Enablement of B2Bi Resources in B2BAC

Sunil Kumar V[1], Girish Rao Salanke N S[2]

[1] M.Tech CNE, Department of CSE, R.V College of Engineering, Bengaluru, Karnataka, 560040, India

[2]Assistant Professor, Department of CSE, R.V College of Engineering, Bengaluru, Karnataka, 560040, India

Abstract: ***IBM sterling Business to Business (B2B) integrator enables security rich integration of complex B2B processes with diverse partner communities. It implements security policies between organization and partners for a better relationship. IBM sterling B2Bi software helps companies to execute a smarter commerce strategy. IBM B2B Advanced Communications (B2BAC) gateway extends, enhances and compliments the capabilities of IBM sterling B2B integrator. It enables high availability and high performance for trading partner networks. It incorporates a new technology stack that segregates communication and protocol support from core message processing. It supports for new Applicability Statement 4(AS4) communication protocol. B2Bi resources related information are stored in the tables of backend database. Identify the tables related to AS2 protocol. Develop an API to fetch the information from the database and provide it as a service to third party clients. B2BAC backend APIs can make use of service provided by B2Bi and obtain the required AS2 related information.***

Keywords: B2B Integrator, B2B Advanced Communications

1. Introduction

IBM Sterling B2B Integrator is a transaction engine and set of components designed to run processes that are defined and managed according to our business needs and it supports high-volume electronic message exchange, complex routing, translation, and flexible interaction with multiple internal systems and external business partners[1]. IBM Sterling B2B Integrator has robust security infrastructure, visual management tools for easy configuration of and visibility into work flows, system and trading partner activities. It integrates applications, processes, data and people, both within and outside an organization

IBM Sterling B2B Integrator includes a perimeter server communication management component which is a single entry point installed in a demilitarized zone (DMZ) to manage communication flows between a perimeter network and Sterling B2B Integrator transport adapters enables an adapter to communicate through an internal firewall with a perimeter server within the demilitarized zone(DMZ).

B2B communication protocols include Web services (SOAP), S/FTP/S client and server, HTTP and HTTP/S, SMTP, AS1, AS2, AS3 and RosettaNet, WebDAV, TCP/IP. Backend connectivity adapters are provided by enterprise applications such as SAP, Oracle, PeopleSoft, Vantive etc.IBM Sterling B2B integrator is designed around a core transaction/business process engine, which orchestrates your message exchange, routing, translation, and other processes. Pre-defined business processes add capabilities on the fly for specific activities as required.

IBM B2BAC is an advanced communication gateway that enables high availability, high performance AS2 and AS4 based trading partner networks. It represents new, modularized communication capabilities that companies can be selectively deployed[2]. It incorporates a new technology stack that segregates communication and protocol support from core message processing and also extends, enhances and compliments the capabilities of IBM Sterling B2B Integrator. IBM B2BAC key capabilities includes modularized communications architecture, support for the new Applicability Statement 4(AS4) communication protocol, high availability, high performance architecture for Applicability Statements AS2 and AS4 protocols.

IBM B2BAC modularize our approach to address complex integration challenges, address changing market requirements including high availability, real-time processing, archival, and standards processing. It provide a best of breed, end to end experience focused on customer excellence since it is built on the strategic IBM technologies enterprises can use to grow their business and improve operational efficiencies.

IBM B2B Advanced communications platform with modular architecture, addresses complex integration challenges with high availability, scalable and real-time message based data exchanges with simplified onboarding based on policy driven configuration, guided tasks and ability to easily manage dependencies. It extends, enhances and compliments IBM Sterling B2B Integrator.

Artifacts such as receivers, destinations, organizations, policies and servers are reusable across exchange profiles. With Intelligent Onboarding, it can add new artifacts in stream if required. Typical B2B software forces, users to cancel out, go create artifacts and then restart their wizard from the beginning.

In this paper we propose an approach to design and develop an API that fetch AS2 related resources for efficient migration from B2Bi into B2BAC. The basic purpose, security of resources exchanged, trading partners communication etc., by making use of some of the existing services and also by querying the resources that are stored in product's database. B2Bi APIs provide AS2 information retrieval to end users.

