Multi-source eBonding *--DRAFT--*

About

eBonding is a term used to describe the electronic passing of information for business-to-business (B2B) operations. Typically eBonding is used to keep two system of records in sync. A record on one system has the same information for a record on a different system. Each system passes information between each other as information on the record changes; hence both records contain the same information. Businesses setup and configure eBonding as a means to maintain a system of record they control automatically saving time and resources; i.e., billing, audit, automation, etc. etc..

Multi-source eBonding is when you have three or more systems coordinating together on the same record from multiple sources. Multi-source eBonding is a mature B2B model for multi-supplier strategies, where one company coordinates with different multiple suppliers that provide various services within an organization. The advantage of a multi-supplier model is the model provides an organization the ability to right-size their enterprise needs across many suppliers that render a variety of services. Versus letting a single supplier render all operational needs within an organization.

This article is to help ServiceNow admins to setup multi-source eBonding framework for incident tickets. Incidents are normally a prime candidate to be shared across multiple suppliers. When an outage to a

service is reported, there could be multiple suppliers working together to resolve the incident. Also if a supplier is utilized as an external call center, then routing incoming incident tickets to the appropriate secondary supplier needs to be performed. Given these two scenarios enterprises typically model, make incidents the first type of ticket to build a multi-source eBond framework. The same framework can be applied to other ticket types in ServiceNow with some minor adjustments.

Progressive thoughts

Here are the progressive thoughts as multi-source eBonding is being developed to go back and refer too.

- You enterprise will eBond with suppliers, so therefore is makes sense to create company records in the core_company table for each supplier.
- Company records need to be associated with the local ServiceNow accounts which are used by the supplier to eBond with ServiceNow. This will help you keep track what ServiceNow account is used by the supplier to make their RESTful API calls.
- eBond enabled companies only need to be associated with accounts that have the role of **eBond Account**.
- eBonding is triggered based on the **Assignment Group** a record to assigned to. Therefore, group (sys_user_group) records need to associated with the companies that ServiceNow are eBonded with. Then in a business rule, if the assignment group company is eBonded, then continue to eBond operations.
- Turning off/on eBonding is as simple as toggling the company record in the core_company table.

Follow along

Each section will include an update set for those that wish to install the multi-source eBonding on their instance and the instructions to create the update set. ServiceNow offers many ways to solve a problem or configure an operational model. These instructions are a way and does not represent the way on creating a multi-source eBonding framework. The best way to use these instructions is to read through them and see how the framework can be adopted and modified to fit your organization.

> **Update sets**
>
> The instructions do not cover how-to use update sets or the nuisances of update sets. You can explore on how to leverage update sets within the online ServiceNow documentation.

> **Terms & Meanings**
>
> - Supplier: The legal entity that ServiceNow connects too. You can substitute supplier for vendor.
> - Company: A record from the core_company table in ServiceNow.

> **Modifying OOTB tables**

> The article focuses on creating a robust framework tied closely to the out of the box (OOTB) tables already existing in ServiceNow. There are plenty of alternative implementation solutions at your disposal to accomplish the same result. Your mileage may vary depending on scope defined by and practices set by your organization.

Foundation setup

The foundation setup is based on the premise that when an incident is created the *Service* and *Service Offering* within an incident are mandatory fields. This is a best practice to follow with a multi-supplier strategy for the enterprise. The reason is this removes the question "who supports this" and focuses on the true issue of "what is broken?" Who solves the incident is not of any concern to most customers and requiring customers or fulfillers to know what assignment group to assign a ticket in large enterprise environments is unreasonable and prone to misrouted tickets.

Service Offering --> **Assignment group** --> **Company** --> **User account**

The *Service Offering* within an incident determines who the ticket is assigned to in the *Assignment group*. The *Assignment group* will be linked to a supplier (company record). This allows enterprises the ability to swap out suppliers across various services within ServiceNow quickly and easily reducing long term run and maintain costs.

> **Assignment groups**
>
> Normally an *Assignment group* will contain all the members that support the *Service Offering** within an incident. In the eBond case it is normal if their are no members of an *Assignment group*. It is also normal to have members in an *Assignment group* that is eBonded with a supplier as well; where the enterprise allows the supplier to log into the enterprise ServiceNow to fulfill tickets. Either scenario still works with eBonding.

The supplier will have a company record that is linked to a user record that represents the eBond account the supplier will use to make inbound RESTful calls. This will become a means to quickly identify what account a company uses for eBonding and creates a "lockout" point for security.

Demo data

This how-to provides demo data that you can use to test the multi-source eBonding framework. The scenario is your organization has hired Alpha Co. and Beta Co. to supply IT services across your enterprise. Alpha Co. supplies data center operations and Beta Co. supplies backup and recovery operations for the data center. An incident occurred at the data center where both Alpha Co. and Beta Co. need to work together to resolve the incident.

Update set

eBond Collegiality v1.0

Add eBond account role

The *eBond Account* role denotes which accounts are used by suppliers to eBond with your instance of ServiceNow. Each supplier should have their own local ServiceNow account that the supplier will use to connect to your instance of ServiceNow passing inbound REST payloads. This role will be used in various ways in the multi-source eBond framework.

1. Navigate to **User Administration** > **Roles** and click **New**.
2. Under *Role New Record*, fill in the following fields:
    - Name: eBond Account
    - Description: Denotes the sys_user account is to be used for eBonding activities.
3. Click **Submit**, which will create the new role.

