Infor Distribution A+ Configuration Guide for Infor ION

Distribution A+ 10.03.02

ION 12.x

Contents

About this guide

This guide provides configuration and implementation information for the integration of Distribution A+ with Infor ION when ION is used to exchange data with another Infor product or third-party product. This document describes configuration requirements and provides setup instructions. It describes the ION connection points that are used in the integration. This document provides information about the business events or user actions in Distribution A+ that send Business Object Documents (BODs) to ION Connect.

The technical chapters of this guide describe integration methodologies and best practice requirements specific to Distribution A+ providing architectural information about Distribution A+ and ION as well as more information about business object documents (BODs) exchange between Distribution A+ and ION.

Use this guide when Distribution A+ is installed on-premises, and Infor OS is installed on-premises or in the cloud. Instructions specific to the Infor OS deployment type are noted where they differ.

Intended audience

This guide is intended for the system administrator or consultant who configures Distribution A+ for use with ION. Before you read this guide, ensure that you are familiar with the other guides listed in "Related documents."

The technical chapters of this guide are intended for developers who implement and extend the integration between Distribution A+ and other ION applications, or who need to troubleshoot and trace data flow between Distribution A+ and ION.

Required knowledge

To configure Distribution A+ with ION, you must have experience or knowledge in these areas:

- Understand the concepts behind ION and BODs, and how the concepts relate to this application. See these topics:
  - "BOD Overview" on page 89

      o   "Infor ION overview" on page 18

Related documents

You can find the documents in the product documentation section of the Infor Support Portal, as described in "Contacting Infor" on page 12.

For more information about Distribution A+, see these documents:

- *Infor Distribution A+ Installation and Software Update Guide*

- *Infor Distribution A+ GUI Installation and Upgrade Guide*

- *Infor Distribution A+ Workflow Management User Guide*, if integrating Distribution A+ Workflow with ION

- *Infor Distribution A+ Integration Guide to Infor Ming.le*

- *Commerce Gateway Installation and Configuration Guide*

- *Infor Distribution A+ Extension Solution User Guide*

- *Infor Distribution A+ Cross Applications User Guide*

- *Infor Distribution A+ Inventory Accounting User Guide*

ION documents

For more information about ION, see these documents:

- *ION Desk User Guide*
- *Infor Operating Service Administration Guide*
- *Infor ION Federation Services Administration Guide*
- *Infor ION Federation Services Installation Guide*
- *Infor ION Installation Guide*
- *Infor Ming.le Administrator Guide*
- *Infor Ming.le User Guide*

Documents related to ION integrations

This table shows the types of documents that are related to ION integrations and their purpose:

Document type	Purpose
Infor Distribution A+ Outbound BOD Mapping and Descriptions	For each BOD that is sent from Distribution A+, this document provides the source and definition of the data in each element that is sent to ION. This document also provides details about the data that is published.
Infor Distribution A+ Inbound BOD Mapping and Descriptions	For each BOD that Distribution A+ has certified, this document provides the details about how each product receives information or processes requests from the inbound BOD. For example, the source might be an extension product, a result of a workflow, or a drillback function from Infor Ming.le. If a BOD updates a source table, the exceptions and handling are described.
Infor Distribution A+ *C*ross BOD mapping and descriptions	These product specific documents provide end-to-end integrated BOD mapping details for BODs that are exchanged between two Infor products or between an Infor product and an Infor partner product. They describe the exceptions and handling requirements by element.
Infor Distribution A+ ION workflow and alert content examples for Infor ION	The Distribution A+ Workflow Management Integration with ION Desk Alerts and Workflows chapter provides information on the event monitors and workflow models that have been created for the BODs that are published by this product. This chapter includes recommendations on how to customize the standard ION content to meet your requirements.
Integration guides for integrations between Distribution A+ and the following products: • Infor CRM • Infor CRM Business Extension • Infor Inforce • Infor Ming.le	These documents provide details about integrations between this product and another product. They contain requirements, configuration instructions, an overview of the integration, and any verification or troubleshooting steps.
Infor Distribution A+ Extension Solution User Guide	This guide provides an overview of the Infor product and its components. This document can include information that can be used to customize the integration.

Not all of these documents are required to set up this integration. For a list of documents that are required for this integration, see Chapter 2 Integration checklist.

Contacting Infor

If you have questions about Infor products, go to the Infor Support Portal at https://support.infor.com/.

If we update this document after the product release, we will post the new version on this website. We recommend that you check this website periodically for updated documentation.

If you have comments about Infor documentation, contact documentation@infor.com.

Chapter 1 Requirements

This section describes the products and other information that you require in order to complete the tasks in this guide.

Required products

Infor ION is one of the products included in the Infor technology platform - Infor Operating Service (OS). Infor Operating Service release 12.0.37 or later, including Infor ION, can be deployed on your own server (on-premises), or in the cloud.

The installation of Infor OS also includes Infor Ming.le. The optional integration of Distribution A+ WEB through Infor Ming.le allows you to capitalize on the ability to share information from Distribution A+ WEB to the company infrastructure for quick response to situations that occur daily within the workplace. See the Infor Distribution A+ Integration Guide for Infor Ming.le.

The server where you install Infor OS must be able to connect to the servers where you install Distribution A+ and other BOD-enabled products. See the *Infor Operating Service Installation Guide*.

If you are integrating Distribution A+ with Infor OS in the cloud, the Infor Cloud team provisions Infor OS and its components for you. You must configure Infor OS to communicate with your on premises application using the instructions in this guide (see chapter *Configuring your application to send and receive BODs using a hybrid ION configuration*).

- Distribution A+ supported version is 10.03.02 or higher.
- Commerce Gateway 7.4.11 or later
- The application with which you are integrating Distribution A+ (i.e. Infor Ming.le, Inforce, ICB, etc.)
  See the *Infor Distribution A+ Installation and Software Update Guide* for instructions on updating to a supported version.

Required Downloads

The following images must be downloaded from the Infor Product Download Center:

- Infor Distribution A+ Connection Point 12 or later
- Commerce Gateway 7.4.11 or later
- Any required software as outlined in the Installation and Configuration guide(s) for the Infor application with which you are integrating

Required information

During the planning phase, you must provide this information:

- Name of the Distribution A+ Base and Environment ID that will be connected to ION
- Logical ID that is used to connect with ION
- Tenant ID that is used to connect the application to ION
- Enterprise organization structure, that is, accounting entities
  - Distribution A+ company number and general ledger account segments

IBM i Access for Windows Limitations

There have been problems reported accessing ION Desk after the installation of some versions of the IBM i Access for Windows, when the IBM i Access for Windows .NET Data Provider is selected as part of the installation.

If you install IBM i Access for Windows on the ION server, do not include the IBM i Access for Windows .NET Data Provider driver during the install. Also, do not perform a 'Complete' installation of the IBM i Access for Windows. If the IBM i Access for Windows .NET Data Provider is needed to integrate with another product, see IBM's website for more information.

Also, if you are integrating ION with Infor ICB, you cannot install both ION and ICB on the same server. ION and ICB must be installed on different servers for a successful integration.

For back release installations prior to Distribution A+ v08.03.03, any purchase orders approved and printed from the ION Activity Deck will be sent to the IBM i User ID's default output queue, if one was not entered in ION.

Chapter 2   Integration checklist

<div style="text-align:right">2</div>

Follow this checklist to integrate this product with ION:

Complete	Task	Reference
☐	Collect the documents listed in the next column from the Infor Support Portal. These documents are required for this configuration.	"ION documents" on page 10
☐	Review the description of the Distribution A+ configuration with ION.	"Integration with other applications through Infor ION" on page 18 "Concepts and definitions specific to this ION configuration" on page 21
☐	Confirm that the version of Commerce Gateway specified in the Required products section of this guide is functional.	*Commerce Gateway Installation and Configuration Guide*
☐	Configure Distribution A+ for integration with ION.	"Configuring Distribution A+ " on page 23
☐	Install and configure the Connection Point to the Infor application to which you are integrating.	Follow the instructions in the Connecting Distribution A+ to ION chapter on page 31 of this document to create the ION Connection Point for that application.
☐	Generate and activate ION document flows between Distribution A+ and other applications.	"Connecting Distribution A+ to ION" on page 31
☐	Publish BODs.	"Publishing BODs" on page 39
☐	Verify the configuration.	"Verifying the configuration" on page 45 *Infor ION Connect Administration Guide*

Complete	Task	Reference
☐	Generate Distribution A+ Workflow Alerts to ION	"Distribution A+ Workflow Management Integration with ION Desk Alerts and Workflows" on page 59

Chapter 3   Overview

This section includes an overview of Infor ION, an overview of

 how Distribution A+ integrates with other applications through ION, and a list of the concepts and definitions that you must understand before you complete the configuration tasks.

Architectural Diagram
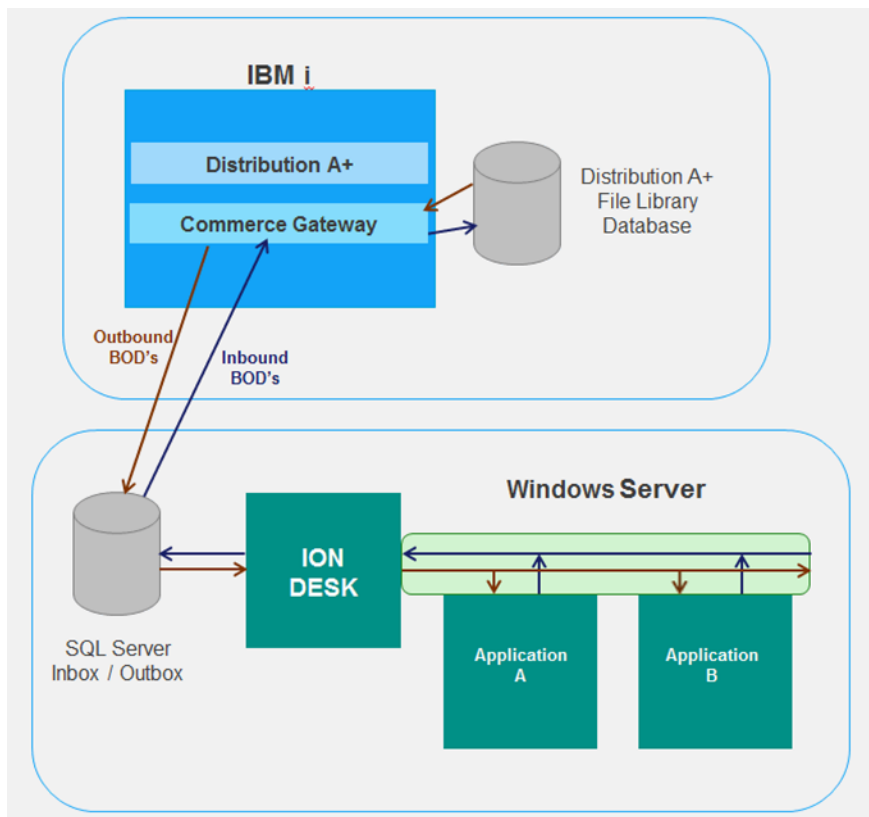
Infor ION overview

Intelligent Open Network (ION) is an enterprise messaging system that integrates Infor applications with each other and with non-Infor applications. Transactional and master data is passed between applications as business object documents (BODs) that are routed through ION. ION also enables customers to set up workflows, design and activate business event monitors, and manage tasks and alerts across applications. ION includes these components:

- ION Desk is a browser-based user interface that you use to work with ION components. You can use ION Desk to configure and manage ION services, configure the routing of messages, monitor message activity, view all errors published by applications, and manage other ION services.
- ION Connect is the component that you use to set up connections to various application databases and create document flows that collect and send data between applications. Through ION Connect, data is shared securely between applications. ION Connect routes and delivers messages to the appropriate applications by using data flows that you define. It communicates with applications by using message inboxes and outboxes.
- ION Workflow is the component that you use to set up workflows for automated business task routings and approvals.
- ION Event Management is the component that you use to monitor the completion of business tasks and to alert users about exceptions.
- ION Activity Deck is the component that you use to view and handle tasks, alerts, and notifications, and to drill down to applications.

For more information about these components, see the Infor ION documentation.

For a conceptual overview of what BODs are and how they are used, see "BOD Overview" on page 89.

Integration with other applications through Infor ION

An outbound operation typically begins when a user performs an action in Distribution A+ that requires a data exchange with another ION-enabled application. A BOD XML message is generated by Distribution A+ and placed in an area designated as the Distribution A+ messages outbox. At scheduled intervals, ION connects to the outbox and retrieves the BODs from it.

In ION, you create application connection points, which define the connections between ION and an application that can send and receive BODs. ION Connect routes BODs according to the document flows between Distribution A+ and other ION-enabled applications.

The document flows between Distribution A+ and other applications represent the business relationship between the databases. You use the ION Document Flow page to define these document flows.

If a document flow is defined from Distribution A+ to another application for a particular BOD, then at specified intervals, ION places the outbound BOD from Distribution A+ in the other application's

designated message inbox. Applications are responsible for validating and incorporating the data in inbound BODs according to their rules.

If a flow is defined from another application to Distribution A+, ION retrieves BODs from the sending application's message outbox and delivers them to Distribution A+'s message inbox for processing. Distribution A+ retrieves, validates, and processes the BODs.

ION Content Packs

In ION Desk, Infor Distribution A+ is defined as an Application Connection Point. The typical connection information for Distribution A+, such as logical ID, hostname, environment, username, and password are specified in ION.

At least one Document Flow for the Aplus Connection Point must be defined in ION.

Distribution A+ delivers a content pack that provides a set of 16 ION alert monitors specially designed for distributors.

Setup and configuration requirements are outlined in the following chapters.

Drill Backs

In some ION-enabled applications that are integrated with Distribution A+, users can drillback from that application to Distribution A+. If the other application is displaying transactional data that was generated based on a BOD sent from Distribution A+, you might be able to drillback from that application to view detailed supporting information in Distribution A+ for the corresponding transactions.

For information about setting up and using the drillback functionality, see the *Infor Distribution A+ Integration Guide for Infor Ming.le*.

Event Triggers

Outbound Distribution A+ Business Object Documents are generated from key application events (Creation of a Sales Order, Maintaining a Customer, Posting AR, etc.). The way Distribution A+ determines how and when to fire an event is determined based on database triggers setup in Distribution A+ Extension Solution BOD Event Registration (MENU ESFILE). This Distribution A+ option will allow you to specify a file to trigger (Customer Master File (CUSMS), Item Master File (ITMST), Open Order Header File (ORHED), etc.), the action to trigger an event (Create, Update, Delete) and when to send the BOD based on the Event Program. Refer to the *Infor Distribution A+ Extension Solution User Guide* for more information.

BOD Outbound Mappers

BOD Mappers are RPG Programs which will pre-process BOD data sent to ION. The BOD Mapper programs will prepare and write data to 3 files: BOD Outbound Header (BODOHD), BOD Outbound Detail (BODODT), and BOD Outbound BOD Attribute (BODOAT). These files are then used by Commerce Gateway routines to build well-formed XML and place the data into the SQL Server Outbox.

Extension Solution On-Ramp Service Processor

The Extension Solution On-Ramp Service Processor is a multi-threaded processor used to send and receive BODs to and from ION and Distribution A+. More information can be found on the Extension Solution Processor in the *Infor Distribution A+ Extension Solution User Guide*.

One additional note about the processor is that any BODs sent to or from Distribution A+ are logged and included in Job Statistics. The Processor will log multiple outbound activities including Event Create, Ready for Create, BOD Created, Sent to Gateway, and Sent to BUS as these are all time stamped logged and can be viewed when drilling into the noun transaction detail.

Commerce Gateway

The Commerce Gateway is a set of java programs used to build XML from the pre-processed tables generated from the BOD Mapper Programs and can update/read from the ION Application Inbox/ Outbox. Commerce Gateway will run on the IBM i in the same subsystem as the ESB Processor and can be started when the ESB Processor starts by configuring the Extension Solution Start up Options (MENU ESFILE). Additional configuration of Commerce Gateway can be done using the gateway.xml file in the Gateway folder on the IBM i. Refer to the *Commerce Gateway Installation and Configuration Guide* for further details on configuring the Commerce Gateway.

Inbox / Outbox

Distribution A+ communicates Business Object Documents (BODs) to and from ION using the standard Infor Application Connector. The Infor Application Connector uses the following set of SQL Server Tables for this communication:

- COR_OUTBOX_ENTRY

- COR_OUTBOX_HEADERS

- COR_INBOX_ENTRY

- COR_INBOX_HEADERS

- ESB_INBOUND_DUPLICATE

Please refer to the *ION Technical Reference Guide* for more details on these tables and how ION uses them to facilitate BOD flow.

Concepts and definitions specific to this ION configuration

To configure the Distribution A+ tenant, logical ID, accounting entity and location correctly, you must understand these terms and how they are defined in Distribution A+. Together, these terms determine where inbound BOD information is processed, or the outbound instance and location to which the BOD information should be associated.

Infor applications integrate by establishing an exchange of business object documents (BODs). These BODs use the logical IDs, tenants, accounting entities and locations to identify the system that sends messages and the system that receives messages. Before you begin the integration process, you need to determine how Distribution A+ environments, companies, and locations relate to these definitions for BOD-enabled applications.

Tenant ID

The tenant is a container for accounting entities and locations and is required in each BOD. The value of the tenant must be the same in all of the applications that exchange BODs. Data is not shared or accessible between tenants.

No data is ever shared or accessible between two tenants, and this is useful in environments where more than one customer is sharing an enterprise software deployment, such as hosting.

For example, tenant can be defined as infor.

Logical ID

The logical ID is the instance of the application that sends or receives BODs. You must provide the logical ID when you define the ION connection point for on premise applications. For cloud applications, the logical ID is defined by the Infor Cloud team and is retrieved automatically when the application connection point is created in ION.

In BOD-enabled applications, the logical ID is a number or name that uniquely identifies an instance of the core product Inbound/Outbound database.

In Distribution A+, the logical ID is defined as infor.aplus.xxxxx; where infor is a tenant, aplus is the application type (Distribution A+), and xxxxx is a unique identifier that no other Infor application instance that Distribution A+ is integrating to is using. For example, logical ID can be defined as infor.aplus.live; it can indicate the Distribution A+ base and environment to which you are integrating and be infor.aplus.BBEE, where BB indicates your Distribution A+ Base ID and EE is the Environment, or it can be defined as infor.aplus.1 with a value as simple as a number to make it unique.

This ID must match the Logical ID specified in the ION Connection Points.

Accounting Entity

In BOD-enabled applications, the accounting entity is a corporation or a subset of a corporation that is independent in regard to one or more operational functions or accounting functions. An accounting entity is an entity which produces a profit and loss and balance sheet from a complete, balanced set of transactions, and is often a legal entity. An accounting entity in Distribution A+ is defined as a company.

Location

A location is the physical location that is associated with data or transactions. The location can be, for example, a warehouse, a manufacturing location, a project location, or an office. Locations are published in the Location BOD. A location is required for all transactional BODs and is usually the location from which a transaction or record is generated.

In BOD-enabled applications, the location is a single geographical site of an organization that is associated with data or transaction. Items are being shipped to or from a specific location. In Distribution A+ location is defined as a warehouse.

Chapter 4 Configuring Distribution A+

4

Use the steps in this section to set up Distribution A+ so that it can generate BODs and place them in a message outbox where ION Connect can retrieve them.

Setting master data to ISO Values

To ensure consistent master data between integrated applications, use the ISO standard values where applicable.

As each ERP can have its own forms of Master Data such as units of measure and currencies, one requirement of the integration enablement is to standardize the common types of master data using standard ISO codes.

