

IBM Commerce

Five key reasons architects should consider IBM Transformation Extender

**Delivering universal transformation
across your IT infrastructure**

IBM



Transforming data and creating value from data transformations can be a formidable task as data and data sources continue to grow exponentially. To add to that, complying with rapidly changing industry and regulatory standards is more challenging than ever before. Organizations need to automate and validate complex data and facilitate the transformation of data to standard formats, without the need for hand-coding.

To accomplish this, you need fast and flexible data transformation systems that can handle current and future workloads. You need to produce and validate data in a wide range of industry standards, and you need tools that can be deployed quickly and offer the fastest throughput available.

With IBM® Transformation Extender, if you can describe your data, you can define it. Specifically designed for high-volume and complex data transformation, Transformation Extender makes mapping and transformation easier. You can process large volumes of data efficiently with validation and transformation capabilities. You can eliminate drag-and-drop, saving time and the need for hand-coding. Transformation Extender offers APIs so you can take advantage of its extensive capabilities within your own code and applications. Transformation Extender provides the core transformation and validation capabilities that you need, to help you meet your business requirements.

In this paper, you'll learn about the data-centric and reusable features of Transformation Extender, including the tools to visually transform data and how the tool provides validation and compliance to industry standards.

Introduction	Any-to-any and many-to-many transformations	Standards and validation	Reusability	Data-centric capabilities of Transformation Extender	Extensive capabilities and built-in functions	Conclusion	◀ 2 ▶
--------------	---	--------------------------	-------------	--	---	------------	-------

Any-to-any and many-to-many transformations

Transformation Extender is designed specifically to help companies perform data transformation. Not only is data represented graphically, but Transformation Extender also has tools and a whole methodology that is built around the structure of the data. Not all relationships between the levels in data are immediately apparent, and the tooling within Transformation Extender is natural for data that sits at multiple hierarchical levels. Transformation Extender is not a “jack of all trades” like a general programming language, rather, it is a set of tools and a design environment that is specifically focused and optimized for data transformation. Once you have used a tool like this, it is difficult to go back and parse data with a regular programming language.

Transformation Extender can transform it all.

The concepts of data transformation are familiar—the data-centric task of changing data from one format to another—whether it is EDI, XML, COBOL Copybook or even if it is in a proprietary format—describe your data, and then define and transform it in Transformation Extender.

Transformation Extender is not limited to just taking one input, reconfiguring it and coming out with one output. You can have multiple data inputs in different formats, and on the output side, Transformation Extender can again create multiple, separate formats. Many-to-many transformations can be performed without requiring a format in between and you do not have to go to XML every time you want to go from one form of EDI to another form of EDI. If you don't need to go to XML or some type of proprietary,

intermediate format, Transformation Extender does not require it—there is no requirement for everything to go through some type of canonical or some type of XML format.

You can take in one file, or multiple files, and within each file you can have multiple different formats. If you have one file or one group of data with different structures within it, you can take whatever you need from the different parts of the structure. An example of this type of transformation is within HIPAA regulations. The HIPAA messages, called clinical attachments are a chunk of EDI data with embedded HL7 XML, can be validated against the schema to see if it is correct and then parsed as part of the same mapping flow that deals with the EDI.

Introduction	Any-to-any and many-to-many transformations	Standards and validation	Reusability	Data-centric capabilities of Transformation Extender	Extensive capabilities and built-in functions	Conclusion	◀ 3 ▶
--------------	---	--------------------------	-------------	--	---	------------	-------

Standards and validation

Transformation Extender has detailed implementations of key industry standards and perhaps more importantly, has a starting place for customers to adapt to changes—changes that naturally come with industry standards. Transformation Extender helps enable organizations and trading partners to make changes as necessary, as the standards change or deviate (as standards tend to do), and to change rules or take rules and structures away.

Through the user interface, you see not only the hierarchical structure but also any rules that are applied to any of the components that you see on the screen, as well as the restrictions and the properties. For example, you can set a restriction to a number, a date, a time, or a date that is year-year, month-month, day-day. All of this is graphically visible

to the user so that if you open a type tree that's been well-defined in Transformation Extender—especially if you open one of the industry standard content artifacts—the tool allows you to support the rules, while also supporting your customers' need to change the rules as needed.