The rest of the paper is organized as follows – Section 2 discusses the Model along with Data flow and Workflow, Section 3 gives the implementation details of enabling various resources. Section 4 contains the experimental results and observations. The paper is concluded with conclusions and future work, followed by acknowledgement and references.

2. Model

A. Overview of the design

Figure 1 shows the Data-Flow Diagram (DFD) Level 0, it shows the detailed visualization of flow of information in and out of each module. When some input is given to a processing module, some computation is carried out. The result is produced with the help of given input.

Level-0 is more general and high level view of input and output. The Level-0 DFD is also known as context diagram. Here the basic level data flow is shown. Here only the main module is included. Input given to the system at this level plays a vital role in understanding the nature of primary input set required for the system. Then the output produced in this Level-0 DFD is the final output and it remain same at all the levels.

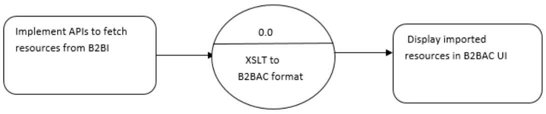

Implement APIs to fetch resources from B2BI
0.0
XSLT to B2BAC format
Display imported resources in B2BAC UI

Figure 1: Data Flow Diagram – Level 0

- From the above figure it is obvious that, Rest APIs are implemented at B2BI to fetch resources from its database. It is converted into the suitable XML format that is accepted by B2BAC. Finally, imported resources are successfully displayed in B2BAC UI.

- Figure 2 shows the Data-Flow Diagram (DFD) Level 1, it shows the immediate next sub-module required for computation. In the below figure the next module being explored are a set of independent sub-processes. These sub-processes run in series where output of one will be given as input to next module. The applicability statement 2 protocol resources which are fetched from B2BI database are in XML format that is accepted by B2BI. It is transformed using XSL transformation to a format that is accepted by B2BAC. Hence, all the resources are successfully imported by B2BAC and is displayed in UI.

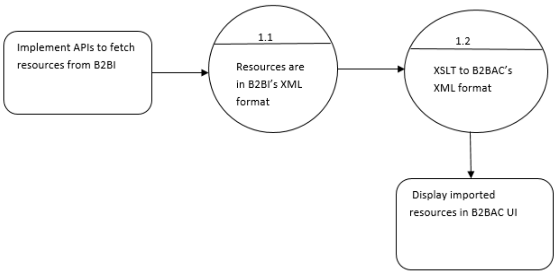

Implement APIs to fetch resources from B2BI
1.1
Resources are in B2BI's XML format
1.2
XSLT to B2BAC's XML format
Display imported resources in B2BAC UI

Figure 2: Data Flow Diagram – Level 1

- Figure 3 shows the Data-Flow Diagram (DFD) Level 2, it shows all the sub-modules are explored. Also, the flow of data between each module is clearly explored.

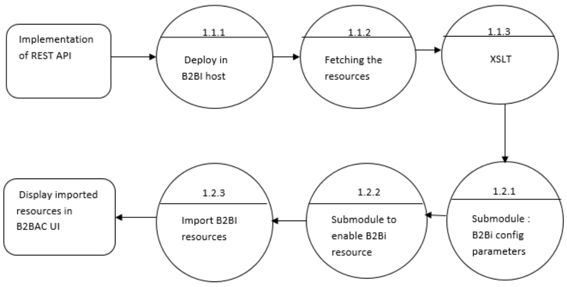

Implementation of REST API
1.1.1
Deploy in B2BI host
1.1.2
Fetching the resources
1.1.3
XSLT
1.2.1
Submodule : B2Bi config parameters
1.2.2
Submodule to enable B2Bi resource
1.2.3
Import B2BI resources
Display imported resources in B2BAC UI

Figure 3: Data Flow Diagram – Level 2

The level 2 DFD explains that the implemented APIs are successfully deployed in B2BI host and all the resources in that particular host could be fetched from its backend database successfully. The output will be in either XML/JSON format and it is converted to a suitable format that is acceptable by B2BAC. There are two submodules implemented at B2BAC UI level, one for providing configuration parameters of B2BI host and other is for enabling the B2BI resources and finally after all resources are imported, it will be displayed in B2BAC UI successfully.

