



Using canonical formats with BizTalk Server
Best Practices

Technical White Paper

motion10
10



Table of Contents

Using canonical formats with BizTalk Server	1
Introduction	3
Canonical format(s)	3
Advantages.....	4
Disadvantages.....	5
When NOT to use.....	5
Design	7
Requirements.....	7
Best practices	7
BizTalk implementation	12
Schema management	12
Use cases.....	13
Conclusion	15





Introduction

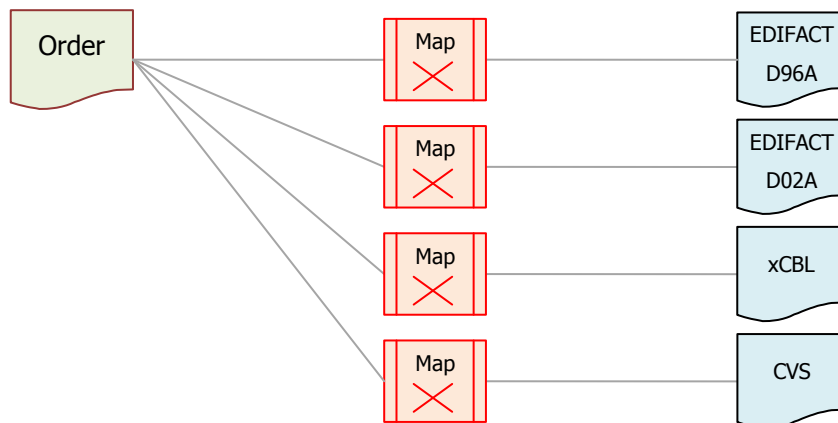
When information needs to be exchanged between applications and/or business partners, each application expects the information to be supplied in a specific format. Although many attempts have been made to define a common structure, especially in the B2B space, the reality is that two applications and/or partners will hardly ever be 100% compatible. Any system for transferring messages overcomes this problem by utilizing a translator to transform the message from the source format to the target format. Within BizTalk this transformation is achieved via a BizTalk map.

As each application may support multiple types of messages and utilize different formats or versions of them, the number of point to point maps grows exponentially. Having a large number of maps is in itself not an issue. However when changes are required then the number of maps effected is a serious issue. This impact applies to support, maintenance, and regression testing and also requires a level of synchronization between all the effected applications/partners.

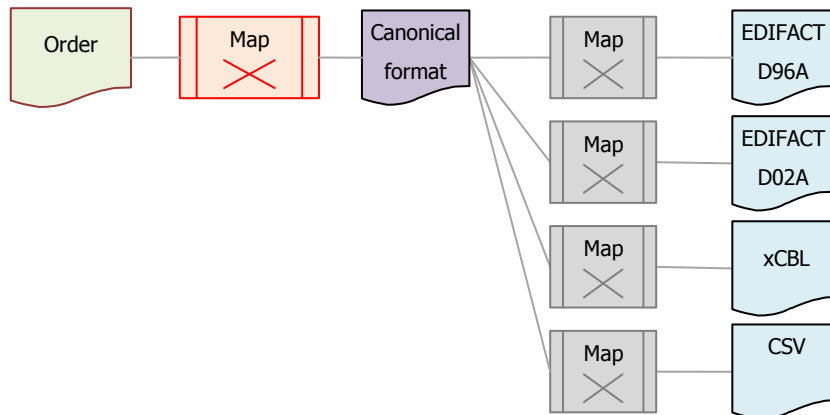
Canonical format(s)

Canonical Formats (CF) offer an extra level of abstraction between different formats. Using a common central format that is independent from all end user formats minimizes the impact of changes to any of the end user formats.

The following figure highlights the impact of a change to the Order: For example to change the structure or mapping of a specific element – all the maps must to be updated to reflect the new Order.



When a central canonical format is used the impact is limited to a single map:



Advantages

Domain expertise separation (Back-office -> CF -> External standards)

As each map only works to/from a single application/partner specific format, the knowledge and expertise required to define and later troubleshoot the map is limited to the single application or partner. An application specialist knowing for example everything about SAP interfaces will hardly ever also have in depth knowledge of for example EDI formats like EDIFACT, X12 or any of the vertical specific standards and how they should be implemented. This implies that a single SME (Subject Matter Expert) is typically required for a given map and avoids delays caused by when attempting to arrange meetings etc between multiple teams/SME, etc.

