PeopleSoft®

EnterpriseOne Xe
Asynchronous Messaging Adapter
Programmer’s Guide PeopleBook

June 2000

J.D. Edwards World Source Company
7601 Technology Way
Denver, CO 80237

Portions of this document were reproduced from material prepared by J.D. Edwards.

SKU XeEAMQ

The information in this guide is confidential and a proprietary trade secret of J.D. Edwards World Source Company. It may not be copied, distributed, or disclosed without prior written permission. This guide is subject to change without notice and does not represent a commitment on the part of J.D. Edwards & Company and/or its subsidiaries. The software described in this guide is furnished under a license agreement and may be used or copied only in accordance with the terms of the agreement. J.D. Edwards World Source Company uses automatic software disabling routines to monitor the license agreement. For more details about these routines, please refer to the technical product documentation.

Table of Contents

Overview

Adapter Overview

Architecture

The OneWorld Asynchronous Messaging Adapter allows you to connect any e-storefront or other application to OneWorld by sending and receiving messages through the message adapter messaging system. The adapter monitors an inbound queue for Request/Reply messages, performs the requested services, and places the results on outbound queues. It also monitors OneWorld for certain activities and publishes the results in an outbound message queue. All messages transported through the messaging system are in the form of XML documents.

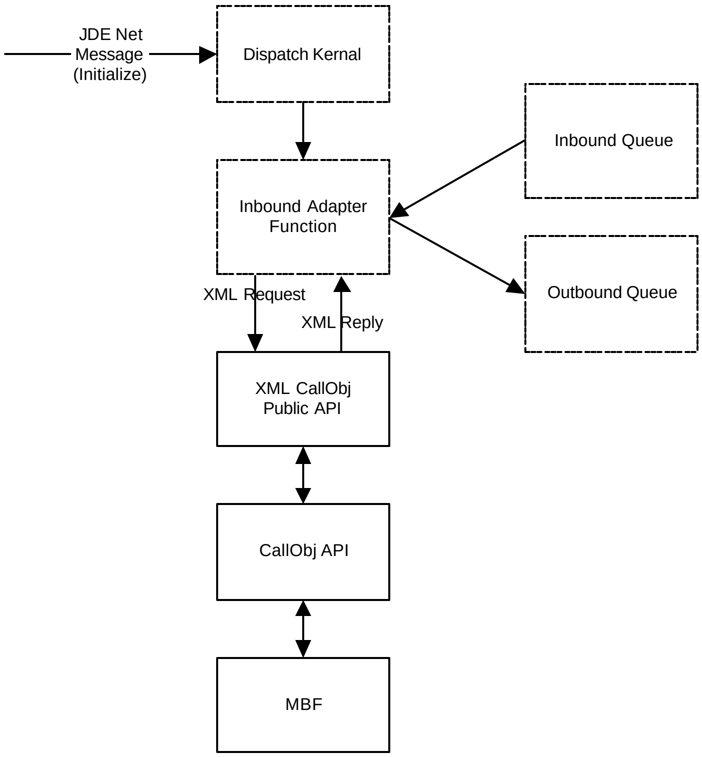
The adapter design is such that it avoids any program-level dependencies on the any particular suite of products. The OneWorld Asynchronous Messaging Adapter is a separate product that can be licensed and installed independently. It can be used to connect OneWorld with any front end that can implement the messaging protocols, and provide/consume XML documents in the prescribed formats. The OneWorld Asynchronous Messaging Adapter is simply designed to export and import XML documents through the messaging system in the prescribed formats.

The J.D. Edwards messaging solution allows you to use multiple adapter processes. This provides better scalability and helps ensure that if an adapter process fails, the failure does not affect OneWorld.

Refer to the appropriate Adapter Installation Guide for more information about setting up the particular adapter you want to use.

Inbound Transactions

The following diagram depicts the inbound process, which uses the XML Call Object methodology for processing inbound transactions such as a Sales Order. The adapter provides a layer over pre-existing functionality. The components of the adapter are indicated in the diagram by dashed lines.



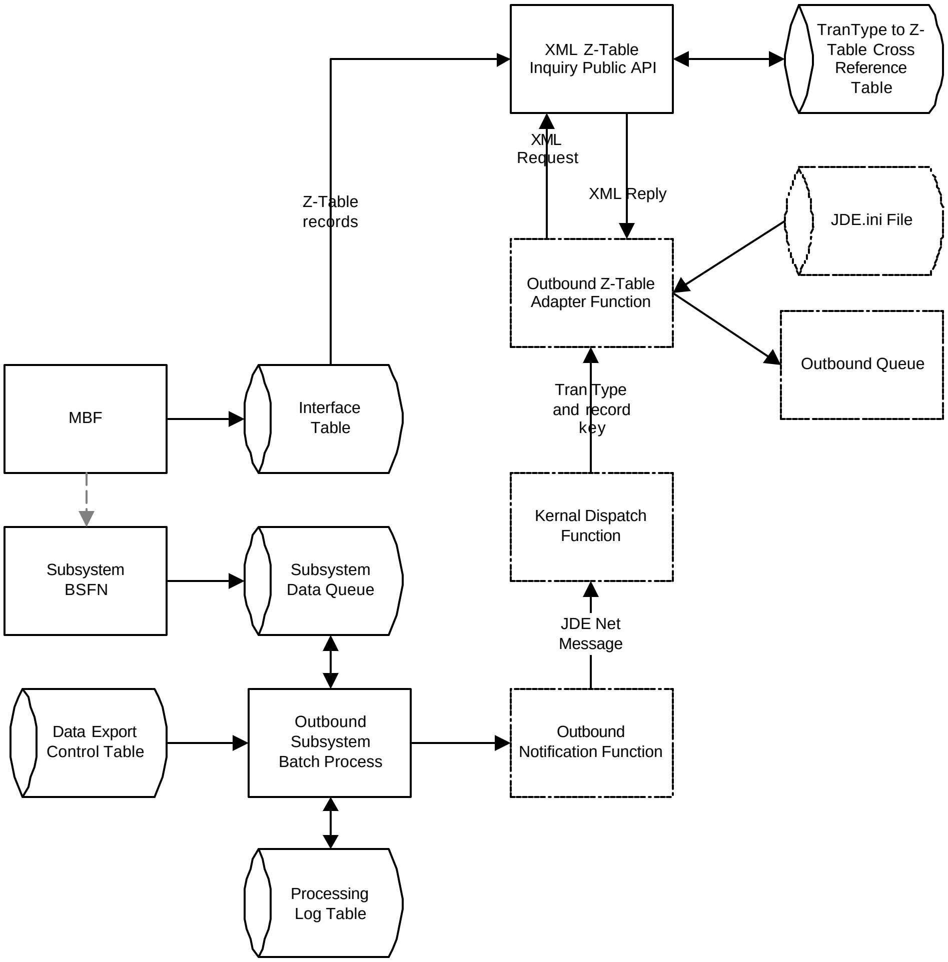
Inbound Process Flow

Typically, inbound processing for XML transactions follows a flow similar to the following:

- The adapter picks up a message in XML format from the message queue.
- The XML document is passed into the jdeXMLCallObject API.
- The OneWorld server processes the message by parsing the XML document. The Session Manager also validates user and password.
- Each requested business function is called separately or within requested transaction boundaries until all calls are processed.
- Transactions are added to the OneWorld database.
- Output data and error messages are merged back into the XML document and a new response document is created.
- The OneWorld adapter puts the response XML document in the queue based on error or success.

Outbound Transactions

The following diagram depicts the outbound process, which uses the Z-Table asynchronous methodology for passing a variety of information to the attached system. The adapter provides a layer over pre-existing functionality. The components of the adapter are indicated in the diagram by dashed lines



Outbound Process Flow

Typically, outbound processing for XML transactions follows a flow similar to the following:

- An outbound message is triggered by an event, for example entry of a sales order.
- Subsystem processing starts processing the transaction and calls the outbound notification function.

  Refer to the Interoperability guide for more information about subsystem processing.
- The outbound notification function sends a net message and the kernel picks it up and call the Z-Table outbound adapter..
- The asynchronous messaging adapter reads the message and calls the jdeRetrieveTransactionInfo API.
- The adapter sends the record key from the net message in the XML request document.
- An XML response document is created.
- The XML document is placed in the outbound queue.

Working with XML Formats

XML CallObject

The following information explains how CallObject works and details the steps involved with using XML CallObject.

Establish Session

A session must be established. This step is addressed by the *session* attribute of the standard *jdeRequest* element. If the *session* attribute is an empty string, this indicates that the user wants a session started. On the server the *SessionManager* singleton class creates a new instance of a *Session* object given the user name, password and environment name.

```
<?xml version='1.0' ?>

<jdeRequest type='callmethod'
user='steve' pwd='xyz'
environment='prod' session=''
sessionidle='1800'>
```

Expire Session

Session expiration is addressed by the *sessionidle* attribute of the standard *jdeRequest* element. This attribute, when given on a session creation request, specifies the amount of time in seconds that this session is allowed to be idle. If the *SessionManager* determines that a session has not had any requests processed in this amount of time, it terminates the session and frees all associated resources.

The session idle default is 30 minutes. You can use a jde.ini file setting to change the default session idle time.

```
<?xml version='1.0' ?>

<jdeRequest type='callmethod'
user='steve' pwd='xyz'
environment='prod' session=''
sessionidle='1800'>
```

Call Object

Tags are used to call business functions on the server

```
<?xml version='1.0' ?>

<jdeRequest type='callmethod' user='steve'
pwd='xyz' environment='prod'>

<callMethod name='myfunc' app='P42101'>

<params>

<param name='CostCtr'>      1001</param>

<param name='ExpDate'>1999/10/31</param>

<param name='Quantity'>12</param>

</params>

</callMethod>

</jdeRequest>
```

The *callMethod* element details which function to call and in what context it is being called. The *name* attribute specifies which business function to call and the *app* attribute allows the business function to know “who” is calling it.

The *params* and *param* elements define the data structure of the business function. Each *param* element describes one data structure member. The caller is only required to give the *name* attribute.

If no *param* element value is given for an input data structure member, then the value will be treated as if it were NULL or zero.

Explicit Transaction

Explicit DB transactions are supported by another element, the *startTransaction* tag. This element specifies whether transactions are going to be manual or automatically committed. The *startTransaction* element is an empty element, meaning that all of its information is in the attributes.

```
<?xml version='1.0' ?>
<jdeRequest type='callmethod' user='steve'
pwd='xyz' environment='prod' session=''>
<startTransaction trans='t1' type='manual'/>
</jdeRequest>
```

Implicit Transaction

A CallObject request is included in a transaction set when the name of a transaction set is referenced in its *trans* attribute. Implicit start transactions can be included in the CallObject request by specifying the name of a transaction set that has not previously been created. If it is an implicit start, the transaction set will be a manual commit set.

```
<?xml version='1.0' ?>
<jdeRequest type='callmethod' user='steve'
pwd='xyz' environment='prod' session=''>
<callMethod name='myfunc' app='P42101' trans='t1'>
<params>
<param name='CostCtr'>      1001</param>
</params>
</callMethod>
</jdeRequest>
```

Prepare/Commit/Rollback

Manual transaction sets can be committed or rolled back. As part of a two phase commit, they can be prepared to commit. These requests to the database are made using the *endTransaction* element. The transaction set is identified by the *trans* attribute and the action attribute indicates the action to take on the transaction set. The value can be ‘prepare’, ‘commit’, or ‘rollback’. This element is always an empty element, as shown by the forward slash.

```
<?xml version='1.0' ?>
<jdeRequest type='callmethod' user='steve'
pwd='xyz' environment='prod' session=''>
<endTransaction trans='t1' action='commit'/>
</jdeRequest>
```

Terminate Session

Session termination is done by the message adapter dispatch function.

Call Object Error Handling

System errors on a call object are reported in the *returnCode* element. The numeric code is returned in the *code* attribute and the corresponding text is returned as a child text node of the *returnCode* element. The standard jdeCallObject return codes are used for the *code* attribute.

```
<?xml version='1.0' ?>
<jdeResponse type='callmethod' user='steve'
pwd='xyz' environment='prod' session=''>
<callMethod name='myfunc' app='P42101' trans='t1'>
<params>
<param name='CostCtr'>      1001</param>
</params>
<returnCode code=’0’>Success</returnCode>
</callMethod>
</jdeResponse>
```

BSFN Error Handling

Business function error handling includes using error text, errors during multiple requests, and errors that occur during call method requests.

Error Text

Business function error messages are returned in the *errors* element. Within that element there can be zero or more *error* elements, containing a *code* attribute for the error code and a child text node containing the error text. The *name* attribute describes which *param* element the error refers to.

```
<?xml version='1.0' ?>
<jdeResponse type='callmethod' user='steve'
pwd='xyz' environment='prod' session=''>
<callMethod name='myfunc' app='P42101' trans='t1'>
<params>
<param name='CostCtr'>      1001</param>
</params>
<returnCode code='2'>Errors</returnCode>
<errors>
  <error code='192' name='CostCtr'>Cost Center not
valid</error>
</errors>
</callMethod>
```

Multiple Requests per Document

Multiple requests can be included in the XML document. By default, requests are not run if there have been any errors on previous requests. If a request should be run even if errors have occurred, then the default behavior can be overridden by using the *runOnError* attribute on the request with a value of *‘yes’*.

```
<?xml version='1.0' ?>
<jdeRequest type='callmethod' user='steve'
pwd='xyz' environment='prod' session=''>
<callMethod name='myfunc' app='P42101' trans='t1'
runOnError=’yes’>
<params>
<param name='CostCtr'>      1001</param>
</params>
</callMethod>
</jdeRequest>
```

On Error Handling

If an error occurs on a *callMethod* request you can add an onError element to the request to take some action, such as calling another business function, to clean up resources. The onError tag can specify an *abort* attribute that specifies if all subsequent requests should be skipped. The allowed values are *‘yes’* or *‘no’*. A “global” onError tag can be specified as a child of the jdeRequest tag, that will be executed if there were errors encountered and no other onError tag with *abort=*’yes’ was executed. The global onError tag should be the last request in the document.

```
<?xml version='1.0' ?>
<jdeRequest type='callmethod' user='steve'
pwd='xyz' environment='prod' session=''>
<callMethod name='myfunc' app='P42101' trans='t1'
runOnError=’yes’>
<params>
<param name='CostCtr'>      1001</param>
</params>
<onError abort=’no’>
     <endTransaction trans=’t1’
action=’rollback’/>
</onError>
</callMethod>
</jdeRequest>
```

ID/IDREF Support

ID type attributes uniquely identify, by a string value, elements in a XML document. IDREF attributes allow other elements to reference the specified element. An IDREF attribute must not be used in a document before the ID it references is defined.

A param element can specify an *id* attribute so that its output value from the callMethod request will be saved and referred to later in another param element by an *idref* attribute. If a param element contains an *idref* attribute, the value of the given parameter is used as the input value for the param element. For example, the output value from referenced parameter is used instead of the value in the XML.