Transformation Extender also offers a quick and efficient pass/fail validation feature. This feature determines if the data is good or bad, and can then perform separate processing based on the results. Even if standards compliance is not a requirement, Transformation Extender has the ability to set up the metadata representations and type trees, which are representations of the data structure, to help define the same data in different ways.

The key is flexibility.

Transformation Extender can save a whole transaction as a series of segments in which you may only care about the segment identifiers. Or, you may care about every character, on every line of every segment in that whole transaction, in which case you have it very specifically defined.

Introduction	Any-to-any and many-to-many transformations	Standards and validation	Reusability	Data-centric capabilities of Transformation Extender	Extensive capabilities and built-in functions	Conclusion	◀ 4 ▶
--------------	---	--------------------------	-------------	--	---	------------	-------



Transformation Extender can have type trees and metadata representations with different levels of rules applied. In this case, you may only care if some of it looks like X12, versus in the next step, where you may be concerned about whether the transformation conforms to the basic rules for each of the segments. The next step is to apply all of the business rules so that you can redefine the same data with different rules, based on how specifically it is defined and how tightly the rules are followed for that data. The pass/fail validation is based on many different criteria without jumping out into code, helping to save the cost of trying to read and understand the whole segment.

You can use the same tools, but with a slightly more rigorous metadata representation, to see what it is and figure out if it is good or bad. With a couple of simple structures, you can take a deeper look to see the next steps. For example, you may know that this is a purchase order, or maybe a laboratory result—how good does it have to be for it to be processed? It may not have to be 100 percent correct throughout the whole transaction, but you know it contains data that is useful and you have to be able to find it. Therefore, the next typical step is to set up a validation map that differentiates the good from the bad, and then decide to send it back to where it came from, create a report or create some other output that can be routed differently so that it is not processed and could choke the transformation process.

Introduction	Any-to-any and many-to-many transformations	Standards and validation	Reusability	Data-centric capabilities of Transformation Extender	Extensive capabilities and built-in functions	Conclusion	◀ 5 ▶
--------------	---	--------------------------	-------------	--	---	------------	-------

Reusability

When using a family of industry standards, you have many common processes between transformations as well as the need to handle multiple maps and multiple transformation needs.

Transformation Extender offers the reusability of functions with differentiated executable maps, which are basically programs, such as MMCs—things that get run and are deployable. Inside, they are made up of building blocks, called functional maps so that you can create and reuse these functions. For example, instead of needing to drag and drop data every time you have a BHT segment, you can be more productive by using the functional maps, taking it from input to output. The robust and mature tooling and functions in Transformation Extender helps ensure that your data proficiently gets from one side to the other.

Transformation Extender allows you to create reusable components that help you to be more productive than hand coding, and you can reuse the methodologies that have been previously applied. When dealing with this type of data, you may have a set group of needs and you can set up those needs as functions and reuse those functions in one map source or across multiple maps sources.

Transformation Extender has sets of common functionality that are based on services around validation. When you look at the direction that Transformation Extender takes with sets of common functionality that are based on the detail within transformation services, and where Transformation Extender offers services around validation, document discernment and services that are specifically for transformation and enveloping and de-enveloping (pulling headers off or putting headers on, depending on the data content that you have) you see that Transformation Extender has the ability to build from the most atomic units, which are defining metadata so that it can be pass/fail validated.

Transformation Extender transforms data so that you can get from one format to the other, and then by using the same tooling that we used for transformation, you can create higher level assets (such as compliance checkers).

Transformation Extender allows users to create their own compliance checkers and services, and create the appropriate level of data validation and reporting based on your customized level of needs—whether you need extensive services, or very little services. For example, if you have a data standard such as HL7 and you know that your trading partners aren't following all the prescribed rules, Transformation Extender gives you the ability to build up from the same components that you used to do the core transformation, with the same tooling set and same set of user interfaces that let you define your artifacts such as your type trees and your transformation maps.

Introduction	Any-to-any and many-to-many transformations	Standards and validation	Reusability	Data-centric capabilities of Transformation Extender	Extensive capabilities and built-in functions	Conclusion	◀ 6 ▶
--------------	---	--------------------------	-------------	--	---	------------	-------

Create maps for specific purposes.

You can set validation maps to look at the input data and have different levels of definition to apply rules to report, before you try to transform it. This is a very common and very powerful capability of Transformation Extender.

