HEXAGON
PPM

Intergraph Smart Data Validator

Getting Started Guide

Version 2018 (3.0)
September 2017

Copyright

U.S. Government Restricted Rights Legend

Use, duplication, or disclosure by the government is subject to restrictions as set forth below. For civilian agencies: This was developed at private expense and is "restricted computer software" submitted with restricted rights in accordance with subparagraphs (a) through (d) of the Commercial Computer Software - Restricted Rights clause at 52.227-19 of the Federal Acquisition Regulations ("FAR") and its successors, and is unpublished and all rights are reserved under the copyright laws of the United States. For units of the Department of Defense ("DoD"): This is "commercial computer software" as defined at DFARS 252.227-7014 and the rights of the Government are as specified at DFARS 227.7202-3.

Unpublished - rights reserved under the copyright laws of the United States.

Intergraph Corporation
305 Intergraph Way
Madison, AL 35758

Documentation

Documentation shall mean, whether in electronic or printed form, User's Guides, Installation Guides, Reference Guides, Administrator's Guides, Customization Guides, Programmer's Guides, Configuration Guides and Help Guides delivered with a particular software product.

Other Documentation

Other Documentation shall mean, whether in electronic or printed form and delivered with software or on Intergraph Smart Support, SharePoint, or box.net, any documentation related to work processes, workflows, and best practices that is provided by Intergraph as guidance for using a software product.

Terms of Use

a. Use of a software product and Documentation is subject to the Software License Agreement ("SLA") delivered with the software product unless the Licensee has a valid signed license for this software product with Intergraph Corporation. If the Licensee has a valid signed license for this software product with Intergraph Corporation, the valid signed license shall take precedence and govern the use of this software product and Documentation. Subject to the terms contained within the applicable license agreement, Intergraph Corporation gives Licensee permission to print a reasonable number of copies of the Documentation as defined in the applicable license agreement and delivered with the software product for Licensee's internal, non-commercial use. The Documentation may not be printed for resale or redistribution.

b. For use of Documentation or Other Documentation where end user does not receive a SLA or does not have a valid license agreement with Intergraph, Intergraph grants the Licensee a non-exclusive license to use the Documentation or Other Documentation for Licensee’s internal non-commercial use. Intergraph Corporation gives Licensee permission to print a reasonable number of copies of Other Documentation for Licensee’s internal, non-commercial use. The Other Documentation may not be printed for resale or redistribution. This license contained in this subsection b) may be terminated at any time and for any reason by Intergraph Corporation by giving written notice to Licensee.

Disclaimer of Warranties

Except for any express warranties as may be stated in the SLA or separate license or separate terms and conditions, Intergraph Corporation disclaims any and all express or implied warranties including, but not limited to the implied warranties of merchantability and fitness for a particular purpose and nothing stated in, or implied by, this document or its contents shall be considered or deemed a modification or amendment of such disclaimer. Intergraph believes the information in this publication is accurate as of its publication date.

The information and the software discussed in this document are subject to change without notice and are subject to applicable technical product descriptions. Intergraph Corporation is not responsible for any error that may appear in this document.

The software, Documentation and Other Documentation discussed in this document are furnished under a license and may be used or copied only in accordance with the terms of this license. THE USER OF THE SOFTWARE IS EXPECTED TO MAKE THE FINAL EVALUATION AS TO THE USEFULNESS OF THE SOFTWARE IN HIS OWN ENVIRONMENT.

Intergraph is not responsible for the accuracy of delivered data including, but not limited to, catalog, reference and symbol data. Users should verify for themselves that the data is accurate and suitable for their project work.

Limitation of Damages

IN NO EVENT WILL INTERGRAPH CORPORATION BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL INCIDENTAL, SPECIAL, OR PUNITIVE DAMAGES, INCLUDING BUT NOT LIMITED TO, LOSS OF USE OR PRODUCTION, LOSS OF REVENUE OR PROFIT, LOSS OF DATA, OR CLAIMS OF THIRD PARTIES, EVEN IF INTERGRAPH CORPORATION HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

UNDER NO CIRCUMSTANCES SHALL INTERGRAPH CORPORATION'S LIABILITY EXCEED THE AMOUNT THAT INTERGRAPH CORPORATION HAS BEEN PAID BY LICENSEE UNDER THIS AGREEMENT AT THE TIME THE CLAIM IS MADE. EXCEPT WHERE PROHIBITED BY APPLICABLE LAW, NO CLAIM, REGARDLESS OF FORM, ARISING OUT OF OR IN CONNECTION WITH THE SUBJECT MATTER OF THIS DOCUMENT MAY BE BROUGHT BY LICENSEE MORE THAN TWO (2) YEARS AFTER THE EVENT GIVING RISE TO THE CAUSE OF ACTION HAS OCCURRED.

IF UNDER THE LAW RULED APPLICABLE ANY PART OF THIS SECTION IS INVALID, THEN INTERGRAPH LIMITS ITS LIABILITY TO THE MAXIMUM EXTENT ALLOWED BY SAID LAW.

Export Controls

Intergraph Corporation's commercial-off-the-shelf software products, customized software and/or third-party software, including any technical data related thereto ("Technical Data"), obtained from Intergraph Corporation, its subsidiaries or distributors, is subject to the export control laws and regulations of the United States of America. Diversion contrary to U.S. law is prohibited. To the extent prohibited by United States or other applicable laws, Intergraph Corporation software products, customized software, Technical Data, and/or third-party software, or any derivatives thereof, obtained from Intergraph Corporation, its subsidiaries or distributors must not be exported or re-exported, directly or indirectly (including via remote access) under the following circumstances:

a. To Cuba, Iran, North Korea, the Crimean region of Ukraine, or Syria, or any national of these countries or territories.

b. To any person or entity listed on any United States government denial list, including, but not limited to, the United States Department of Commerce Denied Persons, Entities, and Unverified Lists, the United States Department of Treasury Specially Designated Nationals List, and the United States Department of State Debarred List (https://build.export.gov/main/ecr/eg_main_023148).

c. To any entity when Customer knows, or has reason to know, the end use of the software product, customized software, Technical Data and/or third-party software obtained from Intergraph Corporation, its subsidiaries or distributors is related to the design, development, production, or use of missiles, chemical, biological, or nuclear weapons, or other un-safeguarded or sensitive nuclear uses.

d. To any entity when Customer knows, or has reason to know, that an illegal reshipment will take place.

e. Any questions regarding export/re-export of relevant Intergraph Corporation software product, customized software, Technical Data and/or third-party software obtained from Intergraph Corporation, its subsidiaries or distributors, should be addressed to PPM's Export Compliance Department, 305 Intergraph Way, Madison, Alabama 35758 USA or at exportcompliance@intergraph.com. Customer shall hold harmless and indemnify PPM and Hexagon Group Company for any causes of action, claims, costs, expenses and/or damages resulting to PPM or Hexagon Group Company from a breach by Customer.

Trademarks

Intergraph®, the Intergraph logo®, Intergraph Smart®, SmartPlant®, SmartMarine, SmartSketch®, SmartPlant Cloud®, PDS®, FrameWorks®, I-Route, I-Export, ISOGEN®, SPOOLGEN, SupportManager®, SupportModeler®, SAPPHIRE®, TANK, PV Elite®, CADWorx®, CADWorx DraftPro®, GTSTRUDL®, and CAESAR II® are trademarks or registered trademarks of Intergraph Corporation or its affiliates, parents, subsidiaries. Hexagon and the Hexagon logo are registered trademarks of Hexagon AB or its subsidiaries. Microsoft and Windows are registered trademarks of Microsoft Corporation. MicroStation is a registered trademark of Bentley Systems, Inc. Other brands and product names are trademarks of their respective owners.

Contents

Preface

This guide introduces you to the basics of Intergraph Smart® Data Validator to help get you started.

Smart Data Validator Product Documentation

Smart Data Validator documentation is available as Help and as PDF files. To view printable guides for Smart Data Validator, click **Help > Printable Guides** in the software.

Hexagon PPM gives its customers permission to print as many copies of the delivered PDF files as they need for their non-commercial use. Do not print the PDF files for resale or redistribution.

Release Bulletin

- *Smart Data Validator Release Bulletin* - Provides information on Smart Data Validator features for the current release.

Installation and Overviews

- *Smart Data Validator Installation and Setup Guide* - Provides installation and setup instructions for Smart Data Validator.
- *Smart Data Validator Getting Started Guide* - Provides overview information to help users know about the Smart Data Validator functionalities.

Administration User's Guide

- *Smart Data Validator Administration User's Guide* - Provides instructions for administering the configuration of Smart Data Validator.

Job Management User's Guide

- *Smart Data Validator Job Management User's Guide* - Provides instructions on the management of jobs that process imported data to a specified workflow and export that data to a target system or target systems.

Customization Guide

- *Smart Data Validator Customization Guide* - Provides instructions for creating and modifying the various areas of customization possible in Smart Data Validator and details on provided samples included with the release.

Troubleshooting Guide

- *Smart Data Validator Troubleshooting Guide* - Provides information about troubleshooting the installation and configuration of Smart Data Validator.

Customer Support

For the latest support information for this product, use a web browser to connect to *http://hexagonppm.com/ppm-support* (*http://hexagonppm.com/ppm-support*). Also, you can submit any documentation comments or suggestions you might have on our support site.

To access the Technical User Forum, go to

http://www.intergraph.com/ppm/customers/tuf/foundation.aspx (*http://www.intergraph.com/ppm/customers/tuf/foundation.aspx*)

SECTION 1

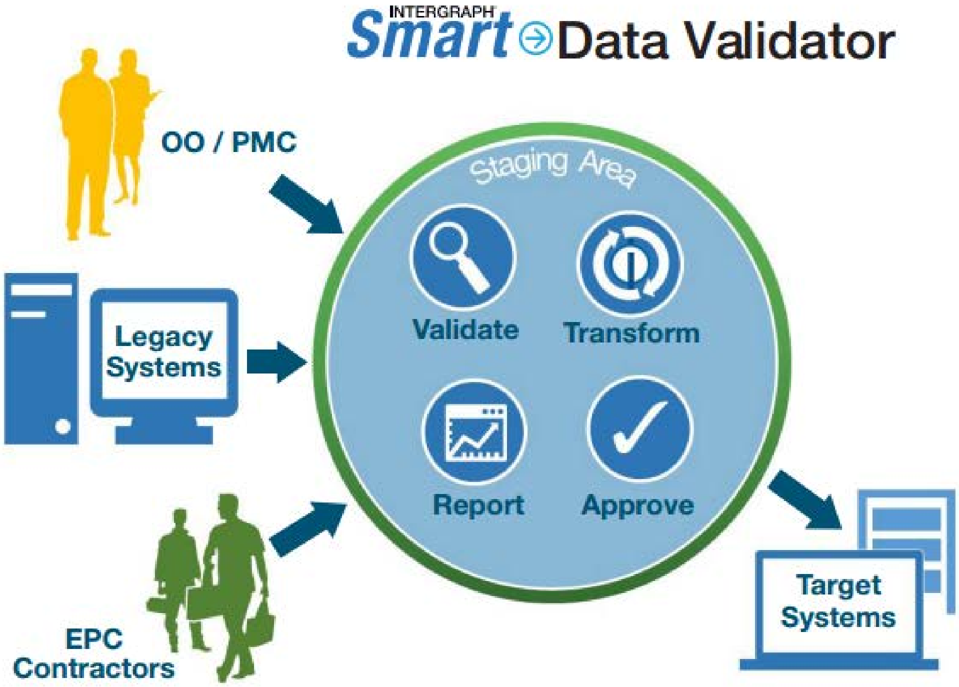
Introduction to Smart Data Validator

Smart Data Validator is a comprehensive data validation, transformation and migration platform that enables companies to ensure the quality of information and significantly reduce the time and costs associated with data take-on.

