

# Event Monitoring

---

© Copyright 2008 Infor

All rights reserved. The word and design marks set forth herein are trademarks and/or registered trademarks of Infor and/or its affiliates and subsidiaries. All rights reserved. All other trademarks listed herein are the property of their respective owners.

### Important Notices

The material contained in this publication (including any supplementary information) constitutes and contains confidential and proprietary information of Infor.

By gaining access to the attached, you acknowledge and agree that the material (including any modification, translation or adaptation of the material) and all copyright, trade secrets and all other right, title and interest therein, are the sole property of Infor and that you shall not gain right, title or interest in the material (including any modification, translation or adaptation of the material) by virtue of your review thereof other than the non-exclusive right to use the material solely in connection with and the furtherance of your license and use of software made available to your company from Infor pursuant to a separate agreement ("Purpose").

In addition, by accessing the enclosed material, you acknowledge and agree that you are required to maintain such material in strict confidence and that your use of such material is limited to the Purpose described above.

Although Infor has taken due care to ensure that the material included in this publication is accurate and complete, Infor cannot warrant that the information contained in this publication is complete, does not contain typographical or other errors, or will meet your specific requirements. As such, Infor does not assume and hereby disclaims all liability, consequential or otherwise, for any loss or damage to any person or entity which is caused by or relates to errors or omissions in this publication (including any supplementary information), whether such errors or omissions result from negligence, accident or any other cause.

### Trademark Acknowledgements

All other company, product, trade or service names referenced may be registered trademarks or trademarks of their respective owners.

### Publication Information

---

<b>Document code</b>	UTEMPA US
<b>Release</b>	Event Monitoring
<b>Publication date</b>	4/18/2008 4:03:34 PM

---

---

# Table of Contents

## About this document

<b>Chapter 1 Introduction.....</b>	<b>1-1</b>
Event Monitoring.....	1-1
<b>Chapter 2 Events.....</b>	<b>2-1</b>
Events (tlbct5530m000).....	2-1
Fields and Commands.....	2-2
Fields.....	2-2
Form Commands.....	2-3
Event Statuses.....	2-4
Event State Transitions.....	2-5
<b>Chapter 3 Event Publishing Settings (tlbct5505m000).....</b>	<b>3-1</b>
Event Publishing Settings (tlbct5505m000).....	3-1
Fields.....	3-2

## Index

---



---

# About this document

## About this Guide

This document describes the new event monitoring functionality in Infor Enterprise Server.

This document contains the following chapters:

- Chapter 1, "Introduction," provides an overview of the basic event monitoring functionality, which was already available in earlier versions, and the new functionality.
- Chapter 2, "Events," describes the *Events (tlbct5530m000)* (p. 2-1) session and provides information on possible event statuses.
- Chapter 3, "Event Publishing Settings," describes the *Event Publishing Settings (tlbct5505m000)* (p. 3-1) session.



## Event Monitoring

Although the status of the event publishing can be monitored, in Infor Enterprise Server you cannot see what happened to a specific triggered event. Has the event already been published, or is the event still in the queue, or did an error occur?

The basic functionality is as follows:

- In the Event Subscriptions (tlbct5500m000) details session, the end commit time of the publisher is visible. The end commit time indicates that events from all transactions committed on or before that moment were processed.
- If an error occurs, it is handled in Integration 6.2. Depending on the configuration and type of error, it is either sent to an error handler bus component, or a retry is done after some time, or the error (including the message as it was when the error occurs) is written to an OpenWorld log file.
- If a subscription exists, an event is never lost. Either the event is still in the queue, or delivered to the specified destination, or is handled as an error in Integration 6.2.

Event Monitoring adds the following:

- An *Events* (tlbct5530m000) (p. 2-1) session is available. You can view the events that are in the queue to be published from ERP LN.
- An *Event Publishing Settings* (tlbct5505m000) (p. 3-1) session is available, where you can configure to keep the event history. If the history is kept, processed events can also be viewed in the Events (tlbct5530m000) session.

Note that this functionality is an add-on and does not replace the basic functionality.





## Events (tlbct5530m000)

Use this session to view the events that are in the queue to be published from ERP LN. If the history is kept, you can also view processed events.

This session can be started from the Event Subscriptions (tlbct5500m000) overview and details session only if a subscription is selected. If the porting set does not support BLOBs, the session cannot be started. Therefore, it will also be invisible in the Event Subscriptions (tlbct5500m000) session.

The session provides an overview of the messages that are in the queue to be published. If event history is switched on, the events that were already processed, either successfully or unsuccessfully, can be viewed.

Note the following:

- If the application transaction that triggered the event is aborted, the event will not be visible because the publishing of the event will also be aborted.
- Audit-based events not triggered directly from the application using PublishEvent or a ShowAndPublishEvent method– but derived from database actions on the application data– are not monitored.
- For BOD-type events, event monitoring will work in the same way as for BDE events. However, the eventRequestID will not be included in the BOD message. Therefore, when using event history, you must find the event for a published BOD by using Timestamp and, if available, business id.

### Note

If many events exist, the user interface responsiveness will decrease. The reason for this is to optimize the runtime event publishing; the events table has only one index or a few indices.

