Infor LN Public Interfaces Reference Guide (On-premises)

Release 10.5.2

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 KB2924522. 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:

1 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    tncmp         iFinancialCompany,
            domain    tcamnt        iSourceAmount,
            domain    tcccur        iSourceCurrency,
            domain    tcrtyp        iExchangeRateType,
            domain    tcdate        iRateDateUTC,
            domain    tcccur        iTargetCurrency,
    ref     domain    tcamnt        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"

                 long          exception.id, i
        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 available from KB2040021.
Syntax:	`Exception.Delete(` `ref long ioExceptionID )`

Usage:	`Expl: This function deletes the exception to free memory.`
	`Pre: ioExceptionID should refer to an Exception.`
	`Post: None`
	`Input:`
	`ioExceptionID - the exception id.`
	`Output:`
	`ioExceptionID - the exception id.`
	`Return: None`

Exception.GetMessage

DLL:	tcextextapi

	This function is available from KB2040021.

Syntax:	`Exception.GetMessage(`
	`  long  iExceptionID,`
	`  long  iMessageIndex,`
	`  ref  domain tcmcs.s999m  oMessageDescription mb )`

Usage:	`Expl:  This function reads message description from all messages in the XML identified by iExceptionID for a certain index.`
	`Pre:  iExceptionID should refer to an Exception.`
	`Post:  None`
	`Input:`
	`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.`
	`Output:`
	`oMessageDescription - The found message.`
	`Return:  None`

Exception.NumberOfMessages

DLL:	tcextextapi This function is available 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:  None`
	`Input:`
	`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.ActionsAfterProcessingIncomingRequest

DLL:	tcextbodapi This function is available from KB2040021.
Syntax:	long BOD.ActionsAfterProcessingIncomingRequest(long iXMLRequest, ref long oResult, ref domain tcmcs.s999m oExceptionMessage mb, ref long oExceptionID)
Usage:	Expl: This function 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.ActionsBeforeProcessingIncomingRequest() function. Preferably, it 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 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.ActionsAfterProcessingIncomingRequestWithAutomaticProcessing

DLL:	tcextbodapi This function is available from KB2040021.
Syntax:	long BOD.ActionsAfterProcessingIncomingRequestWithAutomaticProcessing(long iXMLRequest, string iObjectType(1), string iObject(1), ref long oResult, ref domain tcmcs.s999m oExceptionMessage mb, ref long oExceptionID)
Usage:	Expl: This function 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.ActionsBeforeProcessingIncomingRequest() 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" iObject - 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 available from KB2040021.
Syntax:	long BOD.ActionsBeforeProcessingIncomingRequest(long iXMLRequest, ref boolean oCancel, ref domain tcmcs.s999m oExceptionMessage mb, ref long oExceptionID)
Usage:	Expl: This function validates the incoming request and sets parameters needed during the processing. It validates if the Accounting Entity and the Location (Location is not checked for master data BODs) of the incoming request matches the Accounting Entity and Location of the current company (in Tenant, Accounting Entity and Location setup or in BOD Parameters) and sets output argument oCancel to true in case of a mismatch. It extracts the LastModificationPerson/IDs/ID from the incoming request and determines the linked LN user. If the LN user can be determined, it switches user to the LN user. It sets a parameter that enables the staging of outgoing BODs that are triggered 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 request is processed,
                BOD.ActionsAfterProcessingIncomingRequest() or
                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
                                          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 available from KB2019035.

Syntax:

```
long BOD.ExecuteMethod(
                domain  tcbod.name      iNoun,
                domain  tcmcs.str30     iMethod,
                        long            iXMLRequest,
        ref             long            oXMLResponse,
        ref             long            oXMLResult,
        ref     domain  tcmcs.s999m     oExceptionMessage mb,
        ref             long            oExceptionID )
```

Usage:

```
        Expl:   This function executes a non-batch method for a specified
                BOD.
        Pre:    NA
        Post:   NA
        Input:  iNoun           - The (protected) Noun for which the method
                                  must be executed, e.g.
                                  "ReceiveDeliveryWarehousingBOD". Mandatory
                iMethod         - The method to be executed,
                                  e.g. "Create" or "Change". 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 values:
                0                       - method is executed successfully
                DALHOOKERROR            - on errors
```

BOD.ExecuteOnAcknowledgeForChameleon

DLL: tcextbodapi

This function is available from KB2019035.