You can specify a special request tag called returnParams that can contain one or more param elements. If the param elements contain *idref* attributes, then the referenced values are copied into the response.

```
<?xml version='1.0' ?>
<jdeRequest type='callmethod' user='steve'
pwd='xyz' environment='prod' session=''>
<callMethod name='myfunc' app='P42101' trans='t1'
runOnError=’yes’>
<params>
<param name='CostCtr'>      1001</param>
<param name='Company1' id='1'></param>
<param name='Company2' id='2'></param>
</params>
</callMethod>
<callMethod name='myfunc2' app='P42101' trans='t1'
runOnError=’yes’>
<params>
<param name='Company1' idref='1'></param>
</params>
<returnParams><param idref='2'/></returnParams>
</callMethod>
</jdeRequest>
```

Return NULL Values

By default if a parameter was not specified in the request document, it will not be returned in the response document unless its value is non-blank or non-zero. This behavior can be modified by specifying the *returnNullData* attribute on the callMethod element with a value of *‘yes’*.

```
<?xml version='1.0' ?>
<jdeRequest type='callmethod' user='' pwd=''
environment='prod' session=''>
<callMethod name='myfunc' app='P42101'
returnNullData='yes'>
<params>
     <param name='CostCtr'></param>
     <param name='ExpDate'></param>
     <param name='Quantity'></param>
</params>
</callMethod>
</jdeRequest>
```

Enabling Outbound Z-Table Processes

Outbound Table Adapter

You use the *OutboundZTableAdapter* function to send a message from an outbound Z-Table to an asynchronous messaging adapter queue. The function is invoked from the kernel dispatch function, which then sends the net message data that contains the parameters from the Z-Table subsystem UBE.

```
Void OutboundZTableMessageAdapter(MsgData
*pMsgData)
```

These parameters define the records and the transaction type that will be used to cross reference the tables that contain the data to populate the message to be sent to the message adapter queue. The asynchronous messaging-specific Outbound Z-Table Adapter parses the net message data and calls the jde XML Z-Table Inquiry API to fetch the records from the Z-Table and format the results into an XML string.

You must setup OneWorld to initiate the outbound Z-Table process. The format of the outbound Z-Table message retains an XML based format.

Outbound Notification

The business function sends an outbound notification message to JDENET. This function is called by the standard generic Outbound Subsystem batch process UBE and provides notification that records have been placed in the Z-Tables.

This function passes the key fields for a record in the OneWorld Outbound Transaction Z-Tables, to the outbound adapter. With these keys, you can process the information from the database record into a message queue.

```
void MessageNotificationName(char *szUserID, char
*szBatchNumber,char *szTransactionNumber, double
mnLineNumber,char *szTransactionType, char
*szDocumentType, double mnSequenceNumber )
```

Following is the required parameter list. All parameters are input:

User Id - 11 characters

Batch Number – 16 characters

Transaction Number – 23 characters

Line Number – double

Transaction Type – 9 characters

Document Type – 3 characters

Sequence Number – double

This information is sent on in a JDENET message with the following message packets:

Environment name - use JDE APIs to retrieve environment from the Subsystem batch process.

User Id - key to Z-Table record

Batch Number - key to Z-Table record

Transaction Number - key to Z-Table record

Line Number - key to Z-Table record

Transaction Type - tie to which Z-Table

Document Type - (optional)

Sequence Number - (optional)

The key information in the Net message packets is used by the Outbound Adapter to retrieve the Z-file record from the table. The transaction type allows the adapter to be generic and process other transactions in the future. The transaction type maps to the Flat File Cross Reference Table (F47002) to determine the Z-Tables.

XML Z-Table Inquiry API

The XML Z-Table Inquiry API (jdeRetrieveTransactionInfo) receives an XML string that includes the table record key and returns an XML string for outbound.

This function is called from the Asynchronous Messaging Specific Outbound Z-Table Adapter. It parses the XML string and, based on the transaction type, goes to the F47002 (Flat File Cross Reference Table) to determine from which Z-Table to fetch records. F47002 has a record for each table associated with the transaction type. Using JDB database APIs, XML Z-Table Inquiry then uses the index found in the XML string to fetch records from the Z-Table and returns the results in an XML string.

Refer to the Interoperability Guide and the online Interoperability Interface Tables for more information about specific Z-Tables.

XML Transaction Information Request

The XML transaction information request is created by the outbound adapter and sent to the XML transaction API.

Outbound Order Status XML Request & Response Format

“This format will return all columns for the sales order header (F4201Z1) and detail lines (F4211Z1). “

```
 <?xml version=’1.0’ ?>
<jdeRequest type=’trans’ user=’user’ pwd=’password’ environment=’environment’
session=’
‘ sessionidle=’300’>
      <transaction action=’transactionInfo’ type=’JDESOOUT’>
      <key>
<column name=’EdiUserId’>value</column>
<column name=’EdiBatchNumber’>value</column>
<column name=’EdiTransactNumber’>value</column>
      </key>
   </transaction>
  </jdeRequest>
```

```
<?xml version='1.0' encoding='utf-8' ?>
<jdeResponse type='trans' user='user' session='session1' environment='env'>
   <transaction type='JDESOOUT' action='transactionInfo'>
            <returnCode code='0'>XML Request OK</returnCode>
            <key>
                     <column name='EdiUserId'></column>
                     <column name='EdiBatchNumber'></column>
                     <column name='EdiTransactNumber'></column>
            </key>
            <table name='F4201Z1' type='header'>
                     <column name='EdiUserId'></column>
                     <column name='EdiBatchNumber'></column>
                     <column name='EdiTransactNumber'></column>
                     <column name='EdiLineNumber'></column>
```

```
<column name='EdiDocumentType'></column>
<column name='TypeTransaction'></column>
<column name='EdiTranslationFormat'></column>
<column name='EdiTransmissionDate'></column>
<column name='DirectionIndicator'></column>
<column name='EdiDetailLinesProcess'></column>
<column name='EdiSuccessfullyProcess'></column>
<column name='TradingPartnerId'></column>
<column name='TransactionAction'></column>
<column name='CompanyKeyOrderNo'></column>
<column name='DocumentOrderInvoiceE'></column>
<column name='OrderType'></column>
<column name='OrderSuffix'></column>
<column name='CostCenter'></column>
<column name='Company'></column>
<column name='CompanyKeyOriginal'></column>
<column name='OriginalPoSoNumber'></column>
<column name='OriginalOrderType'></column>
<column name='CompanyKeyRelated'></column>
<column name='RelatedPoSoNumber'></column>
<column name='RelatedOrderType'></column>
<column name='AddressNumber'></column>
<column name='AddressNumberShipTo'></column>
<column name='AddressNumberParent'></column>
<column name='DateRequestedJulian'></column>
<column name='DateTransactionJulian'></column>
<column name='PromisedDeliveryDate'></column>
<column name='DateOriginalPromisde'></column>
<column name='ActualDeliveryDate'></column>
<column name='CancelDate'></column>
<column name='DatePriceEffectiveDate'></column>
<column name='DatePromisedPickJu'></column>
<column name='DatePromisedShipJu'></column>
<column name='Reference1'></column>
```

```
<column name='Reference2Vendor'></column>
<column name='DeliveryInstructLine1'></column>
<column name='DeliveryInstructLine2'></column>
<column name='PrintMessage1'></column>
<column name='PaymentTermsCode01'></column>
<column name='PaymentInstrumentA'></column>
<column name='PriceAdjustmentScheduleN'></column>
<column name='PricingGroup'></column>
<column name='DiscountTrade'></column>
<column name='PercentRetainage1'></column>
<column name='TaxArea1'></column>
<column name='TaxExplanationCode1'></column>
<column name='CertificateTaxExempt'></column>
<column name='AssociatedText'></column>
<column name='PriorityProcessing'></column>
<column name='BackordersAllowedYN'></column>
<column name='SubstitutesAllowedYN'></column>
<column name='HoldOrdersCode'></column>
<column name='PricePickListYN'></column>
<column name='InvoiceCopies'></column>
<column name='NatureOfTransaction'></column>
<column name='BuyerNumber'></column>
<column name='Carrier'></column>
<column name='ModeOfTransport'></column>
<column name='ConditionsOfTransport'></column>
<column name='RouteCode'></column>
<column name='StopCode'></column>
<column name='ZoneNumber'></column>
<column name='ContainerID'></column>
<column name='FreightHandlingCode'></column>
<column name='ApplyFreightYN'></column>
<column name='ApplyFreight'></column>
<column name='FreightCalculatedYN'></column>
<column name='MergeOrdersYN'></column>
```

```
            <column name='CommissionCode1'></column>
            <column name='RateCommission1'></column>
            <column name='CommissionCode2'></column>
            <column name='RateCommission2'></column>
            <column name='ReasonCode'></column>
            <column name='PostQuantities'></column>
            <column name='AmountOrderGross'></column>
            <column name='AmountTotalCost'></column>
            <column name='UnitOfMeasureWhtDisp'></column>
            <column name='UnitOfMeasureVolDisp'></column>
            <column name='AuthorizationNoCredit'></column>
            <column name='AcctNoCrBank'></column>
            <column name='DateExpired'></column>
            <column name='SubledgerInactiveCode'></column>
            <column name='CorrespondenceMethod'></column>
            <column name='CurrencyMode'></column>
            <column name='CurrencyCodeFrom'></column>
            <column name='CurrencyConverRateOv'></column>
            <column name='LanguagePreference'></column>
            <column name='AmountForeignOpen'></column>
            <column name='AmountForeignTotalC'></column>
            <column name='OrderedBy'></column>
            <column name='OrderTakenBy'></column>
            <column name='UserReservedCode'></column>
            <column name='UserReservedDate'></column>
            <column name='UserReservedAmount'></column>
            <column name='UserReservedNumber'></column>
            <column name='UserReservedReference'></column>
            <column name='UserId'></column>
            <column name='ProgramId'></column>
            <column name='WorkStationId'></column>
            <column name='DateUpdated'></column>
            <column name='TimeOfDay'></column>
        </table>
        <table name='F4211Z1' type='detail'>
            <column name='EdiUserId'></column>
            <column name='EdiBatchNumber'></column>
            <column name='EdiTransactNumber'></column>
            <column name='EdiLineNumber'></column>
            <column name='EdiDocumentType'></column>
            <column name='TypeTransaction'></column>
            <column name='EdiTranslationFormat'></column>
            <column name='EdiTransmissionDate'></column>
            <column name='DirectionIndicator'></column>
            <column name='EdiDetailLinesProcess'></column>
            <column name='EdiSuccessfullyProcess'></column>
            <column name='TradingPartnerId'></column>
            <column name='TransactionAction'></column>
            <column name='CompanyKeyOrderNo'></column>
            <column name='DocumentOrderInvoiceE'></column>
            <column name='OrderType'></column>
            <column name='LineNumber'></column>
            <column name='OrderSuffix'></column>
            <column name='CostCenter'></column>
            <column name='Company'></column>
            <column name='CompanyKeyOriginal'></column>
            <column name='OriginalPoSoNumber'></column>
            <column name='OriginalOrderType'></column>
            <column name='OriginalLineNumber'></column>
            <column name='CompanyKeyRelated'></column>
            <column name='RelatedPoSoNumber'></column>
            <column name='RelatedOrderType'></column>
            <column name='RelatedPoSoLineNo'></column>
            <column name='ContractNumberDistributi'></column>
            <column name='ContractSupplementDistri'></column>
            <column name='ContractBalancesUpdatedY'></column>
            <column name='AddressNumber'></column>
            <column name='AddressNumberShipTo'></column>
```

```
                    <column name='AddressNumberParent'></column>
                    <column name='DateRequestedJulian'></column>
                    <column name='DateTransactionJulian'></column>
                    <column name='PromisedDeliveryDate'></column>
                    <column name='DateOriginalPromisde'></column>
                    <column name='ActualDeliveryDate'></column>
                    <column name='DateInvoiceJulian'></column>
                    <column name='CancelDate'></column>
                    <column name='DtForGLAndVouch1'></column>
                    <column name='DateReleaseJulian'></column>
                    <column name='DatePriceEffectiveDate'></column>
                    <column name='DatePromisedPickJu'></column>
                    <column name='DatePromisedShipJu'></column>
                    <column name='Reference1'></column>
                    <column name='Reference2Vendor'></column>
                    <column name='IdentifierShortItem'></column>
                    <column name='Identifier2ndItem'></column>
                    <column name='Identifier3rdItem'></column>
                    <column name='Location'></column>
                    <column name='Lot'></column>
                    <column name='FromGrade'></column>
                    <column name='ThruGrade'></column>
                    <column name='FromPotency'></column>
                    <column name='ThruPotency'></column>
                    <column name='DaysPastExpiration'></column>
                    <column name='DescriptionLine1'></column>
                    <column name='DescriptionLine2'></column>
                    <column name='LineType'></column>
                    <column name='StatusCodeNext'></column>
                    <column name='StatusCodeLast'></column>
                    <column name='CostCenterHeader'></column>
                    <column name='ItemNumberRelatedKit'></column>
                    <column name='LineNumberKitMaster'></column>
                    <column name='ComponentNumber'></column>
```

```
                <column name='RelatedKitComponent'></column>
                <column name='NumbOfCpntPerParent'></column>
                <column name='SalesReportingCode1'></column>
                <column name='SalesReportingCode2'></column>
                <column name='SalesReportingCode3'></column>
                <column name='SalesReportingCode4'></column>
                <column name='SalesReportingCode5'></column>
                <column name='PurchasingReportCode1'></column>
                <column name='PurchasingReportCode2'></column>
                <column name='PurchasingReportCode3'></column>
                <column name='PurchasingReportCode4'></column>
                <column name='PurchasingReportCode5'></column>
                <column name='UnitOfMeasureAsInput'></column>
                <column name='UnitsTransactionQty'></column>
                <column name='UnitsQuantityShipped'></column>
                <column name='UnitsQuanBackorHeld'></column>
                <column name='UnitsQuantityCanceled'></column>
                <column name='UnitsQuantityFuture'></column>
                <column name='UnitsOpenQuantity'></column>
                <column name='QuantityShippedToDate'></column>
                <column name='QuantityRelieved'></column>
                <column name='CommittedHS'></column>
                <column name='OtherQuantity12'></column>
                <column name='AmtPricePerUnit2'></column>
                <column name='AmountExtendedPrice'></column>
                <column name='AmountOpen1'></column>
                <column name='PriceOverrideCode'></column>
                <column name='TemporaryPriceYN'></column>
                <column name='UnitOfMeasureEntUP'></column>
                <column name='AmtListPricePerUnit'></column>
                <column name='AmountUnitCost'></column>
                <column name='AmountExtendedCost'></column>
                <column name='CostOverrideCode'></column>
                <column name='ExtendedCostTransfer'></column>
```

```
<column name='PrintMessage1'></column>
<column name='PaymentTermsCode01'></column>
<column name='PaymentInstrumentA'></column>
<column name='BasedonDate'></column>
<column name='DiscountTrade'></column>
<column name='TradeDiscountOld'></column>
<column name='PriceAdjustmentScheduleN'></column>
<column name='PricingCategory'></column>
<column name='PricingCategoryLevel1'></column>
<column name='DiscountFactor'></column>
<column name='DiscountFactorTypeOr'></column>
<column name='DiscntApplicationType'></column>
<column name='DiscountCash'></column>
<column name='CompanyKey'></column>
<column name='DocVoucherInvoiceE'></column>
<column name='DocumentType'></column>
<column name='OriginalDocumentNo'></column>
<column name='OriginalDocumentType'></column>
<column name='DocumentCompanyOriginal'></column>
<column name='PickSlipNumber'></column>
<column name='DeliveryNumber'></column>
<column name='PromationNumber'></column>
<column name='DraftNumber'></column>
<column name='TaxableYN'></column>
<column name='TaxArea1'></column>
<column name='TaxExplanationCode1'></column>
<column name='AssociatedText'></column>
<column name='PriorityProcessing'></column>
<column name='ResolutionCodeBC'></column>
<column name='BackordersAllowedYN'></column>
<column name='SubstitutesAllowedYN'></column>
<column name='PartialShipmntsAllowY'></column>
<column name='LineofBusiness'></column>
<column name='EndUse'></column>
```

```
                    <column name='DutyStatus'></column>
                    <column name='CommodityCode'></column>
                    <column name='NatureOfTransaction'></column>
                    <column name='PrimaryLastVendorNo'></column>
                    <column name='BuyerNumber'></column>
                    <column name='Carrier'></column>
                    <column name='ModeOfTransport'></column>
                    <column name='ConditionsOfTransport'></column>
                    <column name='RouteCode'></column>
                    <column name='StopCode'></column>
                    <column name='ZoneNumber'></column>
                    <column name='ContainerID'></column>
                    <column name='FreightHandlingCode'></column>
                    <column name='ApplyFreightYN'></column>
                    <column name='ApplyFreight'></column>
                    <column name='FreightCalculatedYN'></column>
                    <column name='RateCodeFrieghtMisc'></column>
                    <column name='RateTypeFreightMisc'></column>
                    <column name='ShippingCommodityClass'></column>
                    <column name='ShippingConditionsCode'></column>
                    <column name='SerialNumberLot'></column>
                    <column name='UnitOfMeasurePrimary'></column>
                    <column name='UnitsPrimaryQtyOrder'></column>
                    <column name='UnitOfMeasureSecondary'></column>
                    <column name='UnitsSecondaryQtyOr'></column>
                    <column name='UnitOfMeasurePricing'></column>
                    <column name='AmountUnitWeight'></column>
                    <column name='WeightUnitOfMeasure'></column>
                    <column name='AmountUnitVolume'></column>
                    <column name='VolumeUnitOfMeasure'></column>
                    <column name='RepriceBasketPriceCat'></column>
                    <column name='OrderRepriceCategory'></column>
                    <column name='OrderRepricedIndicator'></column>
                    <column name='InventoryCostingMeth'></column>
```

```
<column name='AllocateByLot'></column>
<column name='GlClass'></column>
<column name='Century'></column>
<column name='FiscalYear1'></column>
<column name='LineStatus'></column>
<column name='SalesOrderStatus01'></column>
<column name='SalesOrderStatus02'></column>
<column name='SalesOrderStatus03'></column>
<column name='SalesOrderStatus04'></column>
<column name='SalesOrderStatus05'></column>
<column name='SalesOrderStatus06'></column>
<column name='SalesOrderStatus07'></column>
<column name='SalesOrderStatus08'></column>
<column name='SalesOrderStatus09'></column>
<column name='SalesOrderStatus10'></column>
<column name='SalesOrderStatus11'></column>
<column name='SalesOrderStatus12'></column>
<column name='SalesOrderStatus13'></column>
<column name='SalesOrderStatus14'></column>
<column name='SalesOrderStatus15'></column>
<column name='Salesperson1'></column>
<column name='SalespersonCommission1'></column>
<column name='Salesperson2'></column>
<column name='SalespersonCommission2'></column>
<column name='ApplyCommissionYN'></column>
<column name='CommissionCategory'></column>
<column name='ReasonCode'></column>
<column name='GrossWeight'></column>
<column name='UnitOfMeasureGrossWt'></column>
<column name='AcctNoInputMode'></column>
<column name='AccountId'></column>
<column name='PurchasingCostCenter'></column>
<column name='ObjectAccount'></column>
<column name='Subsidiary'></column>
```

```
                  <column name='LedgerType'></column>
                  <column name='Subledger'></column>
                  <column name='SubledgerType'></column>
                  <column name='CodeLocationTaxStat'></column>
                  <column name='PriceCode1'></column>
                  <column name='PriceCode2'></column>
                  <column name='PriceCode3'></column>
                  <column name='StatusInWarehouse'></column>
                  <column name='WoOrderFreezeCode'></column>
                  <column name='CorrespondenceMethod'></column>
                  <column name='CurrencyCodeFrom'></column>
                  <column name='CurrencyConverRateOv'></column>
                  <column name='AmountListPriceForeign'></column>
                  <column name='AmtForPricePerUnit'></column>
                  <column name='AmountForeignExtPrice'></column>
                  <column name='AmountForeignUnitCost'></column>
                  <column name='AmountForeignExtCost'></column>
                  <column name='UserReservedCode'></column>
                  <column name='UserReservedDate'></column>
                  <column name='UserReservedAmount'></column>
                  <column name='UserReservedNumber'></column>
                  <column name='UserReservedReference'></column>
                  <column name='TransactionOriginator'></column>
                  <column name='UserId'></column>
                  <column name='ProgramId'></column>
                  <column name='WorkStationId'></column>
                  <column name='DateUpdated'></column>
                  <column name='TimeOfDay'></column>
          </table>
          <table name='F49211Z1' type='additionalHeader'>
                  <WARNING>No record found</WARNING>
          </table>
  </transaction>
</jdeResponse>
```