B. Workflow of Enabling B2Bi resources

Figure 4 shows the overall workflow of enabling B2Bi resources, The Enable B2Bi Resources submodule is implemented in B2BAC product. On selection, it will list the different communication protocols, list of sterling B2Bi hosts. Once client chooses appropriate B2Bi host and communication protocol. The flow will make a REST API call to Org lister service which will return list of organizations associated with that communication protocol pertaining to that particular sterling B2B integrator host.

Once the organization is selected from the list of organizations, it will in turn make a REST API call to obtain all the trading partners associated with that organization. After all organization, trading partners are listed, required partners will be selected and enabled. On enablement of partners, it will import all the exchange profile details using Exchange Profile Service.

Exchange profile service will return all exchange profile details such as contract details, document exchange details, transport details, delivery channel details. In case, if there is existing relationship is deleted between the partners, this service will return a error message indicating the failure of exchange profile resource import.

XSLT is performed to change a XML format of B2Bi into B2BAC. Since, import of resources is done by B2BAC product. All the attributes at B2BAC UI are mapped accordingly with the attributes returned by web service browser of B2Bi and hence resource import will be successful if there is a relationship between the organization and trading partner.

After all exchange profile resources are imported, members will be enabled and can been seen in the submodule named enabled members that is present in B2BAC UI. All the resource will then be stored into B2BAC backend database so that every time it need not import for the same members, it can directly fetch from its own database improving performance by reducing the delay in enablement of B2Bi resources in B2BAC product.

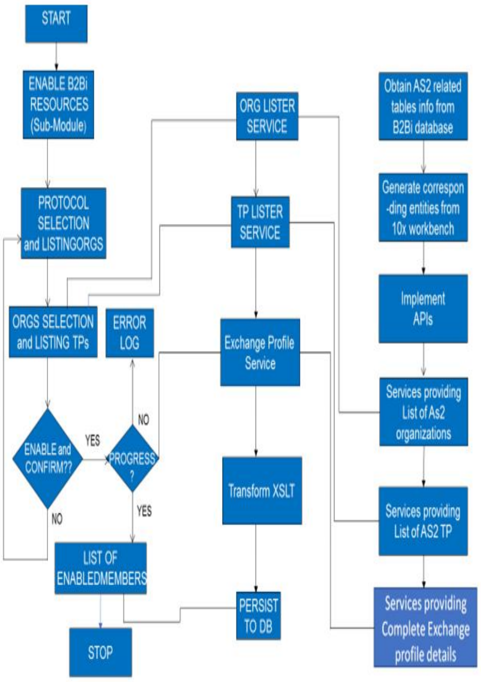

START
ENABLE B2Bi RESOURCES (Sub-Module)
PROTOCOL SELECTION and LISTINGORGS
ORGS SELECTION and LISTING TPs
ERROR LOG
ENABLE and CONFIRM??
YES
NO
PROGRESS ?
NO
YES
LIST OF ENABLEDMEMBERS
STOP
ORG LISTER SERVICE
TP LISTER SERVICE
Exchange Profile Service
Transform XSLT
PERSIST TO DB
Obtain AS2 related tables info from B2Bi database
Generate correspon -ding entities from 10x workbench
Implement APIs
Services providing List of As2 organizations
Services providing List of AS2 TP
Services providing Complete Exchange profile details

Figure 4: Workflow of Enabling B2Bi resources

The next section discusses the implementation of various features of the Performance Analyzer.

3. Implementation

Enable B2Bi resources enables client to make a selection on list of B2Bi hosts from which the AS2 trading partners details to be fetched. Once B2Bi is chosen, all the AS2 related organizations are imported and displayed on B2BAC UI successfully. By selecting one of the organization, trading partners associated with organization are listed. Client will make a choice on organization and its associated partners and clicks on Enable. It will import all the exchange profile details associated between those partners and can be seen in B2BAC UI successfully.