As part of this standardization of the data before sending out BODs from Distribution A+, we need to replace Distribution A+ specific non-standard codes with their standard equivalent. And, when receiving Inbound BODs published by other Infor applications, standard ISO codes will be replaced with Distribution A+ specific codes, if different.

To do this replacement, the following needs to be established in Distribution A+:

- A list of standard ISO codes for each common code type used in your integration(s). Currently common code types are units of measures, currencies and country codes.

- A cross-reference between non-standard codes in Distribution A+ and standard ISO codes.

To accomplish this, the following initial steps need to be followed:

1. Sign on to the IBM i as QPGMR.

2. Run program FXSOACTYP to populate records in the SOA Code Types File (SOACTYP).

   a. From a command line key: CALL FXSOACTYP ENTER

3. On the IBM i IFS\ROOT directory, create a new directory named soacode

   a. From a command line key: CRTDIR DIR ('SOACODE') ENTER

4. Access the IFS\ROOT\soacode directory. Three text files need to be created for the standard ISO codes. You can manually create 3 empty text files using any text editor (i.e. Notepad), one for each SOA Code Type in the soacode directory. The file content should be

a list with standard ISO codes (where each code exists as a separate line). The file name should be [SOA CodeType].txt. Refer to the *Infor Distribution A+ Extension Solution User Guide* Appendix chapter for sample ISO code data.

- UNMS.txt - unit of measure codes
- CURR.txt - currency codes
- CTID.txt – country codes

Or, you can copy pre-filled files from the Distribution A+ installation ISO images. From the Work with Image Catalog Entries Screen, mount the first volume of the install or patch ISO image following the directions in the *Infor Distribution A+ Installation and Software Update Guide*, Install/Update Process Steps of Appendix B. Using the mounted image, select Option 12 to display the Work with Optical Volumes Screen. Select Option 11 to display the Work with Object Links Screen. Page down until you see the CTID.txt, CURR.txt and/or UNMS.txt files listed in the Object link column. Select Option 3 to copy the file from the ISO image to the \IFS\Root\ directory. A message 'Object copied.' displays in the lower left corner of the screen. The text files are placed at the base level of the \IFS \Root directory where you can then move them to the IFS\ROOT\soacode directory.

These text files can be found on the following volumes of Distribution A+ CD images:

- APINSTVOL1 – Volume 1 of the Full installation/upgrade of Distribution A+

5. Import Standard Codes (MENU ESFILE) (quick key ISC)

   This program imports data from ISO files placed in the IFS\ROOT\soacode directory and populates Standard Codes in Distribution A+. Run this program separately for each SOA Code Type (CTID, CURR, and UNMS).

   NOTE: This program also creates a SQL script for each Standard Code added.

   If Integrating to PM Dashboards, this SQL script can be used to insert data into Reporting Database Code table (used for parameters in Metrics/Reports). The script is created in the IFS\ROOT\soacode directory.

6. Non-Standard Codes Report (MENU ESFILE) (quick key NSC)

   This report lists codes in Distribution A+ that do not match any of the Standard ISO Codes. Run this program for each SOA Code Type (CTID, CURR, and UNMS) to detect codes in Distribution A+ that requires a cross-reference to standard ISO codes.

7. Code Cross Reference (MENU ESFILE) (quick key CXREF)

   Use this option to create cross-references between non-standard codes and standard ISO codes. This should be done for each non-standard code listed in the report created in the previous step.

   Select the standard ISO **Code Type** on the prompt screen. On the Code Selection Screen, select the corresponding Distribution A+ code and map it to the Standards Agency Code provided in the list.

Important: Going forward, when a new Non-Standard Code is added in Distribution A+, a Cross-Reference record should be added as well to ensure that Distribution A+ BODs that use this code will be accepted.

Setting up logical IDs, tenants, and accounting entities in Distribution A+

Refer to the *Infor Distribution A+ Extension Solution User Guide* for more detailed information.

Set up Logical ID and Tenant ID Information

1. Access Extension Solution Options Maintenance (MENU ESFILE) (quick key SOAOM)

    a. Key **Option Type S** for system options.

    b. Enter the Tenant ID and Logical ID information.

Set up Accounting Entity

1. Access Extension Solution Options Maintenance (MENU ESFILE) (quick key SOAOM)

    a. Key **Option Type C** for company options.

    b. Select the appropriate company number.

    c. Key the **Accounting Entity ID** (company number).

    d. Repeat for each company to be integrated.

Set up Distribution A+

Use the steps in this section to configure Distribution A+ to generate BODs and place them in the Distribution A+ outbox table of the Commerce Gateway database where ION can retrieve them, and to consume BODs that ION is placing in Distribution A+ / Commerce Gateway inbox table.

Assign Integration Administrator user(s) that will be responsible for monitoring ION Desk, On-Ramp Service Processor Inquiry and Inbound BOD Inquiry to ensure that outbound BODs are published and Inbound BODs are consumed (posted) successfully and without delay.

This section must be done for each Base, Environment, Company and Warehouse that will be enabled for Integration with other Infor applications.

Log in to Distribution A+ as a user who has authority to the following functions and is authorized to the companies and warehouses that will be specified.

Set up Company Address Information

Refer to the *Infor Distribution A+ Cross Applications User Guide* for more detailed information.

1. Access Company Name Maintenance (MENU XAFILE) (quick key CNM)

    a. Select a company.

    b. Enter address information if not already entered.

    c. Repeat for each company to be integrated.

Set up Warehouse Address Information

Refer to the *Infor Distribution A+ Inventory Accounting User Guide* for more detailed information.

1. Access Warehouse Numbers Maintenance (MENU IAFILE) (quick key WNM)

    a. Select a warehouse.

    b. Enter address information if not already entered.

    c. Repeat for each warehouse to be integrated.

Define Extension Solution Options

1. Extension Solution Options Maintenance (MENU ESFILE) (quick key SOAOM)

    This option provides 4 separate set-up options: system options, company options, warehouse options and BOD options. Complete each set-up option for each Distribution A+ value that will be exchanging data with an Infor application(s).

    Refer to the *Infor Distribution A+ Extension Solution User Guide* for additional information.

    a. Select System Options – add the system options data that will establish the matching criteria for exchanging data.

    b. Select Company Options - repeat for each company that will exchange data with other Infor application(s). Select the **Language Code** and **Currency Code** that represents the location of the specific company.

    c. Select Warehouse Options - repeat for each warehouse that will exchange data with other Infor application(s). Select the **Accounting Entity ID**, **Language Code** and **Currency Code** that represents the location of the specific warehouse.

    d. Select BOD Options - use to identify which Incoming and Outgoing BODS will be

exchanged and the setup requirements for those transactions.

Outbound BODs

By default, all active Outbound BODs are sent to Gateway for publishing and related data is purged immediately. Maintain these options only for Outbound BODs that you need to override default settings. This option is useful when testing or debugging a noun or noun/verb. You can optionally choose not to send outbound data out of Distribution A+ and/or purge data from outbound files which can be reviewed to see if the BOD has been created correctly. Reviewing the BOD data in the outbound files is a manual process.

Inbound BODs

There are no default settings for Inbound BODs. You need to provide options for each Inbound BOD that needs to be consumed (posted) by Distribution A+ (updating related Database records in Distribution A+. It is recommended that you complete this step for the following Nouns: CustomerPartyMaster, ContactMaster, and ShipToPartyMaster.

Define Extension Solution Startup Options

1. Extension Solution Startup Options (MENU ESFILE) (quick key SOASO)

Use this option to specify the location of the Commerce Gateway Client files, and if the Commerce Gateway Client Engine should be automatically started.

Refer to the *Infor Distribution A+ Extension Solution User Guide* for additional information.

Register Custom Events

1. Extension Solution BOD Event Registration (MENU ESFILE) (quick key BER)

Within the Distribution A+ application, certain events (user actions) that add, change or delete business entities, like Customers, Items, Sales Orders etc. will trigger generating and publishing of related outbound BODs from Distribution A+. These events are listed in the "Distribution A+ business events that generate publishing of outbound BODs" topic in the Publishing BODs chapter of this guide. Publishing BODs communicates to other integrated applications the latest state of related business entities.

If you have modifications to Distribution A+ that add, change or delete data to files in Distribution A+ that are used to publish BODs, and these modifications do not trigger events listed in the provided list of outbound BODs, you need to create custom events.

Refer to the *Infor Distribution A+ Extension Solution User Guide* for additional information.

Set up the On-Ramp Service Processor

1. On-Ramp Service Processor Setup (MENU ESFILE) (quick key ORSPS)

   This is an interactive job with no prompt screens; your session is inhibited while processing. Your user profile must have Security Administrator authority in order to run this option. A break message will display if you do not have this authority, and the menu option will be canceled; else the job will run interactivity on your screen.

   Refer to the *Infor Distribution A+ Extension Solution User Guide* for additional information.

Configure the On-Ramp Service Processor

1. On-Ramp Service Processor Configuration (MENU ESFILE) (quick key ORSPC)

   a. Select On-Ramp Service Processor Assignments - Repeat for each environment to be connected. Use this option to identify the processors you want to assign/activate for the indicated Environment ID; then configure the Noun Assignments and processing priority for the On-Ramp Service Processor.

   b. Select On-Ramp Service Processor Priorities - Use this option to identify the priority each selected processor should fun at.

   Refer to the *Infor Distribution A+ Extension Solution User Guide* for additional information.

Create Auto Purge job for the On-Ramp Service Processor

1. Auto Purge On-Ramp Service Processor (MENU ESMAST) (quick key APORP)

   Use to set up automatic purges that will run on a regularly scheduled basis. Records will be automatically purged so they don't occupy excessive space on the IBM i. Identify Inbound, Outbound, or Both, the Noun and Verb, and if there are any exceptions to the selected Noun. The auto-purge is submitted to the IBM i Job Scheduler. Refer to the *Infor Distribution A+ Extension Solution User Guide* for additional information.

2. Auto Purge Noun Exceptions – Select by pressing F5=NOUN EXCEPTIONS from the Auto Purge On-Ramp Service Processor Screen. This step does not need to be set up unless there is a need for it.

Create Auto Purge job for the BOD Transactions

1. Access Auto Purge BOD Transactions (MENU ESMAST) (quick key APBT)

   Use to set up automatic purges for the Extension Solution BOD files that will run on a regularly scheduled basis. Records will be automatically purged from the BOD files so they don't occupy excessive space on the IBM i. Identify Inbound, Outbound, or Both, the Noun

and Verb, and if there are any exceptions to the selected Noun. The auto-purge is submitted to the IBM i Job Scheduler. Refer to the *Infor Distribution A+ Extension Solution User Guide* for additional information.

2. Auto Purge Noun Exceptions – Select by pressing F5=NOUN EXCEPTIONS from the Auto Purge BOD Transactions. This step does not need to be set up unless there is a need for it.

Distribution A+ Workflow Management

The Distribution A+ Workflow Management module supports the creation of the AplusWorkflow Business Object Document (BOD). The AplusWorkflow BOD is sent to ION, and subsequently used in ION ActivityDeck tasks and alerts.

There are additional unique set up steps required to complete this interface. Refer to the Distribution A+ Workflow Management Integration with ION Desk Alerts and Workflows chapter of this guide for those specific instructions.

Setting up the system to send initial data

During normal processing, BODs are generated when a user makes a data change or completes a transaction. However, when you start a new integration with another product, you must manually generate BODs that pass the current set of Distribution A+ data to the other application, so that the base systems are synchronized. The BODs are placed in the message outbox, from which ION retrieves them and passes them to the other application.

Distribution A+ Extension Solution Data Refresh

It is a common occurrence in Distribution A+ for a new file environment to be created from an existing file environment or for a test file environment to be created from a live file environment. File Migration Tools exist in Distribution A+ to assist the user with these tasks. However, the Extension Solution integration to Infor ION requires additional steps to be taken when the original file environment was part of an Infor ION Integration.

Additionally, within an Infor ION integrated Distribution A+ file environment, occasionally, Inbound or Outbound BOD transactions are mistakenly created and need to be cleared and removed from the Inbox or Outbox before the transactions are processed.

The Extension Solution module provides tools to clear and validate the Extension Solution files. But additional steps may need to be taken in the Commerce Gateway Server SQL database. Consider the following:

1. When clearing erroneous transactions, there are tables in the Commerce Gateway Server SQL database that will also need to be cleared by accessing the database on the server.

    a. COR_OUTBOX_ENTRY

    b. COR_OUTBOX_HEADERS

    c. COR_INBOX_ENTRY

    d. COR_INBOX_HEADERS

2. When creating a new file environment from an existing integrated file environment, the tables in the Commerce Gateway Server SQL database also need to be cleared (see step 1).

3. When refreshing an existing file environment from an existing integrated file environment, the tables in the Commerce Gateway Server SQL database also need to be cleared (see step 1).

4. When creating a new file environment from an existing integrated file environment, and the new file environment will also be an integrated file environment, the tables in the Commerce Gateway Server SQL database also need to be cleared (see step 1). Additionally, a new Commerce Gateway will need to be loaded to process the transactions for the new environment independently from the original.

Chapter 5 Connecting Distribution A+ to ION

This section describes how to perform these tasks in ION:

- Configure a connection point for the Distribution A+ application.
- Optionally, set up and activate a document flow between applications.

Configuring Distribution A+ to send and receive BODs using a hybrid ION configuration

If you are integrating Distribution A+ with Infor OS in the cloud, the Infor Cloud team provisions Infor OS and its components for you. You must configure Infor OS to communicate with your on premises application.

A hybrid ION integration is required if you want to integrate your Distribution A+ on-premises application with a cloud application, such as CRM. For details about a specific integration, see the appropriate integration guide.

A connection point must be set up for Distribution A+ so that it can send and receive documents to applications running in Infor Operating Service CE. Documents sent and received from Infor OS CE are sent through the Enterprise Connector.

Installing the Enterprise Connector

The Infor Enterprise Connector arranges the connectivity between ION in the cloud and a local, on-premises application. The Enterprise Connector is installed on your local server and is required if you are integrating your on-premises Distribution A+ with Infor Operating Service (OS) in the cloud.

Creating a new Enterprise Location in ION

Before you install the Enterprise Connector, you must create a new Enterprise Location in ION Desk, specifying the **Name** and **Description** fields of the new Enterprise Location.

See the *Infor ION Desk User Guide - Cloud Edition* for instructions on creating a new location.

Installing the Enterprise Connector

Download, install, and configure the Enterprise Connector on your server. The Enterprise Connector service is deployed in your on-premises infrastructure and facilitates communication with the ION CE environment.

You may need to provide a user with read/write access on your server. The user name and password may be required when you configure document flows for the specific integration that uses the Enterprise Connector.

See the *Infor ION Desk User Guide - Cloud Edition* for instructions for installing the Enterprise Connector.

Understanding connection points and document flows

In ION Desk, a configuration page is available to create a list of connection points. Connection points provide the information that ION uses to connect to an application's message inbox and outbox. At least one connection point must be defined for each application instance that integrates to ION.

For each connection point, you select the BOD documents that can be sent or received by the application instance. These correspond to the BODs that are listed in "BODs used in Distribution A+ integrations" on page 107.

In an integration between two BOD-enabled applications, document flows are set up to define the BODs that flow between the application connection points.

A connection point can be reused multiple times in one or more document flows. Connection points can also be created during the modeling of a document flow as part of the details of the document flow elements. The connection points that are created during the modeling of a flow are added to the shared list of connection points, and they can be reused.

Some integrations with Distribution A+ use a solution XML file to set up sample connection points and document flows.

See the *Infor ION Connect Administration Guide* for additional information about connection points and document flows.

Creating Connection Point

1. In ION Desk, select **Connect** > **Connection Points**.

2. Click **Add** > **Infor Application**.

3. In the left pane, specify these values:

   Name

   Specify a name for the connection point, such as Aplus_Base_Env.

   Description

   Optionally, specify a description for the connection point.

When creating a connection point for a hybrid integration, specify the below value.

   Location

   Select the Enterprise Location you created in the chapter *Configuring Distribution A+ to send and receive BODs using a hybrid ION configuration.*

Proceed with Connection Point Details.

1. Click **Details**.
2. Specify this information:

Application Properties	Value
Name	Specify a unique application name. For example, BBEE, where BB indicates your Distribution A+ base and EE indicates your environment.
Logical ID	This is automatically created from the Name and Logical ID Type entries.

Logical ID Type	Logical ID type: aplus.
Tenant	Click the Tenant checkbox and specify the tenant.
Database Type	Specify **SQLSERVER**.
Database - Basic	If you select **Basic** as the database address it allows you to specify properties for the Commerce Gateway database. The properties that you specify depend on the database type that you selected. The URL to connect to the database is built automatically. **Host Name**: Specify the host name of the SQL server that hosts the Commerce Gateway database. This field is enabled if the Basic check box is selected. **Port Number**: Specify the number of the port that is used to connect to the Commerce Gateway database. This field is enabled if the Basic check box is selected. The default value is 1433. **Schema Name**: Specify the name of the Commerce Gateway database. The default value is InforAplusCG.
Database - Advanced	If you select **Advanced** as the database address it allows you to specify the URL to connect to the Commerce Gateway database. Use this option if you require settings that are not available with the Basic option. **URL**: Use a URL format with either a port designation or a named instance. As in these examples, replace the [hostname], [port], and [schemaname] placeholders with the values as described in the list above and replace [NamedInstance] with the SQL Server instance name, if not Default. jdbc:jtds:sqlserver://[hostname]:[port]/[schemaname] OR jdbc:jtds:sqlserver://[hostname]/[schemaname];instance=[NamedInstance]

Connection User Name	Specify the user name to connect to the Commerce Gateway database. If you use SQL Server with Windows authentication, specify a valid Windows domain user name instead of a SQL Server user name.
Connection Password	Specify the password for the selected user to connect to the Commerce Gateway database.
Polling: Delete Process Messages	Set **Delete Processed Messages** to **true** if you want to immediately delete BODs from the Distribution A+ message outbox area after ION processes them. Set the value to **false** if you want the processed BODs to remain in the message outbox for the number of minutes specified in the **Outbox Cleaner Expire Time** field.

3. Click the Test button to verify whether the connection is working. A message is displayed that indicates whether the test is successful. If the test is not successful, correct the connection details and try again.

4. Click the Save icon to save the connection point.

5. Click the Back icon to return to the list.

Connection points are not activated separately. When you activate a document flow, the associated connection points are activated.

Optionally, you can import Connection Points.

The install files for this product are stored on the Distribution A+ Connection Point ISO image available on the Infor Product Download Center. The Distribution A+ Connection Point, aplus.xml, is located in the AplusConnectionPoint folder.

1. Sign onto the ION Desk application.

2. In ION Desk, click Connect > Connection Points.

3. Click Import on the Connection Points page. From the Windows Open dialog box, browse to and select the aplus.xml file from the AplusConnectionPoint folder on the ISO image. Refer to the section "Importing Connection Points" in the *Importing and Exporting Connection Points* chapter of the Infor ION Connect Administrators Guide.

4. Click Open. This will create the Distribution A+ Connection Point in the Desk as an Inactive Connection Point in ION. It will be displayed as "Aplus_Base_Env".

5. Proceed with Connection Point Details as described above.

Defining application documents for the connection point

The Distribution A+ Connection Point, aplus.xml, which can be optionally imported as described in the previous step, contains all the documents required by Distribution A+ to integrate with other products. However, to add a document for a different type of integration, follow these steps:

1. In ION Desk, select **Connect** > **Connection Points**.

2. Select a Distribution A+ connection point and click the Details button.

3. Click the Documents tab of your Distribution A+ connection point and select all of the documents that can be sent or received by this application instance for any integration. For each document, click the scenario option to specify whether it can be sent, received, or both.