Smart Data Validator supports the import, validation, and export of data, providing consistency and traceability of the entire data migration process. It ensures the quality of data delivered from contractors, suppliers and vendors, creating a high-quality data basis for operations.

Smart Data Validator enables Engineering, Procurement, Construction companies (EPCs) and owner operators (OO) to:

- Check the quality of information being transferred
- Keep records of which checks have been performed and results obtained
- Load approved information into target systems
- Route unapproved information back to responsible parties for remediation



Smart Data Validator monitors and maintains the data quality when data is being moved from one source to another, during project handover, brownfield data take-on, data loads and migrations. Smart Data Validator manages the data for quality assurance purposes, and provides reports that can be used for both greenfield and brownfield environment data to help minimize quality issues.

Smart Data Validator Workflows

The validation of data imported into the staging area and the export of the objects to one or more target systems are configured using SmartPlant® Foundation workflows. Smart Data Validator is delivered with three workflow templates:

- Import Validate Export
- Import Validate Implicit Delete Export
- Import Validate Export Delete Job

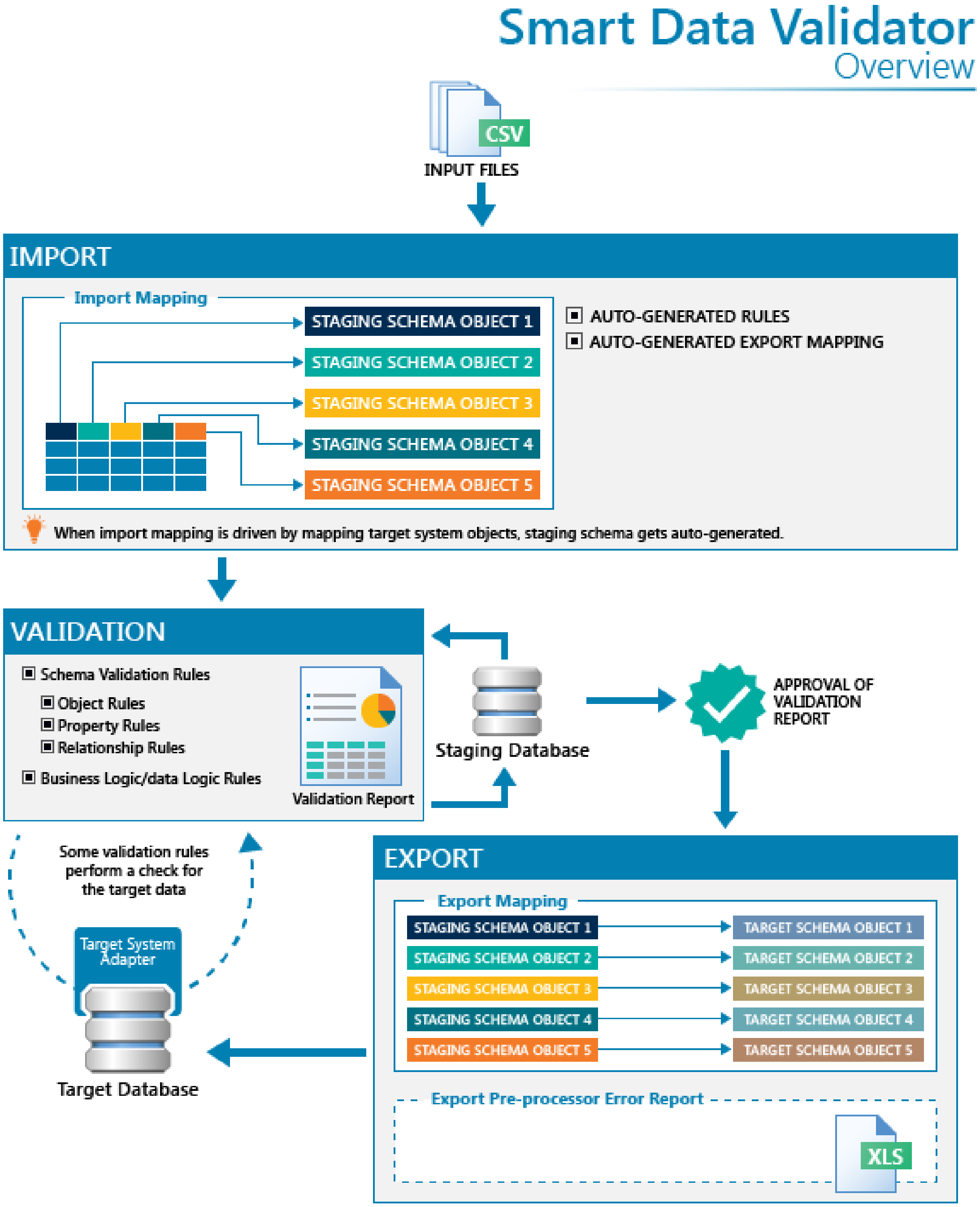
The workflow brings each component together as a process step and ensures that the imported data are validated before export to specified target systems.

Smart Data Validator in a snapshot

In the following graphic, you'll see a high level representation of the core functionalities of Smart Data Validator. This includes:

- Input – Column headers of the input CSV file are mapped to the staging objects.
- Validation – Imported data in the staging database is verified for accuracy with the validation rules.

- Export – After the approval of the validation report, the data is exported into the target system.



Import

Smart Data Validator imports files in CSV format into the staging system where the validation takes place. Smart Data Validator uses import definitions to map the column headers in the input CSV file to the correct column headers for objects, properties, and relationships in the staging area. The import mapping not only uses the physical column headers that already exist in the input data file, but it also supports other column headers that are computed or provided by the user during the import process.

⚠ **CAUTION** We recommend using a text editor like Notepad to modify your CSV files to avoid any unexpected results.

Import definition consists of all the column headers that have defined in it. You can define the column headers for each import definition to map the column data from the imported CSV files to the existing columns in the target system. The mapping can be driven by the target system database and defined manually using the free text option, or you can use existing mapping in the local staging area. You can create, edit, and modify each column header mapping to match the class definitions, objects, properties, relationships, and relationship properties between the imported data objects and target system objects.

Column Headers

Column headers are used to map the data in the CSV file to the correct locations in the staging area and target systems, and supports a number of different column types. The column type you select when defining the column header determines the appearance of the Column Header and Mapping dialog boxes that you use later in the import mapping process. Smart Data Validator supports the following types of columns:

- **Physical** - Maps to a specific column in the input data source.
- **Computed** - Generates a new column using data from other columns or sources, such as environment type variables. This provides the ability to use functions. For example, you can enter a function in the Computed API box to replace original entries with new entries.
- **Constant** - Allows you to use a string constant.
- **Prompted** - Prompts the user for a value when a job is run, based on the job definition. The provided value is stored with the job definition for later use, if necessary.
- **Prompted API** - Generates a list of prompts generated from other columns or sources, such as environment type variables, and displays this list using functions.
- **Prompted Picklist** - Allows the user to pick an entry from a defined enumerated list in the local staging area when running a job.

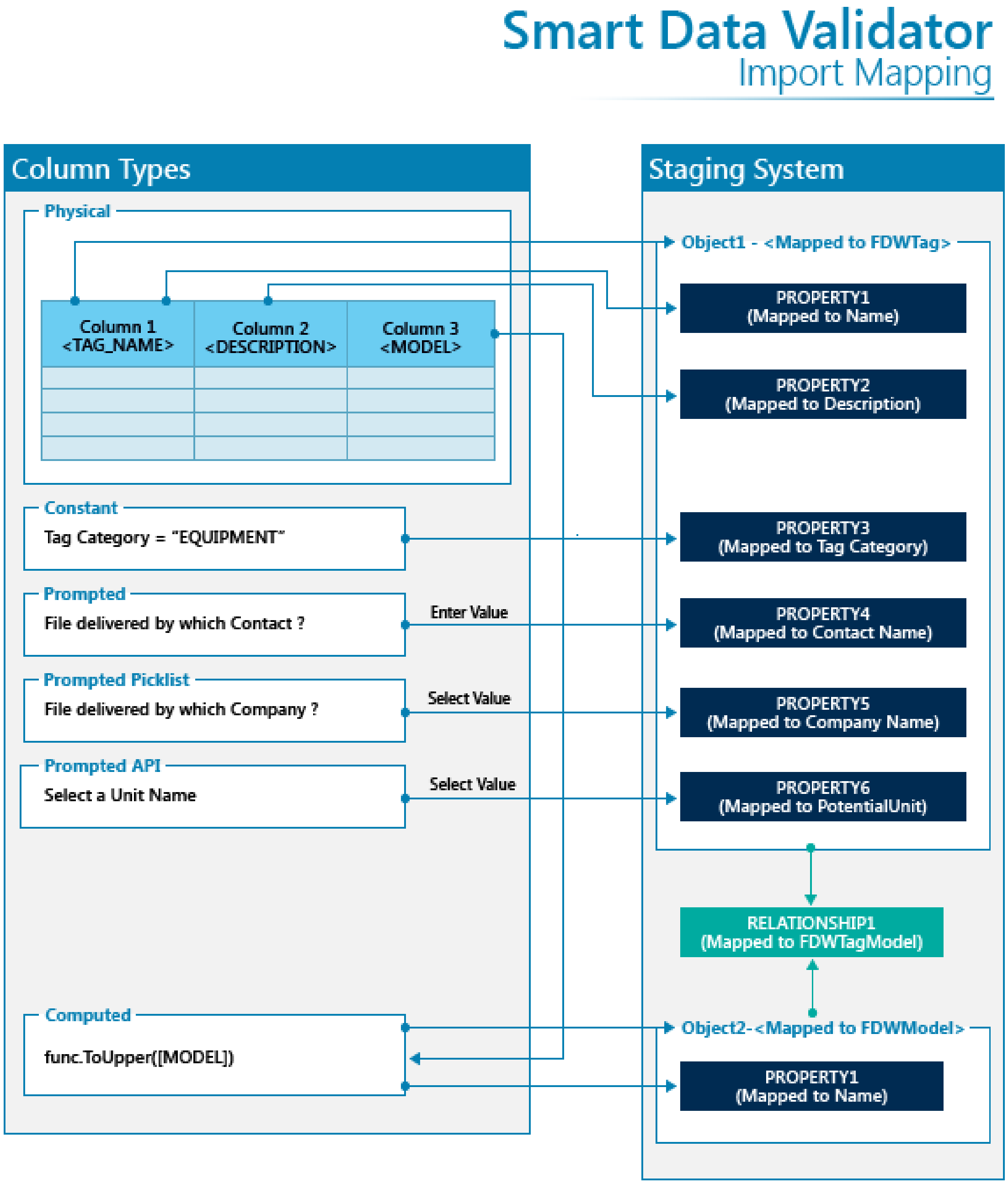
In the following graphic, you'll see that column headers mapped from the input CSV file to the staging objects. Additional column mappings allow users to enter values and the application to compute values during the import process.