> **Demo Data - User accounts**
>
> Setting up two accounts that will be used by two suppliers Alpha Co. and Beta Co. to make RESTful calls for eBonding.
>
> 1. Navigate to **User Administration** > **Users**, and click **New**.
> 2. Under the **User New Record** section, fill in the following field:
>   - User ID: eBondAlpha
>   - First name: Alpha Co
>   - Last name: eBond Account
> 3. In the record header, right-click and select **Save**.
> 4. In the **Roles** tab, click "Edit..."
> 5. Under the **Collection** list, find the *eBond Account* role and add it **>** to the **Roles List**.
> 6. Click **Save**.
> 7. Click **New**.
> 8. Under the **User New Record** section, fill in the following field:
>   - User ID: eBondBeta
>   - First name: Beta Co
>   - Last name: eBond Account
> 9. In the record header, right-click and select **Save**.
> 10. In the **Roles** tab, click "Edit..."
> 11. Under the **Collection** list, find the *eBond Account* role and add it **>** to the **Roles List**.
> 12. Click **Save**.

Changes to core_company table

We need two new columns in the core_company table; *eBond account* and *eBonded*. The *eBond account* is a reference to sys_user table that links which account the company uses to eBond with your instance. The *eBonded* is a true or false field that we can use to turn off eBonding at the company level.

1. Navigate to **System Definition** > **Tables** and open the *core_company* table.
2. Under the **Columns** tab, click **New** to create a new column.
3. Under the **Dictionary Entry New Record** section, fill in the following fields:

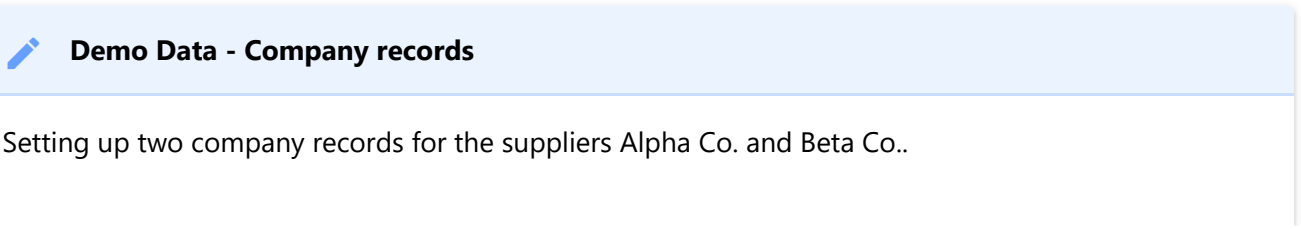
    - Type: Reference
    - Column label: eBond account
4. Under the **Reference Specification** tab, fill in the following fields:
    - Table to reference: sys_user
    - Reference qual condition:
        - Active is true AND
        - Roles is eBond Account
5. Click **Submit**.
6. Under the **Columns** tab, click **New** to create a new column.
7. Under the **Dictionary Entry New Record** section, fill in the following fields:
    - Type: True/False
    - Column label: eBonded
8. Under the **Default value** tab, fill in the following field:
    - Default value: false
9. Click **Submit**.

OPTIONAL: Modify the view on company records to see the *eBond account* and *eBonded* fields.

1. Navigate to **Organization** > **Companies** and click on any of the company records.
2. Navigate to **Additional actions** > **Configure** > **Form Layout**, which will bring up the **Configuring Company form**.
3. Under the **Available** list, find the *eBond account* and *eBonded* fields and add them **>** to the **Selected** list.
4. Move the *eBond account* and *eBonded* fields up the **Selected** list **^** to be below the *Stock price*.
5. Click **Save**.

OPTIONAL: Modify the view on company records to see the *eBond account* only if the *eBonded* field is true.

1. Navigate to **System UI** > **UI Polices** and click **New**.
2. Under the **UI Policy New record** section, fill in the following fields:
    - Table: core_company
    - Short description: Show eBond account in company
3. Under the **When to apply** tab, fill in the following fields:
    - Conditions:
        - eBonded is true
4. In the record header, right-click and select **Save**.
5. In the **UI Policy Actions**, click **New**.
6. Under the **UI Policy Action New record** section, fill in the following fields:
    - Field name: eBond account
    - Mandatory: True
    - Visible: True
7. Click **Submit**.
8. Click **Update**.



1. Navigate to **Organization** > **Companies**, and click **New**.
2. Under the **Company New Record** section, fill in the following field:
    - Name: Alpha Co.
    - eBonded: True
    - eBond account: Alpha Co eBond Account
3. Click **Submit**.
4. Click **New**.
5. Under the **Company New Record** section, fill in the following field:
    - Name: Beta Co.
    - eBonded: True
    - eBond account: Beta Co eBond Account
6. Click **Submit**.

Associate groups with companies

FILL ME IN

1. Navigate to **System Definition** > **Tables** and open the *sys_usr_group* table.
2. Under the **Columns** tab, click **New** to create a new column.
3. Under the **Dictionary Entry New Record** section, fill in the following fields:
    - Type: Reference
    - Column label: Company
4. Under the **Reference Specification** tab, fill in the following fields:
    - Table to reference: core_company
5. Click **Submit**.

OPTIONAL: Modify the view on group records to see the *Company* field.

1. Navigate to **User Administration** > **Groups** and click on any of the group records.
2. Navigate to **Additional actions** > **Configure** > **Form Layout**, which will bring up the **Configuring Group form**.
3. Under the **Available** list, find the *Company* field and add it **>** to the **Selected** list.
4. Move the *Company* field up the **Selected** list **^** to be below the *Name*.
5. Click **Save**.

Demo Data - Assignment Groups

Setting up two support groups; *Data Center Operations Support* and *Backup and Recovery Support*. The reason the group names are not associated with the names of the suppliers is a run and maintain operation decision. Most of the time a supplier will support multiple *Service offerings*, when an organization desires to replace a supplier, instead of updating multiple *Service offerings*, only the support group *Company* field need to be updated. This is a minor detail as many organizations choose to name the assignment groups after the supplier and that is perfectly fine.