Managing Message Adapter Queues

The Asynchronous Messaging Adapter accepts input and produces output by reading and writing to messaging queues. After the message adapter has been loaded, you must create specific queues for the OneWorld Asynchronous Messaging Adapter to use. You must specify the names of these queues in the jde.ini file on the OneWorld Enterprise Server so that the OneWorld Asynchronous Messaging Adapter can find them. The adapter configuration specifications are defined within the jde.ini initialization file that is read upon startup of the OneWorld Enterprise Server. Typically the system administrator will set up the jde.ini file setting, but you may need to change the settings or verify that the settings are correct.

When you install the message adapter you are asked to create several message queues. The following queues reside on your One World Enterprise server:

- Inbound
- Outbound
- Success
- Error
- Default Response

The queue names in the jde.ini file must correspond to the queue names on the server.

Inbound Queue

The inbound queue is used to store all inbound messages to One World. After the message is processed it is removed from the queue. The install suggests calling the queue INBOUND.Q. You can name the queue whatever you want. You must specify the queue name in the jde.ini.

Outbound Queue

The outbound queue is used to store the outbound messages from One World. After the message is processed it is removed from the queue. The install suggests calling the queue OUTBOUND.Q. You can name the queue whatever you want. You must specify the queue name in the jde.ini.

Success Queue

The success queue stores the successfully processed messages into One World. These messages contain return code information for the business function calls and default or calculated parameter information. The messages remain in the queue until you remove them. The install suggests calling the queue SUCCESS.Q. You can name the queue whatever you want. You must specify the queue name in the xml document within the returnParms tag. If you do not specify a success destination queue within the xml document and you leave the QerrorName blank in the jde.ini, the message is not written to any queue.

Error Queue

This error queue is used to store the processed messages into One World that are in error. These messages contain return code information for the business function calls and default/calculated parameter information and error information. These messages remain in the queue until you remove them. The install suggests calling the queue ERROR.Q. You can name the queue whatever you want. You must specify the queue name in the xml document within the returnParms tag. If you do not specify a failure destination queue within the xml document and you leave the QErrorName blank in the jde.ini the message is not written to any queue.

Default Response Queue

The default response queue stores the processed messages into One World. These messages may be in error or successfully processed. The messages contain return code information for the business function calls, default or calculated parameter information, and possibly error information. These messages remain in the queue until you remove them. The install suggests calling the queue DEFRES.Q. You can name the queue whatever you want. You must specify the queue name in the QErrorName in the jde.ini. If you do not specify a success or failure destination queue in the xml document, this queue will be used as the default to place the message in. If this is also blank, the message is not written to any queue.

Setting the jde.ini File

The OneWorld messaging adapter uses settings in the MQSI section or the MSMQ section of the jde.ini file to start, to monitor queues, and to send error messages. The names of queues are case-sensitive. The jde.ini file can be modified for messaging queues and for OneWorld UBE queues. Refer to the appropriate Messaging Adapter Installation Guide for more information about setting up queues and the jde.ini file settings. The queue names you use must correspond with the queue names you have set up on the server.

Creating Custom XML Documents

Creating Custom XML Documents

You can create your own custom XML documents. To create your own custom XML documents:

- Use the business function documentation to help you find the specific business functions you want to use. Refer to Business Function Documentation in this guide for more information about generating documentation and finding information about specific business functions.
- Create an XML template.

Business Function Documentation

Business function documentation explains what individual business functions do and how they should be used. The documentation for a business function should include information such as:

- Purpose
- Parameters (the data structure used)
- Explanation of each individual parameter that indicates: input/output required and explanation of return values
- Related tables (the table accessed)
- Related business functions (business functions called from within the functions itself)
- Special handling instructions

You use Business Function Design and Data Structure Design to document your business functions.

Business function documentation features allow you to:

- Create documentation
- Generate documentation
- View documentation

Creating Business Function Documentation

You can create business function documentation at several levels, including:

Business Function Notes	This shows you the documentation for the specific business function you are using.
Data Structure Notes	This displays notes on the data structure for the business function.
Parameter Notes	This displays notes on the actual parameters in the data structure.

? To create business function documentation

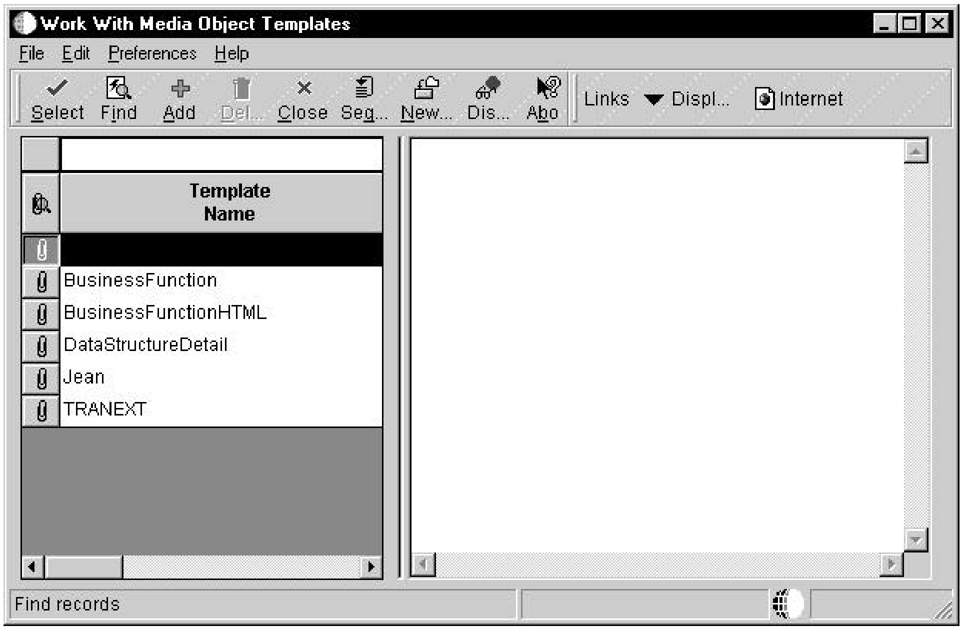
1. On Object Management Workbench, choose the business function you wish to document and click the Design button.

The Object Librarian Business Function Design form appears.

2. Click the Attachments tab.

3. On Media Objects, right click in the icon panel and choose Templates.

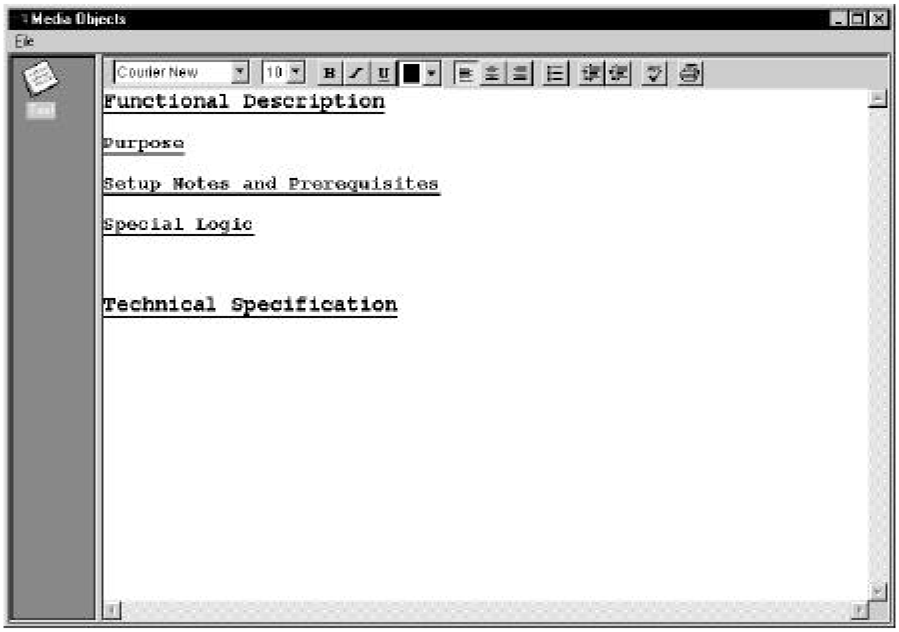
Work with Media Object Templates displays the available templates you can use.



Template Name
BusinessFunction
BusinessFunctionHTML
DataStructureDetail
Jean
TRANEXT

4. Select the template you wish to use.

J.D. Edwards uses the Business Function template as a standard.



Functional Description

Purpose

Setup Notes and Prerequisites

Special Logic

Technical Specification

5. Type in the appropriate information under each template heading.

Business Function Documentation Template

The Business Function documentation template contains the following sections:

Purpose

This section contains a brief summary of what the function does.

Setup Notes and Prerequisites

The section includes any special notes to assist in using the function, including prerequisite functions, special values that need to be initialized, events recommended to run the function, or if memory must be cleared separately after the function is used.

Special Logic

This section contains additional details about the business function logic. It is usually only used for complex functions that require more explanation than the purpose summary.