Future proof

When changes are planned to applications/back-end systems the impact can be minimized as only the appropriate map needs to be updated. The same is true for migrations to newer message formats. Through the use of version control and dynamic maps the roll out to partners can also be performed in a more controlled way.

Mapping and Validation simplified

As the appropriate pipeline disassembler converts the external (wire) format into the XML representation for processing within BizTalk the structure of the document is typically checked for syntax and semantic errors.





Each map needs only to perform mapping and any additional validation specific to the format being created. This avoids duplication and again minimized the impact of changes to the mapping/validation requirements.

Once converted to the canonical format the application of Business Rules can then be applied in a common and consistent manner.

Tracking and BAM

As all formats for a given message type are mapped to a single internal structure, tracking messages through the process flow can be performed in a standard way and becomes much easier. Also the extraction of consistent information for Business Activity Monitoring (BAM) makes the generation of BAM reports much simpler.

Caching

When processing a large batch of similar messages that need to go to/from a large variety of end user formats, the single application to/from Canonical Format map can be kept in cache for improved performance.

Mapping complex structures

For messages that require complex transformations, especially in the structure, using two maps (ERP/CF & CF/External) can greatly simplify the individual maps.

Service Orientation perspective

Reduction of coupling can be achieved, between the source formats and destination formats. Messages only need to be translated to or from the canonical format.

Disadvantages

When looked at from the perspective of a single message flow there is an obvious performance hit as two maps are executed for each instance of a message instead of one.

When trouble-shooting end-to-end issues, the extra step (map) adds another level of complexity to be considered.

When NOT to use

Using a canonical format requires each message pass through two maps which may add an unacceptable overhead when high volume / high throughput is required.

In all other cases it could be argued that canonical formats should always be used as the advantages typically outweigh the disadvantages even when only one or two message types are to be processed.





However in environments where all the partners use the same format and the messaging flows will not be changing in the foreseeable future then canonical formats may not offer sufficient advantages.





Design

Requirements

When planning the implementation of a canonical format, thought should be given to both the needs of the business and partners and the needs of the internal IT systems.

Business Requirements

The business analyst should be able to specify the business data that needs to be present to allow the messages to be successfully processed. However, thought should also be given to additional (optional) data that could be of use to partners.

IT Requirements

Apart from the business information, the addition of internal IT specific information must also be considered. Although of no use to the various applications or partners, this extra information can be of great benefit for simplifying processing, routing, and tracking. Also, information for support and help desks makes trouble shooting easier and more efficient.

Best practices

CF per family of messages or type of business document (invoices, orders, etc.)

Although a single canonical format could be created for all message types, the added complexity of multiple requirements outweighs any benefits. In practice the easiest solution is to create a format for each message type (Invoice, Inventory Report, etc.) or group of related messages (Orders, Order Response, etc.).

Ownership/Management

The size and format of all business related elements must be defined and owned by the relevant business analyst / department.

Any additional attributes of the "business" elements, such as the data type, enumerations lists, etc., plus the structure of the message itself should be owned and defined by the IT department.

Any changes to the final schema must be approved by the business once the IT department has performed an impact analysis.

Structure

The overall structure of the canonical format can be designed in one of three ways:

- Application driven





- B2B driven
- Generic (build from scratch)

Application driven is useful where a single back-office application is used to generate all outgoing messages and process all incoming messages. As such, the canonical format could match the format used by the back-office application (for example: an SAP IDOC). However, this does make the complete system more sensitive to changes in requirements or business logic. Also any changes may need to be implemented within the back-office and that is often more labor intensive than modifying a BizTalk map.

B2B driven takes the other extreme and bases the structure on the B2B format being used by the partners such as EDIFACT or xCBL, etc. The disadvantage with this is that the structure tends to be very large and sometimes overly complex.

I personally prefer the generic approach to build a schema from scratch. This allows the schema to exactly match the requirements of both the business and IT. The structure can also be created to make the mapping as easy as possible by avoiding large differences in structures.