Explanation of the mapping in the graphic:

- Column 1 of input CSV file is mapped to the object called FDWTag and the Name property of it.
- Column 2 of input CSV file is mapped to the Description property of the tag object.
- Column 3 of input CSV file is not mapped directly, but is used in a computed column which is used to map to the object called FDWModel and the Name property of it.

- The constant value “EQUIPMENT” is mapped to the Tag Category property of the tag object.
- A prompted value allows the user to enter a contact name that is mapped to the Contact Name property of the tag object.
- A prompted column allows the user to select a company name from the list, which mapped to the Company Name property of the tag object.
- A prompted API allows the user to select a unit from the list of units available in the staging system. This is mapped to the PotentialUnit property of the tag object.

- A computed column takes the input value from the Column 3 of the input CSV file and changes it to upper case. This is mapped to the object called FDWModel and the Name property of it.


Smart Data Validator
Import Mapping
Column Types
Physical
Column 1 <TAG_NAME>
Column 2 <DESCRIPTION>
Column 3 <MODEL>
Constant
Tag Category = "EQUIPMENT"
Prompted
File delivered by which Contact ?
Enter Value
Prompted Picklist
File delivered by which Company ?
Select Value
Prompted API
Select a Unit Name
Select Value
Computed
func.ToUpper([MODEL])
Staging System
Object1 - <Mapped to FDWTag>
PROPERTY1 (Mapped to Name)
PROPERTY2 (Mapped to Description)
PROPERTY3 (Mapped to Tag Category)
PROPERTY4 (Mapped to Contact Name)
PROPERTY5 (Mapped to Company Name)
PROPERTY6 (Mapped to PotentialUnit)
RELATIONSHIP1 (Mapped to FDWTagModel)
Object2-<Mapped to FDWModel>
PROPERTY1 (Mapped to Name)

A staging object might repesent an object or an object property in the staging system.

You must create column mappings for each column header in the import definition so that imported data matches the corresponding data column format in the staging area or target system. You can search for the objects and properties found locally or add new objects and properties, then create the new mappings to match the target system objects and properties.

There are four types of column header mappings you can create for each column header.

- Object mapping
- Property mapping
- Relationship mapping
- Relationship property mapping

Actions

When you define a column header in Smart Data Validator, you assign it an action. This action indicates what Smart Data Validator will do with the data imported from the related column in the CSV file. The kinds of operations supported varies, depending on whether the column header is mapped to an object, a property, or a relationship.

Object actions

Action	**Description**
Check Exists	Checks to see if the object exists in the target system.
Create Update	Creates a new object or updates the object if it already exists in the target system.
Delete	Deletes an object in the target system.
No Action	No action is taken on the object in the target system.
Rename	Renames the object in the target system.
Terminate	Terminates the object in the target system.
Update	Updates the object in the target system.
Validate Only	Validates the data only. No action is taken on the object in the target system, but if errors are found then they are propagated to the related objects.

Property actions

Action	**Description**
Check Staging File Exists	Checks to see if the staging property file exists in the staging area.

Action	Description
Check Target System File Exists	Checks to see if the target system property exists in the target system.
Compare Create	Invokes the compare value rule, which creates an error if the target system value is different. The target system property is overwritten if the value is not set.
Compare Value	Invokes the compare value rule, which never changes the value in the target system.
Create Update	Creates a new property and updates the existing property in the target system.
Delete	Deletes a property in the target system.
No Action	No action is taken on the property in the target system.
Terminate	Terminates the property in the target system.
Validate Only	Validates the property only. No action is taken on the object in the target system, but if errors are found then they are propagated to the related objects.

NOTE Using Smart Data Validator, you cannot delete or terminate the required properties of an object that exists in the target system.

Relationship actions

Action	Description
Check Exists	Checks to see if the relationship exists in the target system.
Create Update	Creates a new relationship and updates the existing relationship in the target system.
Delete	Deletes a relationship in the target system.
No Action	No action is taken on the relationship in the target system.
Terminate	Terminates the relationship in the target system.

Relationship property actions

Action	Description
Create Update	Creates a new relationship property and updates the existing relationship property in the target system.

Action	Description
No Action	No action is taken on the relationship property in the target system.
Compare Create	Invokes the compare value rule, which creates an error if the target system value is different. The target system relationship property is only overwritten if the value is not set.
Compare Value	Invokes the compare value rule, which never changes the value in the target system.

Validation

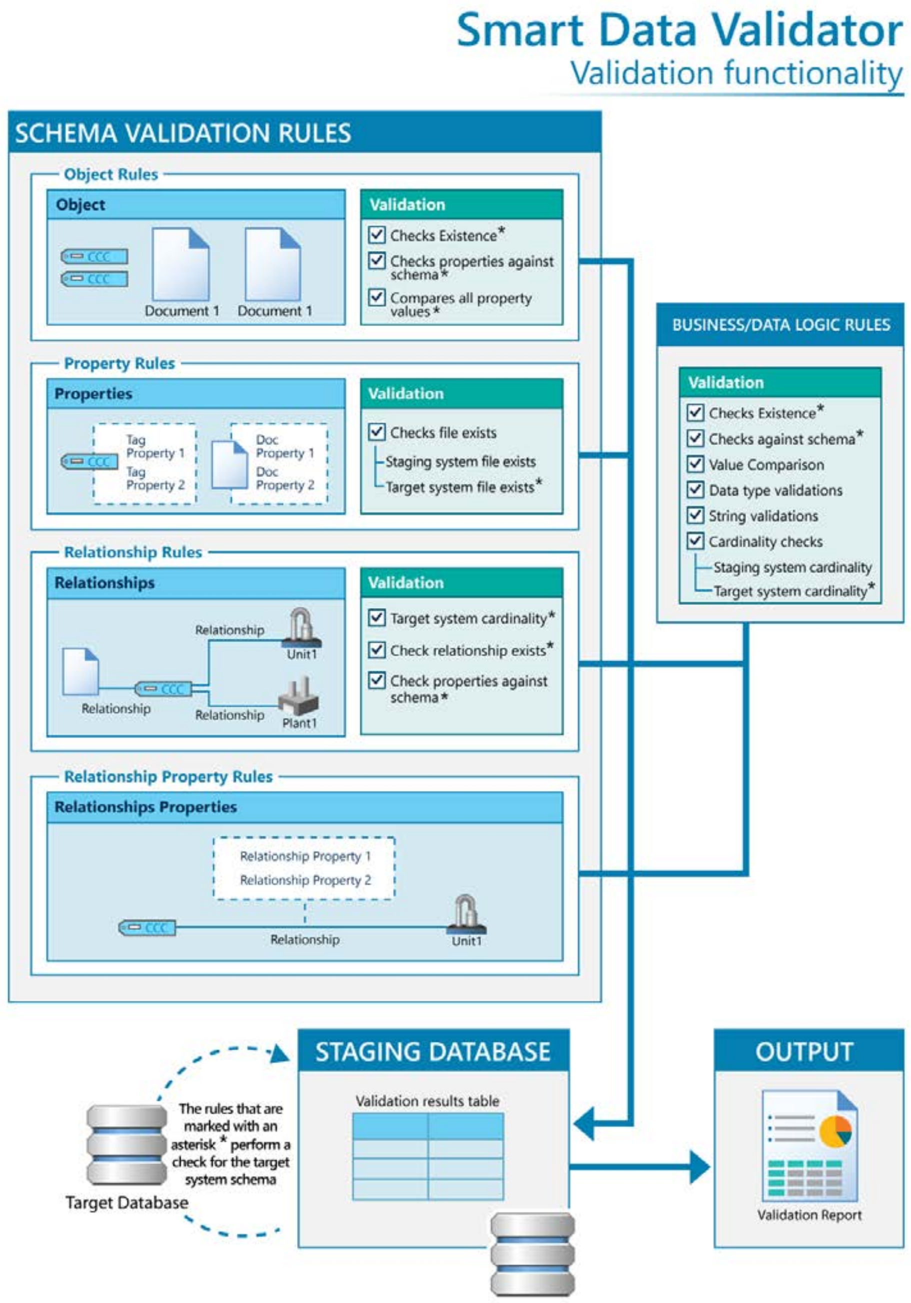
Validation of data is performed on the data that is imported into the staging area. Validation ensures the imported objects meet the criteria mentioned in the selected rule sets.

Validation rules verify that the data in an imported record meets the data standards specified for the target system. Each validation rule contains a formula that evaluates the imported data and verifies that it matches the existing criteria and hierarchy found in the target system database. Each validation rule includes an error message that is recorded in a validation report when the rule returns an invalid value.

All the validation rules that will be used in a job are grouped into a set. You can create a validation rule set, and then create the individual rules that will make up that set.

NOTE When any validation rule is deleted that contains class definitions, relationship definitions, or property definitions no longer used by Smart Data Validator, those schema items are automatically removed from the database.

In the following graphic, you'll see the schema validation and the business/data logic rules validate the imported data in the staging database. Some of the rules also check the target system schema. The validation report displays errors found in the imported data.



If there are more than one target database then only one is used to validate against the target schema.

TIP You can configure Smart Data Validator to automatically generate validation rules when you create your import mapping.

Validation rules and their descriptions

The following table lists all the validation rules and their descriptions delivered with Smart Data Validator.

Rule Name	**Description**
Check Claimed to Sub Configuration	Checks to see if a plant-level item has been claimed to the project.
Check Duplicate Issue Dates	Checks to see if there are revisions with duplicate issue dates for the same master document.
Check File Exists	Checks to see if a file exists in the specified path and that security has been set to configure access permissions.
Check File Exists for Document Revision	Checks to see if any file is attached to the document revision if the document revision is not a planned revision.
Check For Cyclic Relationships	Checks to determine if the same object is set on both End 1 and End 2 of a relationship definition.
Check Is Claimable	Checks to see if the data object is claimable into the SmartPlant Foundation collaboration project and prevents the object from being loaded.
Check Is Revisable	Checks to see if the document is revisable to the SmartPlant Foundation collaboration project and prevents the document revision from being loaded.
Check Object Exists	Checks to see if the object exists in the target system already.
Check Parallel Claim	Checks to see if the item is claimed to a parallel project under the same plant.
Check Revised to Sub Config	Checks to see if a plant-level document has been revised to the project.
Check Revised to Parallel Config	Checks to see if the document item has been revised to a parallel project under the same plant.
Check Properties Against Schema	Checks to see if the target property definitions are valid for the target object type and if the values are valid for this property, including whether required values are present.

Rule Name	Description
Check Relationship Exists	Checks to see if the relationship exists in the target system.
Check Unique Key	Checks to see if a specific unique key already exists elsewhere in the system and cannot be implicitly claimed. This check runs only on the target system and detects duplicates in the source files.
Check Value and UOM	Checks to see if the UOM of the object property is the same as the UOM configured in the target system. It also checks to see if the object property value is defined within a specified range of minimum and maximum values.
Compare All Property Values For Object	Checks to see if all the staging system object property values are the same as the target system property values.
Compare Value	Checks to see if the staging system property value is the same as the target system property value.
Date Time	Checks to see if an object property must validate against some date and time, YMD, or DMY.
Double	Checks to see if an object property value has a minimum and a maximum value.
Integer	Checks to see if an object property value is defined within a specified range by a minimum and maximum value.
Regular Expression	Checks to see if an object property value adheres to a specified pattern.
Staging System Cardinality	Checks for the cardinalities between the related objects.
String Does Not Start With	Checks to see that the value does not start with the given value.
String Length	Checks to see if the length of an object property is within a specified minimum and maximum limit or adheres to a specified pattern.
String Not Equal To	Checks to see that the value is not equal to the given value.