4. See the list of inbound and outbound documents in "BODs used in Distribution A+ integrations" on page 107. The list of documents that you define for this connection point must match that list.

5. To add a document type, click the Add Documents button which will open the Select Documents pop-up window.

6. You can filter the list of documents by specifying your selection criteria in the Filter field or by using these buttons:

   - Standard: Show standard documents

   - Extended: Show standard documents that are customized

   - Custom: Show custom document

7. For an existing document, to add verbs for a document type, select the document and click Add Verbs. The Add Verbs window displays, showing those verbs that have been identified for use. The documents and verbs are selected from the ION registry.

8. To remove a document or verb, select the item and click Remove. When you remove a document or verb that is used in a document flow, an error is reported in the document flow messages pane.

9. Click the Save icon to save your Connection Point.

Create Connection Points for Integrating Products

Create and configure connection points for other products that will be used for BOD message routings with the Distribution A+ application. See the appropriate integration guide for steps on creating the connection point for that product.

Configuring a document flow between applications

This section provides general steps for setting up a document flow in ION Connect.

Note: Usually document flows are configured and activated in conjunction with integrations between this product and another product; in that case, skip this section and follow the steps in the appropriate integration guide. If you want to send BODs to another ION-enabled application but there is not an appropriate integration guide, follow the steps in this section.

See the *Infor ION Connect Administration Guide* for additional information about document flows in ION.

Defining the document flow

Every flow in the modeling canvas has a start and an end. For a new document flow, only the start and end are shown. You can add items to the flow, using this general method:

1. In ION Desk, select **Connect** > **Document Flows**.

2. Select the Application icon in the toolbox. If the Application icon is not visible, you need to select and view the details of a **Flow Name**.

3. Click the line where the application must be added. Alternatively, drag the item from the toolbox to the line where it must be added.

4. An Activity box is added.

5. Click the Activity to view or change the properties in the Activity Task pane.

6. Click Add to add a connection point to the Activity.

7. Repeat steps 2-6 to add another application and its connection point to the flow.

8. Click the document icon between the applications to define the documents that flow from the first application to the second application.

9. Save the document flow.

To delete an item, right-click it and click **Delete**.

Activating the document flow

A new or changed document flow can be used in ION after it is activated. This section describes how to activate a document flow.

Note: If you encounter problems during this setup, you must stop and restart the ION service.

1. In ION Desk, select **Connect** > **Document flows**.

2. Select one or more document flows.

3. Click Activate button.

    a. If you receive a message that the activation was successful, click OK. Your changes

are effective immediately.

  b. If you receive a warning message, the document flow is already activated.

  c. If you receive an error message, the document flow cannot be activated.

See the *Infor ION Connect Administration Guide* for information about activation errors.

Adding users and distribution groups

Set up users and distribution groups in Infor Federation Services (IFS) for Distribution A+ users who require access to ION in order to view or update information or to troubleshoot errors.

See the *Infor ION Connect Administration Guide* for information about adding ION Desk users and setting ION Desk authorization levels.

See the *Infor Federation Services Administration Guide* for information about configuring distribution groups.

Setting up business events and workflows

Use the *Infor ION Process Administration Guide* to create and configure ION events, alerts and workflows.

See the Distribution A+ Workflow Management Integration with ION Desk Alerts and Workflows on page 59 for examples of ION alerts and events that are available for Distribution A+.

Chapter 6   Publishing BODs

6

After you define the Distribution A+ connection point in ION, you publish the associated BODs so that ION can retrieve them from your message outbox. This chapter tells how to publish BODs for general use by any application that subscribes to the BODs through ION.

For an ION-based integration between this product and another product, see the appropriate integration guide to set up the document flows and to perform any additional configuration of the products that is required for that integration. Then use the steps in the integration guide to publish the BODs.

Starting applications and services

Start these applications, services and processes if they are not already running:

- Distribution A+
- Other applications with which you are communicating through ION
- ION Service: Use the ION Grid Management user interface to verify that the ION Service is running. See the *Infor ION Connect Administration Guide.*

Start the Distribution A+ On-Ramp Service Processor

If the On-Ramp Service Processor is already started, you need to first stop the On-Ramp Service Processor.

1. Select On-Ramp Service Processor Inquiry (MENU ESMAIN) or key: ESBJOBS.

2. Press F13=STOPTP from within the inquiry to stop the On-Ramp Service Processor.

3. Press F5=REFRESH until the screen indicates that a restart is needed.

4. Press F6=RESTART to start the On-Ramp Service Processor.

Starting BOD exchange process between Distribution A+ and other Infor applications

Before attempting to start publishing BODs from **Distribution A+,** follow the setup steps in the Integration Guide for each Infor application(s) that **Distribution A+** will be integrating with.

Distribution A+ BOD dependencies

Performing the initial load of Distribution A+ data

The BODs sent from Distribution A+ are picked up by ION, forwarded to subscribing Infor applications, and, if accepted, will end up in the respective Infor applications databases.

The Distribution A+ Integration Guides for any Infor application that Distribution A+ is integrating to, each include a complete list of the applicable BODs for the pertinent integrated application.

The initial population of master data must be performed before you can use the integration. It is critical that you follow these procedures in the order in which they are presented.

WARNING: On the IBM i, if Journaling is turned on for Disaster Recovery, do one of the following to turn journaling off in specific libraries or specific files:

1. Turn off journaling for the following libraries.

    a. Environment File Library (APLUSbbFee), where bb is the Base ID and ee is the Environment ID.

    b. Base Library (APLUSbb)

2. Turn Journaling off for the following files.

    a. in the Environment File Library (APLUSbbFee)

        i. BODOHD

        ii. BODODT

        iii. BODOAT

        iv. SOANTRN

        v. SOADINF

        vi. SOAPPRC

    b. in the Base Library (APLUSbb)

        i. SOAPINQ

WARNING: The initial population of data can take some time, depending upon the number of records (BODs) you are sending.

Before you begin, verify that the Commerce Gateway and On-Ramp Processor are running.

Refer to the following appendices for Noun/Verb requirements:

- Business events that generate outbound BODs on page 93

- Inbound BOD usage on page 105

- BODs used in Distribution A+ integrations on page 107

Initial Load Steps

Refer to the Distribution A+ Integration Guide for each specific Infor application for the list and sequence of the Initial Load of Master Data BODs and Transactional Data BODs.

WARNING: The sequence of Initial Load is critical to ensure that BODs published by Distribution A+ are accepted by other Infor application(s).

For each Noun that Distribution A+ needs to publish and in the exact sequence required, complete the following steps.

Master Data Load Note: a from/to data range should not be entered when performing the initial load of master data. It is up to the user to keep track of previous ranges entered if you choose to do so.

1. Access the Extension Solution BOD Load/Reload (MENU ESMAST) (quick key BLR).

2. In the **Noun** field, type or select the Noun Name (noun names are case-sensitive).

3. Press ENTER.

4. If message **8626: Noun cannot be done yet.** is received, click F5=BYPASS NOUN ORDER CHECK.

5. Click F5=CONTINUE.

6. Make note of the **Transaction Generated** value, indicating the total number of BODs that are processed.

Important: Since some Master Data depends on previously loaded Master Data, it is recommended that after each load you verify that the data actually was accepted by integrating application(s). Use total number of BODs from the **Transaction Generated** value above to make sure that all data is loaded in Databases of integrating application(s).

To verify, sign on to the integrated applications (e.g. PM Dashboards, ICB, etc.) as a business user. Ensure that the Master Data that got loaded appears in the integrated application.

Transactional Data Load Note: Before you populate the transaction data, review the ranges of the Extension Solution BOD Load/Reload (MENU ESMAST) (quick key BLR) as some are specific to the data you are loading.

For example, the company number range is applicable to any document—sales order, purchase order, payable transaction, etc., while the vendor number would only apply to a purchase order, requisition, AP invoice or receive delivery.

You may also consider running the initial load for certain transactions multiple times, using different ranges each time. Be cautious about running Extension Solution BOD Load/Reload (MENU ESMAST) (quick key BLR) "wide open" without beginning and ending ranges. At a minimum, consider entering beginning and ending dates in the Invoice Date, Order Date, Aging Date or Receipt Date ranges.

Range	Nouns										
	Invoice	Payable Transaction	Requisition	Purchase Order	Receivable Transaction	Sales Order	Receive Delivery	Source System GL Movement	Quote	CustomerReturn	Shipment
Company	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Warehouse	✓		✓	✓		✓	✓		✓	✓	✓
Vendor		✓	✓	✓			✓				
Customer	✓				✓	✓			✓	✓	✓
Business Unit		✓									
Voucher Number											
Invoice Date	✓	✓									
Invoice Number	✓	✓			✓	✓				✓	
Order Number	✓					✓			✓	✓	✓
Purchase Order Number				✓			✓				
Requisition Number			✓								
Aging Date					✓						
Order Date			✓	✓					✓	✓	✓
Include Vouchered		✓									
Include Open		✓			✓	✓				✓	✓
Include History						✓				✓	✓
Include Paid		✓			✓						
Item Number							✓				
Receipt Date							✓				

Range	Nouns										
	Invoice	Payable Transaction	Requisition	Purchase Order	Receivable Transaction	Sales Order	Receive Delivery	Source System GL Movement	Quote	CustomerReturn	Shipment
GL Account Segment 1 - 5								✓			
Account Code 1 - 5								✓			
Fiscal Year								✓			
Account Type								✓			
Include Actual								✓			
Include Budget / Budget ID								✓			

Activate Extension Solution

Activate Extension Solution (MENU ESMAST) (quick key AES)

The Activate Extension Solution is only successful if the message Extension Solution has been activated appears after pressing F5=CONTINUE. If the above message is not received, check for the ES990A 'Extension Solution Error Report' in your spooled files to see what needs to be corrected.

The Noun Activation/Deactivation screen will appear if successful. Place a 'Y' next to the nouns that are to be activated at this time. Additional Nouns can be activated at a later time using Noun Activation/Deactivation (MENU ESMAST) (quick key NAD).

Distribution A+ business events that generate publishing of outbound BODs

After the initial data load, Distribution A+ sends BOD updates when users change certain data or when certain transactions occur in Distribution A+.

For a list of the events that generate BODs, see "Business events that generate outbound BODs" on

Chapter 7   Verifying the configuration

7

This section describes the steps to verify that the connections are set up correctly.

Verifying that BODs are generated

1. Use the information in "Business events that generate outbound BODs" on page 93 to determine what user actions generate certain BODs.

2. Perform those user actions in Distribution A+.

3. Check the message outbox in the Commerce Gateway database to see if the BODs were generated. You can also see the XML data for the BOD.

Verifying that ION receives data

To verify that ION is receiving BODs from Distribution A+, monitor these areas in ION Desk:

1. Select **Manage** > **Connect** > **Active Connection Points** to show the message queue counts.

2. Select **Manage** > **Connect** > **Message** Trace to show individual messages, if they are enabled.

If you encounter problems, see "Data is not flowing properly" on page 85.

Verifying the data flow between applications

After you create and activate document flows to other products, you can check the message inbox area of the other products, to ensure that data is flowing between the systems.

Chapter 8   ION Event Monitors

8

This chapter provides instructions for importing the ION Event Monitors which are predefined monitors to initiate business alerts that appear in ION Activity Deck.

Distribution A+ ION Event Monitors

The Distribution A+ ION Event Monitors are 16 alert monitors specially designed for distributors. These monitors are available when you import the DistributionAplusMonitors.xml file from the Distribution_Aplus_ConnectionPoint ISO image and activate at least one document flow for the Distribution A+ Connection Point in ION. The monitors are designed to alert specific individuals in your organization that are assigned to distribution groups in Infor Federated Services (IFS).

Distribution A+ ION Event Monitors do not require any additional setup in Distribution A+ once they are imported; provided that Distribution A+ is configured and setup for ION Integration, with the nouns used in ION Monitor models as well as Distribution A+ events triggering the publishing of BODs with these nouns once they are activated.

The following table lists the ION Event Monitors provided with the Distribution A+. Refer to ION Monitors on page 51 for detailed descriptions of each monitor.

Event Monitor	IFS Distribution Group Notified	Result
Aplus_Customer_AdditionOrUpdate	Aplus Customer Account Manager	Sends an alert when a customer record is created or existing customer updated
Aplus_Customer_Reinstated	Aplus Sales	Sends an alert when a customer is reinstated
Aplus_CustomerReturn_HighValue	Aplus Sales	Sends an alert when a customer return is created or updated that exceeds $1000.00-
Aplus_Invoice_AdditionHighValue	Aplus Finance	Sends an alert when a sales invoice is entered that exceeds $1000.00
Aplus_ItemMaster_AdditionOrUpdate	Aplus Master Data	Sends an alert when an item is added or existing record is updated
Aplus_ItemMaster_WithHazardousMaterial	Aplus Shipping Supervisor	Sends an alert when a hazardous

	Aplus Receiving Supervisor Aplus Warehouse Supervisor	item (requiring an MSDS) is created or updated
Aplus_ReceiveDelivery_Overdue	Aplus Purchase	Sends an alert when a purchase order is not received by 3 days after the planned receipt date
Aplus_ReceiveDelivery_QtyDeviation	Aplus Purchase	Sends an alert when a PO's receipt quantity is less than or greater than 5% of the ordered quantity
Aplus_ReceiveDelivery_ReceivedLate	Aplus Purchase	Sends an alert when a PO is received more than 3 days after the planned receipt date
Aplus_ReceivableTransaction_ InvoiceCreditMemoHighValue	Aplus Finance	Sends an alert when a sales credit memo is created that exceeds $1000.00
Aplus_SalesOrder_HighValue	Aplus Sales	Sends an alert when a sales order greater than $1000.00 is created or updated
Aplus_SalesOrder_AdditionShortTimeFrame	Aplus Sales	Sends an alert when a sales order is created with a delivery date that is less than 14 days from entered date
Aplus_SalesOrder_Due	Aplus Sales	Sends an alert when a sales order shipment is due within 3 days
Aplus_Shipment_Shipped	Aplus Shipping Supervisor Aplus Warehouse Supervisor	Sends an alert when a sales order is shipped
Aplus_Shipment_ShippedLate	Aplus Sales	Sends an alert when a sales order is shipped 3 or more days late
Aplus_Supplier_AdditionOrUpdate	Aplus Purchase	Sends an alert when a vendor record is created or existing vendor is updated

User distribution of alerts

ION Monitors create alerts that are distributed to groups or persons. Distribution groups are configured in Infor Federated Services (IFS). You can change the distribution group without having to change monitors' models.

Since actual users of alerts cannot be known beforehand, Distribution A+ ION Event Monitors included with the Distribution A+ are created with distribution groups. Distribution groups used by the Distribution A+ ION event monitors must be created in Infor Federated Services (IFS) by uploading the Aplus_DistributionGroups.csv file. Then you are ready to assign the groups listed above to users.

At least one user should be added to each distribution group. See Importing Distribution A+ ION Alert Monitors on page 49 for instructions for uploading the csv file.

Distribution Group Name
Aplus_CustomerAccountManager
Aplus_Finance
Aplus_MasterData
Aplus_Purchase
Aplus_ReceivingSupervisor
Aplus_Sales
Aplus_ShippingSupervisor
Aplus_WarehouseSupervisor

Infor ION Alert Monitor definition imports

This section describes the ION alert monitor definitions that are imported from the DistributionAplusMonitors.xml file. Once created or imported, each alert monitor can be modified if needed. Refer to the "Procedures; Modeling Tasks; Defining monitors" procedures in Chapter 3 of the Infor ION Process Administration Guide for the instructions.

Importing Distribution A+ ION Alert Monitors

The Distribution A+ ION Alert Monitors are available on the Distribution_Aplus_ConnectionPoint ISO image in the DistributionAplusMonitors.xml file. Use these instructions to import the Distribution A+ ION Alert monitor definitions in ION Desk, and upload the distribution groups file in Infor Federated Services (IFS).

To import the Distribution A+ ION Alert Monitor Definitions

1. The DistributionAplusMonitors folder contains the Aplus_DistributionGroups.csv and DistributionAplusMonitors.xml files.

2. In Infor Federated Services (IFS), navigate to **Manage** > **Users** and verify all ION users are set up.

3. Log in to Ming.le as Administrator user. Access **User Management** > **Manage > Groups** and import the Aplus_DistributionGroups.csv file.

4. Assign users to the distribution groups.

   Refer to the Infor Federation Services Administration Guide for instructions on Assigning Users to Distribution Groups in the Configuring Distribution Groups section. The master data just loaded will display in the list of available Distribution Groups for a list of the groups uploaded from the Aplus_DistributionGroups.csv file.

5. In ION Desk, select **Monitors & Workflows** > **Monitors**.

6. Click **Import**.

7. Select the DistributionAplus_Monitors.xml file.

8. Click **OK** to save the monitor.

To activate the Distribution A+ ION Event Monitors

WARNING: While performing the initial load (or reload) of Infor Distribution A+ data, ensure that ION Event Monitors are not active to prevent the potential flood of unnecessary ION alerts.

Exception to this may be monitors that refer to several Business Documents, like "Receive Delivery Overdue", where both BODs ReceiveDelivery and PurchaseOrder must be published after event this monitor is activated.

1. Select **Monitors & Workflows** > **Monitors**.

2. Select the monitors you imported above and click **Activate**. Refer to the list in the Distribution A+ ION Event Monitors section on page 47.

3. Update each monitor to include one or more distribution users or groups that you uploaded to Infor Federated Services (IFS).

Note: Refer to the section "Modeling Tasks - Defining monitors" in the *Infor ION Process Administration Guide*.

4. Once you have activated all monitors, you must publish the BODs used by the monitors.

Note: ION alerts will be sent to users when an activated. ION Event Monitor detects Application Business Document(s) defined in this event monitor that are published by Distribution A+ and meet its Condition/Rule. Alerts will be sent to each distribution user.

ION Monitors

The naming convention used for the Distribution A+ ION event monitors is "Aplus_<noun>_<event>." For example, Aplus_Supplier_AdditionOrUpdate.

1. Customer Addition/Update	
Description	An alert when a new customer is added or updated
Monitor Name	Aplus_Customer_AdditionOrUpdate
Application Documents	CustomerPartyMaster
Conditions / Rule	When [ActionCode] is 'Add' or 'Replace'
Action	An alert is sent to notify users when a customer is added or updated
Distribution Group	Aplus_CustomerAccountManager
Message	Customer [CustomerName] is added/updated. Customer ID# [CustomerID]

2. Customer Reinstated	
Description	An alert when customer was reinstated
Monitor Name	Aplus_Customer_Reinstated
Application Documents	CustomerPartyMaster
Conditions / Rule	[Status] changed from 'Hold' to 'Open' within 9999 days
Action	An alert is sent when a suspended customer is reinstated.
Distribution Group	Aplus_Sales
Message	Customer [CustomerID] with the name of '[Name]' has been reinstated

3. Customer Return with High Value	
Description	An alert when Customer return with high value is added or updated
Monitor Name	Aplus_CustomerReturn_HighValue
Application Documents	CustomerReturn
Conditions / Rule	When [Status] <> 'Closed' AND [Status] <>'Canceled' AND [TotalAmount] < -1000.00
Action	An Alert is sent to users when a customer return with a value greater than $1000.00- is added or updated
Distribution Group	Aplus_Sales

Message	Customer return [CustomerReturnID] with the status [Status] exceeds 1000 [currencyID]. The total amount is [TotalAmount] [currencyID].