If the event does not have the status 'To be Published', an event message might be unavailable; this depends on the event history configuration (refer to the online help of the *Event Publishing Settings (tlbct5505m000)* (p. 3-1) session). For Error events, if the error occurred before the actual message was built, an event message might be unavailable. For Irrelevant events, no message will be available.

## Fields and Commands

The session displays the company and business objects from the selected subscription. You cannot change these objects. The existing events for the subscription are listed. For each event, various fields are shown. For details, refer to the *Fields* (p. 2-2) section.

The session's **Specific** menu contains various commands. For details, refer to the *Form Commands* (p. 2-3) section.

### Note

You cannot use the standard **Find** command to search for a specific event. Instead, use the **Sort by** command in the **Specific** menu and then browse through the list of events, or use the standard **Filter** command to view one or more specific events.

## Fields

### Company

The number of the ERP LN company from which events are published through the business object.

### Business Object

The business object that provides the event publishing service.

### Event Timestamp

The time when the event was stored.

### Note

The Timestamp is an indication, and is not used for sequencing; it does not have to be equal to the commit time used in the audit trail.

### Event Key

The identifier of the event.

### Note

The Event Key is not a sequence number. For BDE events, the same id is included as the eventRequestID in the event message control area to facilitate traceability.

### Business ID

The Business ID of the object instance for which the event occurred; for example, an order number or an item code.

---

**Note**

The Business ID is an internal identifier, which does not have to be equal to the public identifier used in the event.

Values are trimmed. In case of multiple identifiers, the elements are separated using a pipeline ('|') character. In case of audit-based events or PublishEvent, it might be empty.

The value might be incomplete because of maximum field length.

**Event Status**

The current state of the event.

For details, refer to *Event Statuses*.

**Status Details**

Additional notes on the status; for example, to indicate that the publisher retried sending the message.

**Last Update Timestamp**

The time when the last status change of the event occurred.

## Form Commands

**Sort by**

Use this command to sort the events in the scope of a company and business object.

You can order the events with the following:

- Timestamp, which is the default ordering.
- Event ID.
- Business ID, Timestamp.

**Event Publishing Settings**

Starts the *Event Publishing Settings (tlbct5505m000)* (p. 3-1) session.

**Show message**

Use this command to view the event message.

The event is displayed in XML format in a separate window. If the event message is large, displaying it might take time and system memory; therefore, the size will be reported beforehand, and the viewing can be canceled.

---

## Event Statuses

The following status values are used:

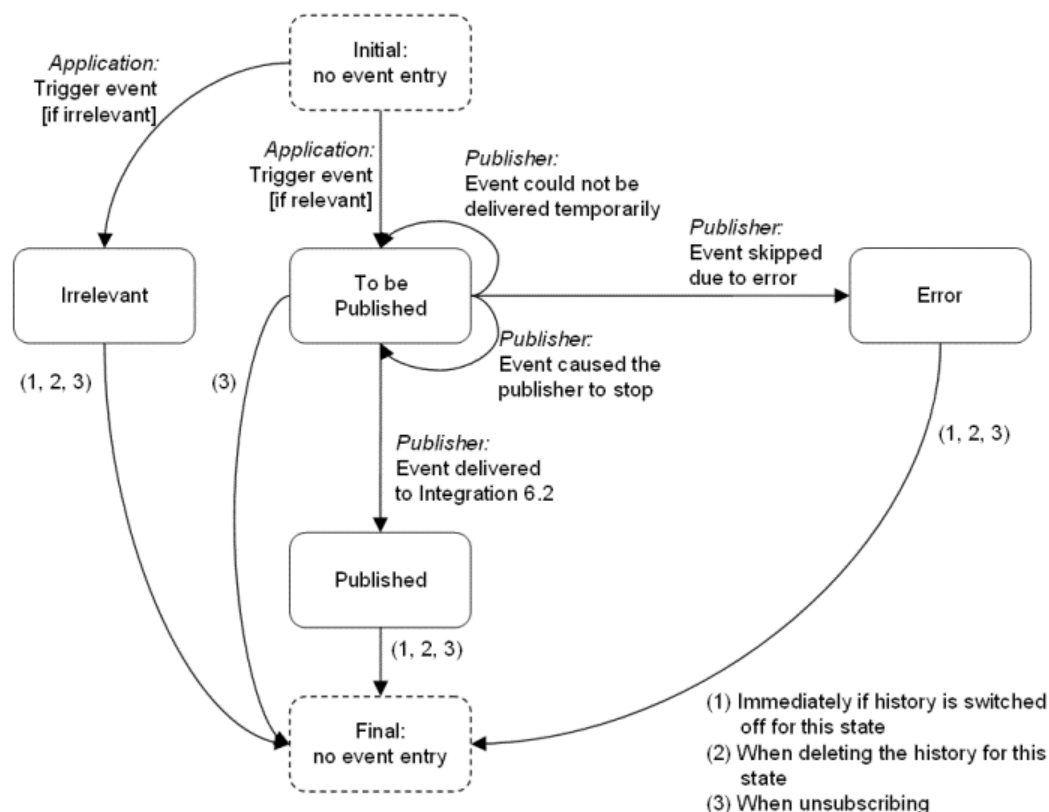
Irrelevant	The event is skipped because it does not match the subscription filter.
To be Published	The event is stored in the queue and ready to be picked up by the publisher.
Published	The event is published. Additionally, this status does not necessarily mean the event is successfully processed at the destination. Either the event is successfully processed at the destination, or a problem occurred outside ERP LN and the event is written to the OpenWorld log, or the event was successfully received at the destination and skipped.
Error	An error occurred, and the publishing continues with the next event. For example, the data could not be retrieved for the event. Note that if the publisher stops or becomes interrupted because of an error, this status is not used, because in that case the publisher will try to send the event next time.