Rule Name	Description
Target System Cardinality	Checks for the minimum and maximum cardinality values between related objects from the target system.

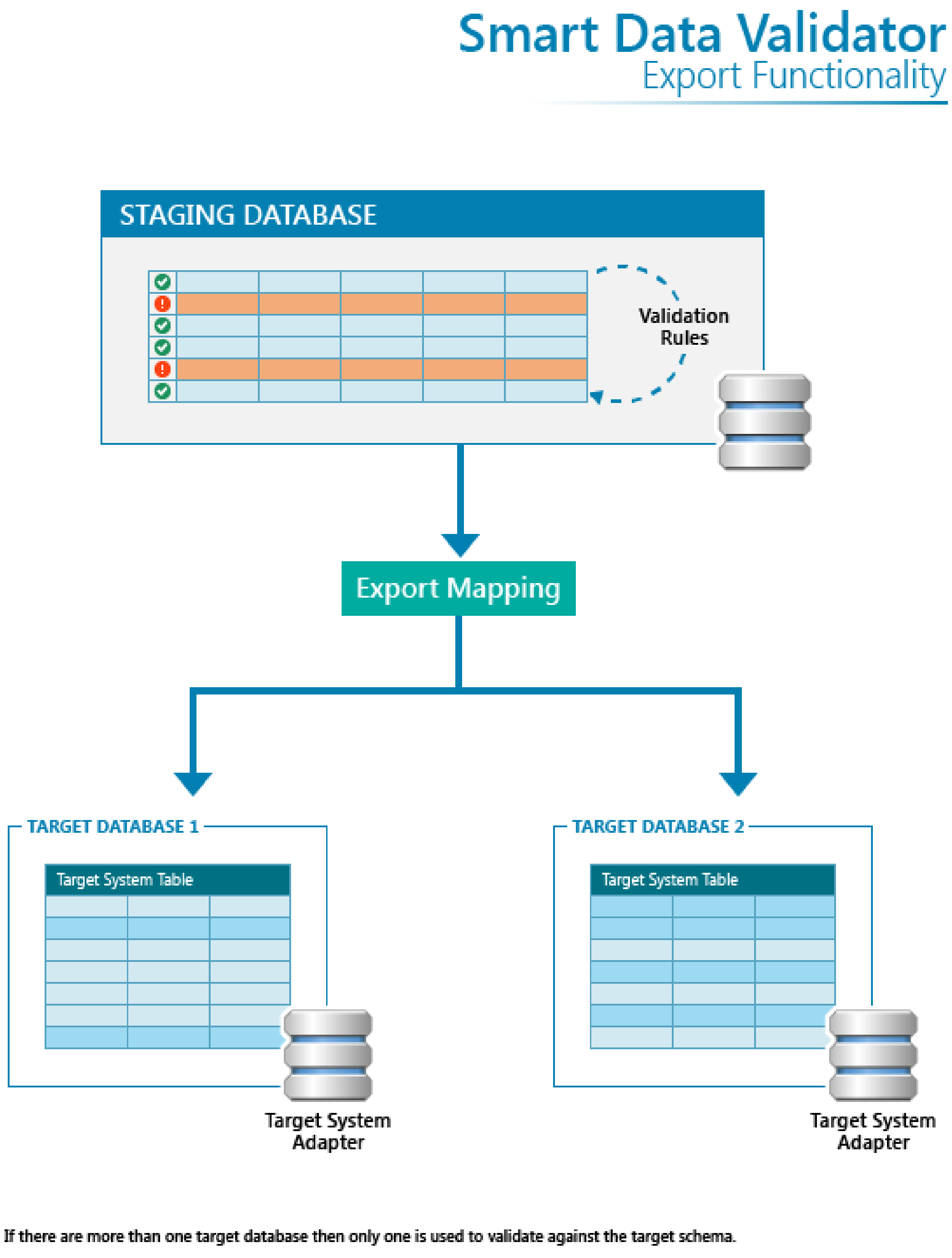
Configure the automatic generation of validation rules

When you create your import mappings from the target system, you can automatically generate validation rules that are set on the import definition. By selecting this option, the system determines which rules it will run for each action, saving configuration time. You can configure which rules are automatically generated for each action in the Smart Data Validator system options. The actions and rules are configurable.

Export

You can export data to a target system by mapping the imported staging area data objects to the existing objects, properties, and relationships found in the target system. You can manually configure the export mapping for each object in the imported data, or you can automatically generate the export mapping from the target system by selecting the **Auto Generate Export Mapping** option when you create a new import definition. The export mapping can be further refined for each stage object to target object mapping by selecting specific property-to-property, relationship-to-property, and relationship-to-relationship mappings.

TIP If you have two target systems that use the same model or schema, they can share the same export mapping.



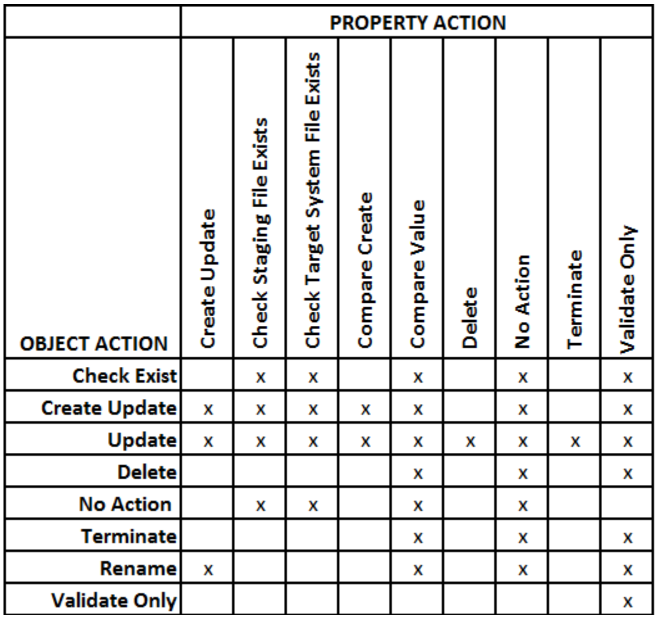
Smart Data Validator
Export Functionality

If there are more than one target database then only one is used to validate against the target schema.

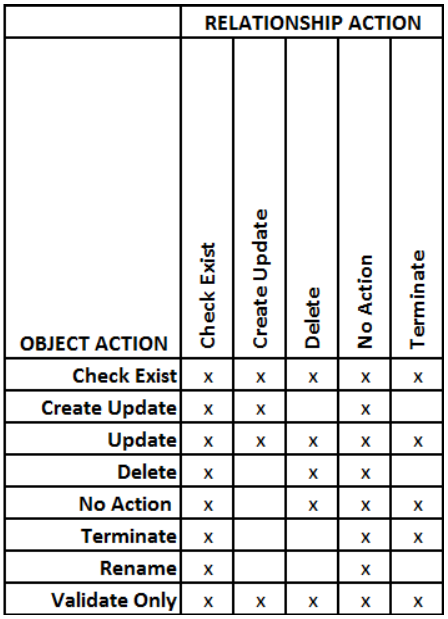
Export mapping actions

Export mappings are restricted by what actions you select when creating the object, as there are permitted combinations of actions. For example, the actions available for the Delete class definition mapping are Compare Value and No Action.

The following table shows the supported combination of default actions available in Smart Data Validator.



	PROPERTY ACTION								
OBJECT ACTION	Create Update	Check Staging File Exists	Check Target System File Exists	Compare Create	Compare Value	Delete	No Action	Terminate	Validate Only
Check Exist		x	x		x		x		x
Create Update	x	x	x	x	x		x		x
Update	x	x	x	x	x	x	x	x	x
Delete					x		x		x
No Action		x	x		x		x		
Terminate					x		x		x
Rename	x				x		x		x
Validate Only									x



	RELATIONSHIP ACTION				
OBJECT ACTION	Check Exist	Create Update	Delete	No Action	Terminate
Check Exist	x	x	x	x	x
Create Update	x	x		x	
Update	x	x	x	x	x
Delete	x		x	x	
No Action	x		x	x	x
Terminate	x			x	x
Rename	x			x	
Validate Only	x	x	x	x	x

★ IMPORTANT Only one **Check Staging File Exists** property action or one **Check Target System File Exists** property action is allowed per object. If there is more than one, the export is blocked.

Target Systems

A target system is a defined system or database that is the destination for validated data imported into Smart Data Validator. Target systems can be created in Smart Data Validator Administration with the URL address for the target system adapter.

When your target system is based on SmartPlant Foundation, you can use it to drive your import mapping, validation, and export mapping. Other target systems can be used just for export mapping.

NOTE When creating a target system, you must define the access groups that will have permission to select that target system during job creation.

Job Definition

A job definition is a combined set of components configured into a template for the import, validation, and export of data to one or more target systems. The job definition manages the quality of the data to be loaded into the target system configuration. You can define the import mappings, export mappings, validation rule sets, and delete rule sets.

When a job definition has been configured, Smart Data Validator Job Management uses it to load jobs containing data into a target system.

Each job definition is configured using the following components:

- **Basic Details** - The job description name and description, as well as the workflow used to export data to the primary target system.
- **Workflow -** The defined set of steps that drive the behavior of the job running through Smart Data Validator. The following workflow templates are delivered with Smart Data Validator: Import Validate Export, Import Validate Delete Export, and Import Validate Export Delete Job.
- **Target Systems -** One or more systems used to validate and export the data.
- **Import Definitions** - The definitions used to import the data.
- **Rule Sets** - The rule sets used to validate and verify the imported data.
- **Implicit Delete Rules** - The rules used to delete data.
- **Validation Report Definitions -** The report definition used to create the validation report.
- **Access Groups** - The access groups that have permission to use the job definition to create a new job in Smart Data Validator Job Management.

SECTION 2

Terms and acronyms

Use the following topics to learn more about key terms and acronyms in Smart Data Validator.

Terms

Term	Meaning
Actions	An indicator of what Smart Data Validator will do with the object data in the validation and export process, such as update and delete. These kinds of operations vary, depending on whether the column header is mapped to an object, a property, or a relationship.
Brownfield data	An existing project or area that has constraints imposed due to prior work and contains existing data.
Column Headers	They are used in Smart Data Validator as the basis for mapping new data in columns to match an existing structured database.
Export Mapping	The process that maps the imported objects, classes, and properties to the existing structure of the objects, classes, and properties found in a target system.
Export Process	Manages the loading of the data into the final destination system using a defined mapped, based on the structure of the target system.
Functions	A computed functional code run at import, where the output value of a function depends only on the arguments that are input to the function.
Greenfield data	A project or area that is completely new and does not have any constraints imposed by prior work or existing data.
Import Definition	A defined mapping of imported file objects, properties, and relationships from existing column headers to objects, properties, and relationships found in the staging area database during the import process.
Import Mapping	The process that maps the imported objects, classes, and properties to the existing structure of the objects, classes, and properties found in a staging system.
Import Process	The process that manages the import of data in the data file or files to the staging area database using a defined mapping.

