

Infor LN Public Interfaces Reference Guide (On-premises)

Release 10.5.2

Copyright © 2024 Infor

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, the terms of which separate agreement shall govern your use of this material and all supplemental related materials ("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.

Without limitation, U.S. export control laws and other applicable export and import laws govern your use of this material and you will neither export or re-export, directly or indirectly, this material nor any related materials or supplemental information in violation of such laws, or use such materials for any purpose prohibited by such laws.

Trademark Acknowledgements

The word and design marks set forth herein are trademarks and/or registered trademarks of Infor and/or related affiliates and subsidiaries. All rights reserved. All other company, product, trade or service names referenced may be registered trademarks or trademarks of their respective owners.

Publication Information

Release: Infor LN 10.5.2

Publication date: March 26, 2024

Contents

About this guide	5
Intended audience	5
Related documents	5
Contacting Infor	5
Chapter 1 Introduction	6
Public Interfaces	6
Public Interfaces explained	6
How to request a new Public Interface	6
Public Interface example	7
Available Public Interfaces	8
Chapter 2 Public Interfaces for Extensibility	9
Public Interfaces for Exception	9
Exception.Delete	9
Exception.GetMessage	9
Exception.NumberOfMessages	10
Chapter 3 Public Interfaces for BOD	11
Public Interfaces for BOD	11
BOD.ActionsAfterProcessingIncomingRequest	12
BOD.ActionsAfterProcessingIncomingRequestWithAutomaticProcessing	12
BOD.ActionsBeforeProcessingIncomingRequest	13
BOD.ExecuteMethod	14
BOD.ExecuteOnAcknowledgeForChameleon	15
BOD.ExecuteOnLoadForChameleon	16
BOD.ExecuteOnProcessForChameleon	16
BOD.ExecuteOnSyncForChameleon	17
BOD.ExecuteOnUpdateForChameleon	18
BOD.ExecutePublishForChameleon	19
BOD.ExecuteShowForChameleon	19

BOD.GetIdAccountingEntity	20
BOD.GetIdLocation	21
BOD.GetIdLogicalID	22
BOD.HandleStagingAfterProcessingIncomingRequest	23
BOD.HandleStagingBeforeProcessingIncomingRequest	23
BOD.Publish	24
BOD.PublishLNMessage	25

About this guide

Intended audience

This guide is intended for IT professionals working in implementation projects or IT optimization phases for Infor LN. Basic knowledge about the Infor LN software structure and Infor LN's 4GL programming language is a pre-requisite.

Related documents

You can find these documents on docs.infor.com:

- Infor LN Studio Application Development Guide
- Infor LN Studio Integration Development Guide
- · Infor LN Extensions Development Guide

You can find the *Infor ES Programmer's Guide* in <u>KB2924522</u>. The content of this guide is also available in the help pages of Infor LN Studio.

Contacting Infor

If you have questions about Infor products, go to Infor Concierge at https://mingle-portal.us2.prd3.inforcloudsuite.com/v2/CONCIERGE_PRD and create a support incident.

For the latest documentation, go to Documentation Central at docs.infor.com. We recommend that you check this website periodically for updated documentation. If you have comments about Infor documentation, contact documentation@infor.com.

Chapter 1 Introduction

Infor LN is a standard ERP application with rich functionality. With its built-in flexibility with parameters, workflows, dynamic processes, it can be adjusted to serve the business processes in the industries Infor LN is designed for. To close the small gaps between the standard functionality and the specific business needs, Infor LN offers a variety of extensibility possibilities. The main goal of extensibility is to develop the last-mile functionality for your organization without changing the core standard software components and using only the public interfaces of the standard application. In this way, you can develop the extensions fully separated from the standard components and upgrading the standard software will therefore not result in additional efforts and costs for upgrading the customizations. Extensions are not influenced by the upgrade process.

Public Interfaces

Public Interfaces are methods with LN application functionality that can be called from extensions.

Public Interfaces explained

The development of extensions can be made easier if methods from the LN Application can be used. This can be done by using the so-called LN Public Interfaces. LN Public Interfaces are functions in the LN Application that are available for anyone who develops extensions on LN. The available LN Public Interfaces are visible in LN Studio and in the Extension Modeler. LN Public Interfaces are generic functions with a certain level of complexity that will likely be used by multiple customers. Simple read actions, for example, needs to be developed by customers or implementation partners themselves. Moreover, LN Public Interfaces will perform something what cannot easily be achieved with one or more standard sessions or processes in LN.

How to request a new Public Interface

Apply the following process if you need a new LN Public Interface:

Evaluate if the new LN Public Interface is generic and contains a certain level of complexity. Make sure the required feature cannot be achieved with personalizing a standard session or process.

- 2 Create an incident and clearly describe the required LN Public Interface. Use the Excel Sheet "Request Template for LN Public Interfaces & Process Extensions)" which is attached to KB2003722 in the Infor Customer Portal.
- 3 Infor Support will create a defect for this incident.
- 4 Infor Development will review the requested LN Public Interface and will either develop the new LN Public Interface or reject the request. If the LN Public Interface is developed it will be released via the regular delivery process.
- 5 After the new solution has been installed the customer can use the new LN Public Interface in the extension modeler and/or in LN Studio.
- 6 Infor Support will complete the incident.

