otherwise it is put into Description 2.

BomPartDetailMerger

Post processor merges BOM part detail records based on unique part numbers.

For each part number except those filtered out from the parameter list, all records in the PM_BOM_PART_DETAILS table having the same level and the parent will be merged together inside a single record.

DuplicatedVendorsRemover

Removes redundant or duplicate AVL records by looking at AVL data for duplicates.

This option takes no parameter.

FieldMerger

Merges multiple fields into one field.

Limitation: This can merge into one field only. Also BOM fields can only be merged into one BOM field, and AVL fields can only be merged into one AVL field.

FirstVPNAsPN

Uses first vendor part number as the item (customer) part number for each record.

This is used where a BOM (or manufacturer parts list) has no item (customer) part numbers, but has MPN-MFR records at a single level. The program populates the first vendor part number as the item (customer) part number for each record in the BOM.

By definition, this can be invoked for any BOM that comes with no customer part numbers and can use the first vendor part number as the customer part number. Thus, if Part Number field does not exist, use the first Vendor Part Number as the Part Number.

While parsing, if TLA Part Number has to be provided by the user, then 'Require TLA' should be selected to Yes in the parser profile.

It takes no parameter.

PartNumberCleaner

Removes special characters from the part numbers.

Sometimes the part number has special characters that need to be stripped away.

For example: If the part number field contains the value " - P4523" and the character "-" is not needed, then PartNumberCleaner can be used to strip this. This program uses whitespace as a tokenizer to tokenize the string and rejects the short string while keeping the long string.

It takes no parameter.

PreviousFieldRetriever

Substitutes the value of a field in the current block by using the value from the previous block.

For example:

Attribute Name	Bom:Level
Attribute Value	No.

The Attribute Name is the field name and the Attribute Value indicates whether to force substitution even if the current block value is non-empty. The program will find out previous block's level number and put it into the current block if the current block's level is empty.

QuantityUOMSplitter

Splits Quantity & UOM from one merged field into two separate fields.

This program is used to split the composite Qty/UOM data into Qty and UOM and update the PM_BOM_PART_DETAILS table for a given PM_BOM_FILE_ID.

This should be sequenced as a post processor after standard parsing.

It takes no parameter.

QuantityCalculator

Calculates the quantity by counting reference designators.

This program updates the quantity column for a given BOM File by counting the number of commas in the Reference Designator field. The data in this field is supposed to be in its expanded form.

It takes only one parameter: QuantitySource.

Possible values for Quantity Source are:

Qty □ Quantity is taken from Qty only.

RefDes □ Quantity is taken from RefDes only.

QtyAndRefDes □ Quantity is taken from Qty first, if Qty is 0, then quantity is calculated from RefDes.

RefDesAndQty □ Quantity is calculated from RefDes , if that is 0, then Quantity is taken from Qty.

RefDesCleaner

Removes special characters from Reference Designator field.

This program looks for the following characters: “:”, “&”, “(”, “)” and converts the reference designators into cleaner format.

The result will look like R12,R45-R23,R4

It also removes any redundant “ - ”, that it finds in this process.

It takes no parameter.

RefDesExpander

Expands Reference Designators to individual items separated by commas.

In cases where the Reference Designators are spread across two rows (i.e. Ref Des 1 and Ref Des 2) this option will Expand Ref Des 1 after appending Ref Des 2 to Ref Des 1 and setting Ref Des 2 to Empty String.

It takes no parameter.

RefDesFilter

Filters out Reference Designators based on the parameters supplied.

For example: It can be used to reject any pure numerical reference designators and accept those that contain at least one alphabetical letter in it. The profile would be as follows:

Attribute Name	Attribute Value
Ref Des 1	^[\\,\\d\\-]+\$
Ref Des 2	^[\\,\\d\\-]+\$

This profile instructs the sub-module to filter out any reference designators that contain only digits, dash, and commas.

VendorDataCopierA

Extracts Vendor Part Number & Vendor Name when contained in a single column.

When multiple Vendors and Vendor Part Numbers are stored in the same field this option provides a mechanism to extract unique fields and populate them in separate columns.

For example:

Attribute Name	Vendor Part#	Vendor Name
Attribute Value	#	#

Vendor Part number and Vendor Name are the columns where the composite data is present and needs to be extracted.

The separator pattern string specifies how to split this composite string into a string array of unique components using regular expression techniques.

The program will tokenize the field in Vendor Part Number and Vendor Name and construct two rows of vendor data populated with the appropriate values. Thus, it will separate the data that follows # character into other columns, say into more vendor part number columns or vendor name columns.

For example:

Vendor Part#	Vendor Name
VP#VP1	VN2# VN2

This will result in 2 rows of vendor data (VP1, VN1) (VP2, VN2)

This works with AVL data only.

VendorDataCopierB

Converts Vendor Name, Vendor Part Number information from horizontal columns to rows.

This program is designed to handle the scenario where multiple vendor data records span horizontally into columns rather than vertically into rows. It handles up to five pairs of vendor names/part numbers.

Based on this, the class will construct two rows of vendor data.

Customer Profile Name	Profile value
AVL: Custom1	Vendor Part Number
AVL: Custom2	Vendor Name
AVL: Custom3	Vendor Part Number
AVL: Custom4	Vendor Name
etc	etc

This works with AVL data only.

VendorDataDuplicater

Duplicates Vendor Name from one row to other rows.

Sometimes only one row of vendor data has the vendor name and other rows do not have it. This program copies the vendor name from one row to other AVL rows that do not have the vendor name.

Attribute Names field has the column that needs to be duplicated, Attribute Values field do not matter.

For example: Attribute Names: "Vendor Name".

VendorDataMerger

Merges vendor data from multiple rows into one vendor record.

This program merges the AVL data and vendor data into a single record. This is needed when vendor data appears in multiple rows, but actually belongs to the same vendor entry.

Attribute Names contains the column, the Attribute Values contains the delimiter.

This works with AVL data only.

VendorDataSplitter

Splits AVL data into two or more vendor data records.

This option is useful for splitting vendor data containing different attributes that were stored in the same column (or position) in the BOM. It splits the composite vendor data in a specific column into the separate columns containing unique vendor attributes. The Delimiter field should always be the last parameter in the profile.

Based on this, the program will construct two rows of vendor data, using the delimiter to split the fields and put them into appropriate vendor data records.

This works with AVL data only.

VendorPartNumberReplacer

Replaces specified string in vendor part number with Item (customer) part number.

This program replaces the specified vendor part number with item (customer) part number.

For example:

Attribute Name	Vendor Part#	Vendor Part#
Attribute Value	"See Description"	"See Reference"

The program will replace the strings "See Description" and "See Reference" with item (customer) part number corresponding to the Vendor Part Number.

TLAHeaderMover

Moves the block header to the string represented by the Regular expression.

This situation will arise when the TLA information block area lies in the bottom of the BOM instead of being in the beginning. The regular expression string defined in the parameter, signals the beginning & end of the TLA header block.

The defined parameters are:

1. patternStringStart
2. patternStringEnd

The Strings defined in the pattern signals the beginning and end of TLA header block.

3. patternREStart
4. patternREEnd

The REgexp string defined in the pattern, signals the beginning and end of the TLA header block.

i.e. -patternREStart []+

LineHorizontalMover

Moves lines horizontally when it matches regular expression.

This program will scan the whole file and whenever the line matches the regular expression, it will be shifted right by that many unit characters.

The unit value can be: \t, \s or other characters. \t is tab. \s is white space. Default is \t.

Line Patterns

Sometimes, it is not possible to profile recognition line, skip line, or TLA attributes with specific strings or column locations. In those cases, Product Interchange provides a capability to use regular expression to identify the patterns for picking up these fields.

RecognitionLine

The unique string of text used to differentiate this BOM from all other BOMs. Future BOMs of this type must contain this exact string.

RecognitionRegExpLine

A regular expression that identifies a row of data in the BOM. Future BOMs of this type must match the regular expression.

SkipLine

Lines of text to skip. Every line in the BOM that contains the specified text will be skipped. Regular expressions may not be used. SkipLine takes precedence over all other Line Patterns except Recognition Line. Therefore, do not use SkipLine to skip lines where you also use TLAPartNumberLine to extract data.

SkipRegExpLine

Same as "SkipLine" but allows the use of regular expressions.

SkipRegExpLineStart

Same as "SkipLineStart" but allows the use of regular expressions and requires a matching "SkipRegExpLineEnd".

SkipLineStart

First line of text block to skip. Requires a matching "SkipLineEnd" to mark the end of the skipped block. Every line in the BOM between "SkipLineStart" and "SkipLineEnd" will be skipped. Regular expressions may not be used. SkipLineStart takes precedence over all other Line Patterns except RecognitionLine. Therefore, do not use SkipLineStart to skip lines where you also use TLAPartNumberLine to extract data.

SkipLineEnd

Last line of text block to skip. Requires a matching "SkipLineStart" to mark the beginning of the skipped block. Every line in the BOM between "SkipLineStart" and "SkipLineEnd" will be skipped. Regular expressions may not be used. SkipLineEnd takes precedence over all other Line Patterns except RecognitionLine. Therefore, do not use SkipLineEnd to skip lines where you also use TLAPartNumberLine to extract data.

SkipRegExpLineEnd

Same as "SkipLineEnd" but allows the use of regular expressions and requires a matching "SkipRegExpLineStart".

FirstLineToParse

The numeric line number where parsing should begin or a text string that, when found, signifies the line to start parsing.

LastLineToParse

The numeric line number where parsing should end or a text string that, when found, signifies the line where parsing should end.

TLAPartNumberLine

LinAssembly Number:es that contain the Top Level Assembly Part Number. Regular expressions are used to extract the part number. For example, if the BOM contains a line such as this:

Assembly Number: 4424-1121

The TLAPartNumberLine regular expression would be:

Assembly Number: \s+(\S+)

TLARevisionLine

Lines that contain the Top Level Assembly Revision. Regular expressions are used to extract the revision.

TLATypeLine

Lines that contain the Top Level Assembly Type. Regular expressions are used to extract the type.