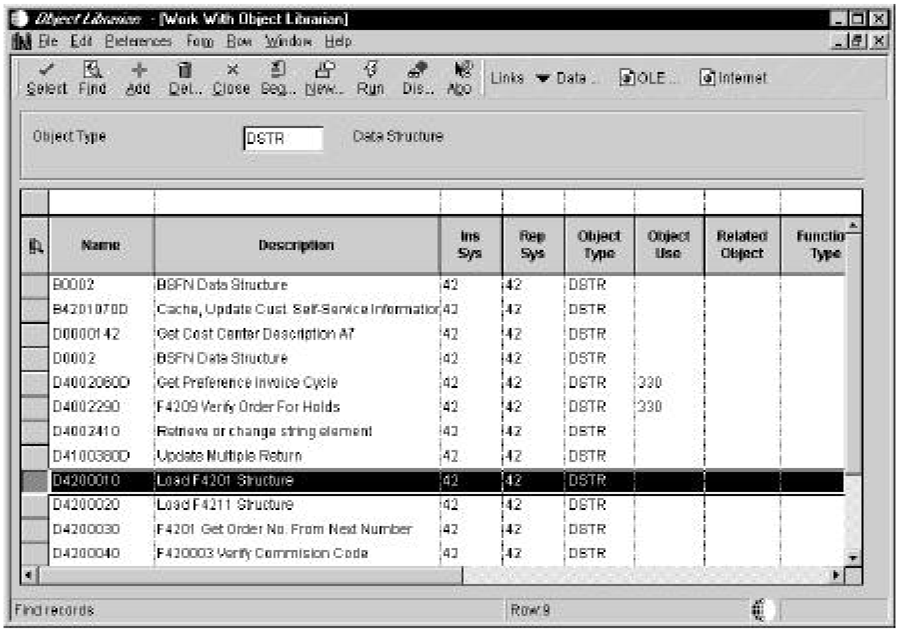
Technical Specification

This section contains the technical specification of the function written in scripted English. This may be a direct copy from an existing word processing document.

Creating Data Structure Documentation

? To create data structure documentation

1. On Object Librarian, choose the data structure you wish to document.

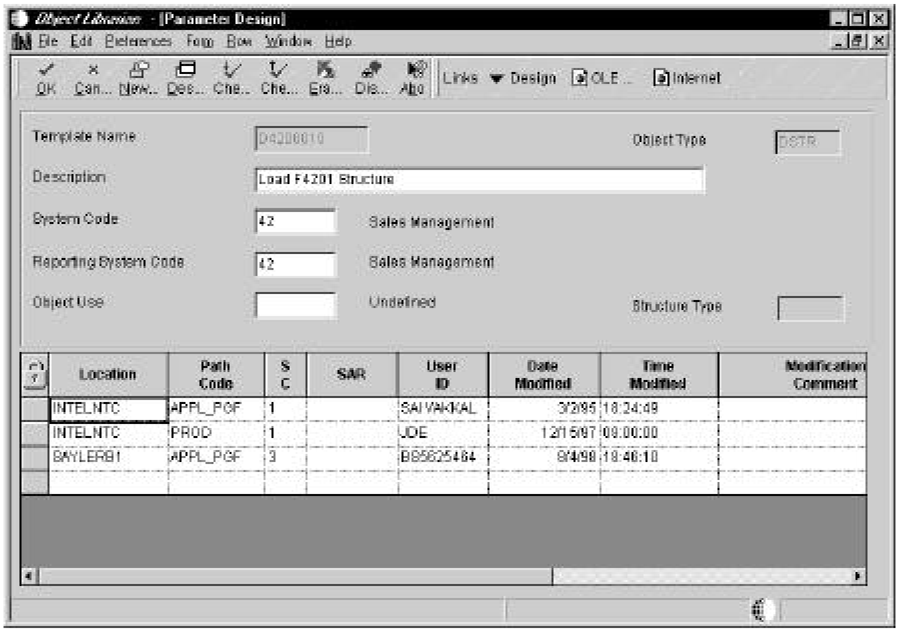

Object Librarian - [Work With Object Librarian]
File Edit Preferences Form Row Window Help
Select Find Add Del... Close Seq... New... Run Dis... Abo
Links Data ... OLE ... Internet

Object Type DSTR Data Structure

	Name	Description	Ins Sys	Rep Sys	Object Type	Object Use	Related Object	Function Type
	B0002	BSFN Data Structure	42	42	DSTR			
	B4201070D	Cache, Update Cust. Self-Service Informatior	42	42	DSTR			
	D0000142	Get Cost Center Description A7	42	42	DSTR			
	D0002	BSFN Data Structure	42	42	DSTR			
	D4002060D	Get Preference Invoice Cycle	42	42	DSTR	330		
	D4002290	F4209 Verify Order For Holds	42	42	DSTR	330		
	D4002410	Retrieve or change string element	42	42	DSTR			
	D4100380D	Update Multiple Return	42	42	DSTR			
	D4200010	Load F4201 Structure	42	42	DSTR			
	D4200020	Load F4211 Structure	42	42	DSTR			
	D4200030	F4201 Get Order No. From Next Number	42	42	DSTR			
	D4200040	F420003 Verify Commision Code	42	42	DSTR			

Find records Row 9

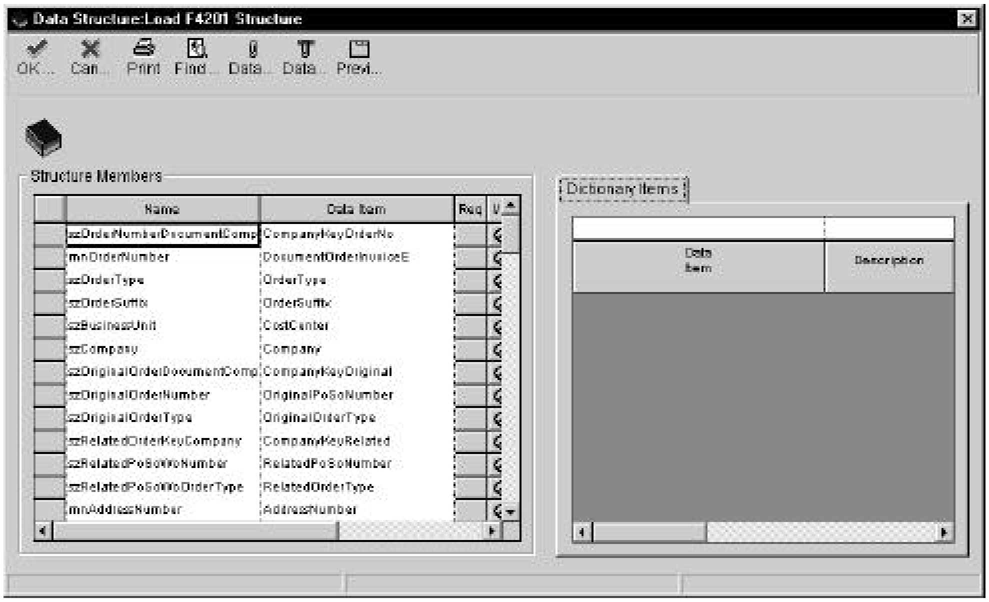
2. Check the data structure out to your workstation.


Object Librarian - [Parameter Design]
File Edit Preferences Form Row Window Help
OK Can... New... Des... Che... Che... Era... Dis... Abo
Links Design OLE ... Internet

Template Name	D4200010	Object Type	DSTR
Description	Load F4201 Structure		
System Code	42	Sales Management	
Reporting System Code	42	Sales Management	
Object Use		Undefined	Structure Type

	Location	Path Code	S C	SAR	User ID	Date Modified	Time Modified	Modification Comment
	INTELNTC	APPL_PGF	1		SAIVAKKAL	3/2/95	18:24:49	
	INTELNTC	PROD	1		JDE	12/15/97	08:00:00	
	SAYLERB1	APPL_PGF	3		BS5625484	8/4/98	18:46:10	

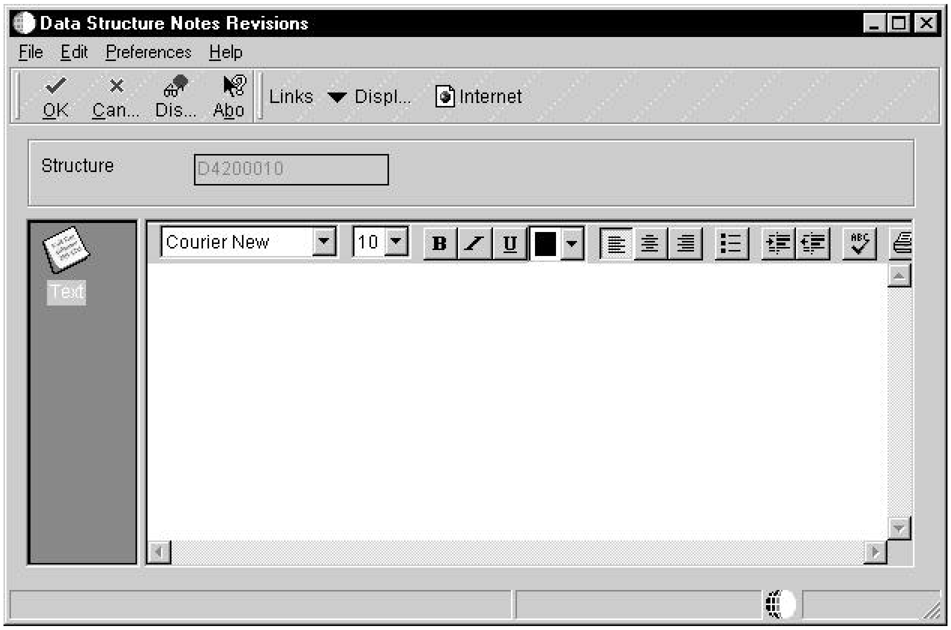
3. On Parameter Design, choose Design from the Form menu.



Data Structure:Load F4201 Structure

Name	Data Item	Req	V
szOrderNumberDocumentComp	CompanyKeyOrderNo		
mnOrderNumber	DocumentOrderInvoiceE		
szOrderType	OrderType		
szOrderSuffix	OrderSuffix		
szBusinessUnit	CostCenter		
szCompany	Company		
szOriginalOrderDocumentComp	CompanyKeyOriginal		
szOriginalOrderNumber	OriginalPoSoNumber		
szOriginalOrderType	OriginalOrderType		
szRelatedOrderKeyCompany	CompanyKeyRelated		
szRelatedPoSoWoNumber	RelatedPoSoNumber		
szRelatedPoSoWoOrderType	RelatedOrderType		
mnAddressNumber	AddressNumber		

4. On Data Structure Design, click the left binder clip for Data Structure Attachments.



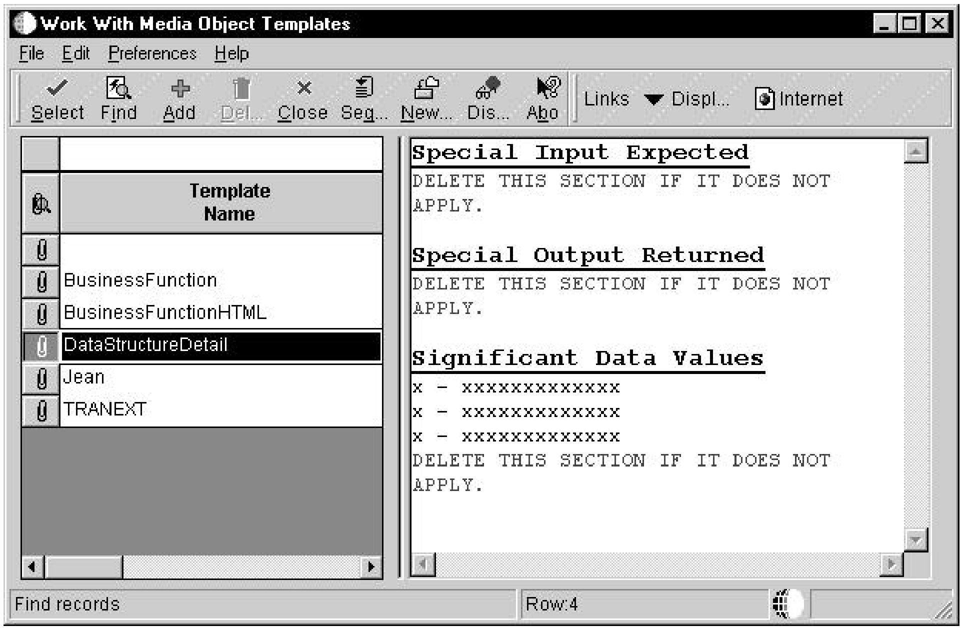
Data Structure Notes Revisions

Structure D4200010

Courier New 10

5. On Media Objects, right click in the icon panel and choose templates.

Work with Media Object Templates displays the available templates you can use.


Work With Media Object Templates
File Edit Preferences Help
Select Find Add Del... Close Seq... New... Dis... Abo Links Displ... Internet
Template Name
Find records
Row:4

BusinessFunction
BusinessFunctionHTML
DataStructureDetail
Jean
TRANEXT

Special Input Expected
DELETE THIS SECTION IF IT DOES NOT APPLY.

Special Output Returned
DELETE THIS SECTION IF IT DOES NOT APPLY.

Significant Data Values
x - xxxxxxxxxxxxx
x - xxxxxxxxxxxxx
x - xxxxxxxxxxxxx
DELETE THIS SECTION IF IT DOES NOT APPLY.

6. Select the template you wish to use.

J. D. Edwards uses the Data Structure Detail template.

7. Type in the appropriate information under each template heading.

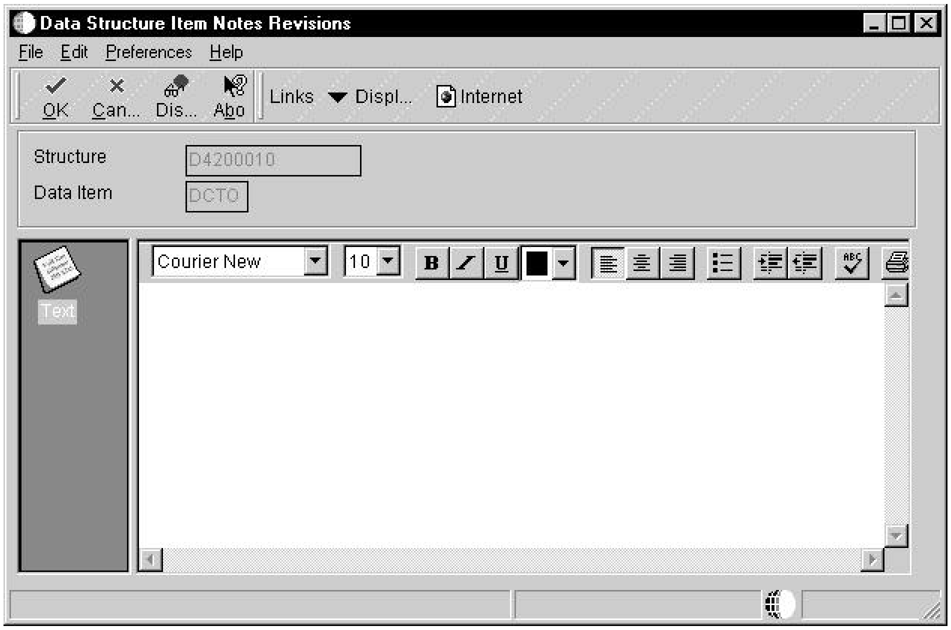
Data Structure Documentation Template

The data structure documentation template contains the following sections:

Special Input Expected	This section should be deleted if it does not apply.
Special Output Returned	This section should be deleted if it does not apply.
Significant Data Values	This section should be deleted if it does not apply. Otherwise it is in the format: x - xxxxxxxxxxxxx.