Public Interface example

Find below an example of one of the LN Public Interfaces. This Public Interface converts an amount to another currency.

```
long Common.ConvertAmount(
         domain tenemp
                               iFinancialCompany,
         domain
                 tcamnt
                               iSourceAmount,
                               iSourceCurrency,
          domain tcccur
         domain tcrtyp
                               iExchangeRateType,
         domain tcdate
                               iRateDateUTC,
         domain
                 tcccur
                               iTargetCurrency,
         domain tcamnt
   ref
                               oTargetAmount,
   ref
         domain tcmcs.s999m oExceptionMessage mb,
   ref
                  long
                               oExceptionID)
```

It is very important to use LN Public Interfaces properly and to catch the errors. For this reason, each Public Interface has the output arguments 'oExceptionMessage' and 'oExceptionID'.

If the Public Interface returns a value unequal to zero, then argument oExceptionMessage is filled for sure with the latest exception message and argument oExceptionID is a reference to an XML-object that contains some more information about the exception. This extra information can be retrieved by using the Public Interface 'Exception' in otcextextapi. It is also important to free up memory by calling Public Interface Exception.Delete(...).

If the Public Interface returns zero, arguments oExceptionMessage and oExceptionID could be filled due to information messages. So, in fact only the return value indicates if a public interface is successful or not.

The next example shows how to implement error handling when using LN Public Interfaces, in this case Common.ConvertAmount to convert an amount to another currency.

```
#pragma used dll "otcextextapi"
#pragma used dll "otcextemmapi"
                               exception.id, i
                 long
        domain
               tcmcs.s999m exception.message
if Common.ConvertAmount(..., exception.message, exception.id) <> 0 then
        |* Exception(s) found.
        for i = 1 to Exception.NumberOfMessages(exception.id)
                dal.set.error.message("@"& Exception.GetMessage(i))
        endfor
        Exception.Delete(exception.id)
        return (DALHOOKERROR)
else
        |* Call was successful. Exception messages could exist!
        Exception.Delete(exception.id)
endif
```

To be able to use the Public Interface in your extensions, you need to add a "#pragma used dll" statement for the DLL that contains the Public Interface. The DLL names can be found in the documentation of the available Public Interfaces. Note that the DLL name needs to be preceded by an "o" in the #pragma-statement.

Available Public Interfaces

The next chapters of this document describe the available Public Interfaces for Infor LN (10.5.2) and what their usage is.

If the Public Interfaces described in this document are not shown in the Extension Modeler or Infor LN Studio in your environment, it may be necessary to apply a Knowledge Base article (KB) that can be found in the Infor Customer Portal. The applicable KB number is mentioned in the Public Interface description.

Chapter 2 Public Interfaces for Extensibility

Public Interfaces for Exception

The following functions are available:

Exception.Delete

Exception.GetMessage

Exception.NumberOfMessages

Exception.Delete

DLL:	tcextextapi		
	This function is av	ailable from KB20	040021.
Syntax:	Exception.Delete	e (
	ref	long	ioExceptionID)
Usage:	Expl: Pre: Post: Input: Output: Return:	<pre>ioExceptionID None ioExceptionID ioExceptionID</pre>	deletes the exception to free memory. should refer to an Exception. - the exception id. - the exception id.

Exception.GetMessage

DLL:	tcextextapi

	This function is ava	ailable from KB2040021.				
Syntax:	X: Exception.GetMessage(
	ref	long iExceptionID, long iMessageIndex, domain tcmcs.s999m oMessageDescription mb)				
Usage:	Expl: Pre: Post: Input:	This function reads message description from all messages in the XML identified by iExceptionID for a certain index. iExceptionID should refer to an Exception. None iExceptionID - the exception id. iMessageIndex - the index for the message to be returned. iMessageIndex should be greater than zero and less than the number of messages. oMessageDescription - The found message.				
	Return:	None				

${\bf Exception. Number Of Messages}$

DLL:	tcextextapi					
	This function is ava	ailable from KB2040021.				
Syntax:	long Exception.	NumberOfMessages(
		long iExceptionID)				
Usage:	Expl:	This function determines the number if messages that have been stored in the XML where iExceptionID refers to.				
	Pre:	iExceptionID should refer to a valid XML with the structure as described above.				
	Post: Input:	None				
	iExceptionID - the exception id that points to the XML. Output:					
		None				
	Return:	The number of found messages. If iExceptionID does not refers to a valid XML the return value is 0.				

