

(presentation?) showing how to process incoming events within campaigns or ERP5 CRM in general.

`\${table_of_content}`

How to Process Incoming Events



by [OSOE Project](#).

As introduced in the previous tutorial "Events", incoming Events are created either from **external or internal** of your company. When your company receives an email, a fax or a phone call from **external**, you can create a new incoming Event by **declaring the incoming message as received**. When it is necessary to exchange between departments of your company an outgoing Event created by **internal** of your company which has been sent, eg, to assign it to a Support Request Ticket, you can also **declare the outgoing message as received**, then follow the Incoming Event Workflow to handle it.

In this tutorial, we will explain in detail the effects of the **Declare as Received**, **Define the Event's Recipient**, **Create a Follow Up Ticket** and **Deliver** actions.

In next tutorial, we will use the Support Requests-a kind of Ticket created from incoming Events, as an example to explain precisely to you how to apply this standard process to ERP5 CRM.

Agenda

- Standard process of incoming Events

The Event workflow in ERP5



Here is the standard Event workflow of ERP5. As introduced in the previous tutorial "Events", the incoming Events (second part of this illustration) can either be created **manually** (eg, you can **input** an email which is a support request or a reply from a customer to ERP5 team as a new incoming Event), or **automatically** (eg, between 2 departments who use the same ERP5 system, an Event can be exchanged by calling the Action "**Declare as Received**". If an Event is initially outgoing for department A-the sender, it becomes incoming for department B-the recipient, once A has sent the Event and declared it as received).

The standard process of how to process incoming Event:

1. Firstly, we will **Declare as Received** an incoming Event from external of our company, or an outgoing Event sent by our company but needs to be exchanged between different departments of our company.
2. Secondly, we have to **Define the Recipient** in order to indicate who in our team will be responsible to handle the Event.
3. Thirdly, if the Event needs to be handled in a serious way, eg, an mail message from a client concerning a product problem, we will need to **Create Follow Up Ticket** in order to record all the interactions related to the original Event which will happen afterwards between the Event operators in our team and the Event sender. We also have to **Define the Ticket's Operation Manager, the Supervisor and the Operators** to indicate who will be responsible for this Ticket.
4. Finally, the person who is defined as the "Recipient" of the Event will see it in his worklist, and **Deliver** the Event to acknowledge that he is responsible for this incoming Event and he will handle it later.

So the final state of incoming Events is "Delivered", which means they are acknowledged by somebody in our team to be taken care of. Then according to the complexity of the incoming Events, they will handle them either by **Create Response directly** from the **Delivered Event** (if the Event can be handled in a simple way), or by **Process the Follow Up Ticket**.

Note that during the process, we can **Change the Recipient** of this Event to assign another team member to handle it.

`\${related_subject_list}`