Syntax:

```
long BOD.ExecuteOnAcknowledgeForChameleon(
                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 OnAcknowledge method of a protected
                noun of the chameleon BOD and can be called from the
                OnExecuteHook of the OnAcknowledge method of an incoming public
                noun. The function is used to route an incoming BOD with
                Acknowledge verb to a protected noun.
        Pre:    NA
        Post:   NA
        Input:  iProtectedNoun  - The (protected) Noun for which the
                                  OnAcknowledge 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.ExecuteOnLoadForChameleon

DLL: tcextbodapi

This function is available from KB2019035.

Syntax:
```
long BOD.ExecuteOnLoadForChameleon(
                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 OnLoad method of a protected noun
                of the chameleon BOD and can be called from the OnExecuteHook
                of the OnLoad method of an incoming public noun. The function
                is used to route an incoming BOD with Load verb to a protected
                noun.
        Pre:    NA
        Post:   NA
        Input:  iProtectedNoun  - The (protected) Noun for which the OnLoad
                                  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.ExecuteOnProcessForChameleon

DLL: tcextbodapi

This function is available from KB2019035.

Syntax:
```
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 function executes the OnProcess method of a protected noun
                of the chameleon BOD and 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.
        Pre:    NA
        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 available from KB2019035.

Syntax:

```
long BOD.ExecuteOnSyncForChameleon(
                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 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:  iProtectedNoun  - The (protected) Noun for which the OnSync
                                  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.ExecuteOnUpdateForChameleon

DLL:	tcextbodapi This function is available from KB2019035.
Syntax:	long BOD.ExecuteOnUpdateForChameleon(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 OnUpdate method of a protected noun of the chameleon BOD and can be called from the OnExecuteHook of the OnUpdate method of an incoming public noun. The function is used to route an incoming BOD with Update verb to a protected noun. Pre: NA Post: NA Input: iProtectedNoun - The (protected) Noun for which the OnUpdate 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.ExecutePublishForChameleon

DLL: tcextbodapi

This function is available from KB2019035.

Syntax:
```
long BOD.ExecutePublishForChameleon(
                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 publishing for a protected noun
                of the chameleon BOD and can be called from the OnExecuteHook
                of the PublishEvent method of a protected noun.
                It translates the protected noun to the public noun and
                executes the PublishEvent method of the public noun
        Pre:    NA
        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 PublishEvent
                                  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.ExecuteShowForChameleon

DLL: tcextbodapi

This function is available from KB2019035.

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 function executes the Show method of the public noun
                of the chameleon BOD 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 XML to the response XML of the protected noun
        Pre:    NA
        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

This function is available from KB1987704.

Syntax:

```
long BOD.GetIdAccountingEntity(
                domain  tcncmp          iCompany,
                domain  tcbod.name      iNoun,
                        long            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.
        Pre:    NA
        Post:   NA
        Input:  iCompany        - Company for which the accounting
                                  entity is determined. Mandatory
                iNoun           - Noun: Mandatory
                iEntityType     - Entity Type: Mandatory for transactional data BODs
                                  Possible values: 1 (Warehouse), 2 (Department),
                                  3 (Project)
                iEntityCode     - Entity Code: Mandatory for transactional data BODs
                                  Possible values: The warehouse, department or
                                  project.
                                  The Entity Type and Entity Code are used to
                                  determine the Tenant, AccountingEntity and
```

```
                                Location.
                iRootTable      - Root Table: Mandatory for master data BODs
        Output: oAccountingEntity       - Accounting Entity
                oAccountingEntityisSet  - Accounting Entity set (true or false).
                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                       - Accounting Entity is determined.
                DALHOOKERROR            - Otherwise.
```

BOD.GetIdLocation

DLL: tcextbodapi

This function is available from KB1987704.

Syntax:

```
long BOD.GetIdLocation(
                domain  tcncmp          iCompany,
                domain  tcbod.name      iNoun,
                        long            iEntityType,
        const           string          iEntityCode(),
                domain  tcmcs.tabl      iRootTable,
        ref     domain  tcbod.lctn      oLocation,
        ref             boolean         oLocationisSet,
        ref     domain  tcmcs.s999m     oExceptionMessage mb,
        ref             long            oExceptionID )
```

Usage:

```
        Expl:   This function determines the Location.
        Pre:    NA
        Post:   NA
        Input:  iCompany        - company for which the location
                                  is determined. Mandatory
                iNoun           - Noun: Mandatory
                iEntityType     - Entity Type: Mandatory for transactional data BODs
                                  Possible values: 1 (Warehouse), 2 (Department),
                                  3 (Project)
                iEntityCode     - Entity Code: Mandatory for transactional data BODs
                                  Possible values: The warehouse, department or
                                  project.
                                  The Entity Type and Entity Code are used to
                                  determine the Tenant, AccountingEntity and
                                  Location.
                iRootTable      - Root Table: Mandatory for master data BODs
        Output: oLocation               - Location
                oLocationisSet          - Location set (true or false).
                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 - Location is determined. DALHOOKERROR - Otherwise.

