Cast to Entity

Introducing a new operator **Cast to Entity** which allows you to create an Entity from the JSON or make your flowchart even more readable by putting Entity to the Variables and then reading it with **Cast to Entity**

	Cast To Entity		
Select module from the list	O module	entity <mark>O</mark>	As a result. there is an Entity
Put Entity or JSON	▼ ○ value		that can be used in Get Entity Attributes, Save Entity, etc.

Example of usage:

1. Creating Entity from the JSON

For example, there is a JSON that describes an object in a similar way as Sugar Entity structure. Instead of operators **Create Entity** and **Set Entity Attributes** following with mapping fields from JSON to Sugar Entity, there is a possibility to use **Cast To Entity**. Let's say there is a JSON like this:

```
{
  "first_name": "Clark",
  "last_name": "Kent"
}
```

With the following flowchart, it is possible to create a new Contact record from this JSON:



As a result, new Contact is added:

Co Clark Kent	☆ Following	<	>	Edit	~
✓ BUSINESS CARD					
(+Title)	+ Mobile				
+ Department	Do Not Call				
+ Account Na	+ Business Ce)				
+ Email Addre					
+ Tags					

2. Get Entity from the Variable

If you have dozens of links from *entity* port:



... and have ever thought of how to remove them to make flowchart even more readable – now it is possible to put Entity in a Variable and then simply use this Variable through Cast To Entity. Put entity into variable:

			Branc	h i		
			o con	true ► dition false ⊳	Set Variable	
On Data Change		i			o name	
O module		-			my_case	
Cases	× •	entity 🔴 🗕			o value	
O event type		previous values O				
After Save	× 🔻	is new 🔵 🗝				
		changed fields 🔲				

And then through Get Variable and Cast To Entity use the entity whenever it is needed:

	Cast To Entity			Get Entity Attributes			i
	O module		entity 🔵 🗕 🗕	entity	Type (type)	× •	•
	Cases	× •			Assigned To: (assigned	i × 🔻	
Get Variable	value					+	_
name value my_case							

New Operators to work with Email Addresses

Out-of-the-box field Email Address is a special type of the field, which allows setting several Email Addresses for one record and marking Email Address as Primary, Invalid, or Opt Out. New Logic Builder Release enables you to work with values in this field - to add new email addresses, remove them, update email addresses attributes, and get all Email Addresses related to the Entity:

Email Addresses
Add Email Address
Get Email Addresses
Remove Email Address

1. Add Email Address

Operator Add Email Address allows adding new Email Address to a Record like Lead, Contact, Account, Target (out-of-the-box entities with Email Address type field or Custom Entities with email type field).

	Add Email Address	
Put Entity (Lead, Contact, Account, etc) to which a new Email Address must be added	▷ ▷	
***************************************	entity entity O	
Specify Email Address to add (e.g.	email address	As a result, there is an Entity with the new Email Address
test@test.example.com)		
	primary	
Set True if Email Address	• reply to	
attributes Primary, Reply		
To, Invalid, and Opt Out→		
must be marked in the Entity	O opt out	

Example of usage:

Let's say that there are Contact details retrieved via REST and you would like to update email address based on the response. The flowchart which implements this may be like this:



In the example above, Email Attributes primary/reply to/invalid/opt out will be set to *false*. In case there is a need to set address as opted out just put **Boolean** true in the input port:



2. Remove Email Address

Operator **Remove Email Address** allows removing Email Address from a Record like Lead, Contact, Account, Target (out-of-the-box entities with Email Address type field or Custom Entities with email type field).



Example of usage:

Let's say that you would like to remove the email address and set it to the value that was received from the 3rd party system. The flowchart that implements email address change may be like this:



3. Get Email Addresses

Operator Get Email Addresses allows you to get a primary email address and an array of emails arrays (each of which consists of email address and all its attributes) related to the Entity.

Put Entity (Lead, Contact,	Get Email Addresses	
Account, etc) from which Email Addresses need retreiving		Array of Email Addresses Objects
	🔵 entity email addresses 🗖	
	primary email address 🔵	
		Primary Email Address (e.g.
		test@test.example.com)

Output Example:

Let's say there is a Lead Record with the following Email Addresses:

Email Address	anastasiia.gubanovych@integroscrm.com	★ ◎ ●
		-
	lbsupport@integroscrm.com	* 0 0
	issupport@integroserni.com	_
		* 0 0
	nastya@integroscrm.com	_

If you use Get Email Addresses from this Lead:



Output port Email addresses contains the following data:

```
Array
```

```
(
    [0] \implies Array
        (
            [email address] => anastasiia.gubanovych@integroscrm.com
            [email address caps] => ANASTASIIA.GUBANOVYCH@INTEGROSCRM.COM
            [invalid email] => 0
            [opt out] => 0
            [date created] => 2021-10-27 11:32:34
            [date modified] => 2021-10-27 11:32:34
            [id] => 945e040c-3719-11ec-b821-000c29791bd4
            [email address id] => 94316e1a-3719-11ec-8e57-000c29791bd4
            [bean id] => 945da0a2-3719-11ec-97aa-000c29791bd4
            [bean module] => Leads
            [primary address] => 1
            [reply_to_address] => 0
            [deleted] => 0
       )
    [1] => Array
        (
            [email address] => lbsupport@integroscrm.com
            [email address caps] => LBSUPPORT@INTEGROSCRM.COM
            [invalid email] => 0
            [opt out] => 1
            [date created] => 2021-10-27 11:32:34
            [date modified] => 2021-10-27 11:32:34
            [id] => 945e4534-3719-11ec-bd31-000c29791bd4
            [email address id] => 9431a448-3719-11ec-94a3-000c29791bd4
            [bean id] => 945da0a2-3719-11ec-97aa-000c29791bd4
            [bean module] => Leads
            [primary address] => 0
            [reply_to_address] => 0
            [deleted] => 0
       )
    [2] => Array
        (
            [email address] => nastya@integroscrm.com
            [email address caps] => NASTYA@INTEGROSCRM.COM
            [invalid email] => 1
            [opt out] => 0
            [date created] => 2021-10-27 11:32:34
            [date modified] => 2021-10-27 11:32:34
```

```
[id] => 945e6dfc-3719-11ec-b53f-000c29791bd4
[email_address_id] => 9431c536-3719-11ec-ab68-000c29791bd4
[bean_id] => 945da0a2-3719-11ec-97aa-000c29791bd4
[bean_module] => Leads
[primary_address] => 0
[reply_to_address] => 0
[deleted] => 0
)
```

Output port primary email address contains only one primary address without attributes:

anastasiia.gubanovych@integroscrm.com

Following Processing of Outputs

)

There are two options how to process results from email addresses output:

Option 1: Use For Each Element to get each Email Address and then Get Array Element or Get Array Elements by Path to get needed Email Address Attribute:



Option 2: Use For Each Element to get each Email Address and then newly added Cast To Entity to treat Email Address as usual Entity:



Change Primary Email Address

Scenario 1: Set Email Address for the new record

For the new record, you may use a field email1 to set a primary email address:



Or new operator Add Email:



Scenario 2: changing Email address if there is an email address

To completely override a Primary Email Address, first remove the current Primary Email Address and then set the new one:



N.B. If there is more than one email address for the Record, removing only the Primary address as in the example above may end up with two primary addresses for the record.

To save all email addresses and change only the primary email address (along with preserving the old primary email address), it's needed to remove all addresses, add a new one, and then return all removed emails, like this:



Other Changes

- Removed deprecated Operators (Flex Docs, User Actions, Behavior)
- Removed non-numeric symbols from packages version number