EDA Readiness - Self Certification - Powered by EDA

What does it mean to be "Powered by EDA"?
A "Powered by EDA" AppExchange app meets the requirements as ***self-certified*** by the ISV partner in that the app :

- Is "EDA Ready"
- Builds upon EDA's data model
- Supports Table Driven Trigger Management (TDTM)

Partners with apps that have successfully gone through the "Powered by EDA" self certification process can use this designation in their app's marketing materials. The EDA Compatibility Self Assessment with the relative questionnaire and the EDA Ready Self Certification are the stepping stone for this next level in the Self Certification process. The guidelines contained in the EDA Ready Self Certification are a requirement to apply for "Powered by EDA" status. Refer to the EDA Ready self certification guidelines document for details about "EDA Ready" .

What do we mean by "the application builds upon EDA's data model"?
When you install an ISV application, it installs additional objects into the org. These objects should not compete with EDA's data model. If an ISV application has a duplicate object (an object with the same name or purpose) to an EDA object, the ISV partner needs to use the EDA object and provide any data mapping or data syncing to those EDA objects. This is what we mean by "builds upon" the EDA data model.

The ISV Application **must** store data in the EDA Data model. All data that can be stored in the data model must be stored in the EDA data model. To be "Powered By EDA" the ISV application stores data in the EDA objects and leverages all applicable parts of the EDA Data Model.

SCENARIOS

The following scenario describes the above:

Scenario 1 : ISV application has a duplicate object (an object with the same name or purpose)
To find out if an ISV application has a duplicate object the partner checks the Object Name and also the Object Description in the managed package.

EDA Object used natively
The ISV application has a Relationship object in their managed package. The ISV application has deprecated their Relationship object in favor of EDA's Relationship object. To continue supporting customers that still use the deprecated object, the ISV application can clearly mark their object as deprecated in its description.The ISV partner can remove the object from their data model providing instructions to customers on how to permanently delete the object from their org as described in here.

ISV Object duplication
The ISV application has an Address object in their managed package, whereas EDA has its own Address object.
The ISV Address object saves addresses entered for purchases in their online registration system and e-commerce store. These addresses must be copied into the EDA address object as required. The ISV application respects the primary Address of the Contact set by EDA.

Scenario 2 : ISV application has no duplicate objects.
In this scenario, there is no overlap between the ISV application's data model and EDA's data model.

REFERENCES

The same reference documentation for the Compatibility Self Assessment in the Self Certification for "EDA Ready" applies for this requirement.

Here we list the most relevant pieces of documentation:

- EDA Account Model
- New Contact creation
- Account record types
- Person Accounts
- Affiliations
- Relationships
- Addresses
- EDA Data Dictionary
- An example of how to document duplicate objects (from the GEM - Gift Entry Management product documentation from Salesforce.org)

ISV Application Supports TDTM

What do we mean when we say "Supports Table Driven Trigger Management (TDTM)"?
Table Driven Trigger Management (TDTM) framework has an important job to play in EDA. It allows for the automation to happen behind the scenes when a user interacts with EDA. EDA applies Inversion of Control (IoC) through the TDTM design, where the name and other information on classes to be run in response to user actions on records are stored in a table/object as documented here.

There are 3 different ways (as described in the following scenarios) where an ISV application can support TDTM.

Scenario 1: The ISV application leverages the Table Data Trigger Management framework.

An ISV application leverages TDTM in that it lets TDTM manage the following aspects of the automation in an org.

- Execution of Trigger Handlers included in their packages.
    - By defining specific pieces of code for the automation of the following EDA objects:
        - Account
        - Address
        - Affiliation
        - Attendance Event
        - Campaign
        - Campaign Member
        - Contact
        - Contact Language
        - Course
        - Course Enrollment
        - Course Offering
        - Course Offering Schedule
        - Plan Requirement

    - Program Enrollment
    - Program Plan
    - Relationship
    - Term
    - Time Block
    - Trigger Handler
- Order of execution of the code contained in the Trigger Handlers.

Scenario 2: The ISV has developed their own trigger management mechanism and is aware of best practices for implementing/integrating a similar framework into EDA.

- The custom code in the ISV application works in conjunction with the above EDA objects.
- An ISV application can deploy their own Apex Triggers on their own objects and manage automation for them separately.

Scenario 3: The ISV application does not include any automation via triggers or any other tools that conflict with the correct functioning of EDA TDTM.
This applies to ISV applications that only integrate via the API or have no code for automation.

How do we know the ISV application supports TDTM?
The ISV application either utilizes TDTM and in that case:

- only one trigger exists per object;
- the TDTM framework will instantiate the necessary classes and run them according to the parameters specified in the TDTM table;
- the customer can manage automation and order of processing of triggers via the same tools provided by EDA;

or does not use TDTM but...

- it uses a similar mechanism for Inversion of Control;
- it provides documentation on how to implement it alongside with TDTM;
- it provides tools or documentation for enabling/disabling automation on EDA;
- it provides tools or documentation for enabling/disabling automation on their package.

The ISV application are advised to disable automation on EDA when importing large sets of data or integrating with external systems. These cases must be justified and documented.

REFERENCES

"Understanding TDTM in EDA" by Kevin Swiggum is a great resource. Check it out here.

More resources to learn about TDTM and how to integrate with it are the following:

- EDA TDTM overview
- EDA Deploy Custom Apex
- TDTM Framework Overview (NPSP related): Technical blog post by Carlos Ramirez Martinez-Eiroa
- How ISVs can use the TDTM framework and create Trigger Handlers in their packages
- Apex Class Descriptions