There are 'n' number sterling B2Bi integrators. It is essential to incorporate or integrate those with B2BAC so that all clients having multiple trading partners will have successful import of resources from B2Bi.[3] Once Enable B2Bi Resources is selected, it will list all B2Bi hosts and client can make a choice of his own. This list is made possible by providing configuration details of each B2Bi hosts in B2BAC UI itself. The submodule used to provide configuration details is B2Bi Integration Config.The submodule named as B2Bi Integration Config is implemented in B2BAC UI and selection of submodule will enable client to provide configuration details of B2Bi hosts. The configuration details include hostname of sterling B2Bi integrator from which the resources need to be fetched, port number through which the connection will be established for that particular host and login credentials for that B2Bi host. New users can be created for the same host and corresponding credentials could be provided in the submodule B2Bi integration config. Once the configuration details are provided, the hostnames will appear in the submodule 'Enable B2Bi Resources' and required host can be selected by the client. The submodule named Enable B2Bi resources also provides an option to select the required protocol which is used by trading partners.

Once configuration parameters are provided using Create B2Bi Integration Param. The list of hosts will appear on Enable B2Bi resources submodule. Based on the selection of host and communication protocol, all the associated organization would be listed. Client will select one of the organization and all trading partners associated with that particular organization would be listed. Client will then make a selection organization and trading partner pair for which the relationship would be established at sterling B2Bi host. After selection, once client clicks on Enable button, all the exchange profile resources between trading partners would be enabled in B2BAC.

After the successful import of exchange profile resources from B2Bi host, list of enabled members could be seen by clicking on "Show Enabled Members' which is highlighted in blue. It is basically used to differentiate between the members for which the resources have been imported and members which are yet to be enabled.

Figure 5 shows the approach that is used for implementation of REST APIs to fetch the resources from B2BI.

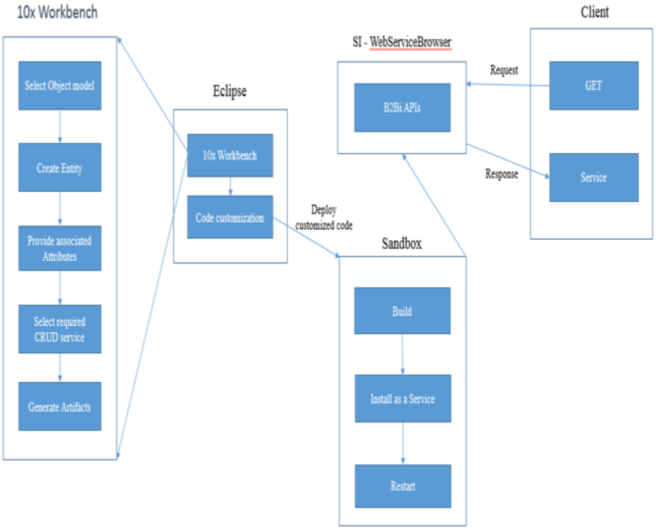


Figure 5: Implementation to fetch resources from B2BI

The 10x workbench is used to generate entity, it ensures that required attributes of that particular resource is defined. The contents of attributes are obtained from corresponding columns from the tables present in database. This can also be done using some of the existing services rather than directly querying on database.

Code customization is done on generated classes to fetch the resource details then these classes are deployed in sandbox. The corresponding results will be displayed in a browser in JSON/XML format. The third party clients can make use of these services by making a rest api call for that particular resource.

The implemented code is deployed in sandbox and in case or errors/exceptions, message logs and console logs need to be analyzed and corresponding errors needs to be resolved. Testing of this code can be done by manually exporting the xml file of that particular resource from the dashboard and comparisons are done. In this way, different APIs are implemented to fetch different resource details.

The next section describes the test environment used for testing the performance evaluation tool.