4. Invoice Addition with High Value

Description	An alert when Sales invoice is added with high value
Monitor Name	Aplus_Invoice_AdditionHighValue
Application Documents	Invoice
Conditions / Rule	When [TotalBaseAmount] > 1000.00 and [Status] = 'Open'
Action	An alert is sent when a sales invoice with a value of greater than $1000.00 is added
Distribution Group	Aplus_Finance
Message	Sales invoice [InvoiceID] exceeds 1000 [BaseCurrency]. The total amount is [TotalBaseAmount] [BaseCurrency].

5. Item Master Addition/Update

Description	An alert when new Item is added/updated
Monitor Name	Aplus_ItemMaster_AdditionOrUpdate
Application Documents	ItemMaster
Conditions / Rule	When [ActionCode] is 'Add' or 'Replace'
Action	An alert is sent when there is addition or an update of an item
Distribution Group	Aplus_MasterData
Message	Item [Item] has been created/updated.

6. Item Added/Updated with Hazardous Materials and MSDS Required

Description	An alert when an item added/updated with MSDS required
Monitor Name	Aplus_ItemMaster_WithHazardousMaterial
Application Documents	ItemMaster
Conditions / Rule	When ([ActionCode = 'Add' or 'Replace' and [Hazard] <> ' ' (blank)
Action	An alert is sent when there is an addition or update of an item that requires an MSDS sheet
Distribution Group	Aplus_ShippingSupervisor Aplus_ReceivingSupervisor Aplus_WarehouseSupervisor

Message	Item [ItemID] of Hazardous Type [HAZARD] is created/updated.

7. Receive Delivery Overdue	
Description	Alert user when no receipts are received on time for a purchase order line
Monitor Name	Aplus_ReceiveDelivery_OverDue
Application Documents	ReceiveDelivery, PurchaseOrder
Conditions / Rule	Reference Document: Purchase Order Condition: [Status] is 'Open' TimeOffset: 3 days Since: [PlannedReceiptDate] OverDue Document: ReceiveDelivery
Action	An alert is sent when no receipts are received on time for a purchase order line
Distribution Group	Aplus_Purchase
Message	Receipts for purchase order [PurchaseOrderNo] are 3 days overdue

8. Receive Delivery Quantity Deviates	
Description	An alert when the Received Quantity deviates more than 5% from the ordered quantity on purchase order line
Monitor Name	Aplus_ReceiveDelivery_QtyDeviation
Application Documents	ReceiveDelivery
Conditions / Rule	When [ReceivedQuantity] deviates more than 5.0% from [OrderQuantity]
Action	An alert is sent when a received quantity deviates by more than 5% from the ordered quantity of the lines on the purchase order
Distribution Group	Aplus_Purchase
Message	Received Quantity of goods against Purchase Order [PurchaseOrderNo] deviates by at least 5% from Ordered Quantity. Receipt ID is [ReceiveDelivery]

9. Receive Delivery Received Late	
Description	Alert user when a receipt is received late
Monitor Name	Aplus_ReceiveDelivery_ReceivedLate

Application Documents	ReceiveDelivery, PurchaseOrder
Conditions / Rule	When [Status] = 'Open' AND [ReceiptStatus] = 'Received' AND [LineNumber] = [ReferenceLineNumber] AND [ReceiptDate] >[PlannedReceiptDate] + 3 days
Action	An alert is sent when an item's receipt is received 3 or more days late
Distribution Group	Aplus_Purchase
Message	Receipt [ReceiveDelivery] for line [LineNumber] of purchase order [PurchaseOrderID] is received more than 3 days late

10. Sales Credit Memo Addition with High Value	
Description	Alert when a sales credit memo is created with certain conditions, for example, over $xxx.
Monitor Name	Aplus_ReceivableTransaction_InvoiceCreditMemoHighValue
Application Documents	ReceivableTransaction
Conditions / Rule	When [ReceivableStyle] = 'CreditMemo' and [TotalBaseAmount] <= -1000
Action	An alert is sent when a sales credit memo is created that is greater than $1000.00-
Distribution Group	Aplus_Finance
Message	Sales credit memo [DocumentID] exceeds 1000 [BaseCurrency]: the total amount is [TotalBaseAmount] [BaseCurrency]

11. Sales Order Addition with High Value	
Description	An alert when Sales Order with high value is added or updated
Monitor Name	Aplus_SalesOrder_HighValue
Application Documents	SalesOrder
Conditions / Rule	When [BaseAmount] > 1000.00 AND ([Status] = 'Open' or [Status] = 'Hold')
Action	Alert is sent when a sales order greater than $1000.00 is added or updated
Distribution Group	Aplus_Sales
Message	Sales order [SalesOrder] that exceeds 1000 [BaseCurrency]: the header amount is [TotalBaseAmount] [BaseCurrency]

12. Sales Order Addition with Short Timeframe	
Description	An alert when New Sales Order with short timeframe delivery window
Monitor Name	Aplus_SalesOrder_AdditionShortTimeFrame
Application Documents	SalesOrder
Conditions / Rule	When ([ActionCode] = 'Replace' or [ActionCode] = 'Add') AND it is 14 days before [PromisedShipDateTime]
Action	An alert is sent when a sales order is entered with a short timeframe delivery window
Distribution Group	Aplus_Sales
Message	New sales order [SalesOrder] for customer [CustomerID] has a short timeframe delivery window (14 days)

13. Sales Order Due	
Description	Alert user when shipments are due within 3 days
Monitor	Aplus_SalesOrder_Due
Application Documents	SalesOrder
Conditions / Rule	It is 3 days before [DueDateTime] and ([Status] = 'Open' or [Status] = 'Hold')
Action	An alert is sent when [Status] ='Open' or Hold', and it is 3 days before [DueDateTime]
Distribution Group	Aplus_Sales
Message	Shipments for sales order [SalesOrder] are due within 3 days

14. Shipment Order Shipped	
Description	An alert when order is shipped
Monitor Name	Aplus_Shipment_Shipped
Application Documents	Shipment
Conditions / Rule	When [Status] = 'Shipped'
Action	Alert is sent to notify user when order is shipped
Distribution Group	Aplus_ShippingSupervisor, Aplus_WarehouseSupervisor
Message	[Warehouse] Shipment Order [Order] is Shipped

15. Shipment Shipped Late	
Description	Alert user when shipment is shipped late
Monitor	Aplus_Shipment_ShippedLate
Application Documents	Shipment, SalesOrder
Conditions / Rule	When [ShipmentStatus] = 'Shipped' AND [LineNumber] = [ReferenceLineNumber] AND [ShipmentDateTime] > [PromisedShipDateTime] + 3 days AND [SaleOrderStatus] = 'Shipped'
Action	Alert is sent to notify user when a shipment is shipped late
Distribution Group	Aplus_Sales
Message	Shipment [ShipmentID] for sales order [SalesOrderID] is shipped 3 or more days late

16. Supplier Addition/Update	
Description	An alert when new Vendor was added/updated
Monitor Name	Aplus_Supplier_AdditionOrUpdate
Application Documents	SupplierPartyMaster
Conditions / Rule	When [ActionCode] is 'Add' or 'Replace'
Action	An alert is sent when a vendor master is added or maintained
Distribution Group	Aplus_Purchase
Message	New Vendor [Supplier] with the name of [Name] was added/updated

Verifying event monitors

After importing and activating the Distribution A+ ION monitors, the users can view Alerts in ION Pulse and Ming.le. You may also receive emails if ION is configured to send emails.

If expected alerts are not received, access the Manage menu.

Check if alerts are listed in ION Activities and verify distribution users.

Otherwise check ION Event Management Active Monitors and review triggers to see if the BODs are received and conditions are met. You can also check the log files in ION.

Refer to the Application Documents row in the tables above to identify the BODs used by each monitor.

Chapter 9 Distribution A+ Workflow Management Integration with ION Desk Alerts and Workflows

9

The Distribution A+ Workflow Management module supports the creation of the AplusWorkflow Business Object Document (BOD). The AplusWorkflow BOD is sent to ION, and subsequently used in ION ActivityDeck tasks and alerts.

This AplusWorkflow BOD is unique to Distribution A+ and is not used by any other ION application. This appendix contains the steps to configure ION to support this custom AplusWorkflow BOD.

We recommend that you read the:

- *Infor Distribution A+ Workflow Management User Guide*, and follow the steps to configure Distribution A+ Workflow Management, before configuring ION to be integrated with Distribution A+ Workflow Management.

- *Infor ION Technical Reference Guide*, specifically the ION Registry chapter, to become familiar with integrating a custom BOD in ION.

- *Infor ION Process Administration Guide*, specifically the ION Workflow chapter, before working with the AplusWorkflow BOD.

Configure Distribution A+ Workflow Management

Before configuring ION to support Distribution A+ workflow alerts, the following steps must be taken in Distribution A+:

1. Register A+ User IDs (MENU XACFIG) for each user that will be using ION ActivityDeck. Refer to the *Infor Distribution A+ Cross Applications User Guide* for additional information.

2. Tailor and activate the alerts which will be sent to ION. Refer to the *Infor Distribution A+ Workflow Management User Guide* for additional information.

3. Activate the AplusWorkflow noun. Refer to the Noun Activation/Deactivation section in the *Infor Distribution A+ Extension Solution User Guide* for more information.

4. Add the AplusWorkflow noun as an Inbound BOD. Refer to the Extension Solution Inbound BOD Options Maintenance section in the *Infor Distribution A+ Extension Solution User Guide* for more information.

5. Restart the On-Ramp Service Processor though the On-Ramp Service Processor Inquiry

(MENU ESMAIN). Refer to the *Infor Distribution A+ Extension Solution User Guide* for more information.

Configure ION Desk

The files required to implement the AplusWorkflow BOD in ION Desk must be downloaded from the Infor Product Download Center. Download the Distribution A+ Connection Point image labeled:

1) Distribution_Aplus_ConnectionPoint_12.iso

The following files on this image are to be used to configure the AplusWorkflow BOD on the ION server.

Download the AplusWorkflow_BOD.zip file and & AplusWorkflows_ION_Content.zip file to separate directories on your system:

- AplusWorkflow BOD folder: AplusWorkflow_BOD.zip - do not extract the content; this zip file will be used in import steps below.

- AplusWorkflow ION Content folder: extract the content of AplusWorkflows_ION_Content.zip Files AplusWorkflows.xml, AplusWorkflowActivationPolicies.xml, AplusCodes.xml, and AplusEventMonitors.xml; these files will be used in import steps below.

Update the ION Catalog

To use the AplusWorkflow BOD in the workflow process, you must add this custom BOD to ION BOD definitions.

1. In ION Desk, select Data Catalog > Custom Object Schemas.
2. Click Import.
3. Browse and select the AplusWorkflow BOD folder]\AplusWorkflow.zip.
4. Click Save.

Add the AplusWorkflow BOD to the Connection Point

In ION Desk, add the AplusWorkflow noun to the Connection Point in order for Workflows to process between Distribution A+ and ION ActivityDeck.

1. In ION Desk, select **Connect** > **Connection Point.**
2. Select the Aplus_Base_Env Connection Point. Note that the Connection Point can only be modified if its Status is Inactive. So, if necessary, first deactivate any Document Flows that are using this Connection Point.

3. Click the Documents tab.

4. Add the AplusWorkflow Noun.

5. Set the AplusWorkflow Noun to **Sync Receive in Application** and **Sync Send from Application.**

6. Save the updated Connection Point.

7. Activate any Document Flows that use this Connection Point.

Work with the AplusWorkflow Sample Files

Sample Workflows, Workflow Activation Policies and Event Monitors are provided on the Infor Distribution A+ Connection Point 12.x.x or higher ISO.

Copy these samples to create your own workflows, activation policies and event monitors.

The following instructions will outline the steps to import and activate the sample Workflow files.

We recommend that you read and understand the *ION Process Administration Guide*, specifically the ION Workflow chapter, before working with AplusWorkflow BODs.

Import the Distribution A+ Workflow Definitions

This section describes the initial workflow definitions that may be imported with files found on the [AplusWorkflow ION Content folder]\AplusWorkflows contents.

1. In ION Desk, select **Monitors & Workflows** > **Workflows.**

2. Click Import.

3. Select the [AplusWorkflow ION Content folder]\AplusWorkflows.xml file and click OK.

ION Workflow Drill Back Definitions

Distribution A+ has created several drill backs that allow Infor Ming.le user's direct access to Distribution A+ screens through hyperlinks on ION Workflow tasks or notifications.

Before proceeding, confirm that Distribution A+ screen views are defined in Ming.le.

Log in to Ming.le as Administrator user. Access Admin Settings > Manage Applications > Manage Drillbacks. Upload Aplus_Views_Standard.xml file delivered by Distribution A+

To add Distribution A+ drill back functionality to a workflow definition, follow the steps in the *Infor ION Process Administration Guide;* beginning in the ION Workflow chapter, find the section Procedures and then Creating a workflow definition.

1. In the Workflow Properties pane, select the Drill Backs tab

    a. Click Add to open the Add Drill Back window and complete the following information

    b. **Name**: Specify the name of this drill back link to identify it in the Task or Notification

    c. **View Set**: Select a View Set from the drop down list. The list of names from the Aplus_Views_Standard.xml file delivered by Distribution A+ will be available if they have been loaded and click Edit.

    d. **View**: Select appropriate **View** from the list based on the View Set selected (ex. If this workflow content has the customer number, you can select CustomerView)

    e. View Parameters-Standard

        i. **Logical ID**: **lid://infor.aplus.1**

        ii. **Accounting Entity** and **Location**: leave blank for Distribution A+

Each view has a set number of required parameters or keys and the values for the first two parameters are always the Distribution A+ Base ID and the Environment ID. Some of the keys have preset values and other parameters will come from the workflow parameters.

Using the CustomerView as an example, the company number and customer number are required workflow parameters. The drill back parameters may be as follows:

2. Edit Drill Back

    a. For **key01**, select the Value radio button and key the **Base ID**, **BB**

    b. For **key02**, select the Value radio button and key the **Environment ID**, **EE**

    c. For **key03**, select the Parameter radio button and select **Company** from the dropdown list

    d. For **key04**, select the Parameter radio button and select **Customer** from the dropdown list

    e. Click OK.

3. Once the drill backs have been added to a workflow, you will need to add the drill backs to the Content of a Task where they will be available to the users.

Import Distribution A+ Workflow Activation Policies

1. In ION Desk, click **Monitors & Workflows** > **Activation Policies**.

2. Click **Import**.

3. Select the [AplusWorkflow ION Content *folder]*| AplusWorkflowActivationPolicies.xml file.

4. Click **OK**.

Activate the ION Workflows

1. In ION Desk, select **Monitors & Workflows** > **Workflows.**

2. Select the imported workflows which you wish to activate, for example:

    Aplus_Approve_Requisition_Print_PO

    Aplus_Release_Sales_Order_From_Hold

    Aplus_Release_Sales_Order_From_New_Customer_Hold

    Aplus_Task_Nofify_by_Email

    Aplus_Task_Nofify_Posted

3. Click **Activate**.

Activate the ION Workflow Activation Policies

1. In ION Desk, select **Monitors & Workflows** > **Activation Policies**.

2. Select the added activation policies to be activated:

    Aplus_Approve_Requisition_Print_PO

    Aplus_Release_Sales_Order_From_Hold

    Aplus_Release_Sales_Order_From_New_Customer_Hold

    Aplus_Task_Notify_by_Email

    Aplus_Task_Notify_Posted

3. Click **Activate**.

Import Distribution A+ Alert Code Lists

1. In ION Desk, click **Configuration** > **Codes** > **Import**.

2. Click **Import**.

3. Select the [AplusWorkflow ION Content folder]\AplusCodes.xml file.

4. Click **OK**.

Import the ION Event Management Monitor Definitions

This section describes the initial alert definitions that may be imported with the XML file provided with the [AplusWorkflowDir] contents.

1. In ION Desk, select **Monitors & Workflows** > **Monitors**.

2. Click Import.

3. Select the [AplusWorkflow ION Content folder]]\AplusEventMonitors.xml.

4. Click OK.

Customize the Distribution A+ Event Management Monitors Distribution List

The distribution list for all Distribution A+ monitors should be updated before activating the monitor. For example, the Aplus_Special_Order_Created Monitor is preconfigured with a sample default user (demo.user@demo.com). Before activating this monitor, the sample default user demo.user@demo.com must be removed from the distribution list and a valid user must be added to the distribution list of the monitor.

NOTE: Distribution lists can be added to any Distribution A+ monitors alert.

1. In ION Desk, select **Monitors & Workflows** > **Monitors**.

2. Select the Aplus_Special_Order_Created Monitor that was imported in the previous step and click Details.

3. Click the Distribution tab and Add.

4. Select User or Group from the drop down list.

5. Select a user from the list of users or select a Group from the list of groups. You can search for the user if they are not shown in the list.

6. Click OK.

7. Now, select the demo.user@demo.com user.

8. Click Remove.

Activate the ION Event Management Monitor(s)

1. In ION Desk, select **Monitors & Workflows** > **Monitors**.

2. Select the Monitors you with to activate, for example:

   Aplus_Alert_Sales_Rep

   Aplus_Order_Placed_On_Backorder

Aplus_Special_Order_Created

Aplus_Task_Notify_Error

Aplus_Generic_Alert

3. Click **Activate**.

Chapter 10 Creating new Outbound Custom Event Driven BOD

The *ION Technical Reference Guide* outlines information related to the ION registry and how to register new custom BODs. Before creating a custom event driven BOD in Distribution A+, you should follow these instructions from this guide in order to understand how to create a custom schema file (xsd) and how to register your schema with ION. Once a schema is registered and available in ION, the BOD can now be routed by ION and available for production or consumption by other applications.

Creating a Custom Event

The first part of creating a custom event driven BOD in Distribution A+ is to define a new custom event. New custom events are defined in Extension Solution BOD Event Registration (MENU ESFILE). After you have selected to add a new event from the Extension Solution Event Registration Screen you are presented with the Event Definition Screen to create a new event. You can use the following steps as a guide to creating a new event. Further detail can also be found in On-line Help or the *Infor Extension Solution User Guide*.

1. Determine the file to trigger.

   Based on the BOD you wish to generate, determine which file is the main file updated in Distribution A+ which should be triggered to send an outbound BOD. For example, if you are maintaining a customer you probably would want to send a BOD when the Customer Master File (CUSMS) is updated.

   The file to trigger should be added to the **Trigger File** field on the Extension Solution Event Definition Screen.

2. Determine the trigger event.

   Outbound BODs can be sent when a record is created, updated or deleted.

   Select the file trigger event by keying **Y** in the **Trigger Event Create**, **Update**, **Delete** fields on the Extension Solution Event Definition Screen.

3. Determine which program the file is in scope when the file is triggered.

   BODs in Distribution A+ are not generated every time a trigger event is fired from a database file trigger. Rather it is the combination of the trigger and the program in scope when the trigger executes (create, update, or delete) that will determine if the BOD Event fires. For

example if the trigger is on the Customer Master File (CUSMS) and Event Program is Customer Master/Ship To File Maintenance (AR800) then the Distribution A+ Extension Solution will look at both the file being triggered (CUSMS) and the event program (AR800) where the file is in scope for the record update.

The RPG Program in scope for the trigger should be added to the **Trigger Event Program** field on the Extension Solution Event Definition Screen.

4. Determine if additional conditions are required to be added to the trigger to create a BOD.

Sometimes additional checks need to be included before creating a BOD. These additional checks can be optionally added in another RPG program called the Trigger Condition Program. This program receives in 3 parameters Skip, EventID, and Buffer. The skip variable is used to determine to process the record; if this value is set to **Y** then the record will not be processed. See the Trigger Conditional Program topic in the Custom BOD Topics appendix for an example of a Conditional Program.