Term	Meaning
Job	A defined object that carries information for the progress of data through a selected workflow in Smart Data Validator Job Management, such as when data is imported, validated, and exported to a target system.
Job Definition	A combined set of components configured for the import, validation, and export of data to a specific target system or multiple target systems.
Mapping	A defined process where data is correlated from existing column headers for the objects, properties, and relationships in the imported data to the correct column headers for the objects, properties, and relationships found in another database, such as the staging database or target system.
Rules	A logical formula used to evaluate and verify whether the data in an imported record meets the requirements and data standards specified for the target database system.
Rule Sets	A combined set of rules that can be run as a set during job processing.
Staging System	The staging system or staging area or staging database used is the first part of a combined set of processes in Smart Data Validator, where imported data is held and validated before being exported to a Smart Data Validator site.
Target System	The system or database used as the destination for exported data that has been validated before export. It is also used as the basis for creating import mappings, implicit delete, and validation rules.
Terminate	To change the status of an object to terminated without removing it from the database. Terminating objects, instead of deleting them, allows you to continue to see the history of the object after termination.
Unique Key	A set of values guaranteed to be unique for each object in a relation, and can be used to identify objects in the target system.
Validation Process	A process that evaluates the imported data against a defined set of rules to ensure the validity of the data.
Validation Rule	A logical formula used to evaluate the imported data in one or more fields to determine whether it matches the existing criteria and hierarchy found in the target system database. There are two types of validation rules: ▪ Rules that determine if the data is valid for the target system schema (these rules can be auto-generated ▪ Rules that determine if the data meets specific business criteria, such as naming format.

Acronyms

Acronym	Meaning
API	Application Programming Interface
CSV	Comma Separated Values
ENS	Engineering Numbering System
EPC	Engineering, Procurement, Construction
GUID	Global Unique Identifier
NLF	Delimited Text Files
PMC	Project Management Consultants
OO	Owner Operators
SDV	Smart Data Validator
SI	International System of Units
UID	Unique Identifier
UOM	Unit of Measure
URL	Uniform Resource Locator
XML	Extensible Markup Language

SECTION 3

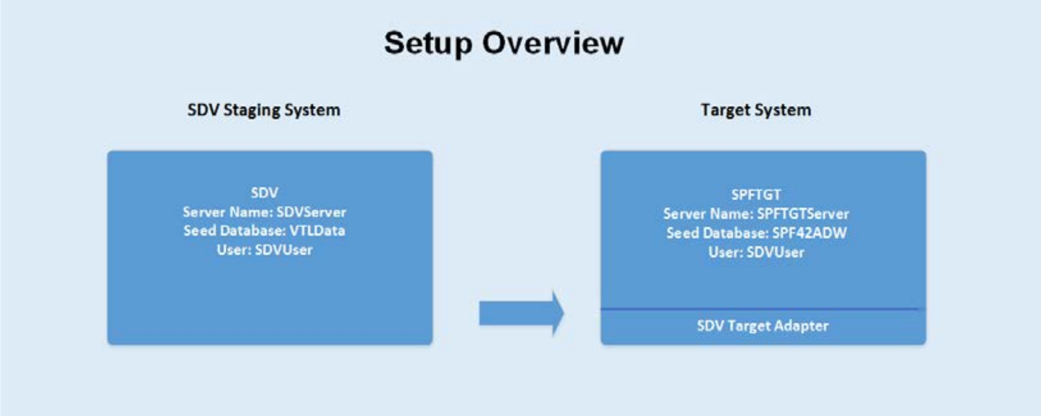
Getting Started

Smart Data Validator provides a hands-on way for you to get started in your new system by providing you with schema files and sample input files. The files help you learn the basics of Smart Data Validator and might give you ideas about how to best approach your project.

This guide, combined with the sample files:

- Helps you create and set up your target system, data, and documents.
- Walks you through how to import the data and documents, validate them with rules, and run a job in Smart Data Validator.

The following diagram is an overview of the two systems you will be setting up to export your data and documents: a staging system and a target system.



Set up the target system

★ IMPORTANT

- The sample schema delivered with Smart Data Validator can affect a production system. We recommend that you create a site and project specifically to load and use the sample schema file.
- You must have the SmartPlant Foundation 2018 server component and Smart Data Validator 2018 target system adapter installed. For more information, see the *Intergraph Smart Data Validator Installation and Setup Guide*.

1. Log on to SmartPlant Foundation Server Manager.
2. Create a site for the sample schema using the SPF42ADW.dmp file, located in the SmartPlant Foundation installation folder.

   TIP Right-click the SmartPlant Foundation Sites node in the tree view, and click **New** on the shortcut menu to open the **New Site Wizard**.

3. Configure the new SmartPlant Foundation site with the Smart Data Validator target system adapter. Name the site SPFTGT. The server is named SPFTGTServer.
4. Log on to the newly created SPFTGT site and load the scheduler schema files. You must load the VTLTargetSystem.xmlldr file, located at: *[drive]*:\Program Files (x86)\Smart\SDV\2018\Model.
5. Load the sample schema files next. You must load the Schema_Tag.xml file first and then load the Admin_Tag.xml file, located at: *[drive]*:\Program Files (x86)\Smart\SDV\2018\SampleData\Tags.
6. Load the project data files onto the Smart Data Validator target site. You must load the FunctionalAreas_PlantA.txt file first and then load the FunctionalUnit_PlantA.txt file, located at: *[drive]*:\Program Files (x86)\Smart\SDV\2018\SampleData.
7. Load the login user file onto the Smart Data Validator target site. You must load the TargetSystem_SDVUser.xml file, located at *[drive]*:\Program Files (x86)\Smart\SDV\2018\SampleData\SDVUser.

For more information, see the *Intergraph Smart Data Validator Installation and Setup Guide*.

Set up the Smart Data Validator site

1. Using SmartPlant Foundation Server Manager, create a site for the sample schema using the VTL_data.dmp file, located at: *[drive]*:\Program Files (x86)\Smart\SDV\2018\Database.

   TIP Right-click the SmartPlant Foundation Sites node in the tree view, and click **SDV** > **Create new SDV site** on the shortcut menu to open the **New Site Wizard**.

2. Log on to the new Smart Data Validator site as Superuser.
3. Load the **StagingSystem_SDVUser.xml** login user file, located at *[drive]*:\Program Files (x86)\Smart\SDV\2018\SampleData\SDVUser, onto the Smart Data Validator staging site.

For more information, see the *Intergraph Smart Data Validator Installation and Setup Guide*.

SECTION 4

Tag Mapping

This section walks you through how to import tags, validate them with validation rules, and run a job in Smart Data Validator.

Create a target system

Target systems can be created in Smart Data Validator Administration with the URL address for the target system adapter. You must create a target system with the URL address pointing to the system that is loaded with the sample schema files. This will allow the software to know which one is the target system and where the tag data or documents available in the input files needs to be exported to.

NOTE When your target system is based on SmartPlant Foundation, you can use it to drive your import mapping, validation, and export mapping. Other target systems can be used just for export mapping.

1. Log on to SmartPlant Dashboard as **SDVUser**.

   NOTE Your system administrator must configure the VTLDataMapper role to your user to enable you to view the SmartPlant Foundation Dashboard.

2. Click **SDV Administration** > **Target Systems**.
3. Click **Create New Target System** ✳.
4. On the **Basic Details** page, enter the name and description of the new target system. You can name it *SPFTGT* with a description of *Target System to load Fusion tags and SPF documents*.
5. In the **Target System URL** box, type **net.tcp://[Target System Name]/[Target Site Name]Server**. For example, net.tcp://[SDVTS]/SPFTGTServer.
6. Click **Next**.
7. On the **Access Groups** page, select the following access groups and move them to the **Selected Access Groups** list.
   - VTLAdmin
   - VTLDataController
   - VTLDataMapper
8. Click **Next**. View the **Summary**, and check the details.
9. Click **Finish** to save the target system.

Create an import definition

Smart Data Validator uses import definitions to map imported CSV file data from existing column headers to the correct column headers for objects, properties, relationships, and relationship properties in the staging area. The import definition helps you specify that the template file is in CSV format and the mapping is of basic format. You can use the sample input file in CSV format to generate column headers.

1. Click **SDV Administration** > **Import Mapping**.
2. Click **Create Import Definition** ✳.
3. In the **Create New Import Definition** dialog box, type a name and description for the import definition. You can name it *ImportDef-FusionTag* with a description of *Import Definition to load Fusion tags*.
4. Select **CSV** as the **Template File**.
5. Click **Browse** to navigate to the PUMP_LIST.csv from the **SampleData** folder.

   You can find the sample file at *[drive]*:\Program Files (x86)\Smart\SDV\2018\SampleData\Tags.
6. Select the character encoding to be used from the **Character Encoding** box. For example, **ISO-8859-15**.
7. Select the **Enforce Column Order** option to use the exact column order as the CSV file that is used for creating the import definition.
8. Select **SPFTGT** option from the **Target System** list.
9. Select the **Auto Generate Export Mapping** option to automatically generate the export mappings. This ensures that the column headers match the objects and properties found in the target system.

   NOTE When the **Auto Generate Export Mapping** option is selected, the import definition maps the imported objects, classes, and properties to the existing objects, classes, and properties in the target system. For more information, see *Create export mapping* (on page 33).
10. Select the **Auto Generate Validation Rules** option if you want to automatically generate the validation rules from the mapping set up in the target system.
11. Select **Basic Format** from the **Mapping Format** list.
12. Click **Save**. This creates ten column headers for each column heading available in the PUMP_LIST.csv file.

You must map the column headers available in the **Column Headers** pane to an object, property, or relationship in the target system.

Create column headers

Column headers are used to map the data in the CSV file to the correct locations in the staging area and target systems. Column headers support a number of different column types. You must create a column header mapping for each column header that is generated from the sample input file. Each of the column headers is mapped to an object or property that is available in the target system. The column headers in the CSV file are mapped to the target system objects that are created by loading the sample schema file into the target system.

Object mapping

1. In the **Import Definitions** pane, select the **ImportDef-FusionTag** import definition.
2. In the **Column Headers** pane, select the **Name** column header and click **Edit Column Header** ✎.
3. In the **Column Header** pane, select **Physical** from the **Column Type** list.
4. Verify that the **Name** and **Column Header Text** boxes are populated with **NAME**.
5. In the **Mappings** pane, select the **Object Mapping** option.
6. On the **Target System** tab, click **Connect to Target System SPFTGT**.

   NOTE The **Target System** tab is not enabled unless you selected the target system while creating the import definition.
7. In the search box, type ***SPFNTag*** and press **ENTER**.
8. Select **SPFNTag** from the class definition list, and click **Add New Mapping** ✚.
9. Select **Create Update** from the **Object Action** list.
10. Verify that **Unique Identifier** is set to **NAME**.
11. Select the **Is Mandatory option** to indicate that the column is required to have a value.
12. Verify the **Property Mapping** option is selected in the **Mappings** pane.
13. Verify that **Name** is populated in the class definition list, and then click **Add New Mapping** ✚ to map the **Name** property.
14. Verify the following in the **Maps to Property Name with action Create Update** pane:
    - **Name** is populated in the **Maps to Property** box.
    - **Property Action** is set to **Create Update**.
    - **Parent Objects** box displays **Name** with **Create Update** action as selected.
15. Click **Save**.