BOD.GetIdLogicalID

DLL:	tcextbodapi This function is available from KB1987704.
Syntax:	long BOD.GetIdLogicalID(domain tcncmp iCompany, domain tcbod.name iNoun, long iEntityType, const string iEntityCode(), domain tcmcs.tabl iRootTable, ref domain tcbod.loid oLogicalID, ref boolean oLogicalIDisSet, ref domain tcmcs.s999m oExceptionMessage mb, ref long oExceptionID)
Usage:	Expl: This function determines the Logical ID. Pre: NA Post: NA Input: iCompany - company for which the logical Id is determined. Mandatory iNoun - Noun: Mandatory iEntityType - Entity Type: Mandatory for transactional data BODs Possible values: 1 (Warehouse), 2 (Department), 3 (Project) iEntityCode - Entity Code: Mandatory for transactional data BODs Possible values: The warehouse, department or project. The Entity Type and Entity Code are used to determine the Tenant, AccountingEntity and Location. iRootTable - Root Table: Mandatory for master data BODs Output: oLogicalId - LogicalID oLogicalIdisSet - LogicalID set (true or false). 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 - LogicalId is determined. DALHOOKERROR - Otherwise.

BOD.HandleStagingAfterProcessingIncomingRequest

DLL:	tcextbodapi This function is available from KB2040021.
Syntax:	long BOD.HandleStagingAfterProcessingIncomingRequest(domain tcbod.name iNoun, ref domain tcmcs.s999m oExceptionMessage mb, ref long oExceptionID)
Usage:	Expl: This function 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.HandleStagingBeforeProcessingIncomingRequest() function. Preferably, it 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.HandleStagingBeforeProcessingIncomingRequest() 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 oExceptionID oExceptionID - An ID that refers to all error information. Use the functions in Exception to get all relevant information. Return: 0 / DALHOOKERROR

BOD.HandleStagingBeforeProcessingIncomingRequest

DLL:	tcextbodapi This function is available from KB2040021.
Syntax:	long BOD.HandleStagingBeforeProcessingIncomingRequest(domain tcbod.name iNoun, ref domain tcmcs.s999m oExceptionMessage mb, ref long oExceptionID)
Usage:	Expl: This function sets a parameter that enables the staging of outgoing BODs that are triggered 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
        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
                                          oExceptionID
                oExceptionID            - An ID that refers to all error
                                          information. Use the functions in
                                          Exception to get all relevant
                                          information.
        Return: 0 / DALHOOKERROR
```

BOD.Publish

DLL: tcextbodapi

This function is available from KB1987704.

Syntax:

```
long BOD.Publish(
                domain  tcbod.name      iNoun,
                domain  tcmcs.tabl      iRootTable,
        const           string          iActionCode(),
                        long            iEntityType,
        const           string          iEntityCode(),
        const           string          iDocumentId(),
        const           string          iProcessingAction(),
        ref     domain  tcmcs.s999m     oExceptionMessage mb,
        ref             long            oExceptionID,
                                        ... )
```

Usage:

```
        Expl:   This function publishes any BOD.
        Pre:    Transaction handling is needed.
        Post:   Transaction handling is needed: A commit or abort must be done.
        Input:  iNoun           - Noun: Mandatory
                iRootTable      - Root Table: Mandatory for master data BODs
                iActionCode     - Action Code: Mandatory
                                  Possible values: "Add", "Change", "Delete",
                                  "Replace", "Canceled".
                iEntityType     - Entity Type: Mandatory for transactional data BODs
                                  Possible values: 1 (Warehouse), 2 (Department),
                                  3 (Project)
                iEntityCode     - Entity Code: Mandatory for transactional data BODs
                                  Possible values: The warehouse, department or
                                  project.
                                  The Entity Type and Entity Code are used to
                                  determine 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:

```
long BOD.PublishLNMessage(
                domain  tcmcs.str30m     iMessageType mb,
        ref     domain  tcmcs.s999m      oExceptionMessage mb,
        ref             long             oExceptionID,
                                         ... )
```

Usage:

```
        Expl:   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".

                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 one of the xml_element names:
                        "DocumentID",           table.key1,
                        "DocumentID",           table.key2,

                The maximum composed DocumentID_ID string length is 132.
                If the "ActionCode" is not set, the default "Add" is used.
                If "DisplayID" is not set the "DocumentID" is used to create the
                DisplayID.

                To create a user area node a UserAreaName, UserAreaType and
                UserAreaValue are mandatory. UserAreaDescription,
                UserAreaStartDate and UserAreaEndDate can also be used.
                --> In the function call, the same order must be used!

                Allowed values for UserAreaType are:
```

```
    -  "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",
        "Element.1"             field.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.
```