Optionally a conditional record check program can be added to the **Trigger Condition Program** field on the Extension Solution Event Definition Screen.

5. Determine the RPG used to map a BOD.

Before XML documents can be generate to ION, Distribution A+ needs to map the data appropriately. This data is mapped into preprocessed tables which then Commerce Gateway builds the XML and sends it to the ION Outbox. More details on the BOD Mappings can be found in the next section Creating a BOD Mapper Program, however for the purpose of Event Registration you will need an RPG program to map the data.

The BOD Mapper RPG program should be added to the **BOD Mapper Program** field on the Extension Solution Event Definition Screen.

6. Determine your noun or verb which will be sent to ION.

ION includes the BOD Registry to store details and definitions of Infor OAGIS Nouns and Verbs and Custom Nouns and Verbs. Before you decide to map existing BODs or create new BODs you should become more familiar with these patterns. These details are found in the Infor ION Technical Reference Guide.

Distribution A+ stores the list of nouns it supports in the SOA Noun Master File (SOANMST). Currently there is not a maintenance option to add additional BODs to support Custom Event Driven BODs. In order to get around this limitation you will need to manually add a record to this file using a data file utility program.

Log on to Distribution A+ and on a command line key:

    UPDDTA FILE(SOANMST) [ENTER]

To add a new record to this file, press:

    F10= ENTRY

If you were adding a noun called MyCustomer, you would complete the fields as shown:

    Noun: **MyCustomer**

    Remove Noun: (leave blank)

Load Seq: **5000**

Noun Active: **Y**

[ENTER]

Press F3=EXIT and save your entries.

The Distribution A+ or Custom noun and verb should be added to the Noun and Verb under the BOD Mapper program.

7. Determine the trigger file values you want passed into the BOD mapper program.

The Distribution A+ Extension Solution BOD Processor can pass values into your mapper program based on the file being triggered. These values are passed in the key 01 – 15 parameters. It uses the start and end position of the file trigger buffer to determine what data to pass in to each key parameter. Typically, these are associated with the key fields to your file which are then used to access the record being triggered in the mapper program.

The best way to fill these values in is to use the Display File Field Description command DSPFFD against your trigger file. Within the Display File Field Description display, for each field there is a Buffer Position and Buffer Length. You can use these values to pass in the appropriate field values into the mapper. For example if you wanted to pass in the Customer Number (CMCSNO) from the Customer Master File (CUSMS) you would put the Buffer Position for CMCSNO (3) and Buffer Length (10) into the **Key Start Position** and **Key Length** fields on the Extension Solution Event Definition Screen.

Log on to Distribution A+ and on a command line key:

DSPFFD CUSMS [Enter]

Press the PAGE DOWN key to display the **Customer Number** field:

Field Level Information			
Field	**Field Length**	**Buffer Length**	**Buffer Position**
CMCSNO	10	10	3

The key values which you wish to be passed to the BOD Mapper program should be entered in the Key 01 - Key15 Start Position(s) and Length(s).

Creating a BOD Mapper Program

BOD Mapper programs must be created using RPG and stored in a library in the Distribution A+ Library List for the Base ID being used. BOD Mapper programs have a fixed signature (Entry parameter list) and will prepare and write data to 3 files, BOD Outbound Header (BODOHD), BOD Outbound Detail (BODODT), and BOD Out bound BOD Attribute (BODOAT). These files are then

used by Commerce Gateway routines to build well-formed XML and place the data into the SQL Server Outbox.

The following are code examples from pieces of a BOD Mapper Program. For a full example, please refer to the BOD Mapper Program appendix.

Entry Parameters

Values are passed into this program from the trigger and Extension Solution On Ramp Processor. If errors exist, they can be passed back in the errtxt@ parameter. Values defined in key 01 – 15 from Event Registration are passed into the mapper programs.

```
D ZESOUTMAP       pi
D shutdown                        1
D errtxt@                       250
D p2bodid                        50
D p2pgnm                         10
D p2verb                         50
D p2noun                         50
D p2action                       50
D p2key01                        50
D p2key02                        50
D p2key03                        50
D p2key04                        50
D p2key05                        50
D p2key06                        50
D p2key07                        50
D p2key08                        50
D p2key09                        50
```

Document ID

Distribution A+ stores a unique BOD Document ID for every BOD sent to ION. All Mapper Programs must create a record in the file (SOADINF) to create a link to this BOD ID and the data sent to ION.

```
// Update BOD Reference
didocid = @@docid;
divarid = 1;
dibodid = p2bodid;
dialtky1 = %editc(zzcono:'X');
dialtky2 = %editc(zzcsno:'X');
dinoun = p2noun;
direvlvl = @@revlvl;
dibodidky = @@bodidky;
diinverb = p2verb;
exsr $chk_action;
diinact = @@action;
diindt = %date;
diintm = %time;
dilsverb = p2verb;
dilsact = @@action;
```

```
dilsdt = %date;
dilstm = %time;
write soadinfr;
```

Begin of Node

Begin of Node is used to start a node of the XML

```
// Begin Group    <DocumentID>
@@Tagtype = 'B';
@@elem = 'DocumentID';
exsr Add_Bod_Detail;
```

Detail Element Tag

Element Detail Tag is used to write out Element Name and Value

```
// Element <ID>100</ID>
@@tagtype = 'D';
@@elem = 'ID';
@@value = %editc(zzcsno : 'X');
exsr Add_Bod_Detail;
```

Attribute Element Tag

Add_Bod_Attribute is used to add attributes to the previous Element Detail Tag

```
// Add Accounting Entity Attribute <ID accountingEntity=01>100</ID>
@@attr = 'accountingEntity';
@@value = %editc(zzcono : 'X');
exsr Add_Bod_Attribute;
```

End Node

End of Node s used to end a node of the XML

```
// End Node
@@tagtype = 'E';
@@elem = 'MyCustomer';
exsr Add_Bod_Detail;
```

Chapter 11 Create Outbound Custom BOD for Distribution A+ using ION Database Connector

Database Connector

ION Database Connector can be configured to read from of write to your Distribution A+ database. It does this is through SQL and defining queries in ION Desk to pull/push data directly from DB2 or call into a DB2 stored procedure to execute this job on the backend server. Details about Database Connectors and DB2 can be found in the *Infor ION Technical Reference Guide* and *Infor ION Connect Administration Guide*.

When you create a new Database Connection point for Distribution A+, you can select DB2_400 as your database type. The Host name is your IBM i Server Name or IP Address. The schema name is the Library you wish included in the DB Connection. Note – library names can also be included in SQL Statements, so the schema name is not required to query Distribution A+. Finally, the connection user and password are your IBM i User ID and Password and should have QPGMR authority.

Creating Documents using Read and SQL.

You can use the ION Desk to read directly from DB2 and details can be found in the *Infor ION Technical Reference Guide.* If you are at version 7.1 of the IBM I operating system you can also use SQL to generate XML documents.

http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_71/rzasp/rzaspkickoff.htm.

One example of this is the following where data is pulled from the Customer Master File (CUSMS) directly and converted to an ION supported BOD.

```
SELECT xmlserialize( content
   xmlelement(name "DataArea",
     xmlagg(
       xmlelement(Name "MyCustomer",
       xmlelement(Name "Company", CMCONO),
       xmlelement(Name "Customer", CMCSNO),
       xmlelement(Name "CustomerName", CMCSNM),
       xmlelement(Name "CustomerAddressLine1", CMCAD1),
       xmlelement(Name "CustomerAddressLine2", CMCAD2),
```

```
      xmlelement(Name "CustomerAddressLine3", CMCAD3),
      xmlelement(Name "CustomerAddressLine4", CMCAD4),
      xmlelement(Name "CustomerCountry", (CMCTID)),
      xmlelement(Name "City", CMCITY),
      xmlelement(Name "State", CMSTAT),
      xmlelement(Name "Postal", CMZIP4)


  ) )
 )
 AS CLOB(1M) ) as "result" FROM CUSMS
```

Creating Documents using Stored Procedures

An ION Database Connector also allows you to call into backend stored procedures. IBM DB2 Stored Procedures can also wrap in RPG program(s) allowing you to add additional business logic and access Distribution A+ programs natively. To accomplish this, you must do the following.

1. Define a Stored Procedure.

    A DB2 Stored Procedure is used to "wrap" an RPG program to allow ION to directly call into an RPG program.

    In the example below we are wrapping the RPG program GETMYORDER in the library ION3P and it will be called with 3 input parameters (Distribution A+ Base ID, Distribution A+ Environment, and Request XML). The RPG Program (bound RPG) is list in the **External Name** parameter of the Stored Procedure. Finally, the Stored Procedure will include a **Result Set** which will allow the RPG to return the XML Response back to ION.

    Example:

    ```
    CREATE PROCEDuRE ION3P/GETMYORDER(
      IN BASE VARCHAR(10),
      IN ENVIRONMENT VARCHAR(10),
      IN REQUEST_XML CLOB)
      EXTERNAL NAME ION3P/GETMYORDER
      LANGUAGE RPGLE
      PARAMETER STYLE GENERAL WITH NULLS
           RESULT SETS 1;;
    ```

Note: To create the Stored Procedure you can use the STRSQL or RUNSQLSTM commands.

2. Create the RPG Program to receive in the parameters and send back XML.

    The program should have the same signature as the Stored Procedure.

    ION can call the program with a base and environment variable to allow you to build the library list accordingly.

    ION can also pass a Request XML into the program by using the [Data] parameter.

From ION Desk (note that BB is the Base ID and EE is the Environment Designator)

```
CALL  ION3P/GETMYORDER('BB', 'EE', '[Data]'
```

The RPG Program should build XML using XML SQL functions or String manipulation.

All Infor BODs require high level `<DataArea>` node and the noun name node, i.e.
`<MyOrder>`.

The program must return a result set in XML so ION can create a BOD from the XML.

Entry Parameters

```
D GETMYORDER      pi
D base                           2    varying options(*varsize)
D env                            2    varying options(*varsize)
D XMLIN                       1024    options(*varsize)
```

Build Library List

```
// build library list
@base = base;
@env = env;
monitor;
  cmd = 'addlible QTEMP';
  QCMDEXC ( cmd : %len(cmd));
on-error *all;
endmon;
monitor;
  cmd = 'addlible QGPL';
  QCMDEXC ( cmd : %len(cmd));
on-error *all;
endmon;
monitor;
  cmd = 'addlible APLUS';
  QCMDEXC ( cmd : %len(cmd));
on-error *all;
monitor;
  cmd = 'addlible APLUS' + %trim(@base);
  QCMDEXC ( cmd : %len(cmd));
on-error *all;
endmon;

@J = 'J';
CFG100P ( @env : @env : @J);
```

Build XML via SQL (requires IBM I 7.1)

```
exec sql declare C1 scroll cursor for
                 SELECT xmlserialize( content
                     xmlelement(name "DataArea",
                        xmlagg(
```

```
                                        xmlelement(Name "MyOrder",
                                        xmlelement(Name "Company", OHCONO),
                                        xmlelement(Name "Order", OHORNO),
                                        xmlelement(Name "OrderGn", OHORGN),
                                        xmlelement(Name "Customer", OHCSNO),
                                        xmlelement(Name "OrderAmt", OHORVL),
                                        xmlelement(Name "InvAmt", OHINAM)
                                        ) )
                                      )
                            AS CLOB(1M) ) as "result" FROM ORHED;

     exec sql open c1;
     exec sql SET RESULT SETS with RETURN TO CLIENT cursor C1;
```

Build XML via RPG String Manipulation and use Temp Table to send back to ION

```
// Use a temp table
EXEC SQL drop TABLE qtemp/TMPXML;
EXEC SQL CREATE TABLE qtemp/TMPXML (XML VARCHAR (1024 ) NOT NULL
        WITH DEFAULT);
    @xml =
        '<DataArea>' +
            '<MyOrder>' +
            '<Company>' + %char(ohcono) + '</Company>' +
            '<Order>' + %trim(ohorno) + '</Order>'  +
            '<OrderGn>' + %editc(ohorgn : 'X') + '</OrderGn>'  +
            '<Customer>' + %char(ohcsno) + '</Customer>'  +
            '<OrderAmt>' + %char(ohorvl) + '</OrderAmt>'  +
            '<InvAmt>' + %char(OHINAM) + '</InvAmt>' +
            '</MyOrder>'  +
        '</DataArea>';

EXEC SQL insert into qtemp/TMPXML values :@xml
         with none;

exec sql declare C2 cursor for Select * from qtemp/TMPXML;
exec sql open c2;
exec sql  SET RESULT SETS with RETURN TO CLIENT cursor C2;
```

3. Add new Document to ION Database Connection Point.

    Add a new connection Point in ION Desk and include to read database and send document.

    In the ION Desk SQL Statement include a call to the stored procedure (note that BB is the Base ID and EE is the Environment Designator).

    CALL ION3P/GETMYORDER('BB', 'EE', '[Data]')

    Press the **Test** Button to validate call.

    Once Test is successful, you can generate a document to live in the ION Registry and be used for Document Flows.

Chapter 12 Map Inbound Custom BODs for Distribution A+ using ION Database Connector

<div style="text-align: right">12</div>

Chapter 11 described mapping outbound Custom BODs from Distribution A+ to ION using the ION Connector. ION Database Connector can also be configured to send BODs to Distribution A+ to write transactions to your Distribution A+ database. When you setup an ION Databases Connector to write to the database, ION will send the XML to a stored procedure which can then handle the XML parsing and any business logic associated with the add or update of the transaction.

Writing a Documents using Stored Procedures

An ION Database Connector will allow you to call into backend stored procedure. IBM DB2 Stored Procedures can wrap in RPG program(s) allowing you to add additional business logic and access to be included when ION sends an inbound document. To accomplish this, you must do the following.

1. Define a Stored Procedure.

    Similar to the database read a DB2 Stored Procedure is used to "wrap" an RPG program to allow ION to directly call into an RPG program. For the cases when you want to return XML and send a reply like an Acknowledgement you should also use the Result Set to return XML just like what was done for the Outbound BODs.

    ```
    DROP PROCEDuRE ION3P/CRTORDERS;
    CREATE PROCEDuRE ION3P/CRTORDERS(
       IN BASE VARCHAR(10),
       IN ENVIRONMENT VARCHAR(10),
       IN REQUEST_XML CLOB)
       EXTERNAL NAME ION3P/CRTORDERS
       LANGUAGE RPGLE
       PARAMETER STYLE GENERAL WITH NULLS
             RESULT SETS 1;;
    ```

    Note: To create the Stored Procedure you can use STRSQL or RUNSQLSTM commands.

2. Create the RPG Program to receive in the XML document from ION.

    The program should have the same signature as the Stored Procedure.

1. ION can call the program with a base and environment variable to allow you to build the library list accordingly.

This program should also receive in a Request XML into the program.

2. The RPG Program should parse XML using RPG XML functions XML-INOT or XML-SAX

The program should return a result set in XML so ION can use it as an Acknowledgement to send to other applications.

Entry Parameters

```
D CRTMYORDER      pi
D base                            2       varying options(*varsize)
D env                             2       varying options(*varsize)
D XMLIN                       32000       options(*varsize)
```

Parse XML via RPG XML functions

IBM SAX Processing is one way to parse an ION XML document. To initiate SAX parsing simply use the key word xml-sax and pass the XML sent from ION into the parser. Once this is done the xmlhandler procedure is run for each node of the XML. Below is an example of using this parsing to create Offline Order Header File (ORHOF) and Offline Order Detail File (ORDOF) records for Inbound Sales Order Processing in Distribution A+.

```
// parse inbound document
xml-sax %handler( xmlHandler : ignoreme )
       %XML( XMLIN : 'doc=string');
// ----------------------------------------------------*
// Function: xmlHandler Export = *yes
P xmlHandler      b

D xmlHandler      PI              10i 0
D   ignore                         1a
D   event                         10i 0 value
D   string                          *   value
D   stringLen                     20i 0 value
D   exceptionId                   10i 0 value
D value           s           65535a    based(string)
D ucs2val         s           16383c    based(string)
D MAX_DEPTH       c                     32
D depth           s              10i 0 inz(0) static
D xPath           s           65535a    varying inz('')
D                                       dim(MAX_DEPTH)
D                                       static
D xPathVal        s           65535a    varying inz('')
D                                       dim(MAX_DEPTH)
D                                       static
D up              c                     'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
D lo              c                     'abcdefghijklmnopqrstuvwxyz'
```

```
  /free
     select;
     when event = *XML_EXCEPTION;
       return 0;
     when event = *XML_START_DOCUMENT;
       depth = 0;
       xPath(*) = '';
       xPathVal(*) = '';

D                                     static
D xPathVal        s         65535a    varying inz('')
D                                     dim(MAX_DEPTH)
D                                     static
D up              c                   'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
D lo              c                   'abcdefghijklmnopqrstuvwxyz'

  /free

     select;
     when event = *XML_EXCEPTION;
       return 0;

     when event = *XML_START_DOCUMENT;
       depth = 0;
       xPath(*) = '';
       xPathVal(*) = '';
     when event = *XML_ATTR_NAME;

        depth += 1;

        xPath(depth) = xPath(depth-1) + '/@'

                      + %subst(%xlate(lo:up:value):1:stringLen);

        xPathVal(depth) = '';


     when event = *XML_END_ELEMENT

       or event = *XML_END_ATTR;

        select;


       // Get Vaules

       when xPath(depth) = '/DataArea/Orders/OrderID';

         Orhofds.xhCONO = 01;

         Orhofds.xhCSNO = 100;

       when xPath(depth) = '/DataArea/Orders/' +
```

```
                                 'Order_x0020_Details/ProductID';

          Ordofds.xditno = %trim(xpathval(depth));



      when xPath(depth) = '/DataArea/Orders/' +
                          'Order_x0020_Details/Quantity';
        Ordofds.xdqtor = getnum(%trim(xpathval(depth)));
      when xPath(depth) = '/DataArea/Orders/' +
                          'Order_x0020_Details/UnitPrice';
        if %trim(xpathval(depth)) <> *blanks;
          @@aslp = getnum(xpathval(depth));
          Ordofds.xdaslp = %editc(@@aslp : 'X');
        endif;

      // end of order
      when xPath(depth) = '/DataArea/Orders';
        @@orsq = 1+ @@orsq;
        Exec SQL
          INSERT Into orhof (xhcono, xhcsno)
                  Values(:orhofDS.xhcono, :orhofds.xhcsno);
          @@orsq = 0;

      // end of order detail
      when xPath(depth) =  '/DataArea/Orders/' +
                            'Order_x0020_Details';
        Exec SQL
          INSERT Into ordof (XDITNO, XDASLP, XDQTOR)
               Values(:ordofDS.xditno,:ordofds.xdaslp,:ordofds.xdqtor);

      endsl;

      depth -= 1;

    when event = *XML_CHARS
      or event = *XML_PREDEF_REF
      or event = *XML_ATTR_CHARS
      or event = *XML_ATTR_PREDEF_REF;
      xPathVal(depth) += %subst(value:1:stringLen);


        when event = *XML_UCS2_REF
                    or event = *XML_ATTR_UCS2_REF;
                     xPathVal(depth) += %char(%subst( ucs2val : 1
                                               : %div(stringLen:2) ));
      endsl;


      return 0;
   /end-free
  P xmlHandler      E
   /free
```

Appendix A Troubleshooting

A

This section describes actions to solve ION integration problems.