1. Navigate to **User Administration** > **Groups**, and click **New**.
2. Under the **Group New Record** section, fill in the following fields:
    - Name: Data Center Operations Support

    - Company: Alpha Co.
3. Click **Submit**.
4. Click **New**.
5. Under the **Group New Record** section, fill in the following fields:
    - Name: Backup and Recovery Support
    - Company: Beta Co.
6. Click **Submit**.

Service, service offering, and assignment group alignment in incidents

Ticket mis-routing is problem in all services organization and surveys show that customer satisfaction reduces the more a ticket is re-routed. To reduce ticket mis-routes, the service, service offering, and assignment group need to be in alignment. It is unreasonable to expect that fulfillers know which support group supports what services. In large enterprises with hundreds of various services, internal teams, and external suppliers ticket mis-routing is common without alignment between all three fields.

> **Optional setup**
>
> This section is optional, but recommended for any ServiceNow environment; eBonded or not. For organizations that employ a multi-supplier strategy, this section is highly recommended.

1. Navigate to **System Definition** > **Dictionary**.
2. Under the list view search, fill in the the following search fields and hit enter:
    - Table: task
    - Column name: business_service
3. Open the *task* record.
4. Click on the **Reference Specification** tab, fill in the following field:
    - Reference qual condition:
        - Class is not Offering
5. Click **Update**.
6. Navigate to **System Definition** > **Business Rules** and click **New**.
7. Under the **Business Rule New Record** section, fill in the following fields:
    - Name: Incident Auto-assignment Group
    - Table: incident
    - Advanced: True
8. Under the **When to run** tab, fill in the following fields:
    - Insert: True
    - Update: True
    - Filter Conditions:
        - Service changes OR
        - Service offering changes
9. Under the **Advanced** tab, fill in the following fields:
    - Condition: gs.isInteractive()
    - Script:

```
(function executeRule(current, previous /*null when async*/ ) {

     if (!gs.nil(current.service_offering.assignment_group))
         current.assignment_group = current.service_offering.assignment_gro
    else
         current.assignment_group = current.business_service.assignment_gro
})(current, previous);
```

10. Click **Submit**.

> **Demo Data - Service and Service Offerings**
>
> Setting up one service and two service offerings for Alpha Co. and Beta Co.
>
> 1. Navigate to **Configuration** > **Services**, and click **New**.
> 2. Under the **Service New Record** section, fill in the following fields:
>   - Name: IT Infrastructure
>   - Service classification: Technical Service
> 3. In the record header, right-click and select **Save**.
> 4. Under the **Offerings** tab, click **New**.
> 5. Under the **Offering New Record** section, fill in the following fields:
>   - Name: Data Center Operations
>   - Support Group: Data Center Operations Support
> 6. Click **Submit**.
> 7. Under the **Offerings** tab, click **New**.
> 8. Under the **Offering New Record** section, fill in the following fields:
>   - Name: Backup & Recovery
>   - Support Group: Backup and Recovery Support
> 9. Click **Submit**.
> 10. Click **Update**.

Associate incidents with eBond companies

Fulfillers will need the ability to eBond incidents with suppliers. And we have partially started the setup that a trigger point will be the *Assignment group* in an incident. Now we will start to add custom fields to the *incident* table to enhance further functionality that will support and aid in multi-source eBonding.

> **Assignment group**
>
> Later on we will configure how changing the *Assignment group* will affect the *eBonded with* field.

There are two main ways to eBond an incident to a supplier; first via the *Assignment group* and the second via a custom field called *eBonded with*. The *eBonded with* will serve a dual purpose. First, it will represent all the suppliers the ticket is eBonded with in a list. A fulfiller will then have the ability to quickly see what

suppliers the ticket is eBonded too. Second, it will toggle eBonding and deBonding operations when suppliers are added or removed from the list. That way a fulfiller can initiate an eBond or deBond with a supplier without having to change the *Assignment group."

> **eBonded with**
>
> This field expects the fulfiller to know which suppliers to eBond with and the service offerings the supplier provides. If the fulfiller does not know the proper supplier, then they should use the *Service* and *Service offering* fields to set the *Assignment group* field to the appropriate group.

> **deBonding**
>
> deBonding is the opposite of eBonding by breaking the connection between the tickets. After a ticket has been deBonded from a company, the ticket can be re-eBonded to the company by adding the company back to the list. In the event of a re-eBonding, there is no guarantee the company ticket will be the same .

1. Navigate to **System Definition** > **Tables** and search for the *incident* table under the **Name** field.
2. Open the *Incident* record and click **New** under the **Columns** tab.
3. Under the **Dictionary Entry New Record** section, fill in the following fields:
    - Type: List
    - Column label: eBonded with
    - Column name: (this should default to u_eBonded_with)
4. Under the **Reference Specification** tab, fill in the following fields:
    - Reference: core_company
    - Reference qual condition:
        - eBonded is true AND
        - eBond Account is not empty
5. Click **Submit**.
6. Navigate to **Incident** > **All** and open any incident record.
7. Navigate to **Additional actions** > **Configure** > **Form Layout**, which will bring up the **Configuring Company form**.
8. Under the **Available** list, find the *eBonded with* field and add it **>** to the **Selected** list.
9. Move the *eBonded with* field up the **Selected** list **^** to be below the *Assigned to*.
10. Click **Save**.

Custom tables

eBonding will require the creation of custom tables or the extension of existing tables. ServiceNow sets a limit on the number of custom tables an instance can have before additional costs are accrued/incurred. This how-to will create custom tables as there are no similar tables in ServiceNow that are proper candidates to extend. You however can take any of the base tables in ServiceNow and extend them adding the custom fields and it will work the same.

eBond relationships

The *u_eBond_relationship* table is a multi-relational table that represents a one to many record mapping for an incident ticket to be mapped to multiple supplier tickets.