Chapter 3 Public Interfaces for BOD

Public Interfaces for BOD

The following functions are available:

BOD.ActionsAfterProcessingIncomingRequest

BOD.ActionsAfterProcessingIncomingRequestWithAutomaticProcessing

BOD.ActionsBeforeProcessingIncomingRequest

BOD.ExecuteMethod

BOD.ExecuteOnAcknowledgeForChameleon

BOD.ExecuteOnLoadForChameleon

BOD.ExecuteOnProcessForChameleon

BOD.ExecuteOnSyncForChameleon

BOD.ExecuteOnUpdateForChameleon

BOD.ExecutePublishForChameleon

BOD.ExecuteShowForChameleon

BOD.GetIdAccountingEntity

BOD.GetIdLocation

BOD.GetIdLogicalID

BOD.HandleStagingAfterProcessingIncomingRequest

BOD.HandleStagingBeforeProcessingIncomingRequest

BOD.Publish

BOD.PublishLNMessage

BOD. Actions After Processing Incoming Request

DLL:	tcextbodapi	cextbodapi				
	This function is ava	ailable fro	m <u>KB2040021</u> .			
Syntax:	long BOD.Actions	sAfterPr	AfterProcessingIncomingRequest(
			long	iXMLRequest,		
	ref		long	oResult,		
	ref	domain	tcmcs.s999m	oExceptionMessage mb,		
	ref		long	oExceptionID)		
Usage:	Expl:	This fu	nction publishes	the staged BODs that were triggered		
			•	he incoming request and resets the		
		paramet	er that enabled t	he staging of outgoing BODs in the		
				ingIncomingRequest() function.		
			± '	called in the AfterExecuteHook		
				, e.g. the OnProcess, OnLoad, OnSync,		
		-	OnAcknowledge et			
	Pre:		_	is processed, function		
			ionsBeforeProcess	ingIncomingRequest() must be called.		
	Post:	NA				
	_	iXMLReq		- XML structure with request. Mandatory		
	Output:	oResult		- 0 if succes, otherwise <> 0		
		oExcept	ionMessage	- The last message if the return		
				value is not equal to 0.		
				If more than one message is		
				given, these are present in the		
				oExceptionID		
		oExceptionID - An ID that refers to all error information. Use the functions in				
				Exception to get all relevant		
				information.		
	Return:	0 / DAL	HOOKERROR			

BOD.ActionsAfterProcessingIncomingRequestWithAutom aticProcessing

DLL:	tcextbodapi							
	This function is av	his function is available from KB2040021.						
Syntax:	long BOD.Action	sAfterPro	cessingIncomingRe	equestWithAutomaticProcessing(
			long	iXMLRequest,				
			string	iObjectType(1),				
			string	iObject(1),				
	ref		long	oResult,				
	ref	domain	tcmcs.s999m	oExceptionMessage mb,				
	ref		long	oExceptionID)				
Usage:	Expl:	This fur	nction publishes	the staged BODs that were triggered				
		from the processing of the incoming request and resets the						
		parameter that enabled the staging of outgoing BODs in the						
		BOD.Act	ionsBeforeProcess	ingIncomingRequest() function.				

Furthermore, it starts the automatic processing (if any) for the object that was created/updated in LN from the incoming request. Preferably, this function should be called in the AfterExecuteHook of every OnEvent section, e.g. the OnProcess, OnLoad, OnSync, OnShow, OnAcknowledge etc. Pre: Before the incoming BOD is processed, function BOD.ActionsBeforeProcessingIncomingRequest() must be called. Post: NA Input: iXMLRequest - XML structure with request. Mandatory iObjectType - object type for which automatic processing must be started, e.g. "PurchaseOrder" i0bject - The identifier of the given ObjectType e.g. the purchase order if ObjectType is "PurchaseOrder" Output: oResult - 0 if succes, otherwise <> 0 oExceptionMessage - The last message if the return value is not equal to 0. If more than one message is given, these are present in the oExceptionID oExceptionID - An ID that refers to all error information. Use the functions in Exception to get all relevant information. Return: 0 / DALHOOKERROR

BOD.ActionsBeforeProcessingIncomingRequest

DLL:	tcextbodapi						
	This function is a	his function is available from KB2040021.					
Syntax:	X:long BOD.ActionsBeforeProcessingIncomingRequest(
			long	iXMLRequest,			
	ref		boolean	oCancel,			
	ref	domain	tcmcs.s999m	oExceptionMessage mb,			
	ref		long	oExceptionID)			
Jsage:	Expl:	paramet It vali is not matches company BOD Par case of It extr request be dete It sets that ar Prefera	ers needed duri dates if the Ac checked for mas the Accounting (in Tenant, Ac ameters) and se a mismatch. acts the LastMo and determines rmined, it swit a parameter th e triggered fro bly, it should	s the incoming request and sets ng the processing. counting Entity and the Location (Location ter data BODs) of the incoming request Entity and Location of the current counting Entity and Location setup or in ts output argument oCancel to true in dificationPerson/IDs/ID from the incoming the linked LN user. If the LN user can ches user to the LN user. at enables the staging of outgoing BODs m the processing of the incoming request. be called in the BeforeExecuteHook on, e.g. the OnProcess, OnLoad, OnSync,			

OnShow, OnAcknowledge etc. Pre: Post: After the incoming request is processed, BOD.ActionsAfterProcessingIncomingRequest() or ${\tt BOD.ActionsAfterProcessingIncomingRequestWithAutomaticProcessing()}$ must be called. Input: iXMLRequest - XML structure with request. Mandatory Output: oCancel - true, if BOD must be cancelled - false, if BOD can be processed oExceptionMessage - The last message if the return $% \left(1\right) =\left(1\right) \left(1\right)$ value is not equal to 0. If more than one message is given, these are present in the oExceptionID oExceptionID - An ID that refers to all error information. Use the functions in Exception to get all relevant information. Return: 0 - No error occurred while executing the actions. DALHOOKERROR - An error occurred while executing the actions, or a mismatch was detected during accounting entity / location validation and BOD parameter Inbound Routing Error Handling is set to Return Error.