Schema details

The schema itself will normally specify the structure, max/min occurrences, data type, length, & enumeration lists. However, when specifying these attributes it is easy to restrict the schema based on existing messages. Thought should also be given to possible future use/expansion and as such any type of limitation should be kept to a minimum. In fact, unless there are specific reasons for validating the internal messages against the schema I would strongly recommend keeping the schema as generic as possible.

Thought should also be given to including a section for internal use. This could include useful meta data and tracking information. Especially when dealing with inbound B2B messages the maps often need to access the message context. By including any required message context values with the message itself makes both the mapping and troubleshooting much easier.

Structure

The structure of the CF schema should be such to minimize the complexity of the maps wherever possible. This is specifically important with regards to the looping





requirements. Typically the target structure should take priority over the source structure when the CF structure is only being used for a single direction (inbound/outbound).

Where possible give all canonical format schemas a similar look and feel using similar sections (Header/Details/Summary etc) and make use of standard naming conventions.

Similar items should also be grouped in repeating sections with an appropriate qualifier/type element to identify the specific instance.

For example: All names and addresses within a given portion of the document (header/detail/summary, etc.) should be grouped together with the appropriate qualifier (Buyer, ShipTo, etc.)

Max/Min Occurrence

Where possible, the maximum occurrence of each parent node should be either 1 or * (unbounded).

Avoid limiting nodes to a specific number of occurrences as this may impact other (future) message flows.

Data type

As the data type for a given element is often different on the various external formats used, specifying specific data formats may limit the possible contents or force the use extra mapping logic to format the data. This is especially true in the B2B environment. Data types that often cause problems include: dates, times, and various numeric types. Even apparently obvious fields such as "Quantity" are not always what they appear when looking at the data type.

If there is no realistic business case to use a specific data type then using a string will typically result in the minimum amount of potential problems. However, this may impact the processing and/or format of numeric values.

Length

When choosing the length of an item, think about the actual length of the expected data with allowances for possible future use. Also bear in mind any padding that may be present in the contents.

Enumeration lists

Specifying a set of values that may appear within a given element severely limits the possible use of the element in the future and should be avoided wherever possible.





Naming conventions

Obviously the naming conventions used by the various end formats will vary depending on the standard used. As such, each application/end user SME will typically use different terms to define a given data value. Therefore the naming convention within the canonical format schemas should be both generic and meaningful. Avoid cryptic abbreviations and where possible format specific labels.

Documentation

One of the key benefits of canonical formats is to allow for reuse. As such development and management of the messaging flows must include standardized documentation on mapping to/from each canonical format.

Also, where each CF schema is used must also be carefully documented to ensure the impact of any change is easily identified.





Paper maps

Many companies already have standards for documentation with regards to layout, format, style, etc. In my experience the best tool for documenting maps is Excel workbooks. A single excel file is used per map and each workbook contains a minimum of four mandatory tabs.

- **Properties**
This contains details of the map, source, and target such as name, namespace, assembly, etc. and also the change register that records all the changes. When this list becomes too large simply outline the older changes and collapse the outline. Final details should include author, approval / review list, etc.
- **Source Schema**
This is just the source schema showing used and unused segments & elements. For example: unused items can be given a grey font. Also the structure should be shown using outlining. The schema should also show the end user format and not the actual BizTalk schema as this tab is often used by SME's and not developers. Examples would be the IDOC parser file or an EDIFACT MIG etc. Also ALL segments should be shown and not just those used. This avoids problems identifying the specific node in the actual schema when mapping.
- **Target Schema**
The same as the Source tab but then for the target.
- **Mapping**
The mapping tab should always be based on the target schema and contains two additional columns: One for source element (showing the full path) and constants; and one for mapping logic. Optional additional columns could be used to cross-reference to the actual BizTalk schema node especially when naming conventions may differ. For example typical EDI MIGs do not use the same naming conventions as the EDI schema supplied by BizTalk.