The *u_eBond_relationship* table will contain the following custom fields:

Field	Description
Type	The type of eBond that was established with the vendor.
Table sys ID	The ServiceNow record sys_id on the instance.
Table name	The table the ServiceNow record resides on.
Status	The status of the eBond.
State	The state of the eBond.
Correlation Number	The supplier's ticket number or designation
Correlation ID	The supplier's ticket sys id or reference
Company	The ServiceNow supplier company record on the instance.
URL	Link to the supplier ticket.
Reflect	Changes made by the supplier are reflected in the ticket.

The fields relate the ServiceNow ticket to the various suppliers the ticket is eBonded too.

1. Navigate to **System Definition** > **Tables**.
2. Click **New**.
3. Under the **Table New record** section, fill in the following fields:
    - Label: Relationship
    - Name: u_ebond_relationship
    - New menu name: eBond
4. In the record header, right-click and select **Save**.
5. In the **Columns** table, click **New**.
6. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: Choice
    - Column label: Type
    - Column name: (this should default to u_type)
7. In the **Choice List Specification** tab, select *Dropdown with --None--*.
8. In the record header, right-click and select **Save**.
9. In the **Choices** tab, add the following records performing these steps:
    1. Click **New**.
    2. In the **Choice New record** section, fill in the field values listed below.
    3. Click **Submit**.

Sequence	Label	Value
100	Mirror	mirror

Sequence	Label	Value
200	Information	information
300	Transaction	transaction

10. Click *Update**.
11. In the **Columns** table, click **New**.
12. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: Table Name
    - Column label: Source table
    - Column name: (this should default to u_source_table)
13. Click **Submit**.
14. In the **Columns** table, click **New**.
15. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: Document ID
    - Column label: Source
    - Column name: (this should default to u_source)
16. Under **Related Links**, click **Advanced view**.
17. Under the **Dependent Field** tab, click **Use dependent field**.
18. In the **Dependent on field**, select **Source table**.
19. Click **Submit**.
20. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: Choice
    - Column label: Status
    - Column name: (this should default to u_status)
21. In the **Choice List Specification** tab, select *Dropdown with --None--*.
22. In the record header, right-click and select **Save**.
23. In the **Choices** tab, add the following records performing these steps:
    1. Click **New**.
    2. In the **Choice New record** section, fill in the field values listed below.
    3. Click **Submit**.

Sequence	Label	Value
100	Queued	queued
200	Success	success
300	Failure	failure

24. Click *Update**.
25. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: Choice
    - Column label: State
    - Column name: (this should default to u_state)
26. In the **Choice List Specification** tab, select *Dropdown with --None--*.
27. In the record header, right-click and select **Save**.
28. In the **Choices** tab, add the following records performing these steps:
    1. Click **New**.

2. In the **Choice New record** section, fill in the field values listed below.
3. Click **Submit**.

Sequence	Label	Value
100	eBond	ebond
200	eBonded	ebonded
300	deBond	debond
300	deBonded	debonded

29. Click *Update**.
30. In the **Columns** table, click **New**.
31. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: String
    - Column label: Correlation Number
    - Column name: (this should default to u_correlation_number)
    - Max length: 40
32. Click **Submit**.
33. In the **Columns** table, click **New**.
34. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: String
    - Column label: Correlation ID
    - Column name: (this should default to u_correlation_id)
    - Max length: 32
35. Click **Submit**.
36. In the **Columns** table, click **New**.
37. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: Reference
    - Column label: Company
    - Column name: (this should default to u_company)
38. In the **Reference Specification** tab, fill in the following field:
    - Reference: Company
    - Reference qual condition:
        - eBonded is true AND
        - eBond Account is not empty
39. Click **Submit**.
40. In the **Columns** table, click **New**.
41. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: URL
    - Column label: URL
    - Column name: (this should default to u_url)
    - Max length: 4000
42. Click **Submit**.
43. In the **Columns** table, click **New**.
44. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: True/False

    - Column label: Reflect
    - Column name: (this should default to u_reflect)
45. Under the **Default Value** tab, , fill in the following field:
    - Default value: false
46. Click **Submit**.
47. Click **Update**.

Relating eBonded tickets

Associating the supplier tickets within the incident requires the creation of a relationship between the incident and u_ebond_relationship table.

1. Navigate to **System Definition** > **Relationships** and click **New**.
2. In the **Relationship New record** section, fill in the following fields:
    - Name: eBond Relationships
    - Applies to table: Global [global]
    - Queries from table: Relationship [u_ebond_relationship]
    - Query with:

```
current.addQuery('source_table', parent.getTableName());
current.addQuery('source', parent.sys_id);
```

3. Click **Submit**.
4. Navigate to **Incident** > **All** and click on any of the incident records.
5. Navigate to **Additional actions** > **Configure** > **Related Lists**, which will bring up the **Configuring related lists on Incident form**.
6. Under the **Available** list, find the *eBond Relationships* relationship and add it **>** to the **Selected** list.
7. Click **Save**.

---

eBond Properties

The *u_ebond_registry* table is similar to that of the *sys_properties* table in that it stores generic properties and values for eBonding. If you choose to use the *sys_properties* table, then make sure to look through the various scripts that reference the *u_ebond_registry* table in this how-to and change as appropriate.

The *u_ebond_registry* table will contain the following custom fields:

Field	Description
Supplier	"All" or the name of the supplier.
Key	The key variable name.
Value	The property variable value.