BOD.ExecuteMethod

DLL:	tcextbodapi			
	This function is ava	ailable from	KB2019035.	
Syntax:	long BOD.Execut	eMethod(
		domain t	cbod.name	iNoun,
		domain t	cmcs.str30	iMethod,
]	long	iXMLRequest,
	ref]	long	oXMLResponse,
	ref]	long	oXMLResult,
	ref	domain t	cmcs.s999m	oExceptionMessage mb,
	ref		long	oExceptionID)
Usage:	Expl:	This fund	ction executes a	non-batch method for a specified
		BOD.		
	Pre:	NA		
	Post:	NA		
	Input:	iNoun	- The (p	rotected) Noun for which the method
			must b	e executed, e.g.
				veDeliveryWarehousingBOD". Mandatory
		iMethod		thod to be executed,
			2	Create" or "Change". Mandatory
		iXMLReque		ructure with request. Mandatory
	Output:	oXMLRespo		ructure with response (if method is ed successfully)
		oXMLResul	lt - XML st	ructure with result (in case of error)

oExceptionMessage	- The last message if the return
	value is not equal to 0.
	If more than one message is
	given, these are present in the
	oExceptionID
oExceptionID	- An ID that refers to all error
	information. Use the functions in
	Exception to get all relevant
	information.
Return values:	
0	- method is executed successfully
DALHOOKERROR	- on errors

BOD. Execute On Acknowledge For Chameleon

DLL:	tcextbodapi					
	This function is available from KB2019035.					
	This function is ava	aliable from NDZ	<u>:019035</u> .			
Syntax:	X: long BOD.ExecuteOnAcknowledgeForChameleon(
		domain tcboo	d.name	iProtectedNoun,		
		long		iXMLRequest,		
	ref	long		oXMLResponse,		
	ref	long		oXMLResult,		
	ref	domain temes	s.s999m	oExceptionMessage mb,		
	ref	long		oExceptionID)		
Usage:	Expl: Pre:	noun of the onexecuteHook noun. The fur	chameleon l cof the On action is a	the OnAcknowledge method of a protected BOD and can be called from the nAcknowledge method of an incoming public used to route an incoming BOD with protected noun.		
	Post:	NA				
	Input:		OnAcl "Sale	(protected) Noun for which the knowledge method must be executed, e.g. esOrderInBOD". Mandatory		
	Output:	iXMLRequest oXMLResponse	- XML :	structure with request. Mandatory structure with response (if method is uted successfully)		
		oXMLResult oExceptionMes		structure with result (in case of error) - The last message if the return value is not equal to 0. If more than one message is given, these are present in the oExceptionID		
		oExceptionID		 An ID that refers to all error information. Use the functions in Exception to get all relevant information. 		
	Return:	0 / DALHOOKER	RROR			

BOD.ExecuteOnLoadForChameleon

DLL:	tcextbodapi		
	This function is ava	ailable from KB2019035.	
Syntax:	long BOD.Execute	eOnLoadForChameleon(
		domain tcbod.name	iProtectedNoun,
		long	iXMLRequest,
	ref	long	oXMLResponse,
	ref	long	oXMLResult,
	ref	domain tcmcs.s999m	oExceptionMessage mb,
	ref	long	oExceptionID)
Usage:	Expl:	This function executes the	ne OnLoad method of a protected noun
		of the chameleon ${\tt BOD}$ and	can be called from the OnExecuteHook
		of the OnLoad method of a	an incoming public noun. The function
		is used to route an incom	ning BOD with Load verb to a protected
		noun.	
	Pre:	NA	
	Post:	NA	
	Input:	·-	rotected) Noun for which the OnLoad
			must be executed, e.g.
			OrderInBOD". Mandatory
		-	ructure with request. Mandatory
	Output:	_	ructure with response (if method is
			ed successfully)
			ructure with result (in case of error)
		oExceptionMessage -	- The last message if the return
			value is not equal to 0.
			If more than one message is
			given, these are present in the
		- Format Laure	oExceptionID
		oExceptionID -	- An ID that refers to all error
			information. Use the functions in
			Exception to get all relevant information.
	Dataman	0 / DATHOOMEDDOD	information.
	keturn:	0 / DALHOOKERROR	
<u> </u>	1		