Data is not flowing properly

After you complete the configuration setup, if BOD data is not flowing to ION, check these areas to determine the issue:

- ION Desk to find failures related to BOD structure, for example, a message that has an invalid format, or a message that is missing a mandatory field

  For more information, see the "Error BODs" section in the *Infor ION Connect Administration Guide.*

- ION Logging to view processed BODs and to find errors due to connection problems
- ION Queue Viewer to check for unprocessed messages for each integrated application in its message inbox or outbox

To avoid business discrepancies between this Distribution A+ and integrated applications, correct any errors as soon possible.

For more information about troubleshooting BOD flows, see the *Infor ION Connect Administration Guide.*

Monitoring and troubleshooting the data flow between Distribution A+ and integrated Infor application(s)

After the completion of the integration, Administrator user(s) that are assigned to monitor Integrations should monitor ION desk, On-Ramp Service Processor Inquiry and Inbound BOD Inquiry to ensure that outbound BODs are published and Inbound BODs are consumed (posted) successfully and without delay.

If the data is not flowing all the way through the system, there are several places you can look for problems.

Monitor the ION Desk application

Use the ION error functionality to find failures related to BOD structure (i.e. a message has an invalid format, or is missing a mandatory field). Refer to "The Error BODs" subsection of the *Infor ION Administration Guide*. The errors should be addressed as soon as possible to avoid any resulting business discrepancies between Distribution A+ and the integrated application(s).

Use ION Queue Viewer to check for unprocessed messages for each integrated application in its inbox/outbox.

Use ION Logging to show processed BODs and errors (i.e. connection problems).

Review the *Infor ION Administration Guide* for information about troubleshooting BOD flow.

Verify the Distribution A+ application

1. Use the On-Ramp Processor Inquiry (MENU ESMAIN) to ensure that jobs are being processed in Distribution A+.

   - Key ESBJOBS from an IBM i command line or access On-Ramp Service Processor Inquiry (MENU ESMAIN) (quick key ORSPI) from Distribution A+.

   - The On-Ramp Service Processor must be showing a status of Active. Use F6=RESTART TP if not Active.

   - If jobs are active, but not completing and the On-Ramp Service Processor is active, this is usually an indication that Commerce Gateway is not active and needs to be started.

   - If jobs are not active but are just remaining at pending status and the On-Ramp Service Processor is active, this is usually an indication that another environment is conflicting with the current one. Use F2=ALL ENV to see all jobs in all environments.

2. Use Inbound BOD Inquiry (MENU ESMAIN) to ensure that inbound BODs are being consumed successfully by Distribution A+. In Distribution A+, Inbound BODs are processed (posted) to the system as Posting Groups. Each inbound Posting Group is edited and then consumed by Distribution A+ (posted), and Distribution A+ database files are updated with BOD data. Ensure that there all inbound BODs are processed in a timely manner, and there are no pending inbound BODs with error statuses.

   - The errors should be addressed as soon as possible to avoid any resulting business discrepancies between Distribution A+ and integrated application(s). Errors should be corrected if possible and Posting Group needs to be posted again. If resolving errors is not possible, then adjustments need to be done manually in Distribution A+ and/or integrated application(s). BOD should be then marked as resolved.

3. Review Commerce Gateway logs – check for any error messages (for example: connection problems).

- Resolve connection problems and restart Commerce Gateway.

4. Review Distribution A+ inbox and outbox tables.

- These tables are in Commerce Gateway Database defined in Distribution A+ Connection Point.

- Inbox table is called COR_INBOX_ENTRY

- Outbox table is called COR_OUTBOX_ENTRY

- All records must have field **c_was_processed** set to **1** restart Commerce Gateway.

Verify the integrated Infor application(s)

Refer to the application specific Infor integration guide(s) for monitoring and troubleshooting functionality. For example, *Infor Distribution A+ Integration Guide for Infor CRM*.

Data maps

In addition to this *Infor Distribution A+ Configuration Guide for Infor ION*, there are additional guides that contain end to end data mapping information. Refer to:

- Infor Distribution A+ to ICB, Cross BOD Mapping and Descriptions

- Infor Distribution A+ to Infor CRM Business Extension, Cross BOD Mapping and Descriptions

- Infor Distribution A+ to Infor Inforce, Cross BOD Mapping and Descriptions

- Infor Distribution A+ BOD Mapping and Descriptions

They are organized by elements in the BOD, according to the BOD schema definition. These documents can be found on the Infor Support Portal under the Documentation tab for **Product Line Aplus**.

Appendix B BOD Overview

This section provides an overview of BODs. It explains how BODs are defined, and how some of the most common XML tags are used. This section also describes the types of BODs that are used in Infor applications.

BOD message structure

ION messages are called Business Object Documents (BODs). These messages are based on standards that are developed by the Open Applications Group, Inc. (OAGi). BODs are XML documents that consist of a verb component and a noun component. The BOD name consists of a verb and a noun, for example, SyncPayableTransaction, ProcessLocation, and ProcessSupplierPartyMaster. A BOD XML includes these areas:

- Verbs specify the action performed on the noun, for example, Process or Sync.
- Nouns specify the business-specific data, for example, Payable Transaction or Location.
- Elements are the XML tags within the BOD that describe the data. They can be hierarchical, as shown in this example:

```
<PaymentTerm>
<Description languageID="EN">2% in 10, Net Due in 30</Description>
<Term>
<ID>2/10N30</ID>
</Term>
</PaymentTerm>
```

- Attributes describe the elements. For example, UnitCode is the attribute in this element:

```
<Quantity UnitCode="EA">
```

- When a value in a BOD is constructed by concatenation of multiple values, an underscore (_) character is used to separate the values.
- Data that is not defined for a noun can be passed in the UserArea of a BOD. With Infor applications, the only element that can be placed in the UserArea is Property. Multiple instances of Property within the UserArea can exist. Only the UserArea fields in the header, detail, and root of a noun are supported by Infor. If you require additional fields, you must define them in the UserArea.

Each Property element contains a NameValue element that holds the value being passed. The NameValue element also contains these attributes:

- @name holds the custom field name
- @type specifies the data type of the value, for example, StringType and DateType

This example shows a UserArea:

```
<UserArea>
    <Property>
        <NameValue name="MyField" type="StringType">Sample Data</NameValue>
    </Property>
</UserArea>
```

- The Document ID element is a unique identification of the represented business object. The ID is used in a BOD to reference other BODs. For the nouns SalesOrder, PurchaseOrder, and CustomerReturn, the ID is the order number. For other nouns, a naming convention is defined that makes the ID unique.
- Party ID elements identify an external entity such as a Customer, Supplier, and ShipTo. Location ID elements identify sites within the enterprise. Within a transactional BOD such as a Shipment, if the ShipToParty/PartyIDs/ID is populated, the ID points to a ShipToPartyMaster record. If the ShipToParty/Location/ID is populated, the ID points to a warehouse or other site within the enterprise.
- All date and time elements within a BOD are presented in Coordinated Universal Time (UTC). The dates are formatted as shown in this example: 2011-08-13T15:30Z. Dates that are received without a time value have the default time of 00:00, which is midnight.

System of record

A piece of information, for example, a customer address, is maintained by only one system, which is the system of record (SOR). The SOR publishes all changes for that piece of information to ION through a Sync message. Other applications that want to add or update that piece of information must make a request to the System of Record by using a Process message.

The system of record does not need to know which applications require the information. The entire contents of the document are published in the Sync message. All applications that subscribe to the message receive a copy of the message from ION.

BOD verbs

Infor uses these verbs:

- A Process message is a point-to-point message that is used to request a service from another application. The Process message is usually sent from a non-system of record to the system of record (SOR). These action codes are supported:

  - Add: Requests that the other application create new document or record.

  - Change: Requests that the other application update an existing document or record. The elements that have a change are required.

- An Acknowledge message is a response to the Process message. The Acknowledge message notifies the requesting system whether the request was accepted or rejected. In many cases, the Acknowledge verb has additional information, such as the document's ID that is assigned by the system of record.

- A Sync message is a broadcast message that is published by the system of record. The Sync message is used to inform other applications about the latest information for the noun. It is published after a business event causes a change in the data. These action codes are supported:

  - Add: Notifies other applications that a document or record has been created.

  - Replace: Notifies other applications that an existing document or record has been updated. The entire document or record is included in the BOD. The current state of the document or record is also shown.

- A Get message is a point-to-point message that requests information about one or more documents or records. When the GetInventoryCount BOD is sent, the inventory balance is requested for only one item. For all other Get requests, the application requests a copy of all documents from the system of record, which is generally for an initial data load or a disaster recovery.

- A Show message is a message that is sent directly to another application by the system of record as a response to a Get message. A Show message can contain multiple records or documents within one message. Show messages are used typically to reinitialize an application, rather than for general initialization.

- A Confirm message is used to notify users of an error condition. The ConfirmBOD is sent to ION and can be viewed in ION Desk.

Note: The OAGIS Delete verb is supported by Infor for special tenant-level master data only. Otherwise, this verb is not supported. For auditing purposes, you should cancel documents and records and not delete them.

Message delivery

Except for the Show message that is used for initial data loads, ION messages are delivered asynchronously. Applications send only one document per message.

Data mapping

Information about how Distribution A+ database elements or calculations are mapped to specific BOD elements is explained in these guides:

- *Distribution A+ Outbound BOD Mapping and Descriptions*: For each BOD that Distribution A+ has certified, this document provides the source and definition of the data in each BOD element that is sent from Distribution A+ through ION. If the source that is used differs based on the specific criteria within Distribution A+, then the document also provides the details about the data that is published.

- *Distribution A+ Inbound BOD Mapping and Descriptions*: For each BOD that Distribution A+ has certified, this document provides the details about how each product receives information or processes requests from the inbound BOD. For example, the source might be an extension product, a result of a workflow, or a drillback function from Infor Ming.le. If a BOD updates a source table, the exceptions and handling are described.

These guides are organized according to the elements in each BOD.

Note: Infor reserves the right to change BOD structures. Although Infor will attempt to provide backward compatibility, custom integrations that are built on a specific version of a BOD might require modification to be compatible with newer versions of Infor components.

Appendix C Business events that generate outbound BODs

C
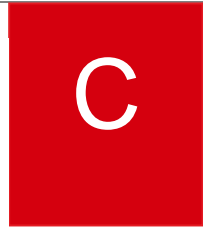
This table shows the events, which are user actions in Distribution A+ that generate an outbound BOD from Distribution A+.

These events may or may not trigger publishing of an outbound BOD depending on Distribution A+ settings, actions taken, and data affected.

This table is a guide on where events can be created.

Verb	Noun	User action to generate the BOD
Sync	AccountingChart	N/A
Sync	AccountingEntity	Company Name Maintenance (quick key CNM), menu XAFILE.
Sync	BillToPartyMaster	Run Customer/Ship to Master Maintenance (quick key CUSM), menu ARFILE. These files when events have been registered for them with the below programs: • Customer Master File CUSMS These programs when run as part of any function: AR800. Inbound Customer Master Post that runs when inbound Process CustomerPartyMaster BOD is consumed.
Sync	ChartOfAccounts	Run G/L Account Maintenance (quick key GLAM), menu GLFILE. Run G/L Account Maintenance (quick key GLAM), menu G2XFER. Run Copy G/L Accounts (quick key CGLA), menu GLFILE. These files when events have been registered for them with the below programs: • G/L Account Master File GLMST These programs when run as part of any function: GL800, GL915.
Sync	CodeDefinition (A/R Terms Codes)	A/R Terms Code Maintenance (quick key ARTCM), menu ARFILE.
Sync	CodeDefinition (Currencies)	Run Currency Code Maintenance (quick key CECM), menu ICFILE.

Verb	Noun	User action to generate the BOD
Sync	CodeDefinition (Customer Classes)	Run Customer Class Maintenance (quick key CCLM), menu ARFILE.
Sync	CodeDefinition (Segment Values)	Run Copy GL Accounts (quick key CGLA), menu GLFILE. Run Segment Value Maintenance (quick key SVM), menu GLFILE. Run G/L Options Maintenance (quick key GLOM), menu GLFIL2. These file when events have been registered for them with the below programs: • G/L Segment Value File GLSEG • Order Control File ORCTL These programs when run as part of any function: GL915, GL830, GL810.
Sync	CodeDefinition (Territories)	Run Territory Maintenance (quick key TM), menu ARFILE.
Sync	ContactMaster	Contact Maintenance that runs from Customer ShipTo Maintenance. Offline Contact Maintenance that runs when inbound ProcessContactMaster is consumed. XACONTC/CONTC trigger
Sync	CustomerPartyMaster	Run Customer/Ship to Master Maintenance (quick key CUSM), menu ARFILE. These files when events have been registered for them with the below programs: • Customer Master File CUSMS These programs when run as part of any function: AR800. Inbound Customer Master Post that runs when inbound Process CustomerPartyMaster BOD is consumed. When a Contact is added

Verb	Noun	User action to generate the BOD
Sync	CustomerReturn	Run Enter, Change and Ship Orders (quick key OE), menu OEMAIN. OE101C
		Run Revalue Open Orders (quick key ROO), menu ICRVAL. IC01A
		Run Invoices quick key (INV), menu OEMAIN. OE526
		Run Reorganize A+ History Files (quick key RHF), menu XAMAST.OE784
		Run Automatic Backorder Release (quick key ABR), menu OEMAST. OE740
		Run Day-End Processing (quick key DEC), menu XAMAST. OE640
		Run Automated Day-End Processing (quick key ADIJL), menu XAFIL3.
		Run RGA Slips (quick key RGA), menu OEMAIN. OE541
		Run Carrier Order Inquiry (quick key COI), menu OEMAIN. OE260
		Run Offline Order Entry (quick key OLOE), menu OEMAIN. OE130
		Run Delete Open Orders (quick key OD), menu OEMAIN. OE121
		Run Release Held Orders, Quotes, Backorders & Futures (quick key OR), menu OEMAIN. OE111
		Run Credit Card Authorization (quick key CCATH), menu EPMAIN. OEP700
		Run Void Authorization (quick key VAUTH), menu EPMAIN OEP670.
		Run EP Transaction Inquiry (quick key EPIQ), menu EPMAIN.
		Run Change the Status of an Invoiced Order. UNINV
		Run Remove In Use Codes from an Order. UNUSE
		These programs when run as part of any function: IC010A, IC012, OEP003, OEP670, OEP700, OEP740, OE101C, OE111, OE121, OE130, OE260, OE526, OE530, OE541, OE640, OE784, UNINV, UNUSE.
Sync	FinancialCalendar	Run Financial Calendar Maintenance (quick key FCM), menu GLFIL2. These files when events have been registered for them with the below programs: • G/L Calendar File GLCAL These programs when run as part of any function: GL840.
Sync	InventoryHold	N/A

Verb	Noun	User action to generate the BOD
Sync	Invoice	Run Print Invoices (quick key INV), menu OEMAIN Run Day-End Processing (quick key DEC), menu XAMAST. Run Enter, Change and Ship Orders (quick key OE), menu OEMAIN. Run Point Of Sale Entry (quick key PSEN), menu PSMAIN. Run Automatic Invoicing (quick key ADIJL), menu OEFIL3. Run Automated Day-End (quick key ADIJL), menu OEFIL3. These files when events have been registered for them with the below programs: • Order History Header File HSHED • Open Order Header File ORHED These programs when run as part of any function: OE526, OE640.
Sync	ItemMaster	Run Item Master Maintenance (quick key IMM), menu IAFILE. These files when events have been registered for them with the below programs: • Item Master File ITMST These programs when run as part of any function: IA810.
Sync	Location	Run Warehouse Numbers Maintenance (quick key WMN), menu IAFILE. These files when events have been registered for them with the below programs: • Order Control File ORCTL These programs when run as part of any function: IA910.

Verb	Noun	User action to generate the BOD
Sync	PayableTransaction	Run Voucher Entry (quick key VT), menu APMAIN.
		Run Create Vouchers from Templates (quick key CVFT), menu APMAIN.
		Run Post Vouchers (quick key VP), menu APMAIN.
		Run Open Payable Maintenance (quick key OPM), menu APMAIN.
		Run Incoming Documents (quick key RSID), menu EIMAIN.
		Run Post Check Reversals (quick key CRP), menu APCHCK.
		Run Reorganize A+ History Files (quick key RHF), menu XAMAST.
		Run Maintain 1099 Information (1099M), menu APMAST.
		Run Record Check Numbers (quick key RCN), menu APCHCK.
		Run Post Manual Checks (quick key MCP), menu APCHCK.
		Run Start Auto Release Incoming/Outgoing EDI Jobs (quick key SAEI), menu EIMAST.
		Run Revalue Open Payables (quick key ROP), menu ICRVAL.
		Run Rebate Extract Posting (quick key REXP), menu OERMAIN.
		These files when events have been registered for them with the below programs: • Accounts Payable History Detail File APHDT • Accounts Payable History Header File APHHD • Accounts Payable Open Payables Header File APOHD • Accounts Payable Voucher Entry Detail File APVDT • Accounts Payable Voucher Entry Header File APVHD These programs when run as part of any function: AP110A, AP110B, AP111, AP112, AP113, AP114, AP115, AP116, AP130B, AP135, AP670, AP671, AP761, AP830, CIDAPOHD, EI120, IC004A, IC004B, IC621A, OER132, OER159E, PO600D, PO600E.
Sync	Person	Run Salesrep Maintenance, menu SAFILE. Run Buyer Maintenance, menu POFILE.

Verb	Noun	User action to generate the BOD
Sync	PurchaseOrder	Run Print Purchase Orders (quick key PO), menu POMAIN.
		Run Work With Special Orders (quick key MSO), menu POMAIN.
		Run Special Order Automatic Req Creation (quick key SOA), menu POMAST.
		Run Automate Suggested Orders (quick key ASO), menu POFIL2.
		Run Day-End Processing (quick key DEC), menu XAMAST.
		Run Automated Day-End Processing (quick key ADIJL), menu XAFIL3.
		Run Post Vouchers (quick key VP), menu APMAIN.
		Run PO Receipts Register quick key (PORR), menu POMAIN.
		Run PO Receipts Register (quick key PORR), menu WOMAIN.
		Run Receipt Post (quick key WORP), menu WOMAIN.
		Run Warehouse Management (quick key WM), menu WMMAIN.
		Run Transaction Manager (quick key TRM), menu RFMAIN.
		Run Enter, Change and Ship Orders (quick key OE), menu OEMAIN.
		Run Delete Open Orders (quick key OD), menu OEMAIN.
		Maintain a Purchase Order through Req/PO Inquiry (quick key ORI), menu POMAIN.
		Run Reorganize A+ History Files (quick key RHF), menu XAMAST.
		Run S/O Change Request Inquiry (quick key SRQP), menu POMAIN if PO affected.
		Run Special Order Automatic Requisition Creation through Application Mail link.
		Run Release Held Orders, Quotes, Backorders and Futures (quick key OR), menu OEMAIN.
		Run Off Line Order Entry (quick key OLOE), menu OEMAIN.
		Run Release Blanket Orders (quick key ORB), menu OEMAIN.
		Run Maintain Special Orders (quick key WOSOM), menu WOMAIN.
		Run Receive Incoming Purchase Orders (850) through Incoming Documents (quick key RSID), menu EIMAIN.
		Create orders through Gateway.
		Create orders through Mobile Order Entry / Mobile Order Management.
		Create orders through Storefront.
		Create Duplicate Order through Customer Order/Shipment (quick key CSI), menu OEMAIN.
		These files when events have been registered for them with the below programs: - Purchase Order History Detail File PHDET - Purchase Order History Header File PHHED - Open Purchase Order Header File POHED These programs when run as part of any function: PO100A, PO170B, PO515, PO615A, PO622, PO796, PO800, SO610.