1. Navigate to **System Definition** > **Tables**.
2. Click **New**.

3. Under the **Table New record** section, fill in the following fields:
    - Label: Registry
    - Name: u_ebond_registry
    - Add module to menu: eBond
4. In the record header, right-click and select **Save**.
5. In the **Columns** table, click **New**.
6. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: String
    - Column label: Supplier
    - Column name: (this should default to u_supplier)
    - Max length: 40
7. Under the **Default Value** tab, fill in the following field:
    - Default value: All
8. Click **Submit**.
9. In the **Columns** table, click **New**.
10. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: String
    - Column label: Key
    - Column name: (this should default to u_key)
    - Max length: 4000
11. Click **Submit**.
12. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: String
    - Column label: Value
    - Column name: (this should default to u_value)
    - Max length: 4000
13. Click **Submit**.
14. Click **Update**.

---

eBond Logs

The *u_ebond_log* table is similar to that of the *syslog* table in that it stores generic log information. If you choose to use the *syslog* table, then make sure to look through the various scripts that reference the *u_ebond_log* table. The *syslog* table has a granulairty of seconds and ServiceNow performs multiple operations per second. Hence the order of the logs messages become out of order. It is recommended that you use the *u_ebond_log* to help with debugging as the *Index* column is a sequential integer to aid in showing the order of operations.

> **Evaluator messages**

The *u_ebond_log* table does not capture *Evaluator* messages from faulted JavaScript execution. Use the *syslog* table to check for failed JavaScript executions.

The *u_ebond_log* table will contain the following custom fields:

Field	Description
Index	Auto-incrementing numeric value.
Supplier	Name of the supplier.
Class	Name of the ServiceNow component, i.e., script include, business rule, workflow, etc. etc..
Source	Label of the ServiceNow component.
Message	The log message.
Level	Choice value of *High*, *Medium*, *Low*.

1. Navigate to **System Definition** > **Tables**.
2. Click **New**.
3. Under the **Table New record** section, fill in the following fields:
    - Label: Log
    - Name: u_ebond_registry
    - Add module to menu: eBond
4. In the record header, right-click and select **Save**.
5. In the **Columns** table, click **New**.
6. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: Integer
    - Column label: Index
    - Column name: (this should default to u_index)
    - Read only: True
7. Under the **Default Value** tab, fill in the following field:
    - Default value: javascript:getNextObjNumberPadded();
8. Click **Submit**.
9. In the **Columns** table, click **New**.
10. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: String
    - Column label: Supplier
    - Column name: (this should default to u_supplier)
    - Max length: 40
11. Click **Submit**.
12. In the **Columns** table, click **New**.
13. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: String
    - Column label: Class
    - Column name: (this should default to u_class)
    - Max length: 40
14. Click **Submit**.
15. In the **Columns** table, click **New**.

16. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: String
    - Column label: Source
    - Column name: (this should default to u_source)
    - Max length: 80
17. Click **Submit**.
18. In the **Columns** table, click **New**.
19. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: String
    - Column label: Message
    - Column name: (this should default to u_message)
    - Max length: 4000
20. Click **Submit**.
21. In the **Columns** table, click **New**.
22. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: Choice
    - Column label: Level
    - Column name: (this should default to u_level)
23. In the record header, right-click and select **Save**.
24. Under the **Choices** tab, add the following records performing these steps:
    1. Click **New**.
    2. In the **Choice New record** section, fill in the field values listed below.
    3. Click **Submit**.

Sequence	Label	Value
100	High	high
200	Medium	medium
300	Low	low

25. Click **Update**.
26. Click **Update**.

eBond Data Map

The *u_ebond_data_map* table is a translation data table for inbound and outbound communication with suppliers.

The *u_ebond_data_map* table will contain the following custom fields:

Field	Description
Module	*All* or the name of the module.
Classification	Name of the classification.
Direction	*Inbound*, *Outbound*, or *Duplex*.
Supplier	*All* or the name of the supplier.

Field	Description
Source Value	The internal value for the instance.
Supplier Value	The supplier's value.
Note	Freeform to document various notes pertaining to the data map record.

1. Navigate to **System Definition** > **Tables**.
2. Click **New**.
3. Under the **Table New record** section, fill in the following fields:
    - Label: Data Map
    - Name: u_ebond_data_map
    - Add module to menu: eBond
4. In the record header, right-click and select **Save**.
5. In the **Columns** table, click **New**.
6. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: String
    - Column label: Module
    - Column name: (this should default to u_module)
    - Max length: 40
7. Click **Submit**.
8. In the **Columns** table, click **New**.
9. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: String
    - Column label: Classification
    - Column name: (this should default to u_classification)
    - Max length: 40
10. Click **Submit**.
11. In the **Columns** table, click **New**.
12. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: Choice
    - Column label: Direction
    - Column name: (this should default to u_direction)
    - Max length: 40
13. In the record header, right-click and select **Save**.
14. Under the **Choices** tab, add the following records performing these steps:
    1. Click **New**.
    2. In the **Choice New record** section, fill in the field values listed below.
    3. Click **Submit**.

Sequence	Label	Value
100	Inbound	inbound
200	Outbound	outbound
300	Duplex	duplex

15. Click **Update**.

16. In the **Columns** table, click **New**.
17. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: String
    - Column label: Supplier
    - Column name: (this should default to u_supplier)
    - Max length: 40
18. Click **Submit**.
19. In the **Columns** table, click **New**.
20. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: String
    - Column label: Source value
    - Column name: (this should default to u_source_value)
    - Max length: 4000
21. Click **Submit**.
22. In the **Columns** table, click **New**.
23. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: String
    - Column label: Supplier value
    - Column name: (this should default to u_supplier)
    - Max length: 4000
24. Click **Submit**.
25. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: String
    - Column label: Note
    - Column name: (this should default to u_note)
    - Max length: 4000
26. Click **Submit**.
27. Click **Update**.

eBond REST Payloads

The *u_ebond_rest_payloads* table records all outbound REST API calls and their statuses. This table is also used to retry REST calls in the event of a failure; see **eBond Scheduled Job**.