The flexibility, responsiveness and reusability of Transformation Extender helps you stay agile, and helps you stand up new maps much more quickly. Often, you're able to take at least one half of the map—one of the type trees that you already have—and use that again. By deploying Transformation Extender across your IT infrastructure, you are able to respond to new business challenges faster, because of the common tooling, and the fact that in

Transformation Extender transformations you can have conditional logic to allow you to deal with variances between one trading partner and another.

The reality is that rules do get broken regardless of set standards, so you'll want to keep the standard trees pure and then create subsets. Transformation Extender has advanced tooling that allows you to pare down a type tree or a metadata representation to the specific components needed, based on what level of detail you use. For example, you'll have the whole X12 standard but maybe only use an 850 for one particular customer and their maps. Instead of using that whole data structure, Transformation Extender has the tooling built in to automatically merge or pull out everything that is associated with the 850 and leave everything else behind.

So you can create a new structure, with only the piece that you need, helping to make you and your mappers a lot more efficient—not only efficient processing wise, because you don't have as big a data structure to traverse to find the data that you need, but also while you're doing the mapping and the transformation, you do not accidentally pick up a segment from an 856 or a loop from the wrong place. This is another one of the many concepts that have been built into Transformation Extender to make it as agile and as quick to develop as is possible.

Introduction	Any-to-any and many-to-many transformations	Standards and validation	Reusability	Data-centric capabilities of Transformation Extender	Extensive capabilities and built-in functions	Conclusion	◀ 7 ▶
--------------	---	--------------------------	-------------	--	---	------------	-------

Data-centric capabilities of Transformation Extender

A common practice for Transformation Extender users is to build systems of patterns. For example, you may receive data, perform some discernment to see what it is, perhaps lightly parse and consider most of this message as a blob except for where you know you should get some type of identification. And then, without parsing and going through the whole message, you can see what you have because Transformation Extender provides the ability to define this data in different ways using this tooling.

There's more than one way to perform the transformations, and you can customize the transformations. For example, you may have a set that is 500 characters, or a group of five sets, that are 100 characters long, or maybe each character in this 500 characters means something. Then you know that every time you see a special character (for

instance, a tilde), you'll know that the tilde means that you have a different field and that is how you separate them. The same data can have multiple ways to be represented and Transformation Extender allows those different ways to be graphically modeled and expressed.

Transformation Extender also allows the multiple definitions to exist in the same sets of maps so that you can set it to transform the set that you need. You may have five records, and you can send each of the five records somewhere and do something with it. When you have the records, you can focus on what's in between those tildes and what rules you are concerned with—its size, its numeric-ness, the values based on restrictions or its values that are based on business rules. You can be as granular as you need to be within the same data and the system of maps and deal with it in different ways.

Picture your next analytics platform project... Can you see it?

It probably involves a large body of different types of data. Picture it as a lake of data. You can imagine that there are several ways to access each piece of information contained in it. You could go bit by bit and use an eyedropper to process the lake, or record by record with a straw — or slightly less granular and use a fire hose. And depending on which way it's processed, hopefully you get out the pieces of data that you cared about while taking varying lengths of time and processing power to do it.

Introduction	Any-to-any and many-to-many transformations	Standards and validation	Reusability	Data-centric capabilities of Transformation Extender	Extensive capabilities and built-in functions	Conclusion	◀ 8 ▶
--------------	---	--------------------------	-------------	--	---	------------	-------

The power of Transformation Extender is that you can use the same set of tools and easily switch out the amount of data you are dealing with at any one point. In this case maybe it would be best to scoop up a swimming pool sized bucket of data from our lake and then spoon out the dozen or so data points which we care about and skip the rest.

If you want a general language to do whatever pieces of coding you want, maybe you use Java™ or something similar. If you're doing some very specific piece of work, maybe something very graphics intensive, then you'd use different language or toolset to do it. When you're performing the data transformation, you can do it in Java, C++ or

whatever other language, but none of these languages are dedicated and designed specifically for solving this data transformation challenge. While Transformation Extender isn't a programming language, it is an environment that has been designed specifically to help you solve the data transformation challenge.

In the same way you would pick a toolset or coding environment to suit the task that you're dealing with, when the task you're faced with is data transformation, then the tool that you should be using is Transformation Extender.