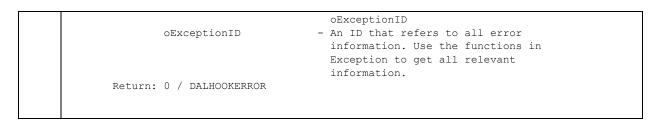
BOD.ExecuteOnProcessForChameleon

DLL:	tcextbodapi								
	This function is av	vailable fro	m <u>KB2019035</u> .						
Syntax:	long BOD.Execut	long BOD.ExecuteOnProcessForChameleon(
		domain	tcbod.name	iProtectedNoun,					
			long	iXMLRequest,					
	ref		long	oXMLResponse,					
	ref		long	oXMLResult,					
	ref	domain	tcmcs.s999m	oExceptionMessage mb,					
	ref		long	oExceptionID)					
Usage:	Expl:	This fu	nction executes	the OnProcess method of a protected noun					
		of the	chameleon BOD an	d can be called from the OnExecuteHook					

```
of the OnProcess method of an incoming public noun. The function
       is used to route an incoming BOD with Process verb to a
       protected noun.
Post:
       NA
Input: iProtectedNoun - The (protected) Noun for which the OnProcess
                       method must be executed, e.g.
                        "SalesOrderInBOD". Mandatory
       iXMLRequest - XML structure with request. Mandatory
Output: oXMLResponse \, - XML structure with response (if method is
                        executed successfully)
       oXMLResult - XML structure with result (in case of error)
       oExceptionMessage
                             - The last message if the return
                                value is not equal to 0.
                                If more than one message is
                                given, these are present in the
                                oExceptionID
       oExceptionID
                               - An ID that refers to all error
                                information. Use the functions in
                                Exception to get all relevant
                                information.
Return: 0 / DALHOOKERROR
```

BOD.ExecuteOnSyncForChameleon

DLL:	tcextbodapi									
	This function is av	nis function is available from KB2019035.								
Syntax:	long BOD.Execut	eOnSyncFor	Chameleon	on (
-		domain t	cbod.name	me iProtectedNoun,						
		1	ong	iXMLRequest,						
	ref	1	ong	oXMLResponse,						
	ref	1	ong	oXMLResult,						
	ref	domain t	cmcs.s999r	99m oExceptionMessage mb,						
	ref	1	ong	oExceptionID)						
Usage:	Expl:	This function executes the OnSync method of a protected noun of the chameleon BOD and can be called from the OnExecuteHook of the OnSync method of an incoming public noun. The function is used to route an incoming BOD with Sync verb to a protected noun.								
	Pre:	NA								
	Post:	NA								
	Input:	iProtecte	I	- The (protected) Noun for which the OnSync method must be executed, e.g. "SalesOrderInBOD". Mandatory - XML structure with request. Mandatory						
	Output:	oXMLRespo	nse - 2	- XML structure with response (if method is executed successfully)						
		oXMLResul	t - 2	- XML structure with result (in case of error)						
		oExceptio	nMessage	e - The last message if the return						
				value is not equal to 0.						
				If more than one message is						
				given, these are present in the						



BOD. Execute On Update For Chameleon

DLL:	tcextbodapi						
	This function is available from KB2019035.						
	This function is ava	ailable from KB201	<u>9035</u> .				
Syntax:	long BOD.Execute	eOnUpdateForCham	eleon(
		domain tcbod.n	ame	iProtectedNoun,			
		long		iXMLRequest,			
	ref	long		oXMLResponse,			
	ref	long		oXMLResult,			
	ref	domain tcmcs.s	999m	oExceptionMessage mb,			
	ref	long		oExceptionID)			
Usage:	Expl:	This function e	xecutes t	he OnUpdate method of a protected noun			
		of the chameleo	n BOD and	can be called from the OnExecuteHook			
		of the OnUpdate	method o	f an incoming public noun. The function			
		is used to rout	ming BOD with Update verb to a protected				
		noun.					
	Pre:	NA					
	Post:	NA					
	Input:	iProtectedNoun	- The (p	rotected) Noun for which the OnUpdate			
			method	must be executed, e.g.			
			"Sales	OrderInBOD". Mandatory			
		iXMLRequest	- XML st	ructure with request. Mandatory			
	Output:	oXMLResponse		ructure with response (if method is			
				ed successfully)			
		oXMLResult		ructure with result (in case of error)			
		oExceptionMessa	ge	- The last message if the return			
				value is not equal to 0.			
				If more than one message is			
				given, these are present in the			
		oExceptionID		oExceptionID - An ID that refers to all error			
		OFXCebriouin		information. Use the functions in			
				Exception to get all relevant			
	Datama	0 / DATHOOMEDDO	D.	information.			
	keturn:	0 / DALHOOKERRO	K				

BOD.ExecutePublishForChameleon

DLL:	tcextbodapi		
	This function is ava	ailable from KB2019035.	
Syntax:	long BOD.Execute	ePublishForChameleon(
		domain tcbod.name	iProtectedNoun,
		long	iXMLRequest,
	ref	long	oXMLResponse,
	ref	long	oXMLResult,
	ref	domain tcmcs.s999m	oExceptionMessage mb,
	ref	long	oExceptionID)
Usage:	Expl:	This function executes t	the publishing for a protected noun
		of the chameleon BOD and	d can be called from the OnExecuteHook
		of the PublishEvent meth	nod of a protected noun.
		cted noun to the public noun and	
		executes the PublishEver	nt method of the public noun
	Pre:	NA	
	Post:	NA	
	Input:	*	cotected Noun, which is the BOD
			calls this function, e.g.
			actionOrderSFCBOD". Mandatory
		•	cructure with request. Mandatory
	Output:	=	tructure with response (if PublishEvent
			d is executed successfully)
			tructure with result (in case of error)
		oExceptionMessage	- The last message if the return
			value is not equal to 0.
			If more than one message is
			given, these are present in the oExceptionID
		oExceptionID	- An ID that refers to all error
		OEXCEPTIONID	information. Use the functions in
			Exception to get all relevant
			information.
	Return.	0 / DALHOOKERROR	INTOTMACTON.
	inccurii.	o , Billioondiator	