The *u_ebond_rest_payloads* table will contain the following custom fields:

Field	Description
Supplier	The name of the supplier.
Retry count	The number of retries performed executing the RESTful call to the supplier.
Retry	True or False.
REST Message	ServiceNow REST message label.
Payload	REST data payload for the supplier.
Format	The format of the payload; e.g., JSON, XML, SOAP, etc. etc..

Field	Description
Number	The source ticket number.
HTTP Status	HTTP return code.
HTTP Response	Entire HTTP response payload.
HTTP Method	ServiceNow REST method label.
Endpoint	The URL endpoint for the REST call.
Active	True or False

1. Navigate to **System Definition** > **Tables**.
2. Click **New**.
3. Under the **Table New record** section, fill in the following fields:
    - Label: REST Payload
    - Name: u_ebond_rest_payload
    - Add module to menu: eBond
4. In the record header, right-click and select **Save**.
5. In the **Columns** table, click **New**.
6. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: String
    - Column label: Supplier
    - Column name: (this should default to u_supplier)
    - Max length: 40
7. Click **Submit**.
8. In the **Columns** table, click **New**.
9. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: Integer
    - Column label: Retry count
    - Column name: (this should default to u_retry_count)
10. Under the **Default Value** tab, , fill in the following field:
    - Default value: 0
11. Click **Submit**.
12. In the **Columns** table, click **New**.
13. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: True/False
    - Column label: Retry
    - Column name: (this should default to u_retry)
14. Under the **Default Value** tab, , fill in the following field:
    - Default value: true
15. Click **Submit**.
16. In the **Columns** table, click **New**.
17. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: String
    - Column label: REST Message
    - Column name: (this should default to u_rest_message)

    - Max length: 40
18. Click **Submit**.
19. In the **Columns** table, click **New**.
20. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: String
    - Column label: Payload
    - Column name: (this should default to u_payload)
    - Max length: 4000
21. Click **Submit**.
22. In the **Columns** table, click **New**.
23. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: String
    - Column label: Format
    - Column name: (this should default to u_format)
    - Max length: 40
24. Click **Submit**.
25. In the **Columns** table, click **New**.
26. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: String
    - Column label: Number
    - Column name: (this should default to u_number)
    - Max length: 40
27. Click **Submit**.
28. In the **Columns** table, click **New**.
29. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: Integer
    - Column label: HTTP Status
    - Column name: (this should default to u_http_status)
30. Click **Submit**.
31. In the **Columns** table, click **New**.
32. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: String
    - Column label: HTTP Response
    - Column name: (this should default to u_http_response)
    - Max length: 4000
33. Click **Submit**.
34. In the **Columns** table, click **New**.
35. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: String
    - Column label: HTTP Method
    - Column name: (this should default to u_http_method)
    - Max length: 40
36. Click **Submit**.
37. In the **Columns** table, click **New**.
38. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: String

- Column label: Endpoint
- Column name: (this should default to u_endpoint)
- Max length: 80

39. Click **Submit**.
40. In the **Columns** table, click **New**.
41. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: True/False
    - Column label: Active
    - Column name: (this should default to u_active)
42. Under **Related Links**, click **Advanced view**.
43. Under the **Calculated Value** tab, fill in the following fields:
    - Calculated: true
    - Calculation:

```
(function calculatedFieldValue(current) {
    var calc = current.u_retry;

    switch (current.u_http_status.toString()) {
        //1×× Informational
        case '100': // Continue
        case '101': // Switching Protocols
        case '102': // Processing
        case '200': // OK
        case '201': // Created
        case '202': // Accepted
        case '203': // Non-authoritative Information
        case '204': // No Content
        case '205': // Reset Content
        case '206': // Partial Content
        case '207': // Multi-Status
        case '208': // Already Reported
        case '226': // IM Used
        case '300': // Multiple Choices
        case '301': // Moved Permanently
        case '302': // Found
        case '303': // See Other
        case '304': // Not Modified
        case '305': // Use Proxy
        case '307': // Temporary Redirect
        case '308': // Permanent Redirect
        case '400': // Bad Request
            calc = false;
            break;
        case '401': // Unauthorized
        case '402': // Payment Required
        case '403': // Forbidden
        case '404': // Not Found
        case '405': // Method Not Allowed
        case '406': // Not Acceptable
        case '407': // Proxy Authentication Required
        case '408': // Request Timeout
        case '409': // Conflict
        case '410': // Gone
```

```
        case '411': // Length Required
        case '412': // Precondition Failed
        case '413': // Payload Too Large
        case '414': // Request-URI Too Long
        case '415': // Unsupported Media Type
        case '416': // Requested Range Not Satisfiable
        case '417': // Expectation Failed
        case '418': // I'm a teapot
        case '421': // Misdirected Request
        case '422': // Unprocessable Entity
        case '423': // Locked
        case '424': // Failed Dependency
        case '426': // Upgrade Required
        case '428': // Precondition Required
        case '429': // Too Many Requests
        case '431': // Request Header Fields Too Large
        case '444': // Connection Closed Without Response
        case '451': // Unavailable For Legal Reasons
        case '499': // Client Closed Request
        case '500': // Internal Server Error
        case '501': // Not Implemented
        case '502': // Bad Gateway
        case '503': // Service Unavailable
        case '504': // Gateway Timeout
            break;
        case '505': // HTTP Version Not Supported
        case '506': // Variant Also Negotiates
        case '507': // Insufficient Storage
        case '508': // Loop Detected
        case '510': // Not Extended
        case '511': // Network Authentication Required
        case '599': // Network Connect Timeout Error
            calc = false;
            break;
        default:
            break;
    }

    // after 48 hours of retrying, turn off active flag
    if (current.u_retry_count >= 48) {
        calc = false;
    }

    return calc; // return the calculated value
})(current);
```

44. Click **Submit**.
45. Click **Update**.

Multi-source Inbound Handling

Inbound incident handling could be setup in an agnostic fashion based on a mutual push-push model setup between the enterprise and suppliers. The mutual push-push model is where the each party sets how they

shall receive messages and no party performs a pull/get operation. There are major advantages for this model over one party dictating both how it will send and receive messages.