4. Experimental Evaluation

A. Test Environment

a) Windows Server running IIS 6 or IIS 7.
b) Third party client software is required to establish connectivity to all different databases pertaining to different host machines.
c) Softwares are required to transfer files from local host into B2Bi's host.
d) Apache Web Server as installed.
e) Microsoft Sql Server as installed per OAT Systems specification.
f) Set up TestNG software to run automation scripts.
g) Set up JUnit to run automation script for UI level automation.
h) IBM web liberty server to run code and generating 10x workbench

Unit testing is done to assure the proper working of each individual modules of the system. If any faults are identified during this test, then they can be fixed easily. This reduces the overhead during integration phase. unit testing carried out for Automation script for entity generation module, along with its sample input, expected output and actual output. unit-test case for AS2 Organization details. It will be used to fetch all the organization details such as organization address, organization name, unique identifier of that organization. For each of these organizations created, separate entity id is created and stored in Entity table referencing organization's unique id.

Unit-Test for fetching AS2 Trading partner details. This test case is used to check whether all the details such as trading partners email address, email port, display name, unique identifier of that trading partner. Unit-Test for fetching AS2 relationship details. This test case is used to check whether all the resources such as transport details, compression techniques and signing algorithms used for communication between partners, contract details, delivery channel details, document exchange details with respect to selected organization and trading partners. URL should be formed by taking hostname, port numbers as the input and same URL should be hit by third party clients by making REST API calls to fetch all relationship or exchange profile details associated with partners.

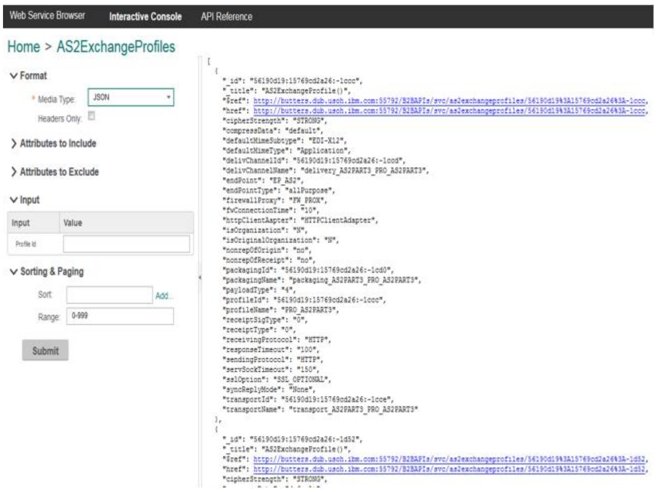
System testing is carried out after integrating all the modules of the systems. It is very much essential that they work together as expected even after integration. Once the integration is done, the internal working is not visible to the user. All the modules are now single entity

B. Results

Experimental analysis refers to systematic evaluation carried out to find out specific features and behavior of the system.

Experimental dataset consists of the different kind of data for which the system designed is tested for. The various datasets used in the enabling the B2Bi resources is listed below:

- Organizations
- Trading partners
- Certificates
- Protocols
- Exchange Profile resources
- Configuration Parameters
- Contracts
- Delivery Channels
- Document Exchange
- Applicability Statement 2
- Transport protocols