Introduction	Any-to-any and many-to-many transformations	Standards and validation	Reusability	Data-centric capabilities of Transformation Extender	Extensive capabilities and built-in functions	Conclusion	◀ 9 ▶
--------------	---	--------------------------	-------------	--	---	------------	-------

Extensive capabilities and built-in functions

Transformation Extender offers a wealth of importers to import data and metadata that you need represented in Transformation Extender, such as PL/I, COBOL Copybook and simple text formats. Transformation Extender lets you bring things as metadata representations that map and that are from other structured data formats that the tooling automatically understands. Transformation Extender also allows for XML schemas as well as pointing to the schemas natively to represent them within the design tools. The Transformation Extender design time APIs allow you to write code to create your Transformation Extender artifacts, so using code to create a type tree from some proprietary metadata source that represents a data format is very powerful and without doing a lot of typing in a user interface, you can perform what would normally be repetitive tasks. Normally, you would only do that for structures and standards that are either proprietary or industry standards outside of what are supported in the Transformation Extender packs.

Transformation Extender offers a powerful feature to save valuable time and resources with the AutoMapper tool. For example, you may have two formats that are very similar and have a common vocabulary (such as the similarities between EDI and XML). Dragging and dropping each of those fields is a tedious task. With Transformation Extender, you can use the AutoMapper that lets you define in the setup while you're running, even before you run it, how closely something has to match for it to be automatically mapped from the input to the output. When you consider the thousands of possible data elements at many, many levels of hierarchy across multiple segments, the time that the AutoMapper can save is amazing.

Create and automatically optimize the functional groups.

The AutoMapper functions can create various hierarchical nested levels of mapping, offering the shortest time to go live, because of the maturity of the Transformation Extender and the functionality that's been built into it.

Introduction	Any-to-any and many-to-many transformations	Standards and validation	Reusability	Data-centric capabilities of Transformation Extender	Extensive capabilities and built-in functions	Conclusion	◀ 10 ▶
--------------	---	--------------------------	-------------	--	---	------------	--------

Conclusion

For the enterprise architect who wants to improve data quality, accelerate data standards compliance and improve IT responsiveness to change, Transformation Extender provides the single tool to deliver data transformation and validation. This tool supports new and evolving industry standards and regulatory requirements and offers the shortest time to go live, lowest TCO and fastest throughput—regardless of data complexity.

With Transformation Extender it is all about data. It does much more than just transforming data from one format to another—it can take in data that is in a complex format, in multiple, different formats and map all of that data within a single map. Transformation Extender can receive one file (or even multiple files), and that file may have multiple different formats within it, with different structures, and Transformation Extender can take what is needed from the different parts of the structure and when it gets to the section that you want drill down into, it can do that irrespective of the fact that it is a completely different structure from what was in the line before.

Simply put, Transformation Extender is a data tamer. Why is it critical to have data tamer critical now? Because the amount of data that your organization deals with and the variety of data formats and the complexity of the data is growing exponentially. In today’s IT ecosystem, it is easy to say “data is power” but in actuality, it isn’t. It is the well-managed and understood data that has the real value.

Analytics and cognitive computing platforms require the ingestion of volumes of data in order to provide insights which are obtained by distilling out the (usually small portion of the) information that is truly important. Transformation Extender is the tool for feeding those systems by understanding and validating the data coming in to an appropriate level, mining for relevant information and creating content for downstream consumption.

For more information, contact your IBM representative or visit: ibm.com/software/products/en/transformation-extender

Introduction	Any-to-any and many-to-many transformations	Standards and validation	Reusability	Data-centric capabilities of Transformation Extender	Extensive capabilities and built-in functions	Conclusion	◀ 11 ▶
--------------	---	--------------------------	-------------	--	---	------------	--------



© Copyright IBM Corporation 2016

IBM Corporation
IBM Commerce
Route 100
Somers, NY 10589

Produced in the United States of America
October 2016

IBM, the IBM logo and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or TM), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Other company, product or service names may be trademarks or service marks of others.



Please Recycle

SWM14024USEN-00

Introduction	Any-to-any and many-to-many transformations	Standards and validation	Reusability	Data-centric capabilities of Transformation Extender	Extensive capabilities and built-in functions	Conclusion	◀ 12
--------------	---	--------------------------	-------------	--	---	------------	------