> **OEM & OSP**
>
> Original equipment manufacturers (OEM) and original service providers (OSP) rarely fit into the mutual push-push model as their clientele is vast and the reliance of the enterprise on the OEMs and OSPs do not warrant a collaborative effort between the enterprise ServiceNow developers and the OEM/OSP. In the case of true partnership between enterprise and suppliers should allow for a mutual push-push model.

The biggest advantage to all parties is long term run and maintain of eBonding operations. If one party changes their fields or data, the other party is not forced into changing their eBonding framework to accommodate; a decision from one party should not impact or cause work for the other. Another advantage is resource utilization. Pull requests are expensive and if an enterprise is being pulled by dozens of suppliers, this places an undue burden on the enterprise. It is far more agreeable that if a change is to occur in one party's record, then that party will pass (push) the delta change to the agreed parties.

Quick Rundown

Suppliers will write to a staging table called *[u_ebond_incident_staging]*. The table will be an extension of the *[sys_import_set_row]* and will have the ability to take advantage of the OOTB ServiceNow transform functionality. The *transform maps* will validate the incoming data and then create or update the appropriate records within the system.

Incident staging table

Normally the REST inbound message handling aspect for eBonding is developed using *Scripted REST API*. For multi-source eBonding a different model will be developed using a staging table. Suppliers will directly write to a staging table instead of a Scripted REST API.

ServiceNow *Transform Maps* provide an elegant and efficient means to do much of what a *Scripted REST API* can do with the use of *Transform Scripts*. The **onStart** is used for preparation of global variables, *onBefore* is used for data validation, and *onAfter* for data operations. All of this efficiently designed by Servicenow in the import set framework to handle inbound REST messages.

> **Staging other tables**
>
> For each type of destination ServiceNow table for multi-source eBonding will require a unique staging table. For example, if the requirement is to have a multi-source eBond with change requests, a similar staging table that that below will need to be created.

The *u_ebond_incident_staging* table holds the field values passed by the supplier for incidents.

The *u_ebond_incident_staging* table will contain the following custom fields:

Field	Description
Work Note	Support work note, customer not visible.
Subcategory	Subcategory for the incident.
State	The state of the incident.
Short Description	The title (or short description) of the incident.
Service Offering	The service offering the incident is impacting.
Service	The service the incident is impacting.
Priority	The priority level of the incident.
Sys ID	The sys_id for the local incident ticket.
Number	The number for the local incident ticket.
Hold Reason	The reason the incident ticket was placed on hold.
Description	The description of the incident.
Contact Type	How the incident was reported.
Comment	Customer visible comment.
External Reference	The supplier's incident ticket unique identifier.
External Number	The supplier's incident ticket number.
CMDB CI	The configuration item the incident is reported against.
Close Notes	Incident closure notes.
Close Code	Incident closure code.
Category	Category for the incident
Caller	The caller who reported the incident.
Assignment Group	The assignment group working the incident.

1. Navigate to **System Definition** > **Tables**.
2. Click **New**.
3. Under the **Table New record** section, fill in the following fields:
    - Label: Incident Staging
    - Name: u_ebond_incident_staging
    - Extends table: Import Set Row [sys_import_set_row]
    - Add module to menu: eBond
4. In the record header, right-click and select **Save**.
5. In the **Columns** table, click **New**.
6. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: String
    - Column label: Work Note
    - Column name: (this should default to u_work_note)

    - Max length: 4000
7. Click **Submit**.
8. In the **Columns** table, click **New**.
9. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: String
    - Column label: Subcategory
    - Column name: (this should default to u_subcategory)
    - Max length: 80
10. Click **Submit**.
11. In the **Columns** table, click **New**.
12. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: String
    - Column label: State
    - Column name: (this should default to u_state)
    - Max length: 40
13. Click **Submit**.
14. In the **Columns** table, click **New**.
15. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: String
    - Column label: Short Description
    - Column name: (this should default to u_short_description)
    - Max length: 1000
16. Click **Submit**.
17. In the **Columns** table, click **New**.
18. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: String
    - Column label: Service Offering
    - Column name: (this should default to u_service_offering)
    - Max length: 255
19. Click **Submit**.
20. In the **Columns** table, click **New**.
21. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: String
    - Column label: Service
    - Column name: (this should default to u_service)
    - Max length: 255
22. Click **Submit**.
23. In the **Columns** table, click **New**.
24. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: String
    - Column label: Priority
    - Column name: (this should default to u_priority)
    - Max length: 40
25. Click **Submit**.
26. In the **Columns** table, click **New**.
27. In the **Dictionary Entry New record** section, fill in the following fields:

    - Type: String
    - Column label: Sys ID
    - Column name: (this should default to u_sys_id)
    - Max length: 40
28. Click **Submit**.
29. In the **Columns** table, click **New**.
30. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: String
    - Column label: Number
    - Column name: (this should default to u_number)
    - Max length: 40
31. Click **Submit**.
32. In the **Columns** table, click **New**.
33. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: String
    - Column label: Hold Reason
    - Column name: (this should default to u_hold_reason)
    - Max length: 40
34. Click **Submit**.
35. In the **Columns** table, click **New**.
36. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: String
    - Column label: Description
    - Column name: (this should default to u_description)
    - Max length: 4000
37. Click **Submit**.
38. In the **Columns** table, click **New**.
39. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: String
    - Column label: Contact Type
    - Column name: (this should default to u_contact_type)
    - Max length: 40
40. Click **Submit**.
41. In the **Columns** table, click **New**.
42. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: String
    - Column label: Comment
    - Column name: (this should default to u_comment)
    - Max length: 4000
43. Click **Submit**.
44. In the **Columns** table, click **New**.
45. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: String
    - Column label: External Reference
    - Column name: (this should default to u_external_reference)
    - Max length: 40