TLADescriptionLine

Lines that contain the Top Level Assembly Description. Regular expressions are used to extract the description.

TLAUOMLine

Lines that contain the Top Level Assembly UOM. Regular expressions are used to extract the unit of measure.

Component Management

This chapter includes the following:

▪ Component Management Tab	33
▪ Data Source Profile.....	34
▪ Mapping Data Source Attributes.....	35
▪ Standard Alias Search.....	37
▪ Standard Alias Bulk Load	39
▪ DataLoad Scripts	40
▪ JDBC Data Source Configuration	45

Component Management Tab

In the Component Management tab, you can perform various administrative actions. These actions include configuration of content recipes and content sources, searches for manufacturer and aliases and parts, and mass population of manufacturers and their associated aliases.

After logging into Product Interchange and accessing the Component Management tab, you are presented with the Submit BOM/AML File(s) screen by default.

The CM Profile provides a means to manage “recipes” for accessing external part content. These recipes are created as part of the Component Management process. It is not intended that even administrative users utilize the CM profile to create or modify recipes.

Note This functionality should only be used to delete recipes which are no longer required.

Component Management Profiles

To search for an existing profile:

1. Select the search field from the drop-down list. The available fields are as follows:
 - Name
 - ID
2. Enter the search characters in the Search field.
 Product Interchange uses two wildcards:
 - The asterisk (*) - The asterisk matches with everything.
 - The percent sign (%) - The percent sign finds matches with any other characters. For example, %r% finds all results that contain the letter “r”.
3. Click Search to display the search results. The Bounce Profile Search screen displays results.

To delete an existing profile:

1. Select the check box of the profile to be deleted.
2. Click the Delete Checked button.

The profile gets deleted.

Data Source Profile

The Data Source Profile is the means by which the 3rd-party content providers are configured within Product Interchange, and where default parameters, which drive behavior within the API are set. This is a critical feature that makes the Component Management process work effectively.

Creating a New Data Source Profile

To create a new Data Source Profile:

1. Click the Create New button from the Data Source Profile Search page.
2. Enter the Profile Name and the URL for the external content source.
3. Enter the User Name and Password for the login account of the external content source.
4. Enter the Driver. This is applicable to JDBC interfaces to external content.
5. Enter the default Match Record Limit. This is an integer value that determines how many matching part records will be pulled from the content source during a part matching process.
6. Enter the Account.
7. Select the Source Type from the drop-down list. The list includes the following, but in all cases, use the Reference_Primary:
 - Reference_Primary
 - Reference_Secondary
 - Reference_Differential
8. Select Organization from the drop-down list.
9. Enter the SQL scripts as required.
 - The first SQL pane is for the extraction from an external source.
 - The second SQL pane is for the translation and manipulation of Mfr Names.
 - The third SQL pane is for translation and manipulation of CPN (customer part numbers or internal part numbers).

They would typically be required only when interfacing with an internal data source, such as a data mart or an ERP system. They would not be required for interface with a 3rd-party content provider.

10. Click the Create button.

Updating an Existing Data Source Profile

To update an existing Data Source Profile:

1. Perform a search for the profile, select and click on the Data Source ID link that needs to be updated.
2. Make necessary modifications to the Data Source Profile from the Update Data Source Profile screen.
3. Click the Update button to complete the modifications.
This search will display the updated Data Source Profiles.
4. To display the profile parameters, select and click Data SourceID profile.

To delete Data Source Profiles:

1. Select the check box of the Data Source Profile to be deleted.
2. Click the Delete Checked button.

Selection of Data Source Fields for MPN Search Display

To select Data Source fields for MPN Search display:

Log in to the Product Interchange application as admin/agile

1. Click the Component Management tab.
2. Click the Data Source tab.
3. Enter * in the text field to display all data sources.
4. Click on the Config Attributes link against the Data Source (i.e. PartMiner) in the search results page.
5. Select the Data Source (i.e. PartMiner) attributes to be displayed in the MPN Search results page from the Left Hand Side Multi-Select box. Use the right arrow to move the Data Source from the Left Hand Side Multi-Select box to the Right Hand Side Multi-Select box
6. Once you are done with moving the attributes, click Save to commit the mapping.

Mapping Data Source Attributes

Agile Product Interchange allows the System Administrator to map the attributes retrieved from specific content sources to selected fields in Agile PLM. It also allows the Administrator to select fields that should be displayed on the 'MPN Search' screen in Agile Product Interchange.

Mapping Standard (Non-Compliance) Fields

To map standard (non-compliance) fields to Agile PLM fields:

Log in to the Product Interchange application as admin/agile

1. Click the Component Management tab.
2. Click the Data Source tab.

3. Enter * in the text field to display all the data sources.
4. Click on the Map Agile Fields link against the Data Source (i.e. PartMiner) in the search results page.

You will see two drop-down menus:

- Enables selection of Data Source (i.e. PartMiner) fields.
- Enables the selection of Agile PLM fields.

You can create the mapping of the above two sets of fields.

5. Click the Save button to commit the mapping.

Note	This mapping is at the Manufacturer Part level and typically includes attributes related to Obsolescence, Part Status, etc retrieved from external data sources.
------	--

Mapping of Partminer (Non-Standard) Attributes

To map Partminer non-standard attributes to Agile fields:

Log in to the Product Interchange application as admin/agile

1. Click Component Management tab.
2. Click Data Source tab.
Enter Partminer in the text field.
3. Click Search. The search will result in the display of Partminer data source profiles.
4. Click on the Map Material Attribute link. The Agile Field map page appears.
5. Select a data source field, from the Data Source Field drop-down list.
6. Select a field, from the Agile Fields drop-down list.
7. Click Add Map.
8. Click Save to commit the mapping.

Note	Mapping of Partminer non-standard attributes to Homogeneous Material Declaration Manufacturer Part Composition Attributes, IPC Manufacturer Part Composition Attributes and Manufacturer Part Composition Substance Attributes enabled.
------	---

Mapping of Total Part Plus (Non-Standard) Attributes

To map Total Part Plus non-standard attributes to Agile fields:

Log in to the Product Interchange application as admin/agile.

1. Click Component Management tab.
2. Click Data Source tab.
3. Enter * in the text field.
4. Click the Search button.

The search will result in the display of all data source profiles.

5. Click on the Map Material Attribute link. The Agile Field map page appears.
6. Select Sub_Mass PPM from the Data Source Field drop-down list.
7. Select a field from the Agile Fields drop-down list.
8. Click Add Map.
9. Click Save to commit the mapping.

Weight_ppm

Weight_ppm is a non-standard Partminer Material attribute in the data source field.

To map weight_ppm attributes to Agile fields:

Log in to the Product Interchange application as admin/agile.

1. Click Component Management tab.
2. Click Data Source tab.
3. Enter * in the text field.
4. Click the Search button.

The search will result in the display of all data source profiles.

5. Click on the Map Material Attribute link. The Agile Field map page appears.
6. Select Weight_ppm from the Data Source Field drop-down list.
7. Select a field from the Agile Fields drop-down list.
8. Click Add Map.
9. Click Save to commit the mapping.

Standard Alias Search

One of the key concepts with Product Interchange is the concept of aliases for manufacturer names, manufacturer part numbers and substances. Product Interchange supports the aliasing of those objects based upon a customer-specific standard and an industry standard derived from a 3rd-party content source.

The Standard Alias Search screen:

- Allows you to search for manufacturers, substances or aliases.
- Provides functionality to manage the relationships.

This could take the form of:

- Adding or deleting manufacturers, substances and/or aliases.
- Merging one or more manufacturers/substances based upon acquisition.
- Data cleansing.
- Other requirements.

Performing Manufacturer Alias Search

To perform a Manufacturer Search:

1. Select the search parameter from the drop-down list. The list includes the following values:
 - Standard Mfr Name.
 - Alias Name.

For example: Let us assume that the manufacturer's name is ACME.

2. Enter the search characteristics in the search field.

Product Interchange uses two wildcards:

- The asterisk (*) - The asterisk matches with everything.
- The percent sign (%) - The percent sign finds matches with other characters. For example, %r% finds all results that contain the letter "r".

3. Click Search to display the search results.

This search will result in the display of all manufacturers with the word "ACME" being displayed.

Note The search is not case sensitive.

Note Please follow the above steps to perform Standard Substance Alias search.

Adding A New Manufacturer/Manufacturer Alias

Though additional manufacturer names and aliases are created automatically as part of the process flow within Product Interchange, there may be situations where new standard manufacturers and aliases may need to be populated manually.

To create a new manufacturer alias:

1. Click the Create New button from the Std. Alias Search screen.
2. Enter the Standard Mfr Name and the Alias Name
3. Click Create Alias to create both the standard manufacturer and associated alias.

Product Interchange will only allow creation of a manufacturer and an alias within this screen. Creation of only standard manufacturer is allowed in this same process through population of the same value in the Standard Mfr Name and Alias Name fields. For population of large volumes of manufacturers and aliases, see [Standard Alias Bulk Load](#) (on page 39).

An additional way to populate manufacturer aliases is by selecting the Add Alias link from the Standard Alias Search screen.

Note Please follow the above steps to Add a New Substance Alias.

Merging Existing Manufacturers and Aliases

Managing lists of manufacturers in the current business environment of acquisitions and takeovers often requires making mass changes to the data. Product Interchange provides the ability to merge

two manufacturers and their associated aliases.

To merge two manufacturers:

1. On the Std. Alias Screen, select Standard Mfr Name from the drop-down list.
2. Click Merge on the Std. Alias Search screen.
3. Select Survivor Mfr from the list of manufacturers.
4. Select Mfr Being Merged / Acquired from the list.
5. Click Update the Aliases to complete the “merger”.

Note Please follow the above steps to Merge Existing Substances and Aliases.

Part Search

Product Interchange allows you to perform a search of 3rd-party industry parts content through a Part Search process.