BOD.ExecuteShowForChameleon

DLL:	tcextbodapi							
	This function is a	vailable fro	m <u>KB2019035</u> .					
Syntax:	long BOD.ExecuteShowForChameleon(
		domain	tcbod.name	iProtectedNoun,				
			long	iXMLRequest,				
	ref		long	oXMLResponse,				
	ref		long	oXMLResult,				
	ref	domain	tcmcs.s999m	oExceptionMessage mb,				
	ref		long	oExceptionID)				
Usage:	Expl:	This fu	nction executes	the Show method of the public noun				
		of the	chameleon BOD a	and can be called from the OnExecuteHook				

of the Show method of a protected noun. It translates the protected noun to the public noun and executes the Show method of the public noun. Then it renames the response ${\tt XML}$ to the response ${\tt XML}$ of the protected noun Pre: Post: NA Input: iProtectedNoun - The protected Noun, which is the BOD that calls this function, e.g. "ProductionOrderSFCBOD". Mandatory iXMLRequest - XML structure with request. Mandatory Output: oXMLResponse - XML structure with response (if method is executed successfully) oXMLResult - XML structure with result (in case of error) oExceptionMessage - The last message if the return value is not equal to 0. If more than one message is given, these are present in the oExceptionID oExceptionID - An ID that refers to all error information. Use the functions in Exception to get all relevant information. Return: 0 / DALHOOKERROR

BOD.GetIdAccountingEntity

DLL:	tcextbodapi	pi						
	This function is av	on is available from KB1987704.						
Syntax:	long BOD.GetIdA	ccountin	αEntity/					
Оутпах.	Tong Bob. Geeran	domain	tenemp	iCompany,				
		domain	tcbod.name	iNoun, iEntityType,				
	const		string	iEntityCode(),				
		domain	tcmcs.tabl	iRootTable,				
	ref	domain	tcbod.acen	oAccountingEntity,				
	ref		boolean	oAccountingEntityisSet,				
	ref	domain	tcmcs.s999m	oExceptionMessage mb,				
	ref		long	oExceptionID)				
Usage:	Expl:	This function determines the Accounting Entity.		the Accounting Entity.				
	Pre:	NA						
	Post:	NA						
	Input:	iCompan	-	y for which the accounting				
			-	v is determined. Mandatory				
		iNoun		Mandatory				
		iEntity		Type: Mandatory for transactional data BODs				
			Possik 3 (Pro	ple values: 1 (Warehouse), 2 (Department), bject)				
		iEntity	-	Code: Mandatory for transactional data BODs ble values: The warehouse, department or				
			projec	et.				
			The Er	tity Type and Entity Code are used to				
			determ	nine the Tenant, AccountingEntity and				

Location. iRootTable - Root Table: Mandatory for master data BODs Output: oAccountingEntity - Accounting Entity oAccountingEntityisSet - Accounting Entity set (true or false). - The last message if the return oExceptionMessage value is not equal to 0. If more than one message is given, these are present in the oExceptionID oExceptionID - $\mbox{\fontfamily{\fontfam$ information. Use the functions in Exception to get all relevant information. Return: 0 - Accounting Entity is determined. DALHOOKERROR - Otherwise.

BOD.GetIdLocation

DLL:	tcextbodapi							
	This function is av	ailable fro	m KB1987704.					
	This function is available from KB1987704.							
Syntax:	long BOD.GetIdLo							
			tcncmp	iCompany,				
		domain		iNoun,				
			long	iEntityType,				
	const	, ,	string	iEntityCode(),				
			tcmcs.tabl	iRootTable,				
	ref	domain	tcbod.lctn	oLocation,				
	ref		boolean tcmcs.s999m	oLocationisSet,				
	ref ref	domain	long	oExceptionMessage mb, oExceptionID)				
Usage:	Expl:	Thie fu	nction determines	•				
Osage.	Pre:	NA	nccion deceimines	the bocation.				
	Post:	NA						
	Input:	iCompan	v - compan	y for which the location				
			= :	ermined. Mandatory				
				Mandatory				
				Type: Mandatory for transactional data BODs				
		Possible values: 1 (Warehouse), 2 (Department), 3 (Project)						
		iEntity		Code: Mandatory for transactional data BODs				
			-	le values: The warehouse, department or				
			projec	=				
				tity Type and Entity Code are used to				
				ine the Tenant, AccountingEntity and				
			Locati	on.				
		iRootTa	ble - Root T	able: Mandatory for master data BODs				
	Output:	oLocati	on	- Location				
		oLocati	onisSet	- Location set (true or false).				
		oExcept	ionMessage	- The last message if the return				
				value is not equal to 0.				
				If more than one message is				
				given, these are present in the				

oExceptionID	<pre>oExceptionID - An ID that refers to all error information. Use the functions in Exception to get all relevant information.</pre>
Return: 0 DALHOOKERROR	- Location is determined Otherwise.