    TIP When a column mapping has been added successfully, a check mark displays in the **Has Mapping** column next to the column header mapping in the main window.

You must create a column mapping for each of the column headers available under the **ImportDef-FusionTag** import definition. This is required to load the data available in the sample input file to the target system.

Property mapping

1. In the **Column Headers** pane, select the **Description** column header and click **Edit Column Header** ✎.
2. In the **Column Header** pane, select **Physical** from the **Column Type** list.
3. Verify that the **Name** and **Column Header Text** boxes are populated with **Description**.
4. In the **Mappings** pane, select the **Property Mapping** option.
5. On the **Target System** tab, select **SPFNTag** from the **Class Definition To Filter Properties** list.
6. Click **Search** 🔍 and select **Description** from the properties list.
7. Click **Add New Mapping** ✚.
8. Select **Create Update** from the **Property Action** list.
9. Verify the following in the **Maps to Property Name with action Create Update** pane:
   - **Description** is populated in the **Maps to Property** box.
   - **Property Action** is set to **Create Update**.
   - **Parent Objects** box displays **Name** with **Create Update** action selected.
10. Click **Save**.
11. Use the information in the following table to complete the property mappings:

Column Header Text in the CSV file	Maps to Property in the Target System	Property Action
DESCRIPTION	Description	Create Update
Pump Type	SPFNTag_Type	Create Update
Design_Pressure_Min	SPFNTag_Design_Pressure_Min	Create Update
Design_Pressure_Max	SPFNTag_Design_Pressure_Max	Create Update
Design_Temperature_Min	SPFNTag_Design_Temperature_Min	Create Update
Design_Temperature_Max	SPFNTag_Design_Temperature_Max	Create Update
OVERALL_LENGTH	SPFNTag_Length	Create Update
OVERALL_HEIGHT	SPFNTag_Height	Create Update
OVERALL_WIDTH_OR_DIA	SPFNTag_Width	Create Update

Create export mapping

You can export data to a target system by mapping the imported staging area data objects to the existing objects, properties, and relationships found in the target system. You can manually configure the export mapping for each object in the imported data, or you can automatically generate the export mapping from the target system by selecting the **Auto Generate Export Mapping** option when you create a new import definition.

1. Click **SDV Administration** > **Export Mapping**.

   The **SPFTGT** export mapping is displayed in the **Export Mapping** pane. This was created because the **Auto Generate Export Mapping** option was selected during the creation of the **ImportDef-FusionTag** import definition. For more information, see *Create an import definition* (on page 30).

2. In the **Export Mapping** pane, select **SPFTGT**.
3. In the **Stage to Target Object Map** pane, select the mapping item and click **Edit Stage To Target Object Mapping** .
4. In the **Edit Stage Object To Target Object Map** dialog box, type **10** in the **Object Weighting** box.

   NOTE You must specify the object weighting for each **Stage to Target Object Map** available under the **SPFTGT** export mapping. This is required to load the data available in the sample input file to the target system. Objects with lower weights are processed first.

5. Type **.Name** in the **Target System Query Definition** box.

   NOTE If an object can be identified in the target system by just its name and class definition, the query definition would be .Name. If the target class definition is defined with a UIDDefinition, then the UID can be defined in the import mapping. If the class definition has a unique key defined, then you can specify it in the **Target Sytem Unique Key** box.If there is no UID Definition or UniqueKey definition for the class definition, then .Name can be defined. For this exercise, the query definition is populated, but not required, as the SPFNTag class definition has a UID Definition defined that was already set in the import mapping.

6. Click **Save**.

Create rule set and rules

Validation rules verify that the data in an imported record meets the data standards specified for the target system. Each validation rule includes an error message that is recorded in a validation report when the rule returns an invalid value. You can create a validation rule set and the individual rules that will make up that set.

NOTE A rule set has already been created called **AutoGeneratedRuleSet** and is auto-populated with rules when items are mapped. For example, if data is mapped to a property in the target system that is scoped by an integer, then an integer rule is auto-created.

Create a rule set

You must create a rule set and create two different rules to validate that the tag name is in the correct format, and that the maximum design pressure is within the required range.

1. Click **SDV Administration** > **RuleSets and Rules**.
2. In the **RuleSets** pane, click **Create a new Rule Set** ✳.
3. In the **Create New Rule Set** dialog box, type a name and description for the rule set. You can name it *Pump-RuleSet* with a description of *Validates the data in the Pump List*.
4. Click **Save**.

Create a regular expression rule to verify the format of the Name property

This rule is created to verify that the tag names are in required format: NNNN-RP-NNN or NNNN-CP-NNN, where N stands for any number between 0 to 9.

1. Select the **Pump-RuleSet**.
2. In the **Rules** pane, click **Create a new Rule** ✳.
3. In the **Create Rule** dialog box, type a name and description for the rule. You can name it *CheckTagName* with a description of *A regular expression rule to verify the tag name is in the required format*.
4. Select **Class Definition** from the **Schema Type** options.
5. Select **Regular Expression** from the **Rule Type** list.
6. Type ***SPFNTag*** in the search box and press **ENTER**.
7. Select **SPFNTag** from the **Class Definition** list.
8. Type **Tag name is invalid** in the **Validation Rule Error Message** box.

   TIP This message appears in the Validation report if the rule does not pass.
9. Select **Error** from the **Validation Rule Severity Level** list.
10. Click **Next**.
11. Type ***Name*** in the search box and press **ENTER**.
12. Select **Name** from the **Property Definition** list.
13. Type **\d{4}-(RP|CP)-\d{3}** in the **Regular Expression** box. This regular expression looks for tag names in the following format: 1234-RP-123, 1234-CP-123, 9999-CP-999, 1111-RP-111.

    NOTE Use the ^ and $ symbols to restrict the data that is returned: **^\d{4}-(RP|CP)-\d{3}$**. Otherwise, formats like 11234-RP-123 and 1234-RP-1234 will also pass the rule.
14. Click **Next**. View **Summary**, and check the details.
15. Click **Finish** to save the validation rule.

Create a Check Value and UOM rule to verify the value of the Design_Pressure_Max property

This rule is created to verify that the value of the maximum design pressure is within the range 15 to 28.

1. Select the **Pump-RuleSet**.

2. In the **Rules** pane, click **Create a new Rule** ✳.
3. In the **Create Rule** dialog box, type a name and description for the rule. You can name it *CheckDesignPressureMax* with a description of *A check value and UOM rule to verify that the design pressure maximum is within a required range*.
4. Select **Class Definition** from the **Schema Type** options.
5. Select **Check Value And UOM** from the **Rule Type** list.
6. Type ***SPFNTag*** in the search box and press **ENTER**.
7. Select **SPFNTag** from the **Class Definition** list.
8. Type **Value is not within the specified range** in the **Validation Rule Error Message** box.
9. Select **Error** from the **Validation Rule Severity Level** list.
10. Click **Next**.
11. Type ***SPFNTag*** in the search box and press **ENTER**.
12. Select **SPFNTag_Design_Pressure_Max** from the **Property Definition** list.
13. Type **15** in the **Minimum Value** box.
14. Type **28** in the **Maximum Value** box.
15. Click **Next**. View **Summary**, and check the details.
16. Click **Finish** to save the validation rule.

Create a job definition

A job definition is a combined set of components configured into a template for the import, validation, and export of data to the target systems. You must select the target system, import definition, and validation rule set that are created in the earlier sections while creating the job definition. Select the default validation report template to generate a validation report after the validation of data in the staging system.

1. Click **SDV Administration** > **Job Definitions**.
2. Click **Create Job Definition** ✳.
3. In the **Create Job Definition** dialog box, type a name and description for the job definition. You can name it *JobDef-PumpList* with a description of *Job definition to load the pump list*.
4. Select **Import Validate Export** from the **Workflow** list.
5. Click **Next**.
6. On the **Target Systems** page, select the system to validate against and the target system. For example, select **SPFTGT** as your target system.
7. Click **Next**.
8. On the **Export Configuration** page, do not select any of the options.
9. Click **Next**.

10. On the **Import Definitions** page, the **ImportDef-FusionTag** is already displayed in the **Selected Import Definitions** list. If it is not displayed, select **ImportDef-FusionTag** from the **Import Definitions** list and move it to the **Selected Import Definitions** list.
11. Click **Next**.
12. On the **Rule Sets** page, move the two rule sets, **AutoGeneratedRuleSet** and **Pump-RuleSet**, to the **Selected Rule Sets** list.
13. Click **Next**.
14. On the **Validation Report Definitions** page, select **DefaultVTLValidationReportDefinition** and move it to the **Selected Validation Report Definitions** list.
15. Click **Next**.
16. On the **Access Groups** page, select the following access groups and move them to the **Selected Access Groups** list.
    - VTLAdmin
    - VTLDataController
    - VTLDataMapper
17. Click **Next**. View **Summary**, and check the details.
18. Click **Finish** to save the job definition.

Create a job

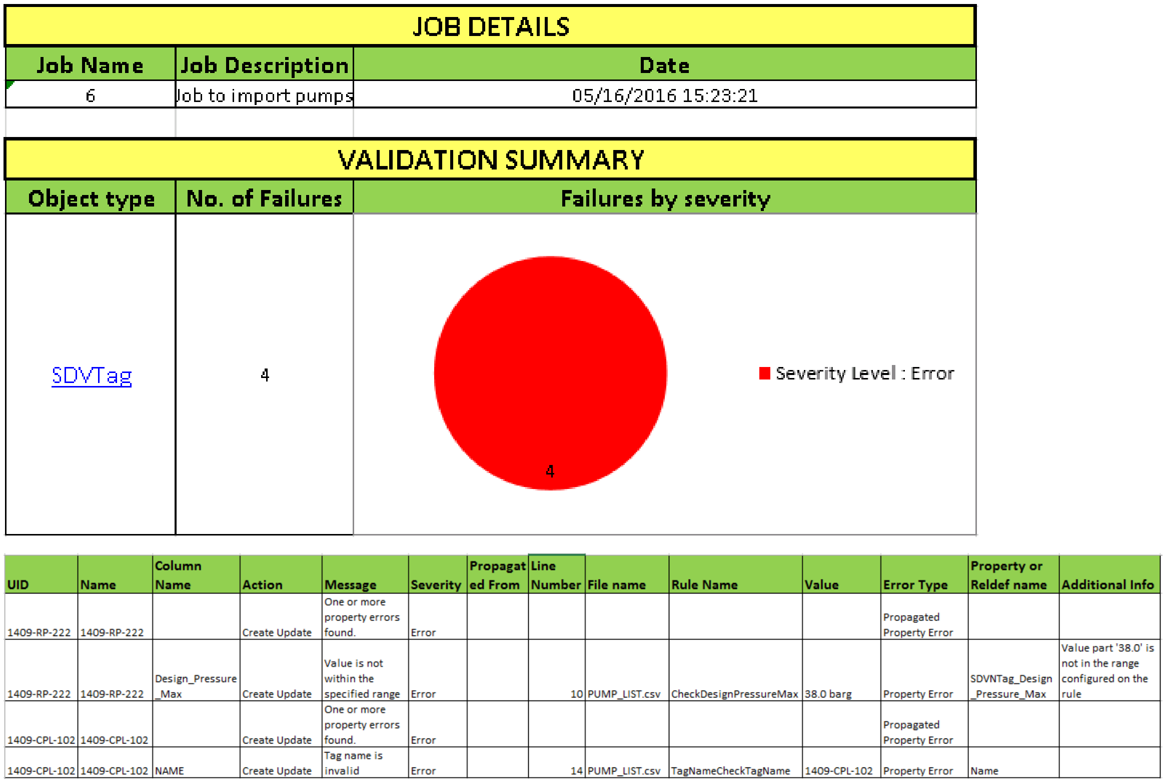
You can create jobs that manage the automation of the import, validation, and export of data into selected target systems, such as SmartPlant Foundation. The basis for each job is the job definition defined in **Smart Data Validator Administration,** which combines all the settings into one job. The JobDef-PumpList job definition includes the import definition, validation rules, export mapping, and target system information that we have created to process the sample input file and load the validated data into the target system.