To perform a part search:

1. Enter a manufacturer part number in the Part Number field.
2. Check the applicable Data Source.
3. Select from the drop-down list the Match Type. The options include:
 - Exact
 - Partial
 - Alternate

The Part Search Result(s) will display one or more part matches, depending on the part and the number of parts specified in the default parameters set in the Data Source Profile. For more information about the manufacturer's part, click on the Part Number link and view the data sheet displayed in .pdf format.

Standard Alias Bulk Load

One of the key steps of Product Interchange is the population of a large volume of manufacturer parts, substances and associated aliases. Component Management enables the import of manufacturers, substances and aliases. Import processes are also available for the population of manufacturer parts, substances and aliases from 3rd-party content providers. As BOM/AML files and mapping substances are validated, the process of resolving manufacturer names will leverage the “standard” manufacturer name and append additional aliases to this standard. Aliases encountered during validation are recognized and converted by Product Interchange, to the standard format for the purpose of export.

To import Manufacturers and Aliases:

1. Browse for the file containing the manufacturers and aliases.

This file should be formatted in an Excel file, with Column A labeled Standard Mfr and containing aliases, and Column B labeled Alias for internal manufacturer name/Column B labeled Data Source manufacturer for data source manufacturer name.

2. Select the Map Type from the drop-down list.
This will be either manufacturers and aliases from internal sources or manufacturers and aliases from 3rd-party content providers.
3. Select a Data Source from the data source drop-down menu for Standard Mfr-Data Source Mfr mapping.
4. Click Load Standard Mfr.

To import Substances and Aliases:

1. Browse for the file containing the substances and aliases.
This file should be formatted in an Excel file, with Column A labeled Standard substances and containing aliases, and Column B labeled alias substances and containing the data source substance name.
2. Select the Map Type from the drop-down list.
This will be standard part and aliases from 3rd-party content providers.
3. Select a Data Source from the data source drop-down menu.
4. Click Load Standard Mfr.

To import Manufacturer Parts and Aliases:

1. Browse for the file containing the manufacturer parts and aliases.
2. This file should be formatted in an Excel file, with Column A labeled Dirty Mfr, Column B labeled Dirty Part, Column C labeled Clean Mfr and Column D labeled Clean Parts and containing their aliases.
3. Select the Map Type from the drop-down list.
This will be standard part and aliases from 3rd-party content providers.
4. Select a Data Source from the data source drop-down menu.
5. Click Load Standard Mfr.

DataLoad Scripts

Agile provides several scripts that can be used for loading data into or extracting information from the Product Interchange database.

Guidelines to unpack and run this script

1. Get *pidataload.zip*:
2. Extract it to a desired location. It would create a folder by name '*pidataload*'
3. Edit the properties file *config.properties* found in the folder *//pidataload/config* for database details as per the guidelines given in the properties file itself.

MFR and MPN Alias Table Format

The Internal Standard MFR Alias Excel file should look like the one in the following example:

- Column A of the Excel file should contain the Standard Mfrs.
- Column B of the Excel file should contain the Aliases.

First row should be the header: Cell A1 should be STANDARD MFR and Cell B1 should be ALIAS.

Internal Standard Mfr	Alias Mfr
PANASONIC	PANASONIC INC
PANASONIC	PANASONIC LTD
MICROSEMI	MICROSEMI DIV
MICROSEMI	MICROSEMI INC

The External Mfr Alias Excel file should look like the following example.

Column A of the Excel file should contain the Standard Mfrs.

Column B of the Excel file should contain the DataSource Mfrs.

First row should be the header: Cell A1 should be STANDARD MFR and Cell B1 should be DATAMFR.

Internal Standard Mfr	Partminer Mfr
TYCO	Tyco Electronics

The MPN Alias Excel file should look like the following.

Column A should be for Dirty Mfr.

Column B should be for Dirty Part.

Column C should be for Clean Mfr.

Column D should be for Clean Part.

Dirty MFR	Dirty Part	Clean MFR	Clean Part
3M	152244-0113GG	3M Interconnect Solutions	1522440113GG
3M	2304-6111TG	3M Interconnect Solutions	23046111TG
3M	2402-6112TB	3M Interconnect Solutions	24026112TB

The Mfr Code Excel file should look like the one in the following.

Column A should be for Mfr Name.

Column B should be Mfr Code.

MFR Name	MFR Code
3L Electronic Corp	7095
Netergy Microelectronics	2018
8 X 8 Inc	4659

MFR and MPN Alias Loader

For Internal Data Source Aliases loading execute a script by calling 'runMfrAlias' from the command prompt at the *//pidataload* folder level by providing two arguments in the following order:

Mfr Aliases data file path
Location of *config.properties* file.

Examples

Example 1 :

```
D:\SCRIPT_HARRIS\pidataload>runMfrAliasLoader.bat
C:\\views\\Agile9PIview\\AgilePI\\dataLoad\\files\\InternalAliasesData.xls
\\pidataload\config\config.properties
```

For External Data Source Aliases loading execute a script by calling 'runMfrAliasLoader.bat' from the command prompt at the *//pidataload* folder level by providing three arguments in the following order:

the Mfr Aliases data file path
the **data source organization** name exactly as configured in the PI
location of *config.properties* file.

Example 2 :

```
D:\SCRIPT_HARRIS\pidataload>runMfrAliasLoader.bat
C:\\views\\Agile9PIview\\AgilePI\\dataLoad\\files\\ExternalAliasesData.xls
PartMiner
\\pidataload\config\config.properties
```

For MPN- Aliases loading execute a script by calling 'runMpnAliasLoader.bat' from the command prompt at the *//pidataload* folder level by providing three arguments in the following order:

the Mfr part Aliases data file path
the **data source profile name** as configured in the PI
location of *torque.properties* file.

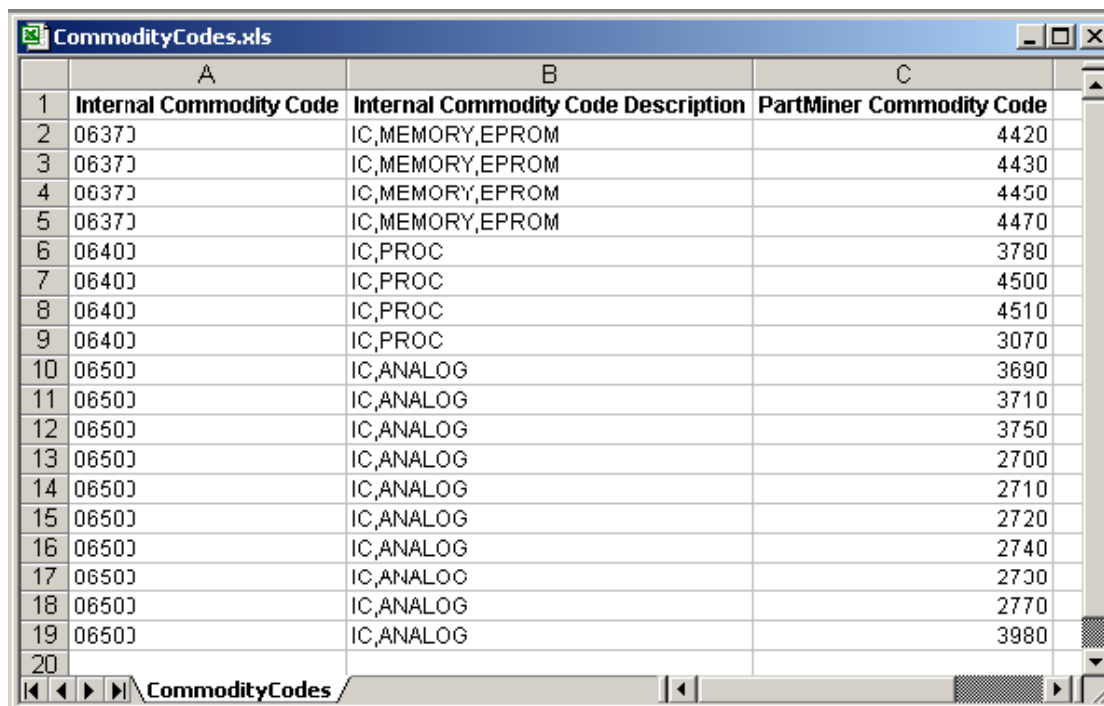
Example 3 :

```
D:\SCRIPT_HARRIS\pidataload>runMpnAliasLoader.bat
C:\\views\\Agile9PIview\\AgilePI\\dataLoad\\files\\mpndl.xls
PartMiner
\\pidataload\config\config.properties
```

Commodity Codes

Format of the Data File

The format of the Commodity Codes data file should look like the one shown in the screen shot.



	A	B	C
	Internal Commodity Code	Internal Commodity Code Description	PartMiner Commodity Code
1			
2	0637J	IC, MEMORY, EPROM	4420
3	0637J	IC, MEMORY, EPROM	4430
4	0637J	IC, MEMORY, EPROM	4450
5	0637J	IC, MEMORY, EPROM	4470
6	0640J	IC, PROC	3780
7	0640J	IC, PROC	4500
8	0640J	IC, PROC	4510
9	0640J	IC, PROC	3070
10	0650J	IC, ANALOG	3690
11	0650J	IC, ANALOG	3710
12	0650J	IC, ANALOG	3750
13	0650J	IC, ANALOG	2700
14	0650J	IC, ANALOG	2710
15	0650J	IC, ANALOG	2720
16	0650J	IC, ANALOG	2740
17	0650J	IC, ANALOG	2730
18	0650J	IC, ANALOG	2770
19	0650J	IC, ANALOG	3980
20			

Column A should be Internal Commodity Code; Column B should be the description of the same; Column C should be the corresponding PartMiner Commodity Code.

To generate the SQL:

Execute the Java Program *CommodityCodes*.

Usage:

```
runCommodityDetails.bat <CommodityDetails Data File> <Configuration File>
```

Where CommodityCode_Data_File is the Commodity Codes data file in the format shown above, Configuration File is the name of the SQL file.

Example:

```
runCommodityDetails.bat "D:\temp\dataload\data\CommodityDetails.xls"
\\PIDataLoad\config\config.properties
```

MFR Code Loader

This program will load codes for Manufacturers.