BOD.GetIdLogicalID

	function is avagranded by BOD.GetIdLo	ogicalID domain		
Syntax: long	g BOD.GetIdLo	domain		
			L	
		domain	Cenemp	iCompany,
		uomain	tcbod.name	iNoun,
			long	iEntityType,
	const		string	iEntityCode(),
			tcmcs.tabl	iRootTable,
	ref	domain	tcbod.loid	oLogicalID,
	ref		boolean	oLogicalIDisSet,
	ref	domain	tcmcs.s999m	oExceptionMessage mb,
	ref		long	oExceptionID)
Usage:	Expl:		nction determines	the Logical ID.
	Pre:	NA		
	Post:	NA		
	Input:	iCompan	•	y for which the logical Id
		iNoun		ermined. Mandatory
		iEntity		Mandatory Type: Mandatory for transactional data BODs
		TEHLLICY		le values: 1 (Warehouse), 2 (Department),
		iEntity(,	Code: Mandatory for transactional data BODs
		TEHICLCY	Possib	le values: The warehouse, department or
			projec	
				tity Type and Entity Code are used to ine the Tenant, AccountingEntity and
			Locati	
		iRootTal		able: Mandatory for master data BODs
	Output.	oLogica		- LogicalID
	oacpac.	_	lIdisSet	- LogicalID set (true or false).
		_	ionMessage	- The last message if the return
				value is not equal to 0.
				If more than one message is
				given, these are present in the
				oExceptionID
		oExcept:	ionID	- An ID that refers to all error
				information. Use the functions in
				Exception to get all relevant
				information.
	Return:	0		- LogicalId is determined.
		DALHOOK	ERROR	- Otherwise.

BOD. Handle Staging After Processing Incoming Request

DLL:	tcextbodapi									
	This function is av	ailable from KB2040021.								
Syntax										
Syritax.	:long BOD.HandleStagingAfterProcessingIncomingRequest(domain tcbod.name iNoun,									
	ref	domain temes.s999m	iNoun, oExceptionMessage mb,							
	ref	long	oExceptionID)							
Usage:	-		nes the staged BODs that were triggered							
ougu.	nnpr.	•	of the incoming request and resets the							
		*	ed the staging of outgoing BODs in the							
		•	oreProcessingIncomingRequest() function.							
			d be called in the AfterExecuteHook							
		of every OnEvent sec	tion, e.g. the OnProcess, OnLoad, OnSync,							
		OnShow, OnAcknowledge	e etc.							
	Pre:	Before the incoming D	BOD is processed,							
		function BOD.HandleS	tagingBeforeProcessingIncomingRequest()							
		must be called.								
	Post:	NA								
	Input:	iNoun	- The noun of the incoming request							
			e.g. "PurchaseOrderBOD"							
	Output:	oExceptionMessage	- The last message if the return							
			value is not equal to 0.							
			If more than one message is							
			given, these are present in the							
		- Book and Court B	oExceptionID							
		oExceptionID	 An ID that refers to all error information. Use the functions in 							
			Exception to get all relevant							
			information.							
	Return:	0 / DALHOOKERROR	INTOTAL CION.							
	icculii.	o , billiooidiator								

BOD. Handle Staging Before Processing Incoming Request

DLL:	tcextbodapi							
	This function is available from KB2040021.							
Syntax:	long BOD.Handle	ong BOD.HandleStagingBeforeProcessingIncomingRequest(
		domain	tcbod.name	iNoun,				
	ref	domain	tcmcs.s999m	oExceptionMessage mb,				
	ref		long	oExceptionID)				
Usage:	Expl:		-	ameter that enables the staging of riggered from the processing of the				

incoming request. Preferably, it should be called in the BeforeExecuteHook of every OnEvent section, e.g. the OnProcess, OnLoad, OnSync, OnShow, OnAcknowledge etc. Pre: NA Post: After the incoming BOD is processed, function BOD.HandleStagingAfterProcessingIncomingRequest() must be called - The noun of the incoming request Input: iNoun e.g. "PurchaseOrderBOD" Output: oExceptionMessage - The last message if the return value is not equal to 0. If more than one message is given, these are present in the oExceptionID oExceptionID - An ID that refers to all error $% \left(1\right) =\left(1\right) \left(1\right)$ information. Use the functions in Exception to get all relevant information. Return: 0 / DALHOOKERROR