1. Log on to SmartPlant Dashboard as **SDVUser**.
2. Click **SDV Job Management**.
3. Click **Create new job** .
4. In the **Create New Job** dialog box, type *Job to import pumps* as the description for the job.
5. Select **JobDef-PumpList** from the **Job Definition** list.
6. Select **SPFTGT** from the **Target System for Validation** list.
7. Select **SPFTGT** from the **Target Systems for Export** list.
8. Click **Next**.
9. On the **Target System Configuration** page, select **PlantA**.
10. On the **Files to Import** page, click **Browse to a file to attach** to locate the PUMP-LIST.csv file, and click **Open** to attach the file to the import definition. You can find the PUMP-LIST.csv file at *[drive]*:\Program Files (x86)\Smart\SDV\2018\SampleData\Tags.
11. Click **Next**. View the **Summary**, and check the details.

12. Click **Finish**.

View job progress and approve the validation step

1. Select the job you previously created, *Job to import pumps*. Click **Refresh jobs** , if the job is not displayed.
2. On the **Progress** tab, click **Refresh the selected job** .
3. Once the validation status is set to **Completed Validation**, click the **Summary** tab to check if any errors or warnings are reported. Four errors and zero warnings are displayed.
4. Click **Download validation report** and save the validation report to a location of your choice. The report is downloaded as a Microsoft Excel file.
5. Open and review details about the job validation summary information in Microsoft Excel.



JOB DETAILS		
Job Name	**Job Description**	**Date**
6	Job to import pumps	05/16/2016 15:23:21

VALIDATION SUMMARY		
Object type	**No. of Failures**	**Failures by severity**
SDVTag	4	4 🟥 Severity Level : Error

UID	Name	Column Name	Action	Message	Severity	Propagated From	Line Number	File name	Rule Name	Value	Error Type	Property or Reldef name	Additional Info
1409-RP-222	1409-RP-222		Create Update	One or more property errors found.	Error						Propagated Property Error		
1409-RP-222	1409-RP-222	Design_Pressure_Max	Create Update	Value is not within the specified range	Error		10	PUMP_LIST.csv	CheckDesignPressureMax	38.0 barg	Property Error	SDVNTag_Design_Pressure_Max	Value part '38.0' is not in the range configured on the rule
1409-CPL-102	1409-CPL-102		Create Update	One or more property errors found.	Error						Propagated Property Error		
1409-CPL-102	1409-CPL-102	NAME	Create Update	Tag name is invalid	Error		14	PUMP_LIST.csv	TagNameCheckTagName	1409-CPL-102	Property Error	Name	

NOTES

- Although only two tags failed validation, the report shows that four rules have failed. This is because the name rule failed for one tag, and this resulted in the error propagating up to the object resulting in another two rules failing. The check pressure rule failed on the other tag, and this resulted in the error propagating up to the object resulting in another two rules failing.
- Although two tags failed validation, it is up to the user running the job to determine what action needs to be taken. The choices are to get the user who submitted the CSV file to correct the issues, and only export the data if it passes all tests, or to export the tags that currently pass the validation now, then correct the failed tags later in a revised CSV

file. In this example, the tags that pass validation will be exported without having the CSV file corrected.

6. Click **Approve Workflow Step** to allow the valid data to load into the target system.
7. Click **OK**.
8. Click **Refresh the selected job** .
9. On completion of the job, the **Status** on the job is set to **Completed**.

View data in SmartPlant Foundation Desktop Client

1. Open SmartPlant Foundation Desktop Client in the target system.
2. Log on to SmartPlant Foundation Desktop Client as **SDVUser** and set the **Create** and **Query** scope to PlantA.
3. Click **Find** > **Tags**.

   Using the sample input file, this Smart Data Validator job loads eight tags, even though there are ten tags in the sample input file. The following tags will be displayed in the search results:

   - 1409-RP-100
   - 1409-RP-101
   - 1409-RP-102
   - 1409-RP-103
   - 1409-RP-104
   - 1409-RP-105
   - 1409-RP-106
   - 1409-RP-107

SECTION 5

Document Mapping

You have now successfully exported data to the target system after following the procedures documented in the previous section. Similarly, you can also export documents to the target system using the sample DESIGN_DOC.csv file and the other supporting files located at: *[drive]*:\Program Files (x86)\Smart\SDV\2018\SampleData\Documents.

This section walks you through how to import documents, validate them with auto-generated rules, and run a job in Smart Data Validator. However, since you are already familiar with tag mapping, this section will not have detailed procedures. It is intended as a guide to enable you to map documents.

- In SmartPlant Foundation, each document consists of three parts, a master, a revision, and a version.
- The supplied Smart Data Validator adapter for SmartPlant Foundation server allows the export of documents to the Smart Data Validator target system using the export mapping for document revisions and masters.
- The import mapping does not have versions, so a version is automatically created for each revision when exported.
- To map version object properties in the target system, you must map the properties to the revision class definition in the staging system. These properties are instantiated only on the version object in the target system.

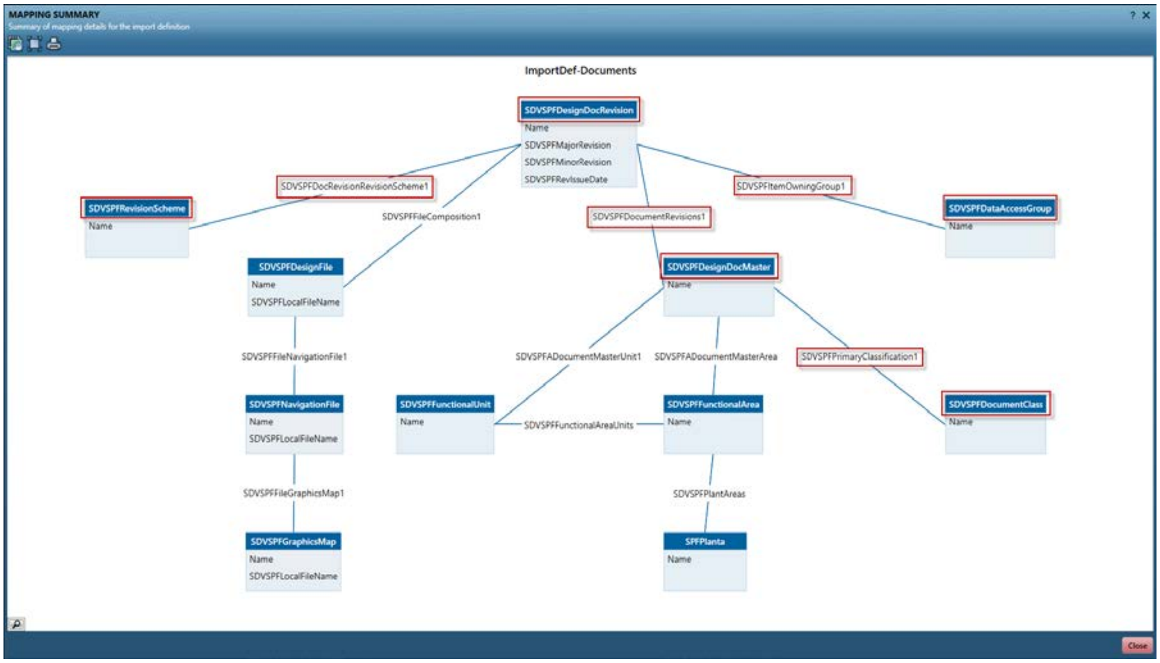
Follow the procedures outlined in *Getting Started* (on page 27) to create the import definition, column headers, and job definition.

NOTES

- Create an import definition using the sample input file located at: *[drive]*:\Program Files (x86)\Smart\SDV\2018\SampleData\Documents. Follow the same procedure outlined in *Create an import definition* (on page 30). You can name the import definition **ImportDef-Documents** with a description *Import definition to load design documents*. Select the **Auto-Generate Export Mapping** and **Auto-Generate Validation Rules** options in the **Create New Import Definition** dialog box.
- You must map the column headers available in the **Column Headers** pane to an object, property, or relationship in the target system. You must create a column mapping for each of the column headers available under the **ImportDef-Documents** import definition. This is required to load the data available in the sample input file to the target system. Use the sample mappings provided in the following tables to create your mappings.
- If an object, property, or relationship in the target system already exists in the staging area, then you must rename the object, property, or relationship while saving the respective mapping. For example, when mapping the Name column header to the SPFDesignDocMaster classification, you must modify the class definition name by adding a prefix, for example SDV, and type **SDVSPFDesignDocMaster** in the **Maps to Object** box.
- The **Relationship Mapping** option is not enabled until at least two objects have been mapped.

- You can use the SmartPlant Foundation SmartConverter for hotspotting drawings. For more information, see *Configuring the SmartConverter for Graphical Navigation* in the *SmartPlant Foundation Server Customization Guide*.

You can use the following illustration and tables to create your own document mapping using the sample DESIGN_DOC.csv file and other attached drawing files in the Documents folder available at: *[drive]*:\Program Files (x86)\Smart\SDV\2018\SampleData.



TIP The objects and relationships highlighted in red are mandatory for document mapping.

- *Object mapping to physical columns* (on page 40)
- *Object mapping to computed columns* (on page 41)
- *Property mapping* (on page 42)
- *Relationship mapping* (on page 44)

Object mapping to physical columns

Before starting object mapping, you must create a column header, **CON_CLASS_OCCURENCE**, of type **Constant**, with a value of 1. This is used to map the **Classification** column header.