To load Mfr codes execute 'runMfrCodesLoader.bat' from the command prompt at the *//pidataload* folder level by providing two arguments in the following order:

the Mfr Aliases data file path
location of *config.properties* file.

Syntax:

```
runMfrCodesLoader.bat < Mfr_Code_Loader_Data_File > <Data Source  
Organization Name> <Configuration File >
```

Example:

```
runInternalMfrAliasesGetter.bat "C:\PI\docs\testFiles\dataload.xls"  
PartMiner \\PIDataLoad\config\config.properties"
```

MFR and MPN Alias Getter

InternalMfrAliasesGetter

This program will return Internal Manufacturer Mapping Table.

For getting Internal Data Source Mfr Mapping execute a script by calling 'runInternalMfrAliasesGetter.bat' from the command prompt at the *//pidataload* folder level by providing two arguments in the following order:

the Mfr Aliases data file path
location of *config.properties* file.

Syntax:

```
runInternalMfrAliasesGetter.bat <Internal_Mfr_Aliases_Generated_File>  
<Configuration File >
```

Example:

```
runInternalMfrAliasesGetter.bat  
"C:\PI\docs\drop3\dataLoad\aliasFiles\InternalMfrAliases.txt"  
\\PIDataLoad\config\config.properties"
```

ExternalMfrAliasesGetter

This program will return External DataSource Manufacturer Mapping Table.

Syntax:

```
runExternalMfrAliasGetter.bat
```

```
<Ext_Mfr_Aliases_Generated_File> <Datasource_Organization_Name> <Configuration File >
```

Example:

```
runExternalMfrAliasGetter.bat  
"C:\PI\docs\drop3\dataLoad\aliasFiles\ExternalMfrAliases.txt" Arrow  
\\PIDataLoad\config\config.properties"
```

MPNAliasGetter

This program will return Mapping Table For PartAliases.

For getting Mapping Table For PartAliases execute a script by calling 'runMpnAliasGetter.bat' from the command prompt at the *//pidataload* folder level by providing two arguments in the following order:

the Mfr Aliases data file path
location of *config.properties* file.

Syntax:

```
runMpnAliasGetter.bat <Part_Aliases_Generated_File> <Configuration File
```

```
> <Datasource_Profile_Name>
```

Example:

```
runMpnAliasGetter.bat
"C:\PI\docs\drop3\dataLoad\aliasFiles\PartAliases.txt"
\\PIDataLoad\config\config.properties" ARROW
```

Note A valid Datasource_Organization_Name is one of the data source Organization names that you see in the drop-down list of Organizations when you create a new Data Source Profile in Agile Product Interchange by logging in as Admin

Note A valid Datasource_Profile_Name is one of the data source profile names that you get as result when you search for Data Source Profile in Agile Product Interchange.

JDBC Data Source Configuration

A typical JDBC data source profile looks like the one shown below.

- * Name: Provide a name for the data source being configured
- * URL: JDBC URL of the data source
- * User Name: Username of the database
- * Password: Password of the database
- * Driver: JDBC Driver
- * Match Record Limit: Limiting the number matched records returned for a given mfr part (default value is 7)
- * Organization: Select the organization of the data source (this list would show the parties of party_type DATA)
- * SQL: We need to embed the SQL that would be used for this JDBC data source part match query. Agile Product Interchange would append the criteria for mfr part (PARTNUMBER) to this SQL and execute the same against the data source.

Leave the following fields blank for JDBC data source profile:

Account, SQL for Mfr Name, and SQL for CPN.

Column Mapping

For any JDBC data source profile, the following are the fields that could be captured by Product Interchange. (SELECT clause of the SQL configured should use these mappings). The following are the valid entries to be used as column names in the Column Mapping section of the data source profile.

- VENDORNAME
- PARTNUMBER
- PARTTYPE
- PARTREV
- PARTDESC

- VENDORCODE
- LEADTIME_IN_DAYS
- LIFECYCLE_STATUS
- PART_URL
- COST
- COMMODITY_CODE

SQL for JDBC Data Source Profile for MPN Search against Agile 90

Select

- MFR.NAME as VENDORNAME
- MPN.PART_NUMBER as PARTNUMBER
- MPN.DESCRPTION as PARTDESC
- MPN.OBJVERSION as PARTREV FROM MANU_PARTS MPN
- MANUFACTURERS MFR where MPN.MANU_ID = MFR.ID

Updating the Compare AML's Data Source Configuration

To update the Compare AML's data source configuration (so that it connects to Agile 90):

1. Search and load the data source profile CPNDB. (This is the profile that would be used by the Compare AML functionality)
2. Update the fields * URL:, * User Name:, and * Password: accordingly with the details of Agile 90 database.
3. Update the query of the field SQL for CPN with the following:

Select

```
DISTINCT MFR.NAME AS VENDORNAME,MPN.PART_NUMBER AS  
PARTNUMBER,IPN.DESCRPTION AS PARTDESC,P2.TEXT02 AS COST FROM ITEM  
IPN,MANU_PARTS MPN, MANUFACTURERS MFR,MANU_BY IPN_CPN_MAP,PAGE_TWO P2  
WHERE IPN.ITEM_NUMBER = CONCAT($PARAM_Prefix, $PARAM_CUSTOMERPARTNUMBER)  
AND IPN_CPN_MAP.AGILE_PART = IPN.ID AND IPN_CPN_MAP.MANU_PART <> 0 AND  
MPN.ID = IPN_CPN_MAP.MANU_PART AND MFR.ID = MPN.MANU_ID AND P2.ID =  
IPN.ID AND MPN.STATUS = 1517
```

Update the Column Mapping as shown below:

- | | |
|----------------------|-----------------|
| • VENDORNAME | MFR.NAME |
| • PARTNUMBER | MPN.PART_NUMBER |
| • CUSTOMERPARTNUMBER | IPN.ITEM_NUMBER |
| • PARTDESC | MPN.DESCRPTION |
| • COST | P2.TEXT02 |

You are done!

Note 'Test this Data Source' may not give a SUCCESS result for this (CPNDB) data source. It would give you a 'java.sql.SQLException: ORA-00911: invalid character' error. If an error other than this occurs, it should be debugged accordingly.

Using Agile PC as a Data Source

For any JDBC data source profile, the following are the fields that could be captured by Product Interchange and the following are the valid entries to be used as column names in the Column Mapping section of the data source profile.

- VENDORNAME
- PARTNUMBER
- PARTTYPE
- PARTREV
- PARTDESC
- VENDORCODE
- LEADTIME_IN_DAYS
- LIFECYCLE_STATUS
- PART_URL
- COST
- COMMODITY_CODE

Agile Import Preferences

This chapter includes the following:

- Agile PC Import Profile 49

Agile PC Import Profile

This part of Agile Product Interchange configuration allows users to select some rules that will be applied at the time of exporting processed files to Agile Product Collaboration. This configuration ensures that the policies implemented by Agile Product Collaboration administrator for the regular Agile Product Collaboration import process are also followed while exporting files from Agile Product Interchange.

This configuration is required only if Agile Product Collaboration is one of the systems to which processed files will be exported.

To configure Agile PC Import Profile:

Configuration of Agile Import involves selecting rules to be applied during export to Agile Product Collaboration.

1. Click Agile Import Preferences tab.
Enables selection of specific rules driving exports to Agile Product Collaboration.
2. Choose from the drop-down lists to indicate preferences for each option.
3. Click Submit when done.
4. Click Reset to return to original settings.

Selecting Import Preferences

Agile Product Interchange allows the system administrator to select preferences that drive import of product information into Agile PLM. These preferences allow the administrator to tailor the use cases related to import and apply the same policies that work with traditional Agile PLM Import.

Parsing and Validation Options

The following describes the functions of specific import preferences:

Log in to the Product Interchange application as admin/agile.

Click Import Preference tab.

This set of import preferences drives rules related to import of specific fields into Agile PLM.

Whitespace Validation action: Reject / Strip

If Product Interchange has been configured to disallow whitespace in any attribute, then it drives the following action during import:

- Reject --> Causes the items with whitespace attributes to be rejected from the import.
- Strip --> Causes the whitespaces to be stripped and allows the item to be imported without the whitespace.

Case Validation action: Reject / Convert

If Agile PLM has been configured to disallow lower case in any attribute, then it drives the following action during import:

- Reject --> Causes the items with lower case attributes to be rejected from the import.
- Convert --> Causes the lower case to be converted to upper case and allows the item to be imported with the conversion.

Length Validation action: Reject / Truncate

This import preference controls the length of the attributes based on the max length specified in Agile PLM. The following actions are taken:

- Reject --> Causes items with attributes exceeding the max length to be rejected from the import.
- Truncate --> Causes truncation of fields exceeding the max length to the length specified by this value.

Business Rule Options

This set of import preferences drives rules related to other aspects of import into Product Interchange.

- Change Mode: Redlining/Authoring

Agile PLM allows import in two specific modes, both of which are supported by Agile Product Interchange.

In Redlining Mode, Product Interchange creates a change order and puts all imported items as Affected Items on the change order. You can route this change order and release it as needed.

In Authoring Mode, Product Interchange does not create a change order, but goes ahead and directly creates / updates items and manufacturer parts.

Note	From a performance perspective, Authoring Mode is much better suited for importing large projects with hundreds or thousands of parts into Agile PLM.
------	---

Note	Authoring Mode is not available for PG&C Bill of Substance imports in Release 9.2
------	---

- Multi Row Update Mode (Complete Replace/ Add/Update Only): Determines how AML is modified as a result of the import.
 - In Complete Replace mode, the existing AML is completely replaced by the AML being imported
 - In Add/Update mode, the existing AML is updated with new attributes and new AML items

are added to the record.