Creating Parameter Documentation

The steps for creating parameter documentation are the same as those for data structure documentation, except that after selecting the data item in the structure you wish to enter notes for, you click the Data Structure Item Attachments binder clip instead of the Data Structure Attachments binder clip. There is no special template for parameter documentation.


Data Structure Item Notes Revisions
File Edit Preferences Help
OK Can... Dis... Abo
Links Displ... Internet
Text
Courier New
10

Structure D4200010

Data Item DCTO

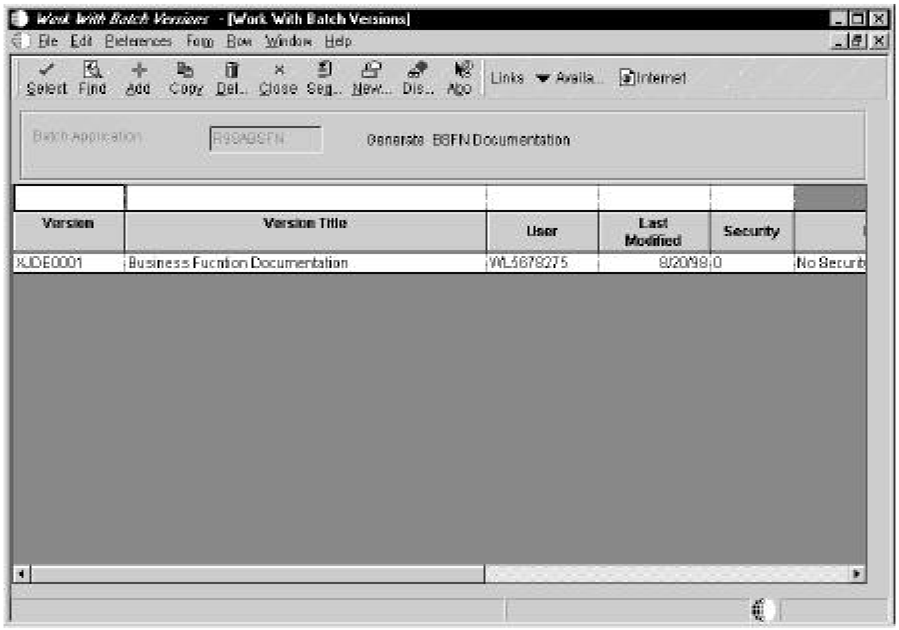
You can enter specific notes about a data structure item to further clarify the information that should be passed in or passed out of the item, for example a mode parameter. The notes should indicate the valid values the function will except when you hook it up and how to use them. For example, 1 = Add mode, or 2 = Delete.

Generating Business Function Documentation

Generating business function documentation provides you with an online list of business function documentation that allows you to view documentation through the Business Function Documentation Viewer (P98ABSFN). Typically the system administrator performs this task, because generating the business function documentation for all business functions takes a long time. If you create new business function documentation you may need to regenerate the business function documentation just for that business function.

? To generate business function documentation

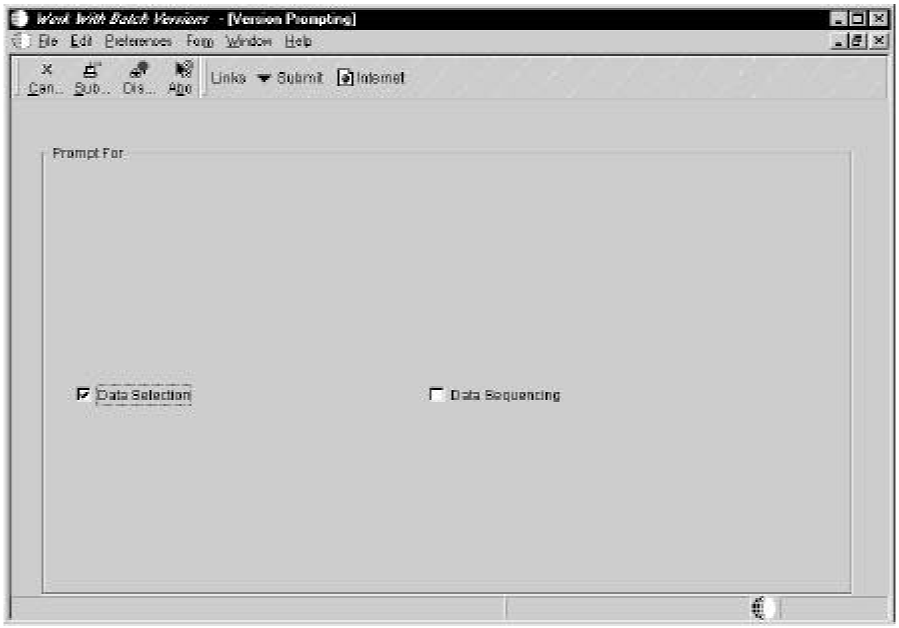
1. From the Cross Application Development Tools menu (GH902), select Generate BSFN Documentation.


Work With Batch Versions - [Work With Batch Versions]
File Edit Preferences Form Row Window Help
Select Find Add Copy Del.. Close Seq.. New... Dis... Abo
Links Availa.. Internet

Batch Application R98ABSFN Generate BSFN Documentation

Version	Version Title	User	Last Modified	Security	
XJDE0001	Business Fucntion Documentation	WL5678275	8/20/98	0	No Securit

2. On Work with Batch Versions, choose version XJDE0001.

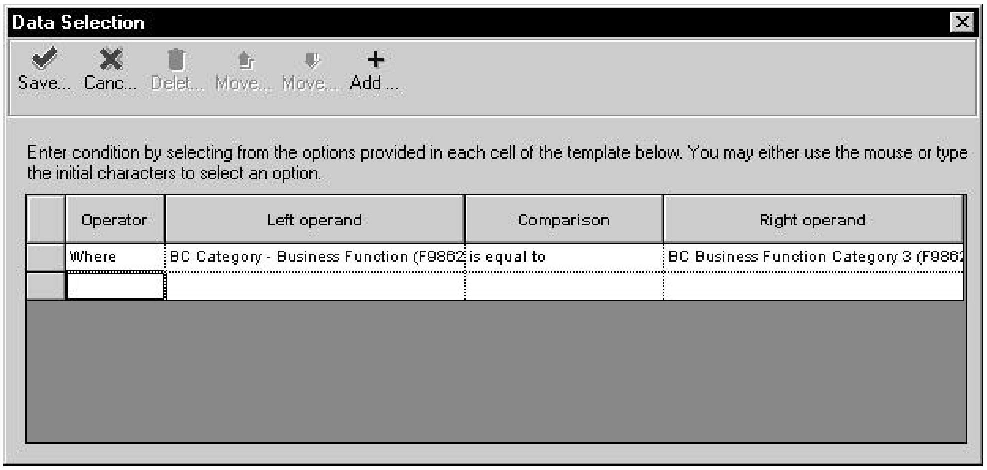

Work With Batch Versions - [Version Prompting]
File Edit Preferences Form Window Help
Can.. Sub.. Dis... Abo
Links Submit Internet

Prompt For

☑ Data Selection ☐ Data Sequencing

3. If you do not want to generate all business function documentation, on Version Prompting, choose the following option:
   - Data Selection

4. Click Submit.

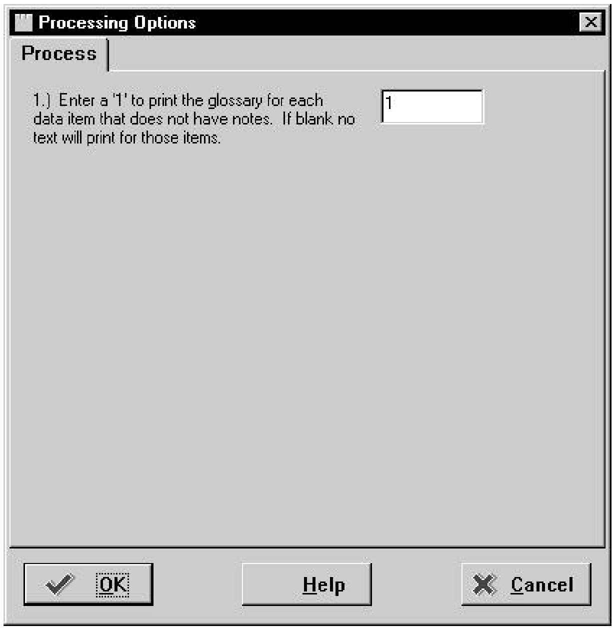


Enter condition by selecting from the options provided in each cell of the template below. You may either use the mouse or type the initial characters to select an option.

Operator	Left operand	Comparison	Right operand
Where	BC Category - Business Function (F9862	is equal to	BC Business Function Category 3 (F986

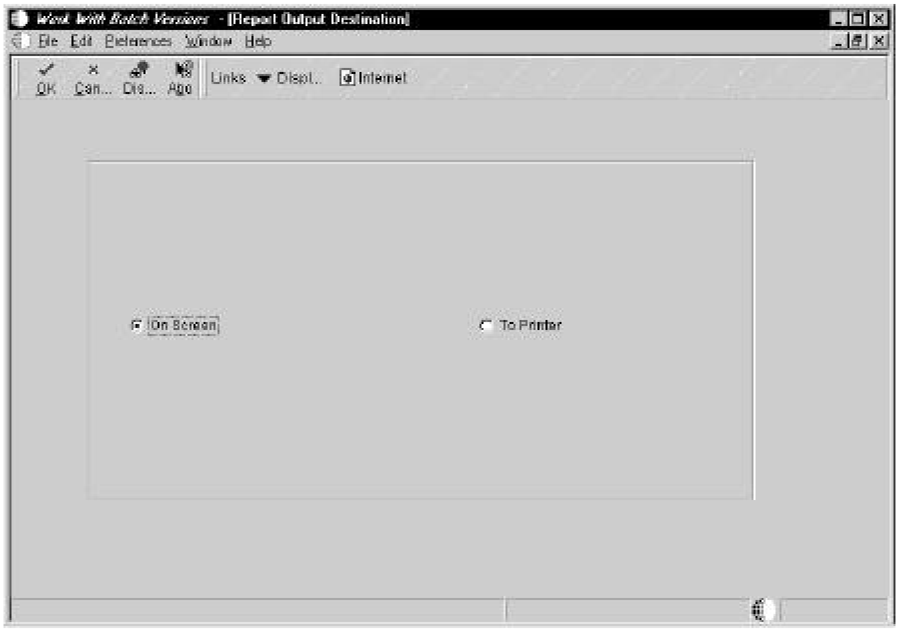
5. On Data Selection, build your criteria for data selection and click OK.

Select only those functions for which you are generating documentation.



1.) Enter a '1' to print the glossary for each data item that does not have notes. If blank no text will print for those items. 1

6. Depending on the criteria you choose, you might also need to designate processing options.


Work With Batch Versions - [Report Output Destination]
File Edit Preferences Window Help
OK
Can...
Dis...
Abo
Links
Displ..
Internet
● On Screen
○ To Printer

7. If you are running your report locally, on Report Output Destination, choose one of the following output destinations:

- On Screen
- To Printer

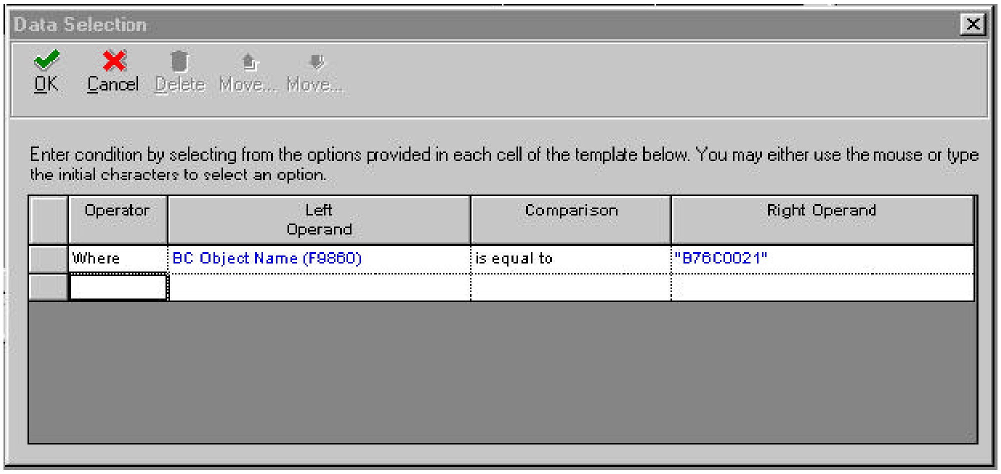
A hypertext markup language (HTML) link is created for each business function for which you generated documentation. An Index HTML file is also created. These HTML files are placed in your output queue directory. Output is in the following format:

Function Name
Function Description from O/L
Parent DLL:
Location:
Language:
Purpose
Special Handling
Data Structure
Parameter Name   dataitem   data type   req/opt   i/o/both

Data Selection Tips

You can use data selection to choose the business functions for which you wish to generate documentation. R98ABSFN uses your data selection criteria to filter the business function documentation. It takes longer to run when you generate documentation you do not need. If you generate documentation for all business functions, the process can take quite a while. You can use data selection to generate documentation for one business function, all business functions, or any combination between.

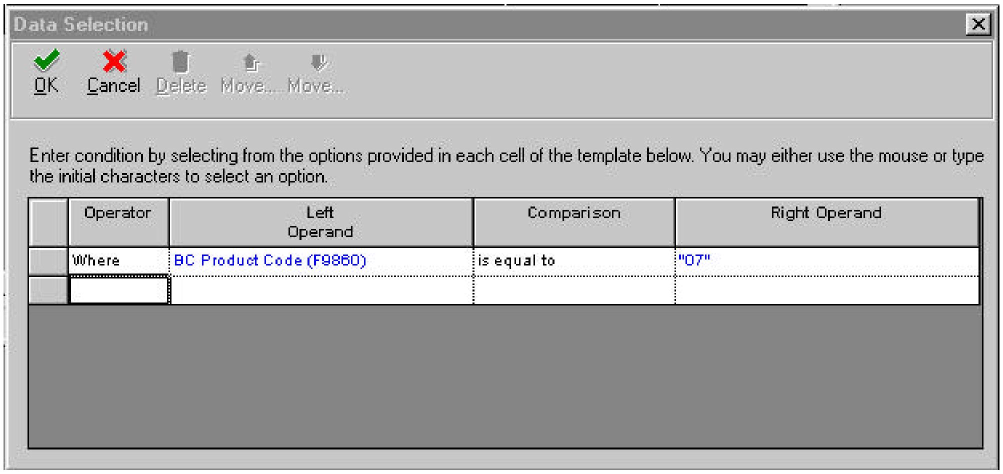
For example, if you want to generate documentation for a single business function you can use the data item BC Object Name (F9860).