BOD.Publish

DLL:	tcextbodapi					
	This function is av	s function is available from KB1987704.				
Syntax:	long BOD.Publis	sh (
		domain	tcbod.name	iNoun,		
		domain	tcmcs.tabl	iRootTable,		
	const		string	iActionCode(),		
			long	iEntityType,		
	const		string	<pre>iEntityCode(),</pre>		
	const		string	iDocumentId(),		
	const		string	iProcessingAction(),		
	ref	domain	tcmcs.s999m	oExceptionMessage mb,		
	ref		long	oExceptionID,		
)		
Usage:	Expl:	This function publishes any BOD.				
	Pre:		tion handling	' I		
	Post:		g is needed: A commit or abort must be done.			
	Input:			oun: Mandatory		
		iActionCode - Action Code: Mandatory		oot Table: Mandatory for master data BODs		
				<u>-</u>		
		Possible values: "Add", "Change", "Delete",				
		=		Replace", "Canceled".		
		iEntity		tity Type: Mandatory for transactional data BODs		
				ossible values: 1 (Warehouse), 2 (Department),		
				(Project)		
		iEntity		ntity Code: Mandatory for transactional data BODs		
				essible values: The warehouse, department or		
			-	roject.		
				ne Entity Type and Entity Code are used to		
			de	termine the Tenant, AccountingEntity and		

```
Location.
       iDocumentId - Document ID: Mandatory
        iProcessingAction - Processing Action: Mandatory
                         Possible values: "OnlyStage", "StageOrPublish",
                         "OnlyPublish".
       Variable arguments are Identifiers
                       - The first Identifier is Mandatory
Output: oExceptionMessage
                              - The last message if the return
                                value is not equal to 0.
                                If more than one message is
                                given, these are present in the
                                 oExceptionID
       oExceptionID
                               - An ID that refers to all error
                                 information. Use the functions in
                                Exception to get all relevant
                                information.
Return: 0
                               - BOD is published.
       DALHOOKERROR
                              - Otherwise.
```

BOD.PublishLNMessage

DLL:	tcextbodapi This function is available from KB2040021.						
Syntax:	tax: long BOD.PublishLNMessage(
	ref long oExc	eptionMessage mb, eptionID,					
Usage:	The first three arguments are	This function publishes the LnMessageBOD with a specific MessageType. The first three arguments are mandatory, in the given order. iMessageType can f.i. have value "PurchaseScheduleLine".					
	pairs, e.g.: "Description",	The other arguments should be 'xml_elementname - xml_elementvalue' pairs, e.g.: "Description", tfgld011.desc, The number of input arguments can vary.					
	The DocumentID is mandatory as "DocumentID", "DocumentID",						
	The maximum composed Document If the "ActionCode" is not se If "DisplayID" is not set the DisplayID.	=					
	To create a user area node a UserAreaValue are mandatory. UserAreaStartDate and UserArea> In the function call, the Allowed values for UserAreaTy	aEndDate can also be used. same order must be used!					

```
"DateTimeType"
    - "IndicatorType"
    - "NumericType"
    - "StringType"
If an xml elementname is not in the list of possible
xml elementnames, the xml element value is added to the
"MessageTypeDetails" node, e.g.:
input arguments:
        "MessageType"
                               "TestType",
                              field.1,
        "Element.1"
        "Element.2"
                               field.2,
result node:
        <MessageTypeDetails>
                <TestTypeGroup>
                       <Element.1>field.1</Element.1>
                       <Element.2>field.2</Element.2>
                </TestTypeGroup>
        </MessageTypeDetails>
xml elementvalues should apply to the following rules:
utc dates should be converted to string representation in ISO.
e.g.: "EffectiveDateTime", utc.to.iso(datefield, UTC ISO Z)
Use the english description associated with a specific value
in an enumerated domain.
e.g.: "TransactionCategory", enum.descr$("tfgld.catg",
                                            tfgld011.catg,
                                            "2")
Attributes of an xml element are reflected using an @ and
must be placed directly under the xml element on which they
are related, e.g.:
        "PurchaseInvoiceAmount",
                                               "10.000",
        "PurchaseInvoiceAmount@currencyID",
                                              "EUR",
        <PurchaseInvoiceAmount currencyID="EUR">10.000
                                        </PurchaseInvoiceAmount>
Possible xml element names:
    - "ActionCode"
    - "Authorization"
    - "Description"
    - "DocumentReference"
    - "DocumentReference ID"
    - "DocumentReference LineNumber"
    - "DocumentReference ScheduleLineNumber"
    - "DisplayID"
      "DistributionGroup_Contact"
    - "DistributionGroup_ContactGroup"
    - "DistributionGroup Email"
    - "DistributionGroup_Person"
    - "DistributionGroup_PersonGroup"
    - "DocumentID"
    - "Message Description"
    - "Message ID"
    - "MessageType"
      "Status Code"
      "Status EffectiveDateTime"
       "Status ArchiveIndicator"
    - "UserAreaDescription"
    - "UserAreaEndDate"
    - "UserAreaName"
```

- "UserAreaStartDate"

- "UserAreaType"

- "UserAreaValue"

Pre: Transaction handling is needed.

Post: Transaction handling is needed: A commit or abort must be done.

Input: iMessageType: Mandatory

Variable arguments are pairs of two arguments with

name and value. At least one argument with name "DocumentID"

is Mandatory.

Output: oExceptionMessage $\,\,$ - The last message if the return

value is not equal to 0.
If more than one message is
given, these are present in the

oExceptionID

oExceptionID - An ID that refers to all error information. Use the functions in

Exception to get all relevant

information.

Return: 0 - BOD is published.

DALHOOKERROR - Error.