- Time Out (minutes): Determines how long the connection to Agile PLM will remain open while exporting. This needs to be set to a larger value for very large exports to Agile PLM.
- Publish Material Weight: Lists material weight and exports it to Agile PLM. If the material weight is not listed it will not export to Agile PLM.
 - For Partminer, the total sum of the substance weight gets listed.
 - For Total Part Plus, only the explicit weight of the material is listed.
- Import BOS: Obtains material details along with part information from the data source.
- AutoCreate Substance: Auto creates a substance in Agile PLM (if it does not already exist in Agile PLM).
- Import Only MFR-MFR Part: Determines the need to create an MCO or not, for all Mfr parts of the substance. If you select the check box, the MCO will not be created.

Note	The Import Only MFR-MFR Part is to be selected only when the Change Mode is set to Redline.
------	---

PG&C Integration

This chapter includes the following:

- PG&C Integration..... 53

PG&C Integration

Product Interchange supports integration with Product Governance & Compliance (PG&C) solution to manage hazardous material information.

Product Interchange provides the following capabilities:

- Retrieval of compliance attributes, including full Bill of Substance information from external content sources and persisting the compliance information in the PI schema.
- Creation of Homogeneous Material Declaration in PG&C solution by exporting compliance information from Product Interchange.
 - Unique declaration created for each external data source.
 - Single declaration created for all parts in a validation project.
 - Substances created in PG&C if they do not exist currently.
 - Multi Level Bill Of Substance tree structure supported.
 - Configurable mapping of compliance attributes from external data source to composition attributes in PG&C.

Note	Materials need not be imported prior to Declaration import since they are not required to be present as global PG&C objects.
------	--

Import Configuration

Please follow the steps outlined below to set up the configurations required for import of Material Declaration into Agile PLM (PG&C).

PG&C Privileges Required for Declaration Import

To be able to export a declaration successfully, a Product Interchange logged in user should have the following privileges in Agile PLM:

- Discover Supplier.
- Create Declarations.
- Delete Declaration.

- Modify Declarations (All attributes).
- Create Substances.
- Modify Substances (All attributes).
- Discover Specifications.

PG&C Configurations Required For Declaration Import

For the declaration of import you need to create the following:

1. Create an active web supplier.
2. Create two specifications.

To create an active web supplier:

Create an active web supplier in Agile PLM using the Supplier creation wizard.

1. Ensure that Supplier has the right number of licensed users.
2. For a web supplier, add a contact user. (This can be created afresh using the wizard).
3. The contact user should have ONLY "my user profile" and "material prov" roles.
4. A particular user CANNOT belong to more than one Supplier.
5. Users need to be created before they are added to the Supplier.

To create two specifications:

Create two specifications labeled: LeadFreeSpec and rohs-spec.

- Since these specifications are used on Homogeneous Material Declaration, "Validation Level" on the title block of the spec general information Page should be set to "Homogeneous Material Level".
- In the substances tab of these specifications, add the appropriate substances and the corresponding threshold ppm values.

PI Configurations For Declarations Import

To map the external data sources to an existing active web-supplier from Product Interchange:

Log in to the Product Interchange application as admin/agile.

1. Click the Administration tab.
2. Click the Organization tab.
3. Enter the org id Or * to lookup data source organization information.
4. Click on the org id of the relevant external data source (for example, 160 for Partminer).
5. Select a supplier from the drop-down against the label Supplier.
6. Click the Update button.

Mapping Compliance-Related Fields

To map Agile PLM fields for compliance-related fields:

This mapping is NOT at the Manufacturer Part level. It is to map the Manufacturer Part Composition (Bill Of Substance) level attributes retrieved from external data sources to the flex fields in Manufacturer Part Composition tab of Homogeneous Material Declaration class in PG&C.

Fields like text01, etc. need to be enabled in PG&C before they will appear in the Product Interchange Materials Fields Mapping page. Product Interchange does not map these fields automatically by default. If you want to export to these fields, you would need to be mapped prior to the export of declaration to PG&C.

Log in to the Product Interchange application as admin/agile.

1. Click the Component Management tab.
2. Click the Data Source tab.
3. Enter * to lookup data source profile information.
4. Click the on the Map Material Attribute link of the relevant external data source (for example, 160 for Partminer).
5. Map the Data Source fields in the left hand drop-down to some text fields (for example, text01, text02 etc.) in the right hand dropdown. The right-hand dropdown displays MDO.MfrParts.Composition fields that are not mapped by default. It also displays Page two fields for Substance.
6. Click Save to persist the mappings.

Note The non-standard flex fields need to be enabled in Agile PLM (PG&C) to appear in the Material Fields Mapping page in Agile Product Interchange. This should be done prior to importing the declaration into PG&C.

Part Miner Field	Agile PLM Attribute	Comments
Material Validate Tags		
PartNo	MDO.MfrParts.Mfr Part Number.	
MfrName	MDO.MfrParts.Mfr Name.	
LeadFree	MDO.MfrParts.Declared Compliance.	
RoHS	MDO.MfrParts.Declared Compliance.	
Material Detail Tags		
Material	MDO.MfrParts.Composition.Substance Name.	When exporting to Agile, this should be treated as a Parent and the Substance as a Child.

Part Miner Field	Agile PLM Attribute	Comments
Substance	Substance.Name (if Product Interchange is going to "auto-create" non-existing substances) & MDO.MfrParts.Composition.Substance Name.	This is the "child" of the "parent" material. All the tags below are for this Substance and for the Material.
Symbol	P2 Text field in Substance (if Product Interchange is going to "auto-create" non-existing substances).	
CasNumber	Substance.CAS Number (if Product Interchange is going to "auto-create" non-existing substances). MDO.MfrParts.Composition.UserEntered.CasNumber..	
Weight_ppm	MDO.MfrParts.Composition.Weight.	
Location	MDO.MfrParts.Composition.Substance Name	When exporting to Agile, this will be treated as a subpart and parent to the material.
Use	Text Flex field in MDO.MfrParts.Composition.	

Note Non-standard Material Validate fields (i.e. fields not mapped by default) should be mapped to Page 2 fields of the manufacturer parts. Once the mapping is done, these attributes will be brought into Product Interchange as part of the standard BOM/AML import step that precedes the creation of a material declaration.

- Each Declaration will have two specifications (rohs-spec, LeadFreeSpec) attached to it and for each manufacturer part in the declaration, there will be two entries under the manufacturer part (one for each spec).
- The import of Material Declaration into PG&C happens immediately after the BOM/AML import. So the export to Agile PLM from Product Interchange is now a 2-phase process involving the Product Interchange project import, substances import and Declaration Import in that order.
- The Declaration Number is generated on the fly and will appear on the Agile Export Results page along with the Change Number (If BOM import was in redline mode).
- The declaration thus created will be in pending state. Once released the substances will be linked to manufacturer parts and will be visible under composition in compliance tab of manufacturer parts.

Regular Expressions

This chapter includes the following:

▪ Introduction	57
▪ Regular Expression Syntax	58
▪ Build a Regular Expression	61
▪ Testing a Regular Expression.....	62
▪ Order of Precedence	62
▪ Ordinary Characters	62
▪ Special Characters	63
▪ Non-Printable Characters	64
▪ Character Matching	64
▪ Bracket Expressions.....	65
▪ Quantifiers	66
▪ Anchors.....	67
▪ Alternation and Grouping.....	69
▪ Back References	70
▪ Character Classes	71

Introduction

A regular expression is shorthand for abbreviating patterns of text. It can also be defined as a pattern that describes a set of strings.

Regular expressions provide a powerful, flexible, and efficient method for processing text. The extensive pattern-matching notation of regular expression allows you to quickly parse large amounts of text to find specific character patterns; to extract, edit, replace, or delete text sub strings; to add the extracted strings to a collection in order to generate a report. For many applications that deal with strings (such as HTML processing, log file parsing, and HTTP header parsing), regular expressions are an indispensable tool.

These regular expressions are sometimes referred to as regex, grep, or pattern matching. You can use a regular expression to perform client-side validation of email addresses, phone numbers, search multiple documents for strings and patterns you wish to change or remove, or extract a list of links from source code. Most languages support Regex. The construction of regular expressions is analogous to the construction of arithmetic expressions. That is, a variety of meta-characters and operators combine small expressions to create large expressions.

You can construct regular expression by putting the various components of the expression pattern between a pair of delimiters.

A typical search and replace operation requires you to provide the exact text that matches the intended search result. Although this technique may be adequate for simple search and replace tasks in static text, it lacks flexibility and makes searching dynamic text difficult if not impossible.

With regular expression, you can:

- Test for a pattern within a string: For example, you can test an input string to see if a telephone number pattern or a credit card number pattern occurs within the string. This is called data validation.
- Replace text: You can use a regular expression to identify specific text in a document and either remove it completely or replace it with other text.
- Extract a sub string from a string based upon a pattern match.
- You can find specific text within a document or input field: For example, you may need to search an entire Web site, remove outdated material, and replace some HTML formatting tags. You can use a regular expression to determine if the material or the HTML formatting tags appears in each file. This process reduces the affected file list to those that contain material targeted for removal or change. It can also be used in a language such as JScript or C that is not known for its string-handling ability.

Regular Expression Syntax

A regular expression is a pattern of text that consists of ordinary characters (for example, letters a through z) and special characters, known as meta-characters. The pattern describes one or more strings to match when searching text.

Literals and Meta-characters

To make the most of the power of regex, one needs to be familiar with a few classifications of characters.

Literals are normal text characters and can include white space (tabs, spaces, new lines, etc.). Unless modified by a meta-character, a literal will match itself on a one-for-one basis. *Meta-characters'* power lies in how they are arranged and interpreted as wildcards.

Meta-characters can be escaped with a backslash (\) to find instances of them, for instance, if it is to find a caret (^) or a backslash, as well as used in nested groups or other combinations.

A "match" can be as simple as a single character or as complex as a sequence of literals and meta-characters in nested and compounded combinations.

The following table contains the complete list of meta-characters and their behavior in the context of regular expressions:

Character	Description
\	Marks the next character as a special character, a literal, a back reference, or an octal escape. For example, 'n' matches the character "n". '\n' matches a new line character. The sequence '\\ matches "\" and "\\(" matches "(".