Media Type: JSON
Headers Only: □

Input	Value
Profile Id	

Sort:
Range: 0-999

```
[
  {
    "_id": "56190d19:15769cd2a26:-1ccc",
    "_title": "AS2ExchangeProfile()",
    "$ref": http://butters.dub.usoh.ibm.com:55792/B2BAPIs/svc/as2exchangeprofiles/56190d19%3A15769cd2a26%3A-1ccc,
    "href": http://butters.dub.usoh.ibm.com:55792/B2BAPIs/svc/as2exchangeprofiles/56190d19%3A15769cd2a26%3A-1ccc,
    "cipherStrength": "STRONG",
    "compressData": "default",
    "defaultMimeSubtype": "EDI-X12",
    "defaultMimeType": "Application",
    "delivChannelId": "56190d19:15769cd2a26:-1ccd",
    "delivChannelName": "delivery_AS2PART3_PRO_AS2PART3",
    "endPoint": "EP_AS2",
    "endPointType": "allPurpose",
    "firewallProxy": "FW_PROX",
    "fwConnectionTime": "10",
    "httpClientAapter": "HTTPClientAdapter",
    "isOrganization": "N",
    "isOriginalOrganization": "N",
    "nonrepOfOrigin": "no",
    "nonrepOfReceipt": "no",
    "packagingId": "56190d19:15769cd2a26:-1cd0",
    "packagingName": "packaging_AS2PART3_PRO_AS2PART3",
    "payloadType": "4",
    "profileId": "56190d19:15769cd2a26:-1ccc",
    "profileName": "PRO_AS2PART3",
    "receiptSigType": "0",
    "receiptType": "0",
    "receivingProtocol": "HTTP",
    "responseTimeout": "100",
    "sendingProtocol": "HTTP",
    "servSockTimeout": "150",
    "sslOption": "SSL_OPTIONAL",
    "syncReplyMode": "None",
    "transportId": "56190d19:15769cd2a26:-1cce",
    "transportName": "transport_AS2PART3_PRO_AS2PART3"
  },
  {
    "_id": "56190d19:15769cd2a26:-1d52",
    "_title": "AS2ExchangeProfile()",
    "$ref": http://butters.dub.usoh.ibm.com:55792/B2BAPIs/svc/as2exchangeprofiles/56190d19%3A15769cd2a26%3A-1d52,
    "href": http://butters.dub.usoh.ibm.com:55792/B2BAPIs/svc/as2exchangeprofiles/56190d19%3A15769cd2a26%3A-1d52,
    "cipherStrength": "STRONG",
```

Figure 6: Output in JSON format.

Figure 6 shows Output of exchange profile resources in JSON format. It can be seen that, resources related to organization and trading partner can be differentiated based on is Organization attribute. It also specifies about compression technique used, delivery channel details and also transport mechanisms that are involved between organization and the trading partner.

5. Conclusion and Future Work

The architectural POC (proof of concept) on the APIs to fetch AS2 related resources information and implementation of submodule to provide B2Bi config parameters in B2BAC UI is accomplished successfully.

The APIs to fetch AS2 related resources information is done successfully. Implementation of submodule to enable B2Bi resources in B2BAC UI is successful. Implementation of submodule to provide B2Bi config parameters in B2BAC UI is successful. Integration of B2Bi with B2BAC is successful with successful resource import.

Following are some of the future enhancements identified for the proposed project.

a) The project can be extended to import the resources of HTTP, SFTP.
b) To enable B2BAC to accept the resources in its own format. Hence, avoiding XSL transformation.

6. Acknowledgment

All the above mentioned narratives and accomplishments are carried out as part of my internship work in IBM for the PG programme (MTech) under Visvesvaraya technological university. We thank the IBM India Private Limited team members for their feedback in improving the tool and the managers for their support. Also, we thank the project committee, Dept. of CSE, R. V. College of Engineering for their timely guidance.

References

[1] C. McGregor, S. Kumaran, “Business process monitoring using web services in B2B e-commerce”, Addison-Wesley Professional, 2002.

[2] T. Vitvar, “Advances in e-business collaboration : semantic Web services in SAP R/3 B2B integration, Communications of B2B, IEEE, 2011.

[3] Liang-Jie Zhang, Haifei Li, H.Chang, Tian Chao, “XML-based advanced UDDI search mechanism for B2B integration”, Proceedings of the IEEE pp. 879-899, 2002.

[4] Leonid Shumskey, Pavel Shapkin, Viacheslav Wolfengagen, “Usage of semantic transformations in B2B integration solutions”, Springer, 2014.

[5] Monika Lobaziewicz, “Integration of B2B systems that supports management of construction processes with ERP systems”, Proceedings of the international workshop on B2B Integrator, 2015.

[6] Cristiano Colombo, Paolo Mauro, “B2B networks : Select, control and improve critical customer-supplier relationships”, Springer Berlin Heidelberg, 2005.