1	2	A	B	C	D	E	F	G	H	I	J	K	L	M
1	2	Pos	Grp	Seg	Comp	Elem	Name	Max	M/c	Type	Length	EFACT_D96A_DESADV	Source: SMD_DISAdvanceShipmentNotice	Mapping Logic
31												UNH7.4		
32	20			BGM			Beginning of Message	1	M			BGM		
33					C002		Document/Message Name	1	C			C002		
34					C002	1001	Document/Message name, coded	1	C	AN	1.3	C00201	"351"	
35					C002	1131	Code list qualifier	1	C	AN	1.3	C00202		
36					C002	3055	Code list responsible agency, coded	1	C	AN	1.3	C00203		
37					C002	1000	Document/Message name	1	C	AN	1.35	C00204		
38						1004	Document/Message number	1	C	AN	1.35	BGM02	Header/DeliveryNumber	
39						1225	Message function, coded	1	C	AN	1.3	BGM03		
40						4343	Response type, coded	1	C	AN	1.3	BGM04		
41	30			DTM			Date/Time/Period Message Date/	10	C			DTM		
42					C507		Date/Time/Period	1	M			C507		
43					C507	2005	Date/Time/period qualifier	1	M	AN	1.3	C50701	"137"	
44					C507	2380	Date/Time/period	1	C	AN	1.35	C50702	system-date + system-time	format = CCYYMMDDHHMM
45					C507	2379	Date/Time/period format qualifier	1	C	AN	1.3	C50703	"203"	
46	30			DTM			Date/Time/Period Expected Deliver	10	C			DTM		
47					C507		Date/Time/Period	1	M			C507		
48					C507	2005	Date/Time/period qualifier	1	M	AN	1.3	C50701	"64"	
49					C507	2380	Date/Time/period	1	C	AN	1.35	C50702	Header/DeadlineDates/PlannedStartDate + Header/DeadlineDates/PlannedStartTime	when Header/DeadlineDates/DateQualifier = "Delivery" format = CCYYMMDDHHMM
50					C507	2379	Date/Time/period format qualifier	1	C	AN	1.3	C50703	"203"	
51	30			DTM			Date/Time/Period Expected Deliver	10	C			DTM		
52					C507		Date/Time/Period	1	M			C507		
53					C507	2005	Date/Time/period qualifier	1	M	AN	1.3	C50701	"63"	
					C507	2380	Date/Time/period	1	C	AN	1.35	C50702	Header/DeadlineDates/PlannedStartDate + Header/DeadlineDates/PlannedStartTime	when Header/DeadlineDates/DateQualifier = "Delivery" format = CCYYMMDDHHMM



In general, avoid the use of multiple colors and only highlight any changes made in the latest version.

Additional tabs can be added if needed but should only contain additional background information such as example files etc.

Mapping guidelines

Although not really related to canonical formats, while discussing documentation I strongly recommend the use of mapping guidelines. This results in all maps also being developed with a common "look and feel". This allows any developer to troubleshoot or modify a map with the minimum time lost in trying to understand the original developer's logic, etc.

BizTalk implementation

Schema management

As the number of schemas and maps increases it is imperative that a central administration is kept which details the use of each schema and map. Also if a central database is used for dynamic mapping that can also be queried to link the maps to the various partners/applications.

This allows for more efficient and less error-prone impact analysis to be performed on potential changes. If possible this should be based on reality and not planning. This implies that the status should be regularly updated based on queries run against the BizTalkMgmtDb database. These queries should also be performed in each tier (Test, QA, & Production).

A typical query for deployed maps

```
SELECT
    ('Test') as "Environment"
    ,bts_item.FullName as "Map Name"
    ,bt_MapSpec.indoc_docspec_name as "Source Schema"
    ,bt_MapSpec.outdoc_docspec_name as "Target Schema"
    ,bt_MapSpec.date_modified
    ,bts_assembly.nvcName as "Assembly Name"
FROM
    BizTalkMgmtDb.dbo.bt_MapSpec
    inner join BizTalkMgmtDb.dbo.bts_item on bt_MapSpec.itemid = bts_item.ID
    inner join BizTalkMgmtDb.dbo.bts_assembly on bt_MapSpec.assemblyid =
bts_assembly.nID
```





A typical query for deployed schemas

```
SELECT
    ('Test') as "Environment"
    , schema_root_name
    , docspec_name
    , bts_assembly.nvcName as "Assembly Name"
    , msgtype
FROM
    BizTalkMgmtDb.dbo.bt_DocumentSpec,
    BizTalkMgmtDb.dbo.bts_assembly
```

Version control

As time passes changes will be required to the various schemas and maps due to changing and/or new requirements. Obviously a change to a common schema or map could impact multiple applications or partners.

