

---

### **Liaison Statement**

**To:** TSG-S1, TSG-S3

**Source:** TSG-T3 chairman

*(Note this LS is sourced from the T3 chairman because, due to a shortage of time at the end of the last T3 meeting to fully discuss the document, the LS was not able to be approved.)*

**Title:** Elaboration of KEY IDENTIFICATION EVENT

**Contact:** Yael Baruch, (yael.baruch@celltick.com)

This liaison to inform S1 and S3 about a new release-4 feature discussed at the last T3 meeting regarding a new USAT "event" KEY IDENTIFICATION.

The goal and background of this feature and some questions raised at the recent T3 #17 meeting in Berlin, Germany are described in more detail in document T3-010093 (attached).

This feature introduces a new "event" that indicates that a key on the MMI has been pressed and includes the key identification, in accordance with the "Get Inkey" command.

This proposal puts the onus of detecting a key being pressed on the mobile, whilst "freeing up" the USAT to perform other activities. Currently, for example, if the USAT wishes to detect a key being pressed, it has to issue the Get Inkey command to the ME and wait for the key to be pressed, and thus cannot carry on with other activities, such as updating or scrolling the display.

As set forth in the T3 documents this enhancement "KEY IDENTIFICATION" opens new user interaction and application possibilities. It adds a useful feature for future advanced services such as new user-friendly, interactive and fast mobile applications

requiring a minimal number of keystrokes. Some of these applications require continuous actions, even if the user does not respond to them.

If S1 agree that this is a useful new feature which should be standardised, T3 believe that the USAT Stage 1 document may need to be updated to reflect this.

T3 would like to point out that concerns were raised regarding user awareness of the activation of this feature, giving the USAT the possibility to intercept key strokes in a standardised manner. T3 seek the opinion of S1 and S3 on this issue, as it raises a potential security threat.

One way of reducing the potential security threat is to offer the user the possibility of disabling this feature.

**Background information for T3-010093**  
**(CR 31.111: A new event Key Identification)**

**Source: Celltick Technologies**

---

The proposed CR is the document T3-010066 and it is related to 3GPP 31.111. It adds the necessary knowledge for future advanced services - the identification of the pushed key by event.

Please note that in the following: when I mention “application” I mean “USAT application.”

**The kinds of services we are talking about**

The services we related to in our CR are services with new user-friendly, interactive and fast (seconds) mobile applications requiring a minimal number of keystrokes.

Some of these applications require continuous actions, even if the user does not respond to them. For example with games: i.e. “Who wants to be a millionaire” where the user is choosing the correct answer from a running set of answers, or in addition - rolling screens, waiting for the user to choose one of them. The application can react differently according to what the user does or does not do, and acts according to different internal time outs.

The services needed to have semi real time (ASAP) and online services, from the point of view of the user (semi “real time” from the user’s point of view is just a few seconds).

When the service is active, it is important to have the possibility to change the text on the screen while waiting for the user to respond.

**Why Get Inkey and Get Input are not enough nor suitable for these services**

The new event differs from the Get Inkey and Get input commands.

The Get Inkey and the Get Inputs commands are used by an application’s menus and need user response or a Timeout to continue to its next step.

On the other hand, applications which use the suggested event can react and show Display Text/ Image, Play Tone, or any other stage in the application, without the need for user response.

Therefore, only those applications which use an event can implement the games we talked about before (i.e. “Who wants to be a millionaire”).

### **Why we chose not used the User Activity event**

The User Activity event is normally used during idle mode, we did not want to mix the two stages (idle and runtime).

### **The necessity of getting all the requested pushed keys**

It might seem that there is no sense to send all requested pushed keys to the SIM. Moreover it could cause a lot of traffic on the interface.

This first impression is not precise. The event will be set only when the service starts working; not during the whole time while the user uses his/her phone. Therefore we assume that each received pushed key is important to the application, otherwise the application will not set the event again (the application needs to set the event after each received key).

If the user is occupied with other actions not related to the service (i.e. dialing a number), it is logical that the service application stops running the service at that moment. Then the service application will not ask for the pushed key identification (it is not necessary, but maybe there is an exception, and the first digit of the dialed number may be received).

### **User confidentiality**

One of the concerns of using event with key identification is the user confidentiality problem. In the case that the application gets the key identification by event, it could perform illegal actions on the received user information. Moreover it could do it without the user's knowledge and thus cause a user confidentiality problem.

We agreed that if the operator or the application providers would like to perform illegal actions, they can do it, with or without the new proposed event. This is true no matter where the service is implemented (Terminal and/or SIM), and it can occur as well in existing services today.

We are going to eliminate the User Confidentiality problem by notifying the user about the usage of this kind of event, or other solutions if will be founds (T3 is still working on that).

### **Conclusions**

There is a need to create a USIM/SIM that can work with new advanced services. As we all know, the advanced services are one of the most important features for the future. Therefore, the main point is that the T3 group should change the standard to insure that the terminal and the USIM could control advanced and future services and we should find the way to do this.

However, the Get Inkey, the Get Input, and the User Activity commands cannot be the solution.