Verb	Noun	User action to generate the BOD
Sync	Quote	Run Enter, Change and Ship Orders (quick key OE), menu OEMAIN. Run Delete Open Orders (quick key OD), menu OEMAIN. Run Release Held Orders, Quotes, Backorders & Futures (quick key OR), menu OEMAIN. Run Offline Order Entry (quick key OLOE), menu OEMAIN. **Note**: All of the above actions to generate the BODs only apply for an order type of "Q" (Quote).
Sync	ReceivableTransaction	Run Cash & Adjustment Post (quick key ARCP), menu ARMAIN. Run Invoice Post (quick key ARIP), menu ARMAIN. Run Mark Disputed Invoices (quick key MDI), menu ARMAIN. Run Customer A/R Inquiry (quick key CUSI), menu ARMAIN. Run Day-End Processing (quick key DEC), menu XAMAST. Run Automated Day-End Processing (quick key ADIJL), menu XAFIL3. Run Credit Card Settlement (quick key CCSTL), menu EPMAIN. Run A/R History Update (quick key ARHUP), menu ARMAST. Run Paid Invoice Purge (quick key PIPT), menu ARMAST. Run Month End Processing (quick key ARMEC), menu ARMAST. Run Reorganize A+ History Files (quick key RHF), menu XAMAST. Run eBill Options Maintenance (quick key EBOM), menu EBILL. Run Auto-Pay Job Maintenance (quick key EMID), menu EPFILE. These files when events have been registered for them with the below programs: • A/R Open Receivables Header File ARDHD • A/R History Invoice Header File ARHHD These programs when run as part of any function: AR102C, AR123, AR150, AR151, AR202, AR603, AR607, AR620, AR642, AR653, AR661, AR722, OEP640.
Sync	ReceiveDelivery	Run PO Receipts Register (quick key PO), menu POMAIN Run Reorganize A+ History Files (quick key RHF), menu XAMAST. These files when events have been registered for them with the below programs: • Purchase Order Receipt History File RCPT These programs when run as part of any function: PO610, AP781.

Verb	Noun	User action to generate the BOD
Sync	Requisition	Run Enter or Change Requisitions (quick key REQ), menu POMAIN.
		Run Print Purchase Orders (quick key PO), menu POMAIN.
		Run Special Order Automatic Req Creation (quick key SOA), menu POMAST.
		Run Work With Special Orders (quick key MSO), menu POMAIN.
		Run Automate Suggested Orders (quick key ASO), menu POFIL2.
		Run to Vendor Processing (quick key RTVP), menu POMAIN.
		Run Create Req From Suggested Orders (quick key CRSO), menu POREPT.
		Run Suggested Order Maintenance (quick key SUOM), menu POREPT.
		Run S/O Change Request Inquiry (quick key SRQP), menu POMAIN.
		Run Special Order Automatic Requisition Creation through Application Mail link.
		Run Release Held Orders, Quotes, Backorders and Futures (quick key OR), menu OEMAIN.
		Run Off Line Order Entry (quick key OLOE), menu OEMAIN.
		Run Release Blanket Orders (quick key ORB), menu OEMAIN.
		Run Maintain Special Orders (quick key WOSOM), menu WOMAIN.
		Run Receive Incoming Purchase Orders (850) through Incoming Documents (quick key RSID), menu EIMAIN.
		Create orders through Gateway.
		Create orders through Mobile Order Entry / Mobile Order Management.
		Create orders through Storefront.
		Create Duplicate Order through Customer Order/Shipment (quick key CSI), menu OEMAIN.
		These files when events have been registered for them with the below programs: • Purchasing Requisition Detail File RQDET • Purchasing Requisition Header File RQHED These programs when run as part of any function: PO100A, PO100B, PO140, PO170B, PO397, PO515, SO610, SO630.

Verb	Noun	User action to generate the BOD
Sync	SalesOrder	Run Enter, Change and Ship Orders (quick key OE), menu OEMAIN. Run Revalue Open Orders (quick key ROO), menu ICRVAL. Run Invoices quick key (INV), menu OEMAIN. Run Reorganize A+ History Files (quick key RHF), menu XAMAST. Run Pick Lists (quick key PCK), menu OEMAIN. Run Transaction Manager (quick key TRM), menu RFMAIN. Run Automatic Backorder Release (quick key ABR), menu OEMAST. Run Inventory Commit/Uncommit (quick key ICU), menu OEMAST. Run Post Vouchers (quick key VP), menu APMAIN. Run Day-End Processing (quick key DEC), menu XAMAST. Run Automated Day-End Processing (quick key ADIJL), menu XAFIL3. Run RGA Slips (quick key RGA), menu OEMAIN. Run Acknowledgements (quick key ACK), menu OEMAIN. Run Carrier Order Inquiry (quick key COI), menu OEMAIN. Run Release Blanket Orders (quick key ORB), menu OEMAIN. Run Release "New Customer" Orders (quick key ORN), menu OEMAIN. Run Offline Order Entry (quick key OLOE), menu OEMAIN. Run Incoming Documents (quick key RSID), menu EIMAIN. Run Ship Confirm Multiple Orders (quick key SC), menu OEMAIN. Run Delete Open Orders (quick key OD), menu OEMAIN. Run Release Held Orders, Quotes, Backorders & Futures (quick key OR), menu OEMAIN. Run PO Receipts Register (quick key PORR), menu POMAIN. Run Warehouse Management (quick key WM), menu WMMAIN. Run Credit Card Authorization (quick key CCATH), menu EPMAIN. Run Void Authorization (quick key VAUTH), menu EPMAIN. Run EP Transaction Inquiry (quick key EPIQ), menu EPMAIN. Run Enter or Change Receivers or PO Receipts (quick key ECPR), menu POMAIN. Run Purchase Orders (quick key PO), menu POMAIN. Run Work With Special Orders (quick key MSO), menu POMAIN. Run Special Order Change Request Inquiry (quick key SRQP), menu POMAIN. Run Confirm Box Shipments (quick key CBS), menu WMMAIN. Run Box Maintenance (quick key BOM), menu WMMAIN. Run Change the Status of an Invoiced Order. Run Remove In Use Codes from an Order. These files when events have been registered for them with the below programs: • Back Order Creation File BOCRT • Order History Header File HSHED • O/E Order Split Control File OESCT • Open Order Detail File ORDET • Open Order Header File ORHED These programs when run as part of any function: IC010A, IC012, OEBACK, OEC110, OEP003, OEP670, OEP700, OEP740, OE101B, OE101C, OE109, OE109A, OE111, OE121, OE125, OE130, OE131, OE151, OE260, OE506, OE512, OE515B, OE515D, OE526, OE530, OE541, OE640, OE700, OE721, OE750, OE784, PO161, PO515, PO610, PO620, PO650, RF120, RF170, SO101B, SO610, SO620, UNINV, UNUSE, WM165, WM185. **Note**: All of the above actions to generate the BODs only apply for an order type of "O" (Order) or 'I' (Invoice).

Verb	Noun	User action to generate the BOD
Sync	ShipFromPartyMaster	Run Vendor Maintenance (quick key WM), menu POFILE. These files when events have been registered for them with the below programs: • Purchasing Vendor Master File VENDR These programs when run as part of any function: PO900.
Sync	Shipment	Ship Sales Order from Enter, Change and Ship Orders (quick key OE), menu OEMAIN. Ship Sales Order from Ship Confirm Multiple Orders (quick key SC), menu OEMAIN. Run Invoice Print, Update Invoice History File, Order History Purge, Confirm Box Shipments Print Pick Slip for Sales Order from Enter, Change and Ship Orders, menu OEMAIN. Print Pick Slip for Sales Order from Pick Slip Print, (quick key OE) menu OEMAIN. Special Order Line Item Change, (quick key MSO) menu POMAIN. Change Special Order Header Change, (quick key MSO) menu POMAIN.
Sync	ShipToPartyMaster	Run Customer/Ship to Master Maintenance (quick key CUSM), menu ARFILE. These files when events have been registered for them with the below programs: • Customer Shipping Address File ADDR These programs when run as part of any function: AR800. Inbound Customer ShipTo Post that runs when inbound Process ShipToPartyMaster BOD is consumed.
Sync	SourceSystemGLMovement	Run Post Transactions (quick key PT), menu GLMAIN. Run Post Bank Account Transactions (quick key BAP), menu APMAIN. Run Post Pre-Receipt Inv Val Group (quick key PRIVP), menu APMAIN. Run Post Vouchers (quick key VP) menu APMAIN. Run Record Check Numbers (quick key RCN), menu APCHCK. Run Post Manual Checks (quick key MCP), menu APCHCK. Run Post Check Reversals (quick key CRP), menu APCHCK. Run Cash & Adjustment Post (quick key ARCP), menu ARMAIN. Run Invoice Post (quick key ARIP), menu ARMAIN. Run Month End Processing (quick key ARMEC), menu ARMAST. Run Post IA Transactions to G/L (quick key IAGLP), menu IAMAST. Run IC Revaluations, menu ICRVAL. Run Rebate Extract Posting (quick key REXP), menu OERMAIN. Run Day-End Processing (quick key DEC), menu XAMAST. Run any job that creates postings to GL. Run Reorganize Infor ERP A+ History Files (quick key RHF), menu XAMAST. Run Load beginning Account Balances (quick key LBAB), menu GLMAST. Run Budget Maintenance (quick key BFM), menu GLFILE. Run Copy GL Budgets (quick key CGLB), menu GLFILE. Run Load Beginning Balances Edit (quick key LBBE), menu GLMAST. These files when events have been registered for them with the below programs: • G/L Summary File GLSUM • G/L Budgets File GLBUD • G/L Current Retained Earnings File GLCRE These programs when run as part of any function: GL130F, GL776, GL860, GL890, GL940, GLCREB.

Verb	Noun	User action to generate the BOD
Sync	SupplierPartyMaster	Run Vendor Maintenance (quick key WM), menu POFILE. These files when events have been registered for them with the below programs: Purchasing Vendor Master File VENDR These programs when run as part of any function: PO900.

Appendix D Inbound BOD usage

D

This table shows the incoming BODs that Distribution A+ can accept. It also indicates, when possible, what area of Distribution A+ shows the processed inbound data.

Verb	Noun	Destination of processed data in Distribution A+
Process	AplusWorkflow	User action from ION Workflow task
Process	ContactMaster	Contact Master File
Process	CustomerPartyMaster	Customer Master File
Process	Quote	Sales Order type 'Q' (Header, Detail and related files)
Process	SalesOrder	Sales Order type 'O' (Header, Detail and related files).
Process	ShipToPartyMaster	Customer/Ship to Master

Appendix E BODs used in Distribution A+ integrations

This appendix contains two tables that list the BODs that are available with Distribution A+ in the rows. The applications columns list the products where the BOD is used in Distribution A+'s integration with that product. Where the application is blank, the BOD is not currently used by interfacing applications, but the BOD is generated by Distribution A+ and is available to be processed through ION and any application that is set up to receive it.

Outbound BODs from Distribution A+ to integrated applications

Verb	Noun	To Applications
Sync	AccountingChart	Infor Distribution Business Analytics
Sync	AccountingEntity	Infor Distribution Business Analytics
Sync	BillToPartyMaster	Inforce / Infor CRM Business Extension / Infor CRM
Sync	ChartOfAccounts	Infor Distribution Business Analytics
Sync	CodeDefinition	Inforce / Infor CRM Business Extension / Infor CRM / Infor Distribution Business Analytics
Sync	ContactMaster	Inforce / Infor CRM Business Extension / Infor CRM
Sync	CustomerPartyMaster	Inforce / Infor CRM Business Extension / Infor CRM / Infor Distribution Business Analytics
Sync	CustomerReturn	Inforce / Infor CRM Business Extension
Sync	FinancialCalendar	Infor Distribution Business Analytics
Sync	InventoryAdjustment	N/A
Sync	InventoryHold	N/A

Sync	Invoice	Inforce / Infor CRM Business Extension / Infor CRM / Infor Distribution Business Analytics
Sync	ItemMaster	Inforce / Infor CRM Business Extension / Infor CRM / Infor Distribution Business Analytics
Sync	Location	Infor CRM / Distribution Business Analytics
Sync	Person	Infor CRM / Distribution Business Analytics
Sync	PurchaseOrder	N/A
Sync	Quote	Inforce / Infor CRM Business Extension / Infor CRM
Sync	ReceivableTransaction	Inforce / Infor CRM Business Extension / Infor CRM
Sync	ReceiveDelivery	N/A
Sync	Requisition	N/A
Sync	SalesOrder	Inforce / Infor CRM Business Extension / Infor CRM / Infor Distribution Business Analytics
Sync	ShipFromPartyMaster	N/A
Sync	Shipment	Inforce / Infor CRM Business Extension / Infor CRM
Sync	ShipToPartyMaster	Inforce / Infor CRM Business Extension / Infor CRM
Sync	SourceSystemGLMovement	Infor Distribution Business Analytics
Sync	SupplierPartyMaster	N/A

Inbound BODs to Distribution A+ from integrated applications

Verb	Noun	From Applications
Process	BillToPartyMaster	Not taken in by A+ ERP
Process	CodeDefinition	Not taken in by A+ ERP
Process	ContactMaster	Inforce / Infor CRM Business Extension
Process	CustomerPartyMaster	Inforce / Infor CRM Business Extension
Process	InventoryAdjustment	N/A
Process	Quote	No integration at this time (future Infor CRM)
Process	ReceivedDelivery	N/A
Process	SalesOrder	No integration at this time (future Infor CRM)
Process	ShipToPartyMaster	Inforce / Infor CRM Business Extension

Appendix F Custom BOD Topics

Trigger Conditional Program

```
//Event Condition PGM
dnewrecord        e ds                    extname(cusms)
doldrecord        e ds                    extname(cusms)    prefix(o_)
dbuffer           ds          32000
D trgfile                         10
D trgevent              31     31
D oldoffset             49     52b00
D oldlength                     9b00
D newoffset             65     68b00
D newlength                     9b00

dskip             s              1
dpos              s              5  0
devid             s              4
D bufferlen       s              4

c     *entry        plist
c                   parm                    pass
c                   parm                    evid
c                   parm                    buffer
 /free

  // save the new values
  pos = newoffset +1;
  newrecord = %subst(buffer:pos:newlength);

  skip = 'Y';
  // save the old values
  pos = oldoffset +1;
  oldrecord = %subst(buffer:pos:oldlength);

  // if create or delete don't skip record
  if trgevent = '1'
  or trgevent = '2';
  skip = 'N';
  endif;
```

```
// Only include record if address changes
if    o_cmcad1  <>   cmcad1 OR
 o_cmcad2  <>   cmcad2 OR
 o_cmcad3  <>   cmcad3 OR
 o_cmcad4  <>   cmcad4;
   pass = 'N';
endif;
*inlr = *on;
```

BOD Mapper Program

```
/COPY QCPYSRC,H_SPEC
//*********************************************************************
//*********************************************************************
//*      program number - ZESOUTMAP                              **
//*      program name   - Custom Mapper                          **
//*      This program will map data from APLUS to a custom noun  **
//*                                                              **
//*                  This is an EXAMPLE and AS IS
//*********************************************************************
//*********************************************************************
     // Custom Customer Master
   Fzcusms    if  e          k disk
    // A+ Outbound BOD Header file
   Fbodohd    uf  e          k disk
    // A+ Outbound BOD Detail file
   Fbododt    uf a e         k disk
    // A+ Outbound BOD Attributes file
   Fbodoat    o  a e         k disk
    // A+ Document ID Information file  pgm prefix: di
   Fsoadinf   uf a e         k disk

   D zesoutmap        pr                extpgm('ZESOUTMAP')
   D shutdown                   1
   D errtxt@                   250
   D p2bodid                    50
   D p2pgnm                     10
   D p2verb                     50
   D p2noun                     50
   D p2action                   50
   D p2key01                    50
   D p2key02                    50
   D p2key03                    50
   D p2key04                    50
   D p2key05                    50
   D p2key06                    50
   D p2key07                    50
   D p2key08                    50
   D p2key09                    50
   D p2key10                    50
   D p2key11                    50
```

```
D p2key12                      50
D p2key13                      50
D p2key14                      50
D p2key15                      50
D p2file                       10
D p2docid                     100
D p2acctentity                  2  0
D p2evid                        4

D ZESOUTMAP       pi
D shutdown                      1
D errtxt@                      250
D p2bodid                      50
D p2pgnm                       10
D p2verb                       50
D p2noun                       50
D p2action                     50
D p2key01                      50
D p2key02                      50
D p2key03                      50
D p2key04                      50
D p2key05                      50
D p2key06                      50
D p2key07                      50
D p2key08                      50
D p2key09                      50
D p2key10                      50
D p2key11                      50
D p2key12                      50
D p2key13                      50
D p2key14                      50
D p2key15                      50
D p2file                       10
D p2docid                     100
D p2acctentity                  2  0
D p2evid                        4
D @@csno         s                       like(zzcsno)
D @@cono         s                       like(zzcono)
D @@tagtype      s                       like(tagtype)
D @@elem         s                       like(elem)
D @@value        s                       like(value)
D @@attr         s                       like(attr)
D @@msg          s                       like(msg)
D @@batchsq      s                       like(batchsq)
D @@bodidsq      s                       like(bodidsq)
D @@bodatsq      s                       like(bodidsq)
D ctlky          s              11
D @@invtype      s              50
D @@duedatetime  s               z
D @@invdatetime  s               z
D edited_dattim  s              50
D datetime       s               z
D dateIn         s               d
D timeIn         s               t
```

```
D @@docid          s                        like(didocid)
D @@docvarid       s              50
D zero_company     s               2  0
D schemename       s             100
D schemeagencyid   s             100
D schemeagencynm   s             100
D languageid       s             100
D agencyrole       s             100
D listid           s             100
D listagencyid     s             100
D @@bodidky        s                        like(bodidky)
D @@revlvl         s                        like(revlvl)
D @@action         s                        like(action)
D @x               s               3  0

 // full key for CUSMS
C     zcm0key       klist
C                   kfld                    @@cono
C                   kfld                    @@csno
 // full key for BODODT
C     xx0key        klist
C                   kfld                    p2bodid
C                   kfld                    @@batchsq
C                   kfld                    @@bodidsq
 // full key for SOADINF
C     di0key        klist
C                   kfld                    p2noun
C                   kfld                    @@docid
C                   kfld                    @@revlvl


 /free

   // Shut Down program
   if shutdown = 'Y';
     *inlr = *on;
     return;
   endif;

   // get data
   @@cono = %int(%subst(p2key01 : 1 : 2));
   @@csno = %int(%subst(p2key02 : 1 : 10));
   Chain zcm0key zcusms;
   if not %found;
     errtxt@ = 'ES204 - Bad or invalid data in event key fields ';
   else;

   // Revision Level
   clear @@revlvl;

   // Document Variation ID
   chain di0key soadinfr;
   if %found(soadinf);
     divarid = divarid + 1;
```

```
      dibodid = p2bodid;
      dilsverb = p2verb;
      exsr $chk_action;
      dilsact = @@action;
      dilsdt = %date;
      dilstm = %time;
      update soadinfr;
    else;
      clear soadinfr;
      didocid = @@docid;
      divarid = 1;
      dibodid = p2bodid;
      dialtky1 = %editc(zzcono:'X');
      dialtky2 = %editc(zzcsno:'X');
      dinoun = p2noun;
      direvlvl = @@revlvl;
      dibodidky = @@bodidky;
      diinverb = p2verb;
      exsr $chk_action;
      diinact = @@action;
      diindt = %date;
      diintm = %time;
      dilsverb = p2verb;
      dilsact = @@action;
      dilsdt = %date;
      dilstm = %time;
      write soadinfr;
    endif;
    // Begin Group
    @@tagtype = 'B';
    @@elem = 'MyCustomer';
    exsr Add_Bod_Detail;
    endif;

    // Begin Group
    @@Tagtype = 'B';
    @@elem = 'DocumentID';
    exsr Add_Bod_Detail;

    @@tagtype = 'D';
    @@elem = 'ID';
    @@value = %editc(zzcsno : 'X');
    exsr Add_Bod_Detail;

    // Add Acouting EnTity Attribute
    @@attr = 'accountingEntity';
    @@value = %editc(zzcono : 'X');
    exsr Add_Bod_Attribute;

    // End Group
    @@tagtype = 'E';
    @@elem = 'DocumentID';
    exsr Add_Bod_Detail;
```

```
        // Begin Group
        @@Tagtype = 'D';
        @@elem = 'MyField';
        @@value = zzfield;
        exsr Add_Bod_Detail;


        // End Group
        @@tagtype = 'E';
        @@elem = 'MyCustomer';
        exsr Add_Bod_Detail;

        return;
        //***********************************************************
        // Add BOD Detail record
        begsr Add_BOD_Detail;

        // Increment BOD Detail Sequence
        @@bodidsq =  @@bodidsq + 1;
        bodidsq = @@bodidsq;

        bodid = p2bodid;
        batchsq = @@batchsq;
        tagtype = @@tagtype;
        hasattr = 'N';
        elem = @@elem;
        value = @@value;
        msg = @@msg;

        write bododtr;

        clear @@tagtype;
        clear @@elem;
        clear @@value;
        clear @@msg;

        endsr;

        //***********************************************************
        // Add BOD Attribute record
        begsr Add_BOD_Attribute;

        // Increment BOD Attribute Sequence
        @@bodatsq =  @@bodatsq + 1;
        bodatsq = @@bodatsq;

        bodid = p2bodid;
        batchsq = @@batchsq;
        bodidsq = @@bodidsq;
        attr = @@attr;
        value = @@value;
        msg = @@msg;

        write bodoatr;
```

```
   clear @@attr;
   clear @@value;
   clear @@msg;

   // Mark BOD Detail as having attributes
   exsr Mark_Bod_Detail;

   endsr;

   //**********************************************************
   // Mark BOD Detail record as having related attribute records
   begsr Mark_BOD_Detail;

   chain xx0key bododtr;
   if %found(bododt);
     if hasattr <> 'Y';
       hasattr = 'Y';
       update bododtr;
     else;
       unlock bododt;
     endif;
   endif;

   endsr;
   //**********************************************************
   // Check if Action code is correct
   begsr $Chk_Action;

   if %found(soadinf);
     if @@action <> 'Change'
       and @@action <> 'Delete';
       @@action = 'Change';
     endif;
   else;
     if @@action <> 'Add'
       and @@action <> 'Delete';
       @@action = 'Add';
     endif;
   endif;
   if @@action = 'Change';
     @@action = 'Replace';
   endif;

   endsr;
```