To minimize this impact version control should be implemented to ensure any updates can be tested and rolled out in easily controlled fashion as and when it suites each application or partners. Also, by creating each schema and/or map in its own project (DLL) implies that unused DLL's can be un-deployed without affecting any existing message flows.

Maps - where/when to execute

BizTalk allows maps can be run in any combination of Receive Pipeline, Orchestration, and Send Pipeline. The use of a canonical format (usually) implies the use of two maps. In theory these could simply be placed on the receive and send pipelines. This works well as long as there are a) not too many maps, and b) there is a single map for each message type.

For larger and more complex messaging scenarios it is much easier to invoke the maps from within an orchestration. For the greatest benefit I would strongly advise the use of dynamic mapping as this allows maps to be added, modified, or deleted "on-the-fly".

Use cases

Use Case 1

This first example was implemented by a multi-national company who had a single back-office system at the head-office that needed to process various messages from partners from all around the world. The formats were primarily various versions of EDI with the XML formats being used as well.

Initially the idea was to base the canonical formats on sub-sets of one of the standard XML formats but this was soon rejected as the resulting schema was too often too





complex. However, care was taken to ensure all the various canonical formats used similar naming conventions and grouped the information in similar structures. For example: A standard NameAddress structure was used even if the given message did not require all the fields.

The back-office system was capable of processing XML directly. So the choice was made to have the back-office read/write directly using the canonical format and thereby removing the need for the additional map except for the occasional situation where a given partner/message required some additional custom processing.

Although many partners required the messages in the same format, version/release (for example EDIFACT D96A), the maps were often not 100% identical. As such the mapping specifications were created showing the variations from a "standard" map. Once an initial map was created for a given format, this was then copied and modified.

As such the development time for a new map was reduced to a few hours. This, together with the use of dynamic mapping which allowed the maps to be deployed without impacting any existing orchestrations, etc. meant that new partner maps could be available for initial testing very efficiently.

Use Case 2

In the next example, the multi-national has multiple back-office systems around the world. However, the intention is to migrate these older systems to a single product. Also, the number of partner formats was very large and diverse as well. This is an obvious candidate for a design involving canonical formats especially with the requirement to migrate legacy back-office systems at some point in the near future.

The initial design of the canonical format was simply based on the partners' requirements plus any additional fields the business thought may be needed in the future.

A single canonical format was used for each business document (Order, Invoice, Manifest, etc.). However the occasional exception was made if a given company division required a large variation in structure and/or fields.

Again the maps themselves were implemented using dynamic mapping with the design also allowing for additional custom processing (orchestrations).

Given the large (and growing) number of maps, schemas, partners, etc. the documentation and administration is very important and must be coordinated centrally.





Conclusion

As a consultant, I have been involved with implementing B2B scenarios for the last 30 years for customers all around the world on a large variety of platforms and products. Since 2002 the work has been focused on BizTalk Server. Obviously not all made use of a canonical format but I feel I can honestly say the majority would have benefited from the use of such a design.

However, it is not something that should be rushed, especially the design of the canonical format itself – both structure and contents.

Also, it is very important that clear standards are used with regards to documentation and development to allow for efficient implementation of new partners, maps, messages, etc. and the ongoing support of existing partners, maps, messages, etc.

About motion10

motion10 is a global provider of consultancy services geared to complex integrated environments. Its service offering also includes implementation, support and training. Unlike other systems integrators motion10 focuses exclusively on the Microsoft Application Platform which supports products such as BizTalk Server, SharePoint Server and SQL Server. The company has its own experts who also advise Microsoft in the further development of these products. Headquartered in Rotterdam (the Netherlands), motion10 has offices in Atlanta and Seattle, and an extensive partner network across Europe, North America and Asia. The motion10 consultants have completed a total of more than 500 BizTalk Server implementations. For more information: www.motion10.com.