Enter condition by selecting from the options provided in each cell of the template below. You may either use the mouse or type the initial characters to select an option.

Operator	Left Operand	Comparison	Right Operand
Where	BC Object Name (F9860)	is equal to	"B76C0021"

If you want to generate documentation for all of the business functions for a specific product code, such as Payroll, you use the data item BC Product Code (F9860).



Enter condition by selecting from the options provided in each cell of the template below. You may either use the mouse or type the initial characters to select an option.

Operator	Left Operand	Comparison	Right Operand
Where	BC Product Code (F9860)	is equal to	"07"

You can also use the right operand on the Data Selection form to choose ranges or lists of values to further refine your filter.

You can filter using any value that is associated with a business function. For example, you can use BC Date - Updated (F9860) if you have already produced the documentation for a previous release of OneWorld and you want only new or modified business function documentation after an upgrade or update of OneWorld.

You use BC Function Type (F9860) to choose Master business function documentation.

You use BC Location Business Function (F9860) to produce documentation for client run business functions.

You use BC Object Type (F9860) to generate documentation for NERs only.

You can use many other informational fields choose the business functions for which you wish to generate documentation.

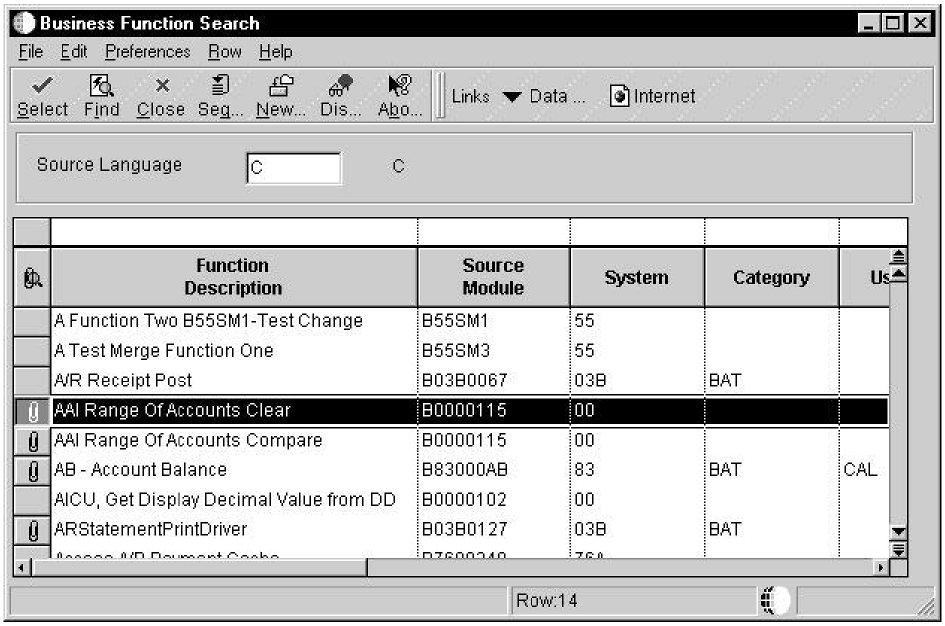
Viewing Business Function Documentation

You can view your business function notes from several different locations, including:

- Business Function Search
- Business Function - Values to Pass
- Business Function Documentation Viewer

Viewing Documentation from Business Function Search

When you make a connection to a business function in event rules, the Business Function Search form appears. You can then select the function you want to call. From the row menu, choose Data Structure Notes or Attachments to view the documentation for the business function.



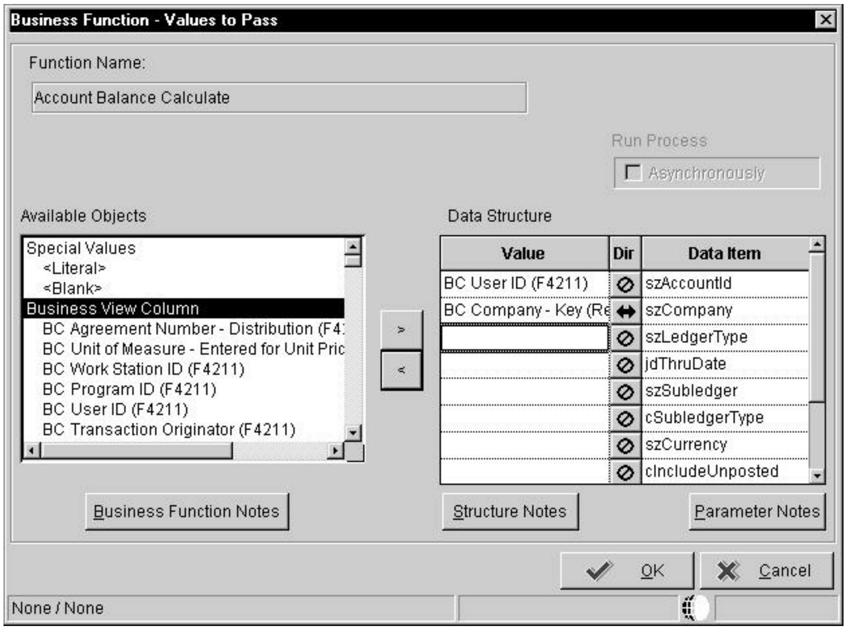
Source Language C C

	Function Description	Source Module	System	Category	Us
	A Function Two B55SM1-Test Change	B55SM1	55		
	A Test Merge Function One	B55SM3	55		
	A/R Receipt Post	B03B0067	03B	BAT	
📎	AAI Range Of Accounts Clear	B0000115	00		
📎	AAI Range Of Accounts Compare	B0000115	00		
📎	AB - Account Balance	B83000AB	83	BAT	CAL
	AICU, Get Display Decimal Value from DD	B0000102	00		
📎	ARStatementPrintDriver	B03B0127	03B	BAT	

Viewing Documentation from Business Function - Values to Pass

You can click one of the following buttons on Business Function - Values to Pass to view documentation for a single business function (refer to *Business Function Event Rules* for more information about accessing this form).

- Business Function Notes
- Structure Notes
- Parameter Notes



Function Name: Account Balance Calculate

Special Values
<Literal>
<Blank>
Business View Column
BC Agreement Number - Distribution (F4
BC Unit of Measure - Entered for Unit Pric
BC Work Station ID (F4211)
BC Program ID (F4211)
BC User ID (F4211)
BC Transaction Originator (F4211)

Value	Dir	Data Item
BC User ID (F4211)	⊘	szAccountId
BC Company - Key (Re	↔	szCompany
	⊘	szLedgerType
	⊘	jdThruDate
	⊘	szSubledger
	⊘	cSubledgerType
	⊘	szCurrency
	⊘	cIncludeUnposted

None / None

BSFN Notes Displays the notes for the business function.

Structure Notes Displays the notes for the whole data structure.

Parameter Notes Displays the notes for a particular parameter.

Viewing Documentation from Business Function Documentation Viewer

You can use Business Function Documentation Viewer to view documentation for all business functions or selected business functions. Once you have generated your report, use menu GH902 to access the Business Function Documentation Viewer (P98ABSFN) to display your information. J.D. Edwards suggests that you use this method to view business function documentation.

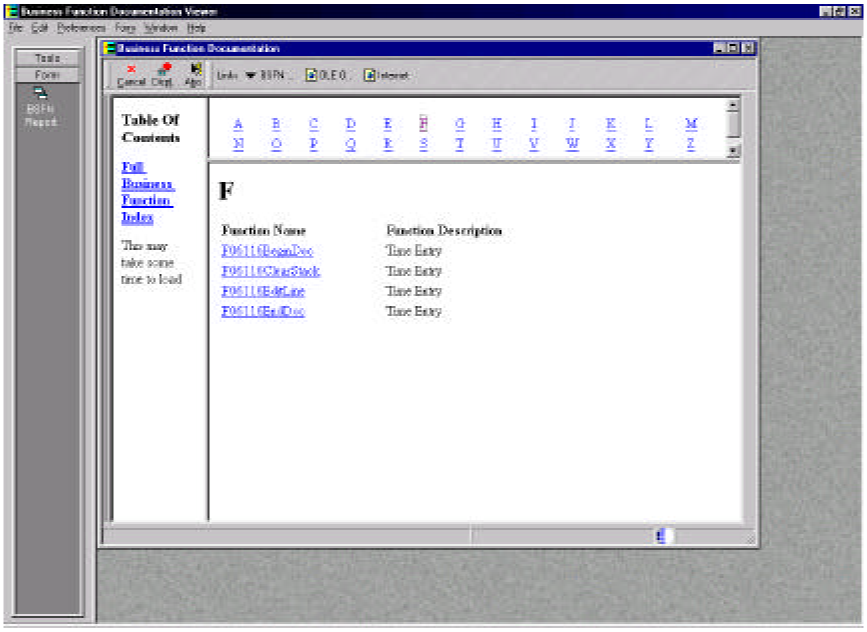

Business Function Documentation Viewer
File Edit Preferences Form Window Help
Tools
Form
BSFN Report
Business Function Documentation
Cancel Clip Abo
Links BSFN OLE 0 Internet
A B C D E F G H I J K L M
N O P Q R S T U V W X Y Z

Table Of Contents

Full Business Function Index

This may take some time to load

F

Function Name	Function Description
F06116BeginDoc	Time Entry
F06116ClearStack	Time Entry
F06116EditLine	Time Entry
F06116EndDoc	Time Entry

The Business Function Documentation form contains the HTML index you generated. You can view either the entire index or select just the functions for a specific letter in the alphabet by clicking on that letter in the index. Double click a business function to view documentation specific to that function.

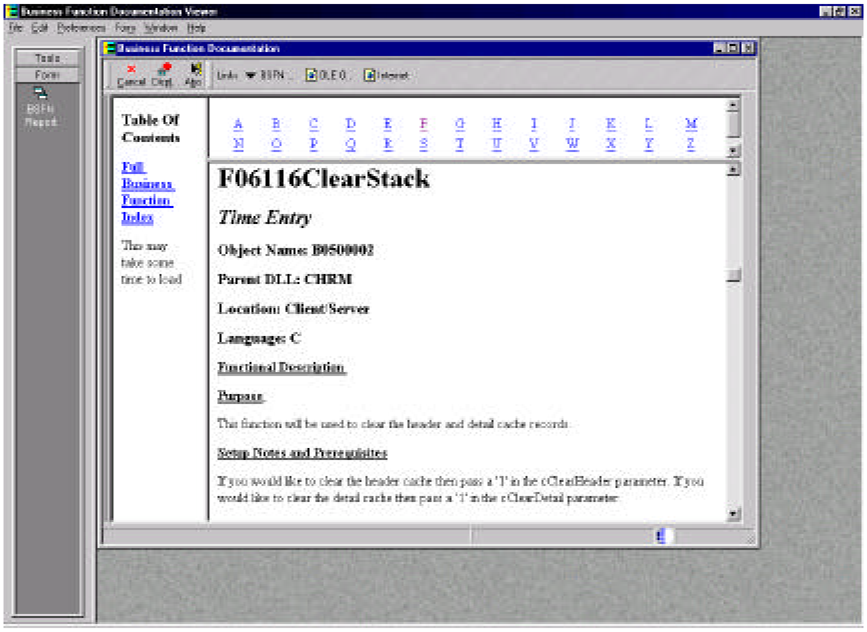

Business Function Documentation Viewer
File Edit Preferences Form Window Help
Tools
Form
BSFN Report
Business Function Documentation
Cancel Clip Abo
Links BSFN OLE 0 Internet
A B C D E F G H I J K L M
N O P Q R S T U V W X Y Z

Table Of Contents

Full Business Function Index

This may take some time to load

F06116ClearStack

Time Entry

Object Name: B0500002

Parent DLL: CHRM

Location: Client/Server

Language: C

Functional Description

Purpose

This function will be used to clear the header and detail cache records.

Setup Notes and Prerequisites

If you would like to clear the header cache then pass a '1' in the cClearHeader parameter. If you would like to clear the detail cache then pass a '1' in the cClearDetail parameter.

The media object loads the HTML index of the business functions based on a media object queue. In the media object queue table a queue named Business Function Doc must be set up. This queue must point to the directory where the business function HTML files are located. The system administrator usually generates the documentation for all business functions. Because the generation process places the documentation files in the local directory, the administrator must then copy the files to a central directory on the deployment server. The files must be copied to the media object queue for media object business function notes. If you are running standalone, this path will usually be the output directory from the Network Queue Settings section of our jde.ini file. If this entry is not in your jde.ini file, it is the print queue directory in your OneWorld directory.

Creating an XML Template

XML CallObject Templates

The caller can request a XML Template for a given business function. This template is an XML document that is a *callMethod* request with information about all the function parameters, but with no data values filled in. The *user, pwd* and *session* attribute values are blanked out so that the caller can cache the response for later use.

This request is an exception to the convention that a *jdeRequest* returns a *jdeResponse* to the caller. Although this makes it easier to directly use the template as a *callMethod* request, this is only true if the request is the only request in the document.

```
<?xml version='1.0' ?>
<jdeRequest type='callmethod' user='steve'
pwd='xyz' environment='prod' session=''>
<callMethodTemplate name='myfunc' app='P42101'/>
</jdeRequest>
```

```
<?xml version='1.0' ?>
<jdeRequest type='callmethod' user='' pwd=''
environment='prod' session=''>
<callMethod name='myfunc' app='P42101'>
<params>
     <param name='CostCtr'></param>
     <param name='ExpDate'></param>
     <param name='Quantity'></param>
</params>
</callMethod>
</jdeRequest>
```

Appendix A – Examples (All Parameters)

The following examples are used for specific formats.