<code>^</code>	Matches the position at the beginning of the input string. If the Regex object's Multiline property is set, <code>^</code> also matches the position following <code>\n</code> or <code>\r</code> .
<code>\$</code>	Matches the position at the end of the input string. If the Regex object's Multiline property is set, <code>\$</code> also matches the position preceding <code>\n</code> or <code>\r</code> .
<code>*</code>	Matches the preceding character or sub expression zero or more times. For example, <code>zo*</code> matches "z" and "zoo". <code>*</code> is equivalent to <code>{0,}</code> .
<code>+</code>	Matches the preceding character or sub expression one or more times. For example, <code>zo+</code> matches "zo" and "zoo", but not "z". <code>+</code> is equivalent to <code>{1,}</code> .
<code>?</code>	Matches the preceding character or sub expression zero or one time. For example, <code>do(es)?</code> matches the "do" in "do" or "does". <code>?</code> is equivalent to <code>{0,1}</code> .
<code>{n}</code>	<i>n</i> is a nonnegative integer. Matches exactly <i>n</i> times. For exam <code>o{2}</code> does not match the 'o' in "Bob," but matches the two o's in "food".
<code>{n,}</code>	<i>n</i> is a nonnegative integer. Matches at least <i>n</i> times. For exam <code>o{2,}</code> does not match the "o" in "Bob" and matches all the o's in "fooooood". <code>{1,}</code> is equivalent to <code>{1,}</code> . <code>{0,}</code> is equivalent to <code>{0,}</code> .
<code>{n,m}</code>	<i>M</i> and <i>n</i> are nonnegative integers, where $n \leq m$. Matches at least <i>n</i> and at most <i>m</i> times. For example, <code>o{1,3}</code> matches the first three o's in "fooooood". <code>{0,1}</code> is equivalent to <code>{0,1}</code> . Note that you cannot put a space between the comma and the num
<code>?</code>	When this character immediately follows any of the other quanti <code>(*, +, ?, {n}, {n,}, {n,m})</code> , the matching pattern is non-greedy. A non-greedy pattern matches as little of the searched string as possible, whereas the default greedy pattern matches as much of the searched string as possible. For example, in the string "oooo", <code>o+?</code> matches a single "o", while <code>o+</code> matches all 'o's.
<code>.</code>	Matches any single character except <code>\n</code> . To match any characincluding the <code>\n</code> , use a pattern such as <code>[s\S]</code> .
<code>(pattern)</code>	A sub expression that matches <i>pattern</i> and captures the match. The captured match can be retrieved from the resulting Matches collection using the <code>\$0...\$9</code> properties. To match parentheses characters <code>(,)</code> , use <code>\(</code> or <code>\)</code> .
<code>(?:pat)</code>	A sub expression that matches <i>pattern</i> but does not capture the match, that is, it is a non-capturing match that is not stored for possible later use. This is useful for combining parts of a pattern with the "or" character <code>()</code> . For example, <code>industr(?:ylies)</code> is a more economical expression than <code>industry industries</code> .

<code>(?=pat)</code>	A sub expression that performs a positive look ahead search, which matches the string at any point where a string matching <i>pattern</i> begins. This is a non-capturing match, that is, the match is not captured for possible later use. For example 'Windows (95 98 NT 2000)' matches "Windows" in "Windows 2000" but not "Windows" in "Windows 3.1". Look heads do not consume characters, that is, after a match occurs, the search for the next match begins immediately following the last match, not after the characters that comprised the look ahead.
<code>(?!pattern)</code>	A sub expression that performs a negative look ahead search, which matches the search string at any point where a string not matching <i>pattern</i> begins. This is a non-capturing match, that is, the match is not captured for possible later use. For example 'Windows (?!95 98 NT 2000)' matches "Windows" in "Windows 3.1" but does not match "Windows" in "Windows 2000". Look heads do not consume characters, that is, after a match occurs, the search for the next match begins immediately following the last match, not after the characters that comprised the look ahead.
<code>x y</code>	Matches either <i>x</i> or <i>y</i> . For example, 'z food' matches "z" or "food". '(z f)ood' matches "zood" or "food".
<code>[xyz]</code>	A character set. Matches any one of the enclosed characters. For example, '[abc]' matches the 'a' in "plain".
<code>[^xyz]</code>	A negative character set. Matches any character not enclosed. For example, '[^abc]' matches the 'p' in "plain".
<code>[a-z]</code>	A range of characters. Matches any character in the specified range. For example, '[a-z]' matches any lowercase alphabetic character in the range 'a' through 'z'.
<code>[^a-z]</code>	A negative range characters. Matches any character not in the specified range. For example, '[^a-z]' matches any character not in the range 'a' through 'z'.
<code>\b</code>	Matches a word boundary, that is, the position between a word and a space. For example, 'er\b' matches the 'er' in "never" but not the 'er' in "verb".
<code>\B</code>	Matches a nonword boundary. 'er\B' matches the 'er' in "verb" but not the 'er' in "never".
<code>\cx</code>	Matches the control character indicated by <i>x</i> . For example, \cM matches a Control-M or carriage return character. The value of <i>x</i> must be in the range of A-Z or a-z. If not, c is assumed to be a literal 'c' character.
<code>\d</code>	Matches a digit character. Equivalent to [0-9].
<code>\D</code>	Matches a nondigit character. Equivalent to [^0-9].
<code>\f</code>	Matches a form-feed character. Equivalent to \x0c and \cL.
<code>\n</code>	Matches a newline character. Equivalent to \x0a and \cJ.

\r	Matches a carriage return character. Equivalent to \x0d and \cM.
\s	Matches any white space character including space, tab, form-feed, and so on. Equivalent to [\f\n\r\t\v].
\S	Matches any non-white space character. Equivalent to [^\f\n\r\t\v].
\t	Matches a tab character. Equivalent to \x09 and \cI.
\v	Matches a vertical tab character. Equivalent to \x0b and \cK.
\w	Matches any word character including underscore. Equivalent to '[A-Za-z0-9_]'.
\W	Matches any nonword character. Equivalent to '[^A-Za-z0-9_]'.
\xn	Matches <i>n</i> , where <i>n</i> is a hexadecimal escape value. Hexadeciescape values must be exactly two digits long. For example, '\x41' matches "A". '\x041' is equivalent to '\x04' & "1". Allows ASCII codes to be used in regular expressions.
\num	Matches num, where num is a positive integer. A reference back to captured matches. For example, '(.)\1' matches two consecutive identical characters.
\nml	Matches octal escape value <i>nm</i> /when <i>n</i> is an octal digit (0-3) and <i>m</i> and <i>l</i> are octal digits (0-7).
\un	Matches <i>n</i> , where <i>n</i> is a Unicode character expressed as four hexadecimal digits. For example, \u00A9 matches the copy

Below are examples of regular expressions:

Expression	Matches
/^\s*\$/	Match a blank line.
/d{2}-ld{5}/	Validate an ID number consisting of 2 digits, a hyphen, and an addi5 digits.
/<\s*(S+)(\s[>]*)?>[ls\S]*<\s*\1\s*>/	Match an HTML tag.

Build a Regular Expression

The construction of Regular Expressions is analogous to the construction of arithmetic expressions. That is, a variety of met characters and operators combine small expressions to create large expressions. You construct a Regular Expression by putting the various components of the expression pattern between a pair of delimiters.

Example:

```
/expression/
```

The components of a Regular Expression can be individual characters, sets of characters, ranges of

characters, choices between characters, or any combination of all of these components.

Testing a Regular Expression

To test Regular Expressions:

Start the browser.

1. Click the Agile Product Interchange bookmark, if one exists, or type the URL of the server where Product Interchange is installed.

Enter your user name and password. Your user name and password are case-sensitive.

You can change your password at any time by clicking the Profile link after you have logged into Product Interchange.

2. Click the Login button.

If you make a mistake, click Clear and retype your user name and password. The login process is complete.

3. Click Parse.

4. Click Regular Expressions Tab.

Order of Precedence

A regular expression is evaluated from left to right and follows an order of precedence, much like an arithmetic expression.

The following table illustrates, from highest to lowest, the order of precedence of the various regular expression operators:

Operators	Description
\	Escape
(), (?:), (?:=), []	Parentheses and Brackets
*, +, ?, {n}, {n,}, {n,m}	Quantifiers
^, \$, \anymetacharacanycharac	Anchors and Sequences
	Alternation

Characters have higher precedence than the alternation operator. This allows 'm|food' to match "m" or "food". To match "mood" or "food", use parentheses to create a sub expression, which results in '(m|f)ood'.

Ordinary Characters

Ordinary characters consist of all printable and non-printable characters that are not explicitly

designated as met characters. This includes all uppercase and lowercase alphabetic characters, all digits, all punctuation marks, and some symbols.

The simplest form of a regular expression is a single, ordinary character that matches itself in a searched string.

For example, a single-character pattern, such as `A`, matches the letter `A` wherever it appears in the searched string.

Single-Character Regular Expression Patterns

```
/a/  
/7/  
/M/
```

You can combine a number of single characters to form a large expression.

Example: The following regular expression combines the single-character expressions: `a`, `7`, and `M`.

```
/a7M/
```

Note There is no concatenation operator. You need to just type one character after another.

Special Characters

A number of met characters require special treatment when trying to match them. To match these special characters, you must first escape *the* characters, that is, precede them with a backslash character (`\`).

The following table lists special characters and their meanings:

Special Characters	Comments
\$	Matches the position at the end of an input string. If the RegExp object's Multiline property is set, <code>\$</code> also matches the position preceding <code>\n</code> or <code>\r</code> . To match the <code>\$</code> character itself, use <code>\\$</code> .
()	Marks the beginning and end of a sub expression. Sub expressions may be captured for later use. To match these characters, use <code>\(</code> and <code>\)</code> .
*	Matches the preceding character or sub expression zero or more times. To match the <code>*</code> character, use <code>*</code> .
+	Matches the preceding character or sub expression one or more times. To match the <code>+</code> character, use <code>\+</code> .
.	Matches any single character except the new line character <code>\n</code> . To match <code>.</code> , use <code>\.</code>
[Marks the beginning of a bracket expression. To match <code>[</code> , use <code>\[</code> .
?	Matches the preceding character or sub expression zero or one time, or indicates a non-greedy quantifier. To match the <code>?</code> character, use <code>\?</code> .