Column Header	Column Type	Object Mappings	Is Mandatory	Action	UID (Unique Identifier)
Name	Physical	SPFDesignDocMaster	Yes	Create Update	Name
Revision Scheme	Physical	SPFRevisionScheme	Yes	Check Exists	Revision Scheme

Column Header	**Column Type**	**Object Mappings**	**Is Mandatory**	**Action**	**UID (Unique Identifier)**
Classification	Physical	SPFDocumentClass	Yes	Check Exists	Classification, CON_CLASS_OCCURRENCE
Area	Physical	SPFFunctionalArea	No	Check Exists	Area
Unit	Physical	SPFFunctionalUnit	No	Check Exists	Unit
Owning Group	Physical	SPFDataAccessGroup	Yes	Check Exists	Owning Group
File	Physical	SPFDesignFile *Optional Interface:* *ISPFNavigationFileComposition*	No	Create Update	File
Navigation File	Physical	SPFNavigationFile *Optional interface:* *ISPFGraphicsMapComposition*	No	Create Update	Navigation File
Graphics Map File	Physical	SPFGraphicsMap	No	Create Update	Graphics Map File

NOTES

- Click ✎ beside **Unique Identifier** to select the **Classification** and **CON_CLASS_OCCURRENCE** column headers for the object identifier while mapping the **Classification** column header.
- Click ✎ beside **Optional Interface** to select the optional interfaces for **File** and **Navigation File** column headers.
- Ensure that the **Name** property has been mapped for every object.

Object mapping to computed columns

There may be cases where a value that is required for a mapping is either not available in the physical file or the physical data supplied needs to be reformatted. In such cases, a computed column can be set up. For example, there is no column for the revision to map to, so a new computed column is required. The document revision has been created using the computed column, COMP_REVISION, which creates a revision object.

1. In the **Import Definitions** pane, select the **ImportDef-Documents** import definition.
2. Click **Create New Column Header**.
3. In the **Column Header** pane, select **Computed** from the **Column Type** list.
4. Type **COMP_REVISION** in the **Name** box.

5. Type **func.** in the **Computed API** box, double-click **Replace** to select it from the drop-down list, and type the parameter string syntax, `func.Replace([Name], "ZZ", "ZZ")`. This function simply copies the data in the **Name** column to this new column.

Column Header	Column Type	Object Mappings	Is Mandatory	Action	UID (Unique Identifier)
COMP_REVISION	Computed `func.Replace([Name], "ZZ", "ZZ")`	SPFDesignDocRevision	Yes	Create Update	COMP_REVISION
COMP_CONFIG	Computed `@CONFIG1@`	SPFPlant	No	Check Exists	COMP_CONFIG

NOTES

- As there is no Copy function available, the Replace function is being used in the COMP_REVISION computed column.
- Ensure that the Name property has been mapped to the objects.

Property mapping

Use the following table to create your property mappings:

Column Header	Column Type	Property Mappings	Action	Parent Object
Title	Physical	SPFTitle	Create Update	SPFDesignDocMaster Name
Major Revision	Physical	SPFMajorRevision	Create Update	SPFDesignDocRevision (COMP_REVISION)
Minor Revision	Physical	SPFMinorRevision	Create Update	SPFDesignDocRevision (COMP_REVISION)
COMP_DESIGNPATH	Computed `func.Concat({[FileDirectory], [File]})`	SPFLocalFileName	Check Staging File Exists	(SPFDesignFile) File

Column Header	Column Type	Property Mappings	Action	Parent Object
COMP_NAVIGATIONFILE	Computed `func.Concat({[File Directory], [Navigation File]})`	SPFLocalFileName	Check Staging File Exists	(SPFNavigationFile) Navigation File
COMP_GRAPHICSPATH	Computed `func.Concat({[File Directory], [Graphics Map File]})`	SPFLocalFileName	Check Staging File Exists	(SPFGraphicsMap) Graphics Map File
COMP_ISSUE_DATE	Computed `func.DateTimeColumn([Issue date], "dd-MM-yyyy", "UTC")`	SPFRevIssueDate	Create Update	(SPFDesignDocRevision) COMP_REVISION

NOTES

- All the objects must be mapped to the **Name** property.
- The file name and the file directory can be in different columns or concatenated into a single computed column.
- The format of the issue date in the CSV file must match the format in the staging server. Otherwise, you can use the computed column function to correct the format in the CSV file.
- The issue date has been mapped to the SPFRevIssueDate property and it decides the state of the document. If different revisions of the same document are loaded into SDV, the following scenarios are possible:
  - If the column has a value, for example 01-01-2017, the revision state is set to CURRENT, and the revision number is 01A.
  - If the column has a value, for example 10-01-2017, the CURRENT revision is changed to SUPERSEDED and the latest revision is set to CURRENT, and the revision number is 03A.
  - If the column has a value, for example 05-01-2017, the revision state is set to SUPERSEDED because a more recent revision is available, and the revision number is 02A.

- If the column has no values in the input CSV, then the revision state is WORKING, and the revision number is 04A.

Relationship mapping

Use the following table to create your relationship mappings:

Relationship Mappings	**End Column 1**	**End Column 2**	**Action**
SPFDocumentRevisions	Name	COMP_REVISION	Create Update
SPFDocRevisionRevisionScheme	COMP_REVISION	Revision Scheme	Create Update
SPFADocumentMasterArea	Name	Area	Create Update
SPFADocumentMasterUnit	Name	Unit	Create Update
SPFItemOwningGroup	COMP_REVISION	Owning Group	Create Update
SPFPrimaryClassification	Classification	Name	Create Update
SPFPlantAreas	Area	COMP_CONFIG	Check Exists
SPFFunctionalAreaUnits	Unit	Area	Check Exists
SPFFileComposition	File	COMP_REVISION	Create Update
SPFFileNavigationFile	File	Navigation File	Create Update
SPFFileGraphicsMap	Navigation File	Graphics Map File	Create Update

Validation rules

The **AutoGeneratedRuleSet** rule set is displayed in the **RulesSets** pane. This was created because the **Auto Generate Validation Rules** option was selected during the creation of the **ImportDef-Documents** import definition. For more information, see *Create an import definition* (on page 30).

Export mapping

The **SPFTGT** export mapping is displayed in the **Export Mapping** pane. This was created because the **Auto Generate Export Mapping** option was selected during the creation of the **ImportDef-Definition** import definition.

You must specify the object weighting for each **Stage to Target Object Map** available under the **SPFTGT** export mapping. This is required to load the data available in the sample input file to the target system. Objects with lower weights are processed first. For example, the object weightings must be set to export the revision before exporting the file to which it is attached.

To identify supporting file objects, use query definition as **.Name** for SPFDesignFile, SPFNavigationFile, and SPFGraphicsMap.

Use the following tables to set the object weights:

Stage Object Name	**Target Object Name**	**Object Weight**	**Target System Unique Key**	**Target System Query Definition**
SDVSPFPlant	SPFPlant	12		
SDVSPFFunctionalArea	SPFFunctionalArea	20	"AR",@CONFIGLEVEL1@,.Name	
SDVSPFDocumentClass	SPFDocumentClass	30		
SDVSPFRevisionScheme	SPFRevisionScheme	40		
SDVSPFDataAccessGroup	SPFDataAccessGroup	50		
SDVSPFFunctionalUnit	SPFFunctionalUnit	60	"UNT",@ConfigLevel1@,+SDVSPFFunctionalAreaUnits.Name,.Name	
SDVSPFDesignDocMaster	SPFDesignDocMaster	70	"DM",.Name	
SDVSPFDesignDocRevision	SPFDesignDocRevision	80	"DM",.Name,.SDVSPFMajorRevision,.SDVSPFMinorRevision	
SDVSPFDesignFile	SPFDesignFile	90		.Name
SDVSPFNavigationFile	SPFNavigationFile	100		.Name
SDVSPFGraphicsMap	SPFGraphicsMap	110		.Name

NOTE You must set a unique object weight for each object you are exporting. The object weight for the objects are set as 12, 20, 30, and so on. This is because 10 is already used as the object weight for SPFNTag while creating the export mapping for tags (*Create export mapping* (on page 33)).

Job definition

You must select the **SPFTGT** target system, **ImportDef-Documents** import definition, and **AutoGeneratedRuleSet** validation rule set while creating the job definition. Select the default validation report template to generate a validation report after the validation of data in the staging system.

1. Click **SDV Administration** > **Job Definitions**.
2. Click **Create Job Definition**.
3. In the **Create Job Definition** dialog box, type a name and description for the job definition. You can name it *JobDef-DesignDocs* with a description of *Job definition to load the design documents*.
4. Select **Import Validate Export** from the **Workflow** list.
5. Click **Next**.
6. On the **Target Systems** page, select the system to validate against and the target system. For example, select your Smart Data Validator application server as your staging system and select SPFTGT as your target system.
7. Click **Next**.
8. On the **Export Configuration** screen, do not select any of the options.
9. Click **Next**.
10. On the **Import Definitions** page, **ImportDef-Documents** is displayed in the **Selected Import Definitions** list. If it is not displayed, select **ImportDef-Documents** and move it to the **Selected Import Definitions** list.
11. Click **Next**.
12. On the **Rule Sets** page, select the **AutoGeneratedRuleSet** and move it to the **Selected Rule Sets** list.
13. Click **Next**.
14. On the **Validation Report Definitions** page, select **DefaultVTLValidationReportDefinition** and move it to the **Selected Validation Report Definitions** list.
15. Click **Next**.
16. On the **Access Groups** page, select the following access groups and move them to the **Selected Access Groups** list.
    - VTLAdmin
    - VTLDataController
    - VTLDataMapper
17. Click **Next**. View **Summary**, and check the details.
18. Click **Finish** to save the job definition.

Job

Create a job in the Job Management module using the JodDef-DesignDocs job definition that you created earlier. Enter the description as *Job to import design documents*. The job definition includes the import definition, validation rules, export mapping, and target system information to process the sample input file and load the validated documents into the target system. You can find the sample input file, **DESIGN_DOC.csv**, at *[drive]*:\Program Files (x86)\Smart\SDV\2018\SampleData\Documents.

Ensure that the files attached to the documents are present in the location mentioned in the FileDirectory column in the sample input CSV file. If they are not, update the CSV file with the correct location.

View job progress and approve the validation step

1. Select the job you previously created, *Job to import design documents*. Click **Refresh jobs** if the job is not displayed.
2. On the **Progress** tab, click **Refresh the selected job** .
3. Once the Validation status is set to **Completed Validation**, click the **Summary** tab to check if any errors or warnings are reported. Zero errors and zero warnings are displayed.
4. Click **Approve Workflow Step** to allow the loading of the valid data into the target system.
5. Click **OK**.
6. Click **Refresh the selected job** .
7. On completion of the job, the **Status** on the job is set to **Completed**.

View documents and attached files in SmartPlant Foundation Desktop Client

1. Open SmartPlant Foundation Desktop Client in the target system.
2. Log on to SmartPlant Foundation Desktop Client as **SDVUser** and set the **Create** and **Query** scopes to **PlantA**.
3. Click **Find** > **Design Documents**.

   Using the sample input file, this Smart Data Validator job loads five documents with files attached to them. The following documents will be displayed in the search results:

   - 1409-10-015
   - 1409-10-026
   - 1409-10-052
   - 1409-10-054
   - 1409-10-061

NOTE The document 1409-10-061 has been loaded as **Issued** because it had an issue date defined in the CSV file.

SECTION 6

Help, support, and training

Check out these resources to make the most of Smart Data Validator.

Help and documentation

Learn more about how to use Smart Data Validator from the product documentation. To view, click **Help** in the software.

Support

Visit the Smart Support Web site (*https://smartsupport.intergraph.com* (*https://smartsupport.intergraph.com*)) to contact Intergraph Support.

Training

To find out more information about Smart Data Validator courses that are being offered, visit http://www.intergraph.com/ppm/training.aspx.

Index