Inbound Sales Order XML Format (All Parameters)

This example illustrates an inbound XML format with all of the parameters.

```
            ”<?xml version='1.0' ?>
            <jdeRequest type=’callmethod’ user=’userid’ pwd=’password’
            environment=’environment’ >
  <callMethod name=’ GetLocalComputerId’ app=’NetCommerce’ runOnError=’no’>
     <params>
       <param name=’ szMachineKey’id=’2’ >< /param>
     <params>
                <callMethod>
  <callMethod name=’ F4211FSBeginDoc’ app=’NetCommerce’ runOnError=’no’>
     <params>
       <param name=’mnCMJobNumber‘ id=’1‘></param>
       <param name=’cCMDocAction’ >A</param>
       <param name=’cCMProcessEdits’ <1>/param>            ( 1 = Full)
       <param name=’szCMComputerID’ idref=’2’ >< /param>
       <param name=’cCMErrorConditions’ >value< /param> (1=Warnings, 2=Errors)
       <param name=’cCMUpdateWriteToWF’ >value< /param>       (1=wf,2=cache)
        <param name=’szCMProgramID’ >value< /param>
        <param name=’szCMVersion’ >value< /param>
       <param name=’szOrderCo’ <value< /param>
       <param name=’mnOrderNo’ >value< /param>
       <param name=’szOrderType’ >value< /param>        (If blank def Proc Opt)
       <param name=’szBusinessUnit’ >value< /param>        (If blank def Proc Opt)
       <param name=’szOriginalOrderCo’ >value< /param>   (used copy/blanket
       function)
       <param name=’szOriginalOrderNo’ >value< /param>    (used copy/blanket
       function)
```

```
<param name=’szOriginalOrderType’ >value< /param>  (used copy/blanket
function)
<param name=’mnAddressNumber’ >value< /param>          (Required if ship to
= 0)
<param name=’mnShipToNo’ >value< /param>          (Required if sold to = 0)
<param name=’jdRequestedDate’ >value< /param>
<param name=’jdOrderDate’ >value< /param>
 <param name=’jdPromisedDate’ >value< /param>
<param name=’jdCancelDate’ >value< /param>
<param name=’szReference’ >value< /param>
<param name=’szDeliveryInstructions1’ >value< /param>
 <param name=’szDeliveryInstructions2’ >value< /param>
<param name=’szPrintMesg’ >value< /param>
 <param name=’szPaymentTerm’ >value< /param>
<param name=’cPaymentInstrument’ >value< /param>
 <param name=’szAdjustmentSchedule’ >value< /param>
 <param name=’mnTradeDiscount’ >value< /param>
 <param name=’szTaxExplanationCode’ >value< /param>
 <param name=’szTaxArea’ >value< /param>
 <param name=’szCertificate’ >value< /param>
 <param name=’cAssociatedText’ >value< /param>
 <param name=’szHoldOrdersCode’ >value< /param>
<param name=’cPricePickListYN’ >value< /param>
<param name=’mnInvoiceCopies’ >value< /param>
<param name=’mnBuyerNumber’>value</param>
<param name=’mnCarrier’ >value</param>
<param name=’szRouteCode’ >value</param>
<param name=’szStopCode’ >value</param>
<param name=’szZoneNumber’ >value</param>
<param name=’szFreightHandlingCode’ >value</param>
<param name=’cApplyFreightYN’ >value</param>
<param name=’mnCommissionCode1’ >value</param>
<param name=’mnCommissionRate1’ >value</param>
<param name=’mnCommissionCode2’ >value</param>
<param name=’mnCommissionRate2’ >value</param>
```

```
<param name=’szWeightDisplayUOM’ >value</param>
<param name=’szVolumeDisplayUOM’ >value</param>
<param name=’szAuthorizationNo’ >value</param>
<param name=’szCreditBankAcctNo’ >value</param>
<param name=’jdCreditBankExpiredDate’ >value</param>
<param name=’cMode’ >value</param>
<param name=’szCurrencyCode’ >value</param>
<param name=’mnExchangeRate’ >value</param>
<param name=’szOrderedBy’ >value</param>
<param name=’szOrderTakenBy’ >value</param>
<param name=’szUserReservedCode’ >value</param>
<param name=’jdUserReservedDate’ >value</param>
<param name=’mnUserReservedAmnt’ >value</param>
<param name=’mnUserReservedNo’ >value</param>
<param name=’szUserReservedRef’ >value</param>
<param name=’jdDateUpdated’ >value</param>
<param name=’szUserID’ >value</param>
<param name=’szWKBaseCurrency’ >value</param>
<param name=’cWKAdvancedPricingYN’ >value</param>
<param name=’szWKCreditMesg’ >value</param>
<param name=’szWKTempCreditMesg’ >value</param>
<param name=’cWKInvalidSalesOrderNo’ >value</param>
<param name=’cWKSourceOfData’ >blank</param>   (Required, blank = parms
)
<param name=’cWKProcMode’ >blank</param>   (blank = reg order)
<param name=’mnWKSuppressProcess’ >0</param>       (0 = def, 2=P/O)
<param name=’mnSODDocNo’ >value</param>
<param name=’szSODDocType’ >value</param>
<param name=’szSODOrderCo’ >value</param>
<param name=’mnTriangulationRateFrom’ >value</param>
<param name=’mnTriangulationRateTo’ >value</param>
<param name=’cCurrencyConversionMethod’ >value</param>
<param name=’cRetrieveOrderNo’ >value</param>
<param name=’szPricingGroup’ >value</param>
<param name=’cCommitInvInED’ >value</param>
```

```
        <param name=’cSpotRateAllowed’ >value</param>
        <param name=’cGenericChar2_EV02’ >value</param>
        <param name=’szGenericString1_DL01’ >value</param>
        <param name=’szGenericString2_DL02’ >value</param>
        <param name=’mnGenericMathNumeric1_MATH01’ >value</param>
        <param name=’mnGenericMathNumeric2_MATH02’ >value</param>
        <param name=’szLongAddressNumberShipto’ >value</param>
        <param name=’szLongAddressNumber’ >value</param>
        <param name=’mnProcessID’ >value</param>
        <param name=’mnTransactionID’ >value</param>
      </params>
      <onError abort=’yes’>\
        <callMethod name=’ F4211ClearWorkFile’app=’NetCommerce’
        runOnError=’yes’>
          <params>
             <param name=’mnJobNo‘ idref=’1‘></param>
             <param name=’szComputerID’ idref=’2’ >< /param>
             <param name=’mnFromLineNo’ >value</param>
             <param name=’mnThruLineNo’ >value</param>
             <param name=’cClearHeaderWF’ >value</param>
             <param name=’cClearDetailWF’ >value</param>
             <param name=’szProgramID’ >value</param>
             <param name=’mnWKRelatedOrderProcess’ >value</param>
             <param name=’szCMVersion’ >value</param>
             <param name=’cGenericChar1_EV01’ >value</param>
             <param name=’szGenericString1_DL01’ >value</param>
             <param name=’mnSODRelatedJobNumber’ >value</param>
             <param name=’mnProcessID’ >value</param>
             <param name=’mnTransactionID’ >value</param>
           </params>
          </callMethod>
      </onError>
  </callMethod>
  <callMethod name=’ F4211FSEditLine’app=’NetCommerce’ runOnError=’yes’>
  (each line)
```

```
<params>
    <param name=’mnCMJobNo ‘ idref=’1‘></param>
    <param name=’cCMLineAction’ >value</param>
    <param name=’cCMProcessEdits ’ >value</param>
    <param name=’cCMWriteToWFFlag ’ >value</param>
    <param name=’cCMRecdWrittenToWF ’ >value</param>
    <param name=’szCMComputerID’ idref=’2’ >< /param>
    <param name=’cCMErrorConditions’ >value</param>
    <param name=’szOrderCo’ >value</param>
    <param name=’mnOrderNo’ >value</param>
    <param name=’szOrderType’ >value</param>
    <param name=’mnLineNo’ >value</param>
    <param name=’szBusinessUnit’ >value</param>
    <param name=’mnShipToNo’ >value</param>
    <param name=’jdRequestedDate’ >value</param>
    <param name=’jdPromisedDate’ >value</param>
    <param name=’jdCancelDate’ >value</param>
    <param name=’jdPromisedDlvryDate’ >value</param>
    <param name=’szItemNo’ >value</param>
    <param name=’szLocation’>value</param>
    <param name=’szLotNo’ >value</param>
    <param name=’szDescription1’ >value</param>
    <param name=’szDescription2’ >value</param>
    <param name=’szLineType’ >value</param>
    <param name=’szLastStatus’ >value</param>
    <param name=’szNextStatus’ >value</param>
    <param name=’mnQtyOrdered’ >value</param>
    <param name=’mnQtyShipped’ >value</param>
    <param name=’mnQtyBackordered’ >value</param>
    <param name=’mnQtyCanceled’ >value</param>
    <param name=’mnExtendedPrice’ >value</param>
    <param name=’mnExtendedCost’ >value</param>
    <param name=’szPrintMesg’ >value</param>
    <param name=’cPaymentInstrument’ >value</param>
    <param name=’szAdjustmentSchedule’ >value</param>
```

```
        <param name=’cSalesTaxableYN’ >value</param>
        <param name=’cAssociatedText’ >value</param>
        <param name=’szTransactionUOM’ >value</param>
        <param name=’szPricingUOM’ >value</param>
        <param name=’mnItemWeight’ >value</param>
        <param name=’szWeightUOM’ >value</param>
        <param name=’mnForeignUnitPrice >value</param>
        <param name=’mnForeignExtPrice’ >value</param>
        <param name=’mnForeignUnitCost’ >value</param>
        <param name=’mnForeignExtCost’ >value</param>
        <param name=’szPricingCategoryLevel’ >value</param>
        <param name=’mnDiscountFactor’ >value</param>
        <param name=’mnCMLineNo’ >value</param>
        <param name=’szCMProgramID’ >value</param>
        <param name=’szCMVersion’ >value</param>
        <param name=’mnSupplierNo’ >value</param>
        <param name=’szRelatedKitItemNo’ >value</param>
        <param name=’mnKitMasterLineNo’ >value</param>
        <param name=’mnComponentLineNo’ >value</param>
        <param name=’mnRelatedKitComponent’ >value</param>
        <param name=’mnNoOfCpntPerParent’ >value</param>
        <param name=’cOverridePrice’ >value</param>
        <param name=’cOverrideCost’ >value</param>
        <param name=’szUserID’ >value</param>
        <param name=’jdDateUpdated’ >value</param>
        <param name=’mnWKOrderTotal’ >value</param>
        <param name=’mnWKForeignOrderTotal’ >value</param>
        <param name=’mnWKTotalCost’ >value</param>
        <param name=’mnWKForeignTotalCost’ >value</param>
        <param name=’cWKProcessingType’ >value</param>
        <param name=’cWKSourceOfData’ >value</param>
        <param name=’cWKCheckAvailability’ >value</param>
        <param name=’mnLastLineNoAssigned’ >value</param>
        <param name=’cStockingType’ >value</param>
        <param name=’szOriginalOrderKeyCo’ >value</param>
        <param name=’szOriginalOrderNo’ >value</param>
        <param name=’szOriginalOrderType’ >value</param>
        <param name=’mnOriginalOrderLineNo’ >value</param>
        <param name=’cParentItmMethdOfPriceCalcn’ >value</param>
        <param name=’szLandedCost’ >value</param>
        <param name=’mnWKSuppressProcess’ >value</param>
        <param name=’mnShortItemNo’ >value</param>
        <param name=’mnWKRelatedOrderProcess’ >value</param>
        <param name=’mnSODLineNo’ >value</param>
        <param name=’mnPriceAdjRevLevel’ >value</param>
        <param name=’szSalesOrderFlags’ >value</param>
        <param name=’mnSODDocNo’ >value</param>
        <param name=’szSODDocType’ >value</param>
        <param name=’szSODOrderCo’ >value</param>
        <param name=’szTransferOrderToBranch’ >value</param>
        <param name=’mnDomesticDetachedAdj’ >value</param>
        <param name=’mnForeignDetachedAdj’ >value</param>
        <param name=’mnSODWFLineNo’ >value</param>
        <param name=’szGeneric2CharString’ >value</param>
        <param name=’mnTOEPOExchangeRate’ >value</param>
        <param name=’szTOEPOCurrencyCode’ >value</param>
        <param name=’mnDRPKeyId’ >value</param>
        <param name=’mnSoldToCust’ >value</param>
        <param name=’szF4201BranchPlant’ >value</param>
        <param name=’szSoldToCurrencyCode’ >value</param>
        <param name=’cConsolidationFlag’ >value</param>
        <param name=’jdPriceEffectiveDate’ >value</param>
        <param name=’mnWOWFLineNo’ >value</param>
        <param name=’mnLineNoIncrement’ >value</param>
        <param name=’mnParentWFLineNo’ >value</param>
        <param name=’cStatusInWarehouse’ >value</param>
        <param name=’cBypassCommitments’ >value</param>
        <param name=’szProductSource’ >value</param>
        <param name=’szProductSourceType’ >value</param>
        <param name=’mnSequenceNumber’ >value</param>
```

```
<param name=’szAgreementNumber’ >value</param>
<param name=’mnAgreementSupplement’ >value</param>
<param name=’mnAgreementsFound’ >value</param>
<param name=’szModeOfTransport’ >value</param>
<param name=’szDutyStatus’ >value</param>
<param name=’szLineofBusiness’ >value</param>
<param name=’jdPromisedShip’ >value</param>
<param name=’szEndUse’ >value</param>
<param name=’mnTOEPOExchangeRate’ >value</param>
<param name=’szPriceCode1’ >value</param>
<param name=’szPriceCode2’ >value</param>
<param name=’szPriceCode3’ >value</param>
<param name=’szItemFlashMessage’ >value</param>
<param name=’szCompanyKeyRelated’ >value</param>
<param name=’szRelatedPoSoNumber >value</param>
<param name=’szRelatedOrderType >value</param>
<param name=’mnRelatedPoSoLineNo>value</param>
<param name=’cGenericChar3 >value</param>
<param name=’mnProfitMargin >value</param>
<param name=’mnQuantityAvailable >value</param>
<param name=’cRequestScheduleFlag >value</param>
<param name=’cOrderProcessType >value</param>
<param name=’cGenericChar2 >value</param>
<param name=’mnSODRelatedJobNumber >value</param>
<param name=’szGenericString >value</param>
<param name=’mnCarrier >value</param>
<param name=’szGenericString2_DL02 >value</param>
<param name=’mnGenericMathNumeric1_MATH01 >value</param>
<param name=’mnGenericMathNumeric2_MATH02 >value</param>
<param name=’mnItemVolume_ITVL >value</param>
<param name=’szVolumeUOM_VLUM >value</param>
<param name=’szRevenueBusinessUnit >value</param>
<param name=’szCustomerPO_VR01 >value</param>
<param name=’szReference2Vendor_VR02 >value</param>
<param name=’mnProcessID >value</param>
```

```
          <param name=’mnTransactionID >value</param>
      </params >
        <onError abort=’no’>\
        </onError >
  </callMethod >
  <callMethod name=’ F4211FSEndDoc’app=’NetCommerce’ runOnError=’no’>
      <params>
            <param name=’mnCMJobNo ‘ idref=’1‘></param>
            <param name=’mnSalesOrderNo’ >value</param>
            <param name=’szCMComputerID’ idref=’2’ >< /param>
            <param name=’cCMErrorCondition’ >value</param>
            <param name=’szOrderType’ >value</param>
            <param name=’szKeyCompany’ >value</param>
            <param name=’mnOrderTotal’ >value</param>
            <param name=’mnForeignOrderTotal’ >value</param>
            <param name=’szBaseCurrencyCode’ >value</param>
            <param name=’szProgramID’ >value</param>
            <param name=’szWorkstationID’ >value</param>
            <param name=’szCMProgramID’ >value</param>
            <param name=’szCMVersion’ >value</param>
            <param name=’mnTimeOfDay’ >value</param>
            <param name=’mnTotalCost’ >value</param>
            <param name=’mnForeignTotalCost’ >value</param>
            <param name=’cSuppressRlvBlnktFlag’ >value</param>
            <param name=’cWKSkipProcOptions’ >value</param>    (Skip Proc Opt,
            1=Yes)
            <param name=’mnWKRelatedOrderProcess’ >value</param>
            <param name=’cCMUseWorkFiles’ >value</param>(Req,Work File=1,
            Cache =2)
            <param name=’mnEDIDocNo’ >value</param>
            <param name=’szEDIKeyCo’ >value</param>
            <param name=’szEDIDocType’ >value</param>
            <param name=’cCMProcessEdits’ >value</param>
            <param name=’cGenericChar2’ >value</param>
            <param name=’mnSODRelatedJobNumber’ >value</param>
```

```
    <param name=’cGenericChar1_EV01’ >value</param>
    <param name=’mnGenericMathNumeric2_MATH02’ >value</param>
    <param name=’szGenericString1_DL01’ >value</param>
    <param name=’szGenericString2_DL02’ >value</param>
    <param name=’mnProcessID’ >value</param>
    <param name=’mnTransactionID’ >value</param>
  <params/>
  <onError abort=’no’>\
    <callMethod name=’ F4211ClearWorkFile’app=’NetCommerce’
            runOnError=’yes’>
      <params>
        <param name=’mnJobNo‘ idref=’1‘></param>
        <param name=’szComputerID’ idref=’2’ >< /param>
         <param name=’mnFromLineNo’ >value</param>
        <param name=’mnThruLineNo’ >value</param>
        <param name=’cClearHeaderWF’ >value</param>
        <param name=’cClearDetailWF’ >value</param>
        <param name=’szProgramID’ >value</param>
        <param name=’mnWKRelatedOrderProcess’ >value</param>
        <param name=’szCMVersion’ >value</param>
        <param name=’cGenericChar1_EV01’ >value</param>
        <param name=’szGenericString1_DL01’ >value</param>
        <param name=’mnSODRelatedJobNumber’ >value</param>
        <param name=’mnProcessID’ >value</param>
        <param name=’mnTransactionID’ >value</param>
      </params>
    </callMethod>
  </onError>
</callMethod>
<returnParams version=’value’ messagetype=’messsage name’’failure
Destination=’queuename’ successDestination=’queuename>
      <param name=’long description’ idref=’value’/param>
</returnParams>
<onError>
    <callMethod name=’ F4211ClearWorkFile’app=’NetCommerce’
```

```
               runOnError=’yes’>
          <params>
            <param name=’mnJobNo‘ idref=’1‘></param>
            <param name=’szComputerID’ idref=’2’ >< /param>
             <param name=’mnFromLineNo’ >value</param>
            <param name=’mnThruLineNo’ >value</param>
            <param name=’cClearHeaderWF’ >value</param>
            <param name=’cClearDetailWF’ >value</param>
            <param name=’szProgramID’ >value</param>
            <param name=’mnWKRelatedOrderProcess’ >value</param>
            <param name=’szCMVersion’ >value</param>
            <param name=’cGenericChar1_EV01’ >value</param>
            <param name=’szGenericString1_DL01’ >value</param>
            <param name=’mnSODRelatedJobNumber’ >value</param>
            <param name=’mnProcessID’ >value</param>
            <param name=’mnTransactionID’ >value</param>
          </params>
        </callMethod>
    </onError>
        </jdeRequest>
```