46. Click **Submit**.
47. In the **Columns** table, click **New**.
48. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: String
    - Column label: External Number
    - Column name: (this should default to u_external_number)
    - Max length: 40
49. Click **Submit**.
50. In the **Columns** table, click **New**.
51. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: String
    - Column label: CMDB CI
    - Column name: (this should default to u_cmdb_ci)
    - Max length: 40
52. Click **Submit**.
53. In the **Columns** table, click **New**.
54. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: String
    - Column label: Close Notes
    - Column name: (this should default to u_close_notes)
    - Max length: 1000
55. Click **Submit**.
56. In the **Columns** table, click **New**.
57. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: String
    - Column label: Close Code
    - Column name: (this should default to u_close_code)
    - Max length: 40
58. Click **Submit**.
59. In the **Columns** table, click **New**.
60. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: String
    - Column label: Category
    - Column name: (this should default to u_category)
    - Max length: 80
61. Click **Submit**.
62. In the **Columns** table, click **New**.
63. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: String
    - Column label: Caller
    - Column name: (this should default to u_caller)
    - Max length: 40
64. Click **Submit**.
65. In the **Columns** table, click **New**.
66. In the **Dictionary Entry New record** section, fill in the following fields:
    - Type: String

    - Column label: Assignment Group
    - Column name: (this should default to u_assignment_group)
    - Max length: 40
67. Click **Submit**.
68. Click **Update**.

Inbound Transform

The inbound *transform map* uses *transform scripts* to perform various operations on the data directly passed by the supplier. The *transform scripts* can be set to *when* they are executed against the data in the transform.

- **onStart:** Sets up the global variables used by the other transform scripts. Global variables will be capitalized for ease of development.
- **onBefore:** Validates security and the data passed by the supplier. The order of scripts are important as pre-checks must complete before continuing the transform.
- **on:After** Creates or updates the needed fields within the records.
- **on:Complete** Verifies all operations and updates return information to the supplier.

1. Navigate to **System Import Sets** > **Transform Maps**.
2. Click **New**.
3. In the **Table Transform Map New record** section, fill in the following fields:
    - Name: eBond Incident Transform
    - Source table: Incident Staging [u_ebond_incident_staging]
    - Target table: Relationship [u_ebond_relationship]
4. In the record header, right-click and select **Save**.
5. In the **Field Maps** tab, click **New**.
6. In the **Field Map New record** section, fill in the following fields:
    - Source field: External Number [u_external_number]
    - Target field: Correlation Number [u_correlation_number]
    - Coalesce: True
7. Click **Submit**.

> **Coalesce field not indexed**
>
> If a pop-up box appears titled *Coalesce field not indexed*, then click **OK**, fill in the followup contact information and click **OK**.

8. In the **Field Maps** tab, click **New**.
9. In the **Field Map New record** section, fill in the following fields:
    - Source field: External reference [u_external_reference]
    - Target field: Correlation ID [u_correlation_id]
    - Coalesce: false

10. Click **Submit**.
11. In the **Field Maps** tab, click **New**.
12. In the **Field Map New record** section, click **Use source script** and fill in the following fields:
    - Choice action: ignore
    - Target field: company [core_Company]
    - Coalesce field: true
    - Script:

```
answer = (function transformEntry(source) {

    // coalesce script is called at various times in the transform
    // ignore the time(s) when the row is not even being evaluated
    if (source.sys_created_by == "") {
        return;
    }

    // find the company the account is associated with
    var company = new GlideRecord('core_company');
    company.addQuery('u_ebond_account', source.sys_created_by);
    company.query();
    if (company.next()) {
        // make sure the supplier is still eBond approved
        if (company.u_ebonded) {
            return company;
        }

        // supplier is not eBond enabled
        var log = new GlideRecord('u_ebond_log');
        log.u_class = 'eBond Incident Transform map';
        log.u_source = '[Field Map] Company';
        log.u_supplier = company.name;
        log.u_message = 'Security Violation: The supplier ' + company.name
        log.u_level = 'Low';
        log.insert();

        error = true;
        error_message = "Security violation.";
        return "";
    }

    // creator is not associated with a supplier
    var log = new GlideRecord('u_ebond_log');
    log.u_class = 'eBond Incident Transform map';
    log.u_source = '[Field Map] Company';
    log.u_supplier = 'Unknown';
    log.u_message = 'Security Violation: An eBond account (' + source.sys_c
    log.u_level = 'High';
    log.insert();

    error = true;
    error_message = "Security violation.";
```

```
    return ""; // return the value to be put into the target field

})(source);
```

13. Click **Submit**.

> **Coalesce fields**
>
> Only the *target field* fields *u_correlation_number* and *u_company* fields are coalesce fields. The *u_company* is derived internally from the *sys_created_by* field that holds the account that wrote the record. This removes the means to supplant company information. The *u_correlation_number* by itself is not sufficient to make it unique enough with multiple eBonded suppliers. The combination of the *u_correlation_number* and *u_company* is required; as the assumption is the supplier will not duplicate their ticket numbers. However, two or more suppliers could have the same ticket numbers unrelated to each other.

14. In the **Transform Scripts** tab, click **New**.
15. In the **Transform Script New record** section, fill in the following fields:
    - When: onStart
    - Script:

```
// Initialize global variables
//
var LOGING = false;
var COMPONENT = "Transform Map";
var SOURCE = "eBond Incident Transform";
```

16. Click **Submit**.

Multi-source Outbound Handling