\	Marks the next character as a special character, a literal, a back reference, or an octal escape. For example, the character <code>n</code> matches the character <code>n</code> . <code>\n</code> matches a new line character. The sequence <code>\\</code> matches <code>\</code> and <code>\(</code> matches <code>(</code> .
^	Matches the position at the beginning of an input string except when used in a bracket expression where it negates the character set. To match the <code>^</code> character itself, use <code>\^</code> .
{	Marks the beginning of a quantifier expression. To match <code>{</code> , use <code>\{</code> .
	Indicates a choice between two items. To match <code> </code> , use <code>\ </code> .

Non-Printable Characters

Non-printing characters may also be part of a regular expression. The following table lists the escape sequences that represent non-printing characters:

Character	Meaning
<code>\cx</code>	Matches the control character indicated by <code>x</code> . For example, <code>\cM</code> matches a Control-M or carriage return character. The value of <code>x</code> must be in the range of A-Z or a-z. If not, <code>c</code> is assumed to be a litc character.
<code>\f</code>	Matches a form-feed character. Equivalent to <code>\x0c</code> and <code>\cL</code> .
<code>\n</code>	Matches a new line character. Equivalent to <code>\x0a</code> and <code>\cJ</code> .
<code>\r</code>	Matches a carriage return character. Equivalent to <code>\x0d</code> and <code>\cM</code> .
<code>\s</code>	Matches any white space character including space, tab, form-feed, and so on. Equivalent to <code>[\f\n\r\t\v]</code> .
<code>\S</code>	Matches any non-white space character. Equivalent to <code>[^\f\n\r\t\v]</code> .
<code>\t</code>	Matches a tab character. Equivalent to <code>\x09</code> and <code>\cI</code> .
<code>\v</code>	Matches a vertical tab character. Equivalent to <code>\x0b</code> and <code>\cK</code> .

Character Matching

The period (`.`) matches all but one single printing or non-printing character in a string. The exception is a new line character (`\n`).

Example: The following regular expression matches `aac`, `abc`, `acc`, `adc`, and so on, as well as `a1c`, `a2c`, `a-c`, and `a#c`:

```
"/a.c/"
```

To match a string containing a file name in which a period (`.`) is part of the input string, precede the

period in the regular expression with a backslash (\) character.

Example: The following regular expression matches filename.ext:

```
"/filename\\.ext/"
```

Bracket Expressions

The above expressions only let you match any single character. You may want to match specific characters from a list. For example, you might want to find chapter headings that are expressed numerically (Chapter 1, Chapter 2, and so on). To create a list of matching characters, place one or more individual characters within square brackets ([and]).

When characters are enclosed in brackets, the list is called a bracket expression.

Within brackets, as anywhere else, an ordinary character represents itself, that is, it matches an occurrence of itself in the input text. Most special characters lose their meaning when they occur inside a bracket expression.

Here are some exceptions:

- The] character ends a list if it is not the first item. To match the] character in a list, place it first, immediately following the opening [.
- The \ character continues to be the escape character. To match the \ character, use \\.
- Characters enclosed in a bracket expression match only a single character for the position in the regular expression.

Example: The following regular expression matches Chapter 1, Chapter 2, Chapter 3, Chapter 4, and Chapter 5:

```
/Chapter [12345]/
```

Note	The word Chapter and the space that follows are fixed in position relative to the characters within brackets. The bracket expression is used to specify only the set of characters that matches the single character position immediately following the word Chapter and a space. That is the ninth character position.
-------------	---

- To express the matching characters using a range instead of the characters themselves, use the hyphen (-) character to separate the beginning and ending characters in the range. The character value of the individual characters determines the relative order within a range.

Example: The following regular expression contains a range expression that is equivalent to the bracketed list shown above.

```
/Chapter [1-5]/
```

When a range is specified in this manner, both the starting and ending values are included in the range. It is important to note that the starting value must precede the ending value in Unicode sort order.

- To include the hyphen character in a bracket expression, do one of the following:
- Escape it with a backslash:

1. [-]

- Put the hyphen character at the beginning or the end of the bracketed list. The following expressions match all lower case letters and the hyphen:

2. [-a-z]
3. [a-z-]
 - Create a range in which the beginning character value is lower than the hyphen character and the ending character value is equal to or greater than the hyphen. Both of the following regular expressions satisfy this requirement:
4. [!-~]
5. [!~]

To find all characters not in the list or range, place the caret (^) character at the beginning of the list. If the caret character appears in any other position within the list, it matches itself.

Example: The following regular expression matches chapter headings with numbers greater than 5:
`/Chapter [^12345]/`

In the examples above, the expression matches any digit character in the ninth position except 1, 2, 3, 4, or 5. So, for example, Chapter 7 is a match and so is Chapter 9.

The above expressions can be represented using the hyphen character (-):

`/Chapter [^1-5]/`

A typical use of a bracket expression is to specify matches of any upper- or lower case alphabetic characters or any digits.

The following expression specifies such a match:

`/[A-Za-z0-9]/`

Quantifiers

If you cannot specify the number of characters that comprise a match, regular expressions support the concept of quantifiers. These quantifiers let you specify how many times a given component of a regular expression must occur for a match to be true.

The following table illustrates the various quantifiers and their meanings:

Character	Description
*	Matches the preceding character or sub expression zero or more times. For example, <code>zo*</code> matches <code>z</code> and <code>zoo</code> . <code>*</code> is equivalent to <code>{0,}</code> .
+	Matches the preceding character or sub expression one or more times. For example, <code>zo+</code> matches <code>zo</code> and <code>zoo</code> , but not <code>z</code> . <code>+</code> is equivalent to <code>{1,}</code> .
?	Matches the preceding character or sub expression zero or one time. For example, <code>do(es)?</code> matches the <code>do</code> in <code>do</code> or <code>does</code> . <code>?</code> is equivalent to <code>{0,1}</code> .
{ <i>n</i> }	<i>n</i> is a nonnegative integer. Matches exactly <i>n</i> times. For example, <code>o{2}</code> does not match the <code>o</code> in <code>Bob</code> but matches the two <code>o</code> 's in <code>food</code> .
{ <i>n</i> ,}	<i>n</i> is a nonnegative integer. Matches at least <i>n</i> times. For example, <code>o{2,}</code> does not match the <code>o</code> in <code>Bob</code> and matches all the <code>o</code> 's in <code>foooooo</code> . <code>o{1,}</code> is equivalent to <code>o+</code> . <code>o{0,}</code> is equivalent to <code>o*</code> .

$\{n,m\}$	m and n are nonnegative integers, where $n \leq m$. Matches at least n and at most m times. For example, $o\{1,3\}$ matches the first three o's in fofooooo. $o\{0,1\}$ is equivalent to $o?$. Note that you cannot put a space between the comma and the numbers.
-----------	--

Since chapter numbers could easily exceed nine in a large input document, you need a way to handle two or three digit chapter numbers. Quantifiers give you that capability.

Example: The following regular expression matches chapter headings with any number of digits:

```
/Chapter [1-9][0-9]*/
```

Note The quantifier appears after the range expression. Therefore, it applies to the entire range expression that, in this case, specifies only digits from 0 through 9, inclusive.

The `+` quantifier is not used here because there does not necessarily need to be a digit in the second or subsequent position.

The `?` Character also is not used because it limits the chapter numbers to only two digits. You want to match at least one digit following Chapter and a space character.

If you know that chapter numbers are limited to only 99 chapters, you can use the following expression to specify at least one but not more than two digits.

```
/Chapter [0-9]{1,2}/
```

The disadvantage of the above expression is that a chapter number greater than 99 will still only match the first two digits. Another disadvantage is that Chapter 0 would match.

Better expressions for matching only two digits are the following:

```
/Chapter [1-9][0-9]?/ or /Chapter [1-9][0-9]{0,1}/
```

The `*`, `+`, and `?` quantifiers are all referred to as greedy because they match as much text as possible. However, sometimes you just want a minimal match.

Example: You may be searching an HTML document for an occurrence of a chapter title enclosed in an H1 tag. That text appears in your document as:

```
<H1>Chapter 1 - Introduction to Regular Expressions</H1>
```

The following expression matches everything from the opening less than symbol (`<`) to the greater than symbol (`>`) that closes the H1 tag.

```
/<.*>/
```

If you only want to match the opening H1 tag, the following, non-greedy expression matches only `<H1>`.

```
/<.*?>/
```

By placing the `?` after a `*`, `+`, or `?` quantifier, the expression is transformed from a greedy to a non-greedy, or minimal, match.

Anchors

Examples in previous topics in this section have only been concerned with finding chapter headings. Any occurrence of the string Chapter followed by a space and a number could be an actual chapter heading, or it could also be a cross-reference to another chapter. Since true chapter

headings always appear at the beginning of a line, it may be useful to devise a way to find only the headings and not the cross-references.

Anchors provide that capability. Anchors allow you to fix a regular expression to either the beginning or end of a line. They also allow you to create regular expressions that occur within a word, at the beginning of a word, or at the end of a word.

The following table contains the list of regular expression anchors and their meanings:

Character	Description
^	Matches the position at the beginning of the input string. If the RegExp object's Multiline property is set, ^ also matches the position following \n or \r.
\$	Matches the position at the end of the input string. If the RegExp object's Multiline property is set, \$ also matches the position pre\n or \r.
\b	Matches a word boundary, that is, the position between a word and a space.
\B	Matches a nonword boundary.

You cannot use a quantifier with an anchor. Since you cannot have more than one position immediately before or after a new line or word boundary, expressions such as ^* are not permitted.

- To match text at the beginning of a line of text, use the ^ character at the beginning of the regular expression. (Do not confuse this use of the ^ with the use within a bracket expression.)
- To match text at the end of a line of text, use the \$ character at the end of the regular expression.