If the event history is not kept (refer to the online help of the *Event Publishing Settings (tlbct5505m000)* (p. 3-1) session), only events with the 'To be Published' state will exist.

---

## Event State Transitions

The following figure shows the transitions of the event status:



The 'To be Published' state usually means the event was not yet processed by the publisher. However, two situations exist in which it was processed, but could not yet be successfully completed:

- The 'To be Published (Retrying)' state means the publisher tried to publish the event, but it could not be delivered. The cause is probably temporary, so retrying will be done. However, note that this is the event state, not the publisher process state. The publisher might be stopped while retrying, so it does not have to be running at that moment.
- The 'To be Published (Halted)' state means that while processing the event, an error occurred that stopped the publisher process. When the publisher process is continued, the event will be picked up.

In case of status Error, the error details are unavailable in the Events session. The error message is not linked to the event, but has been published. Depending on the Integration 6.2 configuration, the error message is sent to an error handler bus component or written to the OpenWorld log file.

The Error state is not directly reached from the application. If an error on event publishing occurs in the application process, the process will not enqueue an event message with status Error, but will enqueue an error message with status

'To be Published' if no subscription exists for the business object. Additionally, if the problem occurred in the generated business object library, the business ID is lost; the business ID can only be stored if the error occurs in the standard event publishing logic, available in the Tools AddOn package.

When unsubscribing, all events for the subscription are deleted. If events to be published are still waiting in the queue—because the publisher is not running or has a backlog—they will also be deleted. When unsubscribing from the user interface, you will be warned beforehand and asked to confirm the unsubscribe action. Note that this cannot be done when an UnsubscribeEvent request comes from outside ERP LN, because no user interaction is possible.

---

## Event Publishing Settings (tlbct5505m000)

Use this session to configure the Event Publishing Settings.

This session can be started from the *Events (tlbct5530m000)* (p. 2-1) session and the Event Subscriptions (tlbct5500m000) overview session. If the porting set does not support BLOBs, the session cannot be started; therefore, it will also be invisible in the Events (tlbct5530m000) and Event Subscriptions (tlbct5500m000) session.

For each event status other than 'To be Published', you can configure the following:

- Whether the events must be kept after they reach this state.
- Whether the event message (XML) must be kept (for Error and Published state only).
- How long the messages must be kept.

The last two settings can only be configured if the first one is switched on.

The settings apply to all companies.

To store the updated settings, you can choose **OK**; to undo the changes and close the session, choose **Cancel**. When selecting **OK**, obsolete events are deleted if according to the new settings less history events must be kept.

At runtime, deleting the history of a company is only done when at least one publisher is running for that company. The publisher Bshell will delete obsolete history data when starting; while running, it will recheck on an hourly basis. If the delete action fails, an error message is published. Changed settings are automatically picked up by the publisher Bshell at the next delete action; therefore, when changing the settings, you do not have to restart any publishers.

When deleting the history, the last update time of an event is used because this is the time the event reached its final state, whether Irrelevant, Error, or Published. Therefore, if an event was triggered three hours ago and published one hour ago, it will not yet be deleted if it is configured to be deleted when older than two hours.

**Warning**

Keeping event history increases the database size and will reduce event publishing performance, especially when high volumes of messages occur or the 'Also Keep Event Message' is switched on. Additionally, if no publisher is running for a company, no deletion will be carried out. Note that the events are stored per company, so a table instance exists per company.

**Fields****Irrelevant Events****Keep History**

If this check box is selected, events will be kept after they reach the Irrelevant state.

**Delete if older than**

The number of hours after which the events will be deleted.

**Error Events****Keep History**

If this check box is selected, events will be kept after they reach the Error state.

**Also Keep Event Message**

If this check box is selected, the event messages (XML) will also be kept.

**Delete if older than**

The number of hours after which the events and messages will be deleted.

**Published Events****Keep History**

If this check box is selected, events will be kept after they reach the Published state.

**Also Keep Event Message**

If this check box is selected, the event messages (XML) will also be kept.

**Delete if older than**

The number of hours after which the events and messages will be deleted.

---



---

# Index

**Event**, 2-1, 2-4, 3-1

    monitor, 1-1, 2-1, 3-1

    publish, 3-1

    status, 2-4

**Event Monitoring**, 1-1, 2-1, 2-4, 3-1

**Event Publishing**, 3-1

---