Outbound Customer Create XML Format (all fields)

This example illustrates an outbound XML format with all of the parameters.

Outbound Customer Create XML Request and Response Format

```
“This format will return all columns for the customer master (F0101Z2)“

<?xml version=’1.0’ ?>
 <jdeRequest type=’trans’ user=’user’ pwd=’password’ environment=’environment’
     session=’ ‘ sessionidle=’300’>
   <transaction action=’transactionInfo’ type=’JDEAB’>
              <key>
                    <column name=’EdiUserId’>value</column>
                    <column name=’EdiBatchNumber’>value</column>
                    <column name=’EdiTransactNumber’>value</column>
              </key>
    </transaction>
              </jdeRequest>
```

```
<?xml version='1.0' encoding='utf-8' ?>
<jdeResponse type='trans' user='user' session='session'   environment='env'>
      <transaction type='JDEAB' action='transactionInfo'>
              <returnCode code='0'>XML Request OK</returnCode>
            <key>
                   <column name='EdiUserId'></column>
                   <column name='EdiBatchNumber'></column>
            </key>
            <table name='F0101Z2' type='detail'>
                   <column name='EdiUserId'></column>
                  <column name='EdiBatchNumber'></column>
                  <column name='EdiTransactNumber'></column>
                  <column name='EdiLineNumber'></column>
                  <column name='EdiDocumentType'></column>
                  <column name='TypeTransaction'></column>
                <column name='EdiTranslationFormat'></column>
               <column name='EdiTransmissionDate'></column>
               <column name='DirectionIndicator'></column>
               <column name='EdiDetailLinesProcess'></column>
               <column name='EdiSuccessfullyProcess'></column>
               <column name='TradingPartnerId'></column>
               <column name='TransactionAction'></column>
               <column name='AddressNumber'></column>
               <column name='AlternateAddressKey'></column>
               <column name='TaxId'></column>
               <column name='NameAlpha'></column>
               <column name='DescripCompressed'></column>
               <column name='CostCenter'></column>
               <column name='StandardIndustryCode'></column>
               <column name='LanguagePreference'>< /column>
```

```
                    <column name='AddressType1'></column>
                    <column name='CreditMessage'></column>
                    <column name='PersonCorporationCode'></column>
                    <column name='AddressType2'></column>
                    <column name='AddressType3'></column>
                    <column name='AddressType4'></column>
                    <column name='AddressTypeReceivables'></column>
                    <column name='AddressType5'></column>
                    <column name='AddressTypePayables'></column>
                    <column name='AddTypeCode4Purch'></column>
                    <column name='MiscCode3'></column>
                    <column name='AddressTypeEmployee'></column>
                    <column name='SubledgerInactiveCode'></column>
                    <column name='DateBeginningEffective'></column>
                    <column name='AddressNumber1st'></column>
                    <column name='AddressNumber2nd'></column>
                    <column name='AddressNumber3rd'></column>
                    <column name='AddressNumber4th'></column>
                    <column name='AddressNumber6th'></column>
                    <column name='AddressNumber5th'></column>
                    <column name='ReportCodeAddBook001'></column>
                    <column name='ReportCodeAddBook002'></column>
                    <column name='ReportCodeAddBook003'></column>
                    <column name='ReportCodeAddBook004'></column>
                    <column name='ReportCodeAddBook005'></column>
                    <column name='ReportCodeAddBook006'></column>
                    <column name='ReportCodeAddBook007'></column>
                    <column name='ReportCodeAddBook008'></column>
                    <column name='ReportCodeAddBook009'></column>
                    <column name='ReportCodeAddBook010'></column>
                    <column name='ReportCodeAddBook011'></column>
                    <column name='ReportCodeAddBook012'></column>
                    <column name='ReportCodeAddBook013'></column>
                    <column name='ReportCodeAddBook014'></column>
                    <column name='ReportCodeAddBook015'></column>
                    <column name='ReportCodeAddBook016'></column>
                    <column name='ReportCodeAddBook017'></column>
                    <column name='ReportCodeAddBook018'></column>
                    <column name='ReportCodeAddBook019'></column>
                    <column name='ReportCodeAddBook020'></column>
                    <column name='CategoryCodeAddressBook2'></column>
                    <column name='CategoryCodeAddressBk22'></column>
                    <column name='CategoryCodeAddressBk23'></column>
                    <column name='CategoryCodeAddressBk24'></column>
                    <column name='CategoryCodeAddressBk25'></column>
                    <column name='CategoryCodeAddressBk26'></column>
                    <column name='CategoryCodeAddressBk27'></column>
                    <column name='CategoryCodeAddressBk28'></column>
                    <column name='CategoryCodeAddressBk29'></column>
                    <column name='CategoryCodeAddressBk30'></column>
                    <column name='GlBankAccount'></column>
                    <column name='TimeScheduledIn'></column>
                    <column name='DateScheduledIn'></column>
                    <column name='ActionMessageControl'></column>
```

```
                <column name='NameRemark'></column>
                <column name='CertificateTaxExempt'></column>
                <column name='TaxId2'></column>
                <column name='Kanjialpha'></column>
                <column name='UserReservedCode'></column>
                <column name='UserReservedDate'></column>
                <column name='UserReservedAmount'></column>
                <column name='UserReservedNumber'></column>
                <column name='UserReservedReference'></column>
                <column name='NameMailing'></column>
                <column name='SecondaryMailingName'></column>
                <column name='AddressLine1'></column>
                <column name='AddressLine2'></column>
                <column name='AddressLine3'></column>
                <column name='AddressLine4'></column>
                <column name='ZipCodePostal'></column>
                <column name='City'></column>
                <column name='Country'></column>
                <column name='State'></column>
                <column name='CountyAddress'></column>
                <column name='PhoneAreaCode1'></column>
                <column name='PhoneNumber'></column>
                <column name='PhoneNumberTyp1'></column>
                <column name='PhoneAreaCode2'></column>
                <column name='PhoneNumber1'></column>
                <column name='PhoneNumberTyp2'></column>
                <column name='TransactionOriginator'></column>
                <column name='UserId'></column>
                <column name='ProgramId'></column>
                <column name='WorkStationId'></column>
                <column name='DateUpdated'></column>
                <column name='TimeOfDay'></column>
                <column name='TimeLastUpdated'></column>
            </table>
      </transaction>
</jdeResponse>
```

Appendix B – Examples (Default Values)

Inbound Sales Order XML Format

This example uses the OneWorld default values. It omits the parameters that an external entity chooses not to fill.

```
  <?xml version="1.0" encoding="utf-8" ?>
- <jdeRequest type="callmethod" user="JDE" pwd="JDE"
     environment="PRD733">
  - <callMethod name="GetLocalComputerId" app="NetComm"
       runOnError="no">
    - <params>
        <param name="szMachineKey" id="2" />
      </params>
      <onError abort="yes" />
    </callMethod>
  - <callMethod name="F4211FSBeginDoc" app="NetComm"
       runOnError="no">
    - <params>
        <param name="mnCMJobNumber" id="1" />
        <param name="cCMDocAction">A</param>
        <param name="cCMProcessEdits">1</param>
        <param name="szCMComputerID" idref="2" />
        <param name="cCMUpdateWriteToWF">2</param>
        <param name="szCMProgramID">NetComm</param>
        <param name="szCMVersion">ZJDE0001</param>
        <param name="szOrderType">SO</param>
        <param name="szBusinessUnit">M30</param>
        <param name="mnAddressNumber">4242</param>
        <param name="jdOrderDate">2000/01/21</param>
        <param name="szReference">10261</param>
        <param name="cApplyFreightYN">Y</param>
        <param name="szCurrencyCode">CAD</param>
        <param name="cWKSourceOfData" />
        <param name="cWKProcMode" />
        <param name="mnWKSuppressProcess">0</param>
      </params>
    - <onError abort="yes">
      - <callMethod name="F4211ClearWorkFile" app="NetComm"
           runOnError="yes">
        - <params>
            <param name="mnJobNo" idref="1" />
            <param name="szComputerID" idref="2" />
            <param name="mnFromLineNo">0</param>
```

```
              <param name="mnThruLineNo">0</param>
              <param name="cClearHeaderWF">2</param>
              <param name="cClearDetailWF">2</param>
              <param
                name="szProgramID">NetComm</param>
              <param
                name="szCMVersion">ZJDE0001</param>
            </params>
          </callMethod>
        </onError>
      </callMethod>
    - <callMethod name="F4211FSEditLine" app="NetComm"
        runOnError="yes">
      - <params>
          <param name="mnCMJobNo" idref="1" />
          <param name="cCMLineAction">A</param>
          <param name="cCMProcessEdits">1</param>
          <param name="cCMWriteToWFFlag">2</param>
          <param name="szCMComputerID" idref="2" />
          <param name="szItemNo">1001</param>
          <param name="mnQtyOrdered">1</param>
          <param name="cSalesTaxableYN">N</param>
          <param name="szTransactionUOM">EA</param>
          <param name="szCMProgramID">NetComm</param>
          <param name="szCMVersion">ZJDE0001</param>
          <param name="cWKSourceOfData" />
        </params>
        <onError abort="no" />
      </callMethod>
    - <callMethod name="F4211FSEditLine" app="NetComm"
        runOnError="yes">
      - <params>
          <param name="mnCMJobNo" idref="1" />
          <param name="cCMLineAction">A</param>
          <param name="cCMProcessEdits">1</param>
          <param name="cCMWriteToWFFlag">2</param>
          <param name="szCMComputerID" idref="2" />
          <param name="szItemNo">1001</param>
          <param name="mnQtyOrdered">10</param>
          <param name="cSalesTaxableYN">N</param>
          <param name="szTransactionUOM">EA</param>
          <param name="szCMProgramID">NetComm</param>
          <param name="szCMVersion">ZJDE0001</param>
          <param name="cWKSourceOfData" />
        </params>
        <onError abort="no" />
      </callMethod>
    - <callMethod name="F4211FSEndDoc" app="NetComm"
        runOnError="no">
      - <params>
          <param name="mnCMJobNo" idref="1" />
          <param name="szCMComputerID" idref="2" />
          <param name="szCMProgramID">NetComm</param>
          <param name="szCMVersion">ZJDE0001</param>
          <param name="cCMUseWorkFiles">2</param>
        </params>
      - <onError abort="no">
        - <callMethod name="F4211ClearWorkFile" app="NetComm"
            runOnError="yes">
          - <params>
              <param name="mnJobNo" idref="1" />
              <param name="szComputerID" idref="2" />
              <param name="mnFromLineNo">0</param>
              <param name="mnThruLineNo">0</param>
              <param name="cClearHeaderWF">2</param>
              <param name="cClearDetailWF">2</param>
              <param
                name="szProgramID">NetComm</param>
              <param
                name="szCMVersion">ZJDE0001</param>
            </params>
          </callMethod>
        </onError>
      </callMethod>
      <returnParams failureDestination="ERROR.Q"
        successDestination="SUCCESS.Q" runOnError="yes" />
    - <onError abort="yes">
      - <callMethod name="F4211ClearWorkFile" app="NetComm"
          runOnError="yes">
        - <params>
            <param name="mnJobNo" idref="1" />
            <param name="szComputerID" idref="2" />
            <param name="mnFromLineNo">0</param>
            <param name="mnThruLineNo">0</param>
            <param name="cClearHeaderWF">2</param>
            <param name="cClearDetailWF">2</param>
            <param name="szProgramID">NetComm</param>
            <param name="szCMVersion">ZJDE0001</param>
          </params>
        </callMethod>
      </onError>
    </jdeRequest>
```