To use anchors when searching for chapter headings, the following regular expression matches a chapter heading that contains no more than two following digits and that occurs at the beginning of a line:

```
/^Chapter [1-9][0-9]{0,1}/
```

Not only does a true chapter heading occur at the beginning of a line, it is also the only text on the line. It occurs at beginning of the line and also at the end of the same line.

The following expression ensures that the specified match only matches chapters and not cross-references. It does so by creating a regular expression that matches only at the beginning and end of a line of text.

```
/^Chapter [1-9][0-9]{0,1}$/
```

Matching word boundaries is a little different but adds a very important capability to regular expressions. A word boundary is the position between a word and a space. A non-word boundary is any other position.

The following expression matches the first three characters of the word *Chapter* because the characters appear following a word boundary:

```
/\bCha/
```

The position of the \b operator is critical. If it is at the beginning of a string to be matched, it looks for

the match at the beginning of the word. If it is at the end of the string, it looks for the match at the end of the word.

Example: The following expression matches the string `ter` in the word `Chapter` because it appears before a word boundary:

```
/ter\b/
```

Example: The following expression matches the string `apt` as it occurs in `Chapter` but not as it occurs in `aptitude`:

```
^Bapt/
```

The string `apt` occurs on a nonword boundary in the word `Chapter` but on a word boundary in the word `aptitude`.

For the `\B` nonword boundary operator, position is not important because the match is not relative to the beginning or end of a word.

Alternation and Grouping

Alternation uses the `|` character to allow a choice between two or more alternatives.

For example, you can expand the chapter heading regular expression to return more than just chapter headings. However, it is not a straightforward activity. Alternation matches the largest possible expression on either side of the `|` character.

Example:

```
/^Chapter|Section [1-9][0-9]{0,1}$/
```

This regular expression matches either the word `Chapter` at the beginning of a line, or the word `Section` and whatever numbers follow `Section` at the end of the line. If the input string is `Chapter 22`, the above expression only matches the word *Chapter*. If the input string is `Section 22`, the expression matches `Section 22`.

To make the regular expressions more responsive, you can use parentheses to limit the scope of the alternation. That is, to make sure that, it applies only to the two words *Chapter* and *Section*. However, parentheses are also used to create sub expressions and possibly capture them for later use. By adding parentheses in the appropriate places of the above regular expression, you can make the regular expression match either `Chapter 1` or `Section 3`.

Example:

```
/^(Chapter|Section) [1-9][0-9]{0,1}$/
```

The above regular expression uses parentheses to group `Chapter` and `Section` so the expression works properly:

Although these expressions work properly, the parentheses around `Chapter|Section` also cause either of the two matching words to be captured for future use. Since there is only one set of parentheses in the above expression, there is only one captured *sub match*.

In the above example, you merely want to use the parentheses to group a choice between the words *Chapter* and *Section*. To prevent the match from being saved for possible later use, place `?:` before the regular expression pattern inside the parentheses.

The following modification provides the same capability without saving the sub match:

```
/^(?:Chapter|Section) [1-9][0-9]{0,1}$/
```

In addition to the `?:` met characters, two other non-capturing met characters create something called *look ahead* matches.

A positive look ahead, which is specified using `?=`, matches the search string at any point where a matching regular expression pattern in parentheses begins.

A negative look ahead, which is specified using `?!`, matches the search string at any point where a string not matching the regular expression pattern begins.

Example:

```
/Windows(?:95 |98 |NT )/
```

Suppose you have a document that contains references to Windows 3.1, Windows 95, Windows 98, and Windows NT. Suppose further that you need to update the document by changing all references to Windows 95, Windows 98, and Windows NT to Windows 2000. The above regular expression, which is an example of a positive look ahead, matches Windows 95, Windows 98, and Windows NT:

Once a match is found, the search for the next match begins immediately following the matched text without including the characters in the look-ahead. For example, if the above expression matched Windows 98, the search resumes after Windows not after 98.

Back References

One of the most important features of regular expressions is the ability to store part of a matched pattern for later reuse. As you may recall, placing parentheses around a regular expression pattern or part of a pattern causes that part of the expression to be stored into a temporary buffer. You can override the capture by using the non-capturing met characters `?:`, `?=`, or `?!`.

Each captured sub match is stored as it is encountered from left to right in a regular expressions pattern. The buffer numbers begin at one and continue up to a maximum of 99 captured sub expressions. Each buffer can be accessed using `\n` where `n` is one or two decimal digits identifying a specific buffer.

One of the simplest, most useful applications of back references provides the ability to locate the occurrence of two identical, adjacent words in text.

Example: Is the cost of gasoline going up up?

The above sentence clearly has several duplicated words. It would be nice to devise a way to fix that sentence without looking for duplicates of every single word.

The following regular expression uses a single sub expression to do that:

```
/\b([a-z]+) \1\b/gi
```

- The captured expression, as specified by `[a-z]+`, includes one or more alphabetic characters.
- The second part of the regular expression is the reference to the previously captured sub match, that is, the second occurrence of the word just matched by the parenthetical expression.
- `\1` specifies the first sub match.
- The word boundary met characters ensure that only whole words are detected. Otherwise,

a phrase such as "is issued" or "this is" would be incorrectly identified by this expression.

- The global flag (g) following the regular expression indicates that the expression is applied to as many matches as it can find in the input string.
- The case insensitivity (i) flag at the end of the expression specifies case insensitivity. The multiline flag specifies that potential matches may occur on either side of a new line character.

You can use \$1 to specify string within the replace method which refers to the first saved sub match. If you had more than one sub match, you would refer to them consecutively by using \$2, \$3, and so on.

Back references can also break down a Universal Resource Indicator (URI) into its component parts. Assume that you want to break down the following URI to the protocol (ftp, http, and so on), the domain address, and the page/path:

`http://msdn.microsoft.com:80/scripting/default.htm`

Example: The following regular expressions provide that functionality:

```
/(\w+):\/\/([^\/:]+)(:\d*)?([^\# ]*)/
```

- The first parenthetical sub expression captures the protocol part of the Web address. That sub expression matches any word that precedes a colon and two forward slashes.
- The second parenthetical sub expression captures the domain address part of the address. That sub expression matches any sequence of characters that does not include ^, /, or : characters.
- The third parenthetical sub expression captures a port number if one is specified. That sub expression matches zero or more digits following a colon.
- Finally, the fourth parenthetical sub expression captures the path and/or page information specified by the Web address. That sub expression matches one or more characters other than # or the space character.

Applying the regular expression to the above URL, the sub matches contain the following:

- `RegExp.$1` contains "http"
- `RegExp.$2` contains "msdn.microsoft.com"
- `RegExp.$3` contains ":80"
- `RegExp.$4` contains "/scripting/default.htm"

Character Classes

Character class shorthands are shortcuts to specify character types like digits, whitespace, and so on.

Character Class	Match
<code>\d</code>	Matches a digit, same as <code>[0-9]</code>
<code>\D</code>	Matches a non-digit, same as <code>[^0-9]</code>
<code>\s</code>	Matches a white space character (space, tab, new line, etc.)
<code>\S</code>	Matches a non-white space character
<code>\w</code>	Matches a word character

\W	Matches a non-word character
\b	Matches a word-boundary (NOTE: within a class, matches a backspace)
\B	Matches a non-word boundary

EXAMPLES

String: Assembly Number: 0010-06845 Rev. 004 Quote QuantityExtended Quantity NOTES

Pattern: Assembly Number:\s+(\S+)\s+

Result: 0010-06845

String: Assembly Number: 0010-06845 Rev. 004 Quote QuantityExtended Quantity NOTES

Pattern: Assembly Number:.*Rev.\s+(\d+)\s+

Result: 004

String: ! ! ! !7040501 !K-PL-7040501-0-DBP ! 0 !

Pattern: \!\s+\!\s+\!\s+\!(\d+)?\s+\!*

Result: 7040501

String: ! ! ! !7Description: Test Phantom Board

Pattern: Description:(.*)

Result: Test Phantom Board

String: Assembly: BRB-99812

Pattern: Assembly:(.*)

Result: BRB-99812

String: Rev: 02

Pattern: Rev:(.*)

Result: 02

String: Item: 540-3655-01 ASSY,CHASSIS,OUTSRCD,BASE,MD Engineering Bill: No

Pattern: Item:\s+[0-9\-\]+\s+([\w,]+)

Result: ASSY,CHASSIS,OUTSRCD,BASE,MD

String: Item: 540-3655-01 ASSY,CHASSIS,OUTSRCD,BASE,MD Engineering Bill: No

Pattern: Item:\s+([0-9\-\-]+)

Result: 540-3655-01

String: Alternate: Primary UOM: EA Revision: 50 Date: 13-MAR-01 14:57

Pattern: UOM:\s+(\w+)

Result: EA

String: 061C553-01 BOM as of 10/01/01

Pattern: ^(\S+)-(\d+)\s\D+

Result: 061C553-01

String: Assy Name KIT, PCA, ROUTING ENGINE CONTROLLER II (REC-II)

Pattern: Assy Name(.*)

Result: KIT, PCA, ROUTING ENGINE ONTROLLER II (REC-II)

String: Assy # K92-000001-002

Pattern: Assy #(.*)

Result: K92-000001-002

String: Description: CORE2.0 MODULE DUAL VOLTAGE

Pattern: Description:\s+(\S+)

Result: CORE2.0

String: ERP ID: NTBW30BA

Pattern: ERP ID:\s+(\S+)

Result: NTBW30BA

String: Design Rev/CIL: 0010

Pattern: Design Rev/CIL:\s+(\S+)

Result: 0010

String: |PCBA,POLARIS BASE ASSY Valid: 24.08.2001|

Pattern: \s+(.*)\sValid.*

Result: PCBA,POLARIS BASE ASSY

String: Material SCPLRSXXXXXXPC3 01 Alt. Usage 1

Pattern: .*Material\s+(\S+)\s+(\S+)\s.*

Result: SCPLRSXXXXXXPC3