Outbound ION DB Connector Program

```
/COPY QCPYSRC,H_SPEC
```

```
   // a+ open order header  pgm prefix: oh
 Forhed      if  e           k disk


 D CFG100P         pr                      extpgm('CFG100P')
 D @curenv                        2
 D @env                          2
 D @J                           1
 D QCMDEXC         pr                      extpgm('QCMDEXC')
 D Cmd_                       1024    options(*VARSIZE)
 D                                         CONST
 D CmdLen                       15P 5 CONST

 D GETMYORDER      pr
 D base_                         2    varying options(*varsize)
 D env_                          2    varying options(*varsize)
 D XMLIN_                     1024    options(*varsize)
 D GETMYORDER      pi
 D base                          2    varying options(*varsize)
 D env                           2    varying options(*varsize)
 D XMLIN                      1024    options(*varsize)
  //-----------------------------------------------*
  // Stand Alone Fields  - TOP
  //-------------------------------------------------------------
 d*@xml           s            a    LEN(1600000) varying
 d @xml           s         1024
 D @curenv        s            2
 D @J             s            1
 D cmd            s         1024
 D option         S            1  0  inz(2)
 D X              S            2  0
 D @X             S            5  0
 D @base          s            2
 D @env           s            2
 D
  * 10000 record array
 D result          DS                  Qualified
 D                                     OCCURS(10000)
 d xml                      1024    varying

  /free
   // read IFS file
   exec sql Set Option CLOSQLCSR=*ENDACTGRP;

   // build library list
   @base = base;
   @env = env;
   monitor;
     cmd = 'addlible QTEMP';
     QCMDEXC ( cmd : %len(cmd));
   on-error *all;
   endmon;
   monitor;
```

```
    cmd = 'addlible QGPL';
    QCMDEXC ( cmd : %len(cmd));
on-error *all;
endmon;
monitor;
    cmd = 'addlible APLUS';
    QCMDEXC ( cmd : %len(cmd));
on-error *all;
endmon;

monitor;
    cmd = 'addlible APLUS' + %trim(@base);
    QCMDEXC ( cmd : %len(cmd));
on-error *all;
endmon;

@J = 'J';
CFG100P ( @env : @env : @J);

select;
//################################
// option 1 use straight SQL
//################################
when option = 1;

exec sql close c1;
exec sql declare C1 scroll cursor for
                    SELECT xmlserialize( content
                        xmlelement(name "DataArea",
                            xmlagg(
                                xmlelement(Name "MyOrder",
                                xmlelement(Name "Company", OHCONO),
                                xmlelement(Name "Order", OHORNO),
                                xmlelement(Name "OrderGn", OHORGN),
                                xmlelement(Name "Customer", OHCSNO),
                                xmlelement(Name "OrderAmt", OHORVL),
                                xmlelement(Name "InvAmt", OHINAM)
                                ) )
                            )
                        AS CLOB(1M) ) as "result" FROM ORHED;

exec sql open c1;
exec sql SET RESULT SETS with RETURN TO CLIENT cursor C1;

//################################
// option 2 use an array
// - only returns 1 record
//################################
when option = 2;

// read order header
setll *loval orhed;
read orhed;
dow not %eof;
```

```
        @x = 1 + @x;
        @xml =
              '<DataArea>' +
                    '<MyOrder>' +
                    '<Company>' + %char(ohcono) + '</Company>' +
                    '<Order>' + %trim(ohorno) + '</Order>'   +
                    '<OrderGn>' + %editc(ohorgn : 'X') + '</OrderGn>'  +
                    '<Customer>' + %char(ohcsno) + '</Customer>'  +
                    '<OrderAmt>' + %char(ohorvl) + '</OrderAmt>'  +
                    '<InvAmt>' + %char(OHINAM) + '</InvAmt>' +
                    '</MyOrder>'  +
              '</DataArea>';
          result.xml = @xml;
          leave;
     read orhed;
   enddo;
   result.xml = @xml;

   EXEC SQL SET RESULT SETS with RETURN TO CLIENT
        ARRAY :result FOR 1 ROWS;
   //#####################################################
   // option 3 - Use a temp table
   //           - create a temp table with 1 field for XML
   //           - Use RPG logic to insert additinoal business logic
   //           - and map data
   //           - Use a XML Select to create XML from temp table for ION
   //#####################################################
   when option = 3;
   // Use a temp table
   EXEC SQL drop TABLE qtemp/TMPXML;
   EXEC SQL CREATE TABLE qtemp/TMPXML (XML VARCHAR (1024 ) NOT NULL
           WITH DEFAULT);


   // read order header
   @x = 0;
   setll *loval orhed;
   read orhed;
   dow not %eof;

     @x = 1 + @x;
     @xml =
           '<DataArea>' +
                   '<MyOrder>' +
                   '<Company>' + %char(ohcono) + '</Company>' +
                   '<Order>' + %trim(ohorno) + '</Order>'   +
                   '<OrderGn>' + %editc(ohorgn : 'X') + '</OrderGn>'  +
                   '<Customer>' + %char(ohcsno) + '</Customer>'  +
                   '<OrderAmt>' + %char(ohorvl) + '</OrderAmt>'  +
                   '<InvAmt>' + %char(OHINAM) + '</InvAmt>' +
                   '</MyOrder>'  +
           '</DataArea>';
        EXEC SQL insert into qtemp/TMPXML values :@xml
                  with none;
```

```
      if @x = 10;
        leave;
      endif;

   read orhed;
 enddo;
 exec sql close c2;
 exec sql declare C2 cursor for Select * from qtemp/TMPXML;
 exec sql open c2;
 exec sql  SET RESULT SETS with RETURN TO CLIENT cursor C2;

 //###################################################
 endsl;

 *inlr = *on;
```

Inbound ION DB Connector Program

```
H BNDDIR('GW720')
 /COPY QCPYSRC,H_SPEC
 //
D lda             uds
D @rejct                         1
D wsid                    5       6
D user                   13      20
D ohiseq                102     106  0
D llcono                108     109  0
D llorno                         5
D llorgn                         2  0

D               sds
D @@dsid                244     253
D @suser                254     263
D jobnbr                264     269
D @sdate                276     281  0

d xml            s               a   LEN(1600000) varying
D QCMDEXC         pr                  extpgm('QCMDEXC')
D Cmd_                   1024    options(*VARSIZE)
D                                     CONST
D CmdLen                     15P 5 CONST

D CFG100P         pr                  extpgm('CFG100P')
D @curenv                 2
D @env                    2
D @J                      1

D CRTCREDIT       pr
D base                    2     varying  options(*varsize)
```

```
     D envid                          2    varying  options(*varsize)
     D XMLquestXML                         like(xml) options(*varsize)

     D oe130          pr                   extpgm('OE130')

     D xmlHandler     Pr          10i 0
     D   ignore_                    1a
     D   event_                    10i 0 value
     D   string_                    *    value
     D   stringLen_                20i 0 value
     D   exceptionId_              10i 0 value
     D/COPY QCPYSRC,P_GETNUM
      /COPY QCPYSRC,P_TPCKOBJP

     D mbrnam         s           10
     D @curenv        s            2
     D @J             s            1
     D Cmd            s         1024
     D xmllog         s        32000    varying
     D @base          s            2
     D @env           s            2
     D ignoreme       s            1a
     D xlibp          s           10
     D ReturnID       s           10
     D xpflag         s            1
     D xpgmp          s           10
     D xtypp          s            7
     D @@aslp         s           15  5
     D orhofDS        e ds             qualified ExtName(orhof)
     D ordofDS        e ds             qualified ExtName(ordof)
     D @@orsq         s            5  0
     D CRTCREDIT      pi
     D base                        2    varying options(*varsize)
     D env                         2    varying options(*varsize)
     D XMLIN                            like(xml) options(*varsize)
     D xmlDS          DS               qualified
     d xml                      5000    varying
     D result         DS               likeds(xmlds)
     D                                 OCCURS(1)


     C     *dtaara      define    *lda          lda
      /free
       in lda;

       exec sql Set Option COMMIT=*NONE;
       exec sql Set Option CLOSQLCSR=*ENDMOD;

       mbrnam = 'ION' + jobnbr;

       // build library list
       @base = base;
       @env = env;
       monitor;
```

```
    cmd = 'addlible QTEMP';
    QCMDEXC ( cmd : %len(cmd));
on-error *all;
endmon;
monitor;
    cmd = 'addlible QGPL';
    QCMDEXC ( cmd : %len(cmd));
on-error *all;
endmon;
monitor;
    cmd = 'addlible APLUS';
    QCMDEXC ( cmd : %len(cmd));
on-error *all;
endmon;

monitor;
    cmd = 'addlible APLUS' + %trim(@base);
    QCMDEXC ( cmd : %len(cmd));
on-error *all;
endmon;

@J = 'J';
CFG100P ( @env : @env : @J);

// Log Values
xmllog = XMLIN;
xlibp  = 'ION3P';
xpgmp = 'ION3PLOG';
xtypp = '*FILE';
xpflag = '1';
TPCKOBJP ( xpgmp : xlibp : xtypp :
xpflag );
if xpflag <> '1';
  EXEC SQL
  CREATE TABLE ION3P/ION3PLOG
      (C_ID BIGINT GENERATED ALWAYS AS IDENTITY,
       C_TIMESTAMP TIMESTAMP NOT NULL WITH DEFAULT,
       C_USER VARCHAR (100) NOT NULL WITH  DEFAULT,
       C_JOBNAME VARCHAR ( 10) NOT NULL WITH DEFAULT,
       C_JOBNBR VARCHAR ( 6) NOT NULL WITH DEFAULT,
       C_XML VARCHAR(32000) NOT NULL WITH DEFAULT);
endif;
exec SQL INSERT Into ION3P/ION3PLOG
           ( C_TIMESTAMP, C_USER, C_JOBNAME, C_JOBNBR, C_XML)
         Values(Current Timestamp,
             :@suser,  :@@dsid, :jobnbr,
             :xmllog)
         with none;

// add physical file members
monitor;
    cmd = 'ADDPFM FILE(ORDOF) MBR(' + MBRNAM + ')';
 QCMDEXC ( cmd : %len(cmd));
on-error *all;
```

```
     endmon;

     monitor;
       cmd = 'ADDPFM FILE(ORHOF) MBR(' + MBRNAM  + ')';
       QCMDEXC ( cmd : %len(cmd));
     on-error *all;
     endmon;

     monitor;
       cmd = 'ADDPFM FILE(ORLOF) MBR(' + MBRNAM  + ')';
       QCMDEXC ( cmd : %len(cmd));
     on-error *all;
     endmon;

     monitor;
       cmd = 'ADDPFM FILE(ICOFHD) MBR(' +  MBRNAM + ')';
       QCMDEXC ( cmd : %len(cmd));
     on-error *all;
     endmon;

     monitor;
       cmd = 'ADDPFM FILE(ICOFDT) MBR(' + MBRNAM  + ')';
       QCMDEXC ( cmd : %len(cmd));
     on-error *all;
     endmon;

     // clear members
     monitor;
       cmd = 'CLRPFM FILE(ORDOF) MBR(' + MBRNAM + ')';
       QCMDEXC ( cmd : %len(cmd));
     on-error *all;
     endmon;

     monitor;
       cmd = 'CLRPFM FILE(ORHOF) MBR(' + MBRNAM + ')';
       QCMDEXC ( cmd : %len(cmd));
     on-error *all;
     endmon;

     monitor;
       cmd = 'CLRPFM FILE(ORLOF) MBR(' + MBRNAM + ')';
       QCMDEXC ( cmd : %len(cmd));
     on-error *all;
     endmon;

     monitor;
       cmd = 'CLRPFM FILE(ICOFHD) MBR(' + MBRNAM + ')';
       QCMDEXC ( cmd : %len(cmd));
     on-error *all;
     endmon;

     monitor;
       cmd = 'CLRPFM FILE(ICOFDT) MBR(' + MBRNAM + ')';
       QCMDEXC ( cmd : %len(cmd));
```

```
on-error *all;
endmon;

// Overrides
monitor;
  cmd = 'DLTOVR FILE(*ALL)';
  QCMDEXC ( cmd : %len(cmd));
on-error *all;
endmon;

mbrnam = 'ION' + jobnbr;
cmd = 'OVRDBF     FILE(ORDOF) TOFILE(ORDOF) MBR(' + MBRNAM +
                 ') OVRSCOPE(*CALLLVL)';
QCMDEXC ( cmd : %len(cmd));

cmd = 'OVRDBF     FILE(ORHOF) TOFILE(ORHOF) MBR(' + MBRNAM +
                 ') OVRSCOPE(*CALLLVL)';
QCMDEXC ( cmd : %len(cmd));

cmd = 'OVRDBF     FILE(ORLOF) TOFILE(ORLOF) MBR(' + MBRNAM +
                 ') OVRSCOPE(*CALLLVL)';
QCMDEXC ( cmd : %len(cmd));

cmd = 'OVRDBF     FILE(ICOFHD) TOFILE(ICOFHD) MBR(' + MBRNAM +
                 ') OVRSCOPE(*CALLLVL)';
QCMDEXC ( cmd : %len(cmd));

cmd = 'OVRDBF     FILE(ICOFDT) TOFILE(ICOFDT) MBR(' + MBRNAM +
                 ') OVRSCOPE(*CALLLVL)';
QCMDEXC ( cmd : %len(cmd));

// parse inbound document
xml-sax %handler( xmlHandler : ignoreme )
        %XML( XMLIN : 'doc=string');

// call offline OE
out lda;
oe130();
in  lda;

// send back response
xmlds.xml =

             '<DataArea>' +
             '<CreditMemo>' +
             '<CreditMemoRequest>' +
             '<Header>' +
             '<ReturnID>' + LLOrno  + '</ReturnID>'   +
             '</Header>' +
             '</CreditMemoRequest>' +
             '</CreditMemo>' +
             '</DataArea>';
```

```
     result = xmlds;

  EXEC SQL SET RESULT SETS with RETURN TO CLIENT
       ARRAY :result FOR 1 ROWS;


 *inlr = *on;

/end-free
// -------------------------------------------------*
// Function: xmlHandler Export = *yes
P xmlHandler      b

D xmlHandler      PI            10i 0
D   ignore                       1a
D   event                       10i 0 value
D   string                       *    value
D   stringLen                   20i 0 value
D   exceptionId                 10i 0 value
D value          s          65535a   based(string)
D ucs2val        s          16383c   based(string)

D MAX_DEPTH      c                  32
D depth          s          10i 0 inz(0) static
D xPath          s          65535a   varying inz('')
D                                    dim(MAX_DEPTH)
D                                    static
D xPathVal       s          65535a   varying inz('')
D                                    dim(MAX_DEPTH)
D                                    static
D up             c                  'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
D lo             c                  'abcdefghijklmnopqrstuvwxyz'

 /free

    select;
    when event = *XML_EXCEPTION;
      return 0;

    when event = *XML_START_DOCUMENT;
      depth = 0;
      xPath(*) = '';
      xPathVal(*) = '';
      return 0;
    when depth >= MAX_DEPTH
      and ( event = *XML_START_ELEMENT
      or event = *XML_ATTR_NAME );
      return -1;
    when event = *XML_START_ELEMENT;

      // build XPath
      depth += 1;
      if depth = 1;
        xPath(depth) = '/' + %subst(value:1:stringLen);
```

```
   else;
     xPath(depth) = xPath(depth-1) + '/'
                      + %subst(value:1:stringLen);
   endif;
   xPathVal(depth)  = '';

when event = *XML_ATTR_NAME;
  depth += 1;
  xPath(depth) = xPath(depth-1) + '/@'
                       + %subst(%xlate(lo:up:value):1:stringLen);
  xPathVal(depth) = '';

when event = *XML_END_ELEMENT
  or event = *XML_END_ATTR;
   select;

 // Get Vaules
 when xPath(depth) = '/DataArea/CreditMemo' +
                      '/CreditMemoRequest/Header/ReturnID';
   ReturnID = %trim(xpathval(depth));

 when xPath(depth) = '/DataArea/CreditMemo' +
                      '/CreditMemoRequest/Header' +
                      '/TrilogieCustomerNumber';

   Orhofds.xhCSNO = getnum(%trim(xpathval(depth)));


 when xPath(depth) = '/DataArea/CreditMemo' +
                      '/CreditMemoRequest/LineItem' +
                    '/TrilogieProduct';

   Ordofds.xditno = %trim(xpathval(depth);

 when xPath(depth) = '/DataArea/CreditMemo' +
                      '/CreditMemoRequest/LineItem' +
                    '/Quantity';
   Ordofds.xdqtor = getnum(%trim(xpathval(depth))) * -1;

 // TODO get original order number

 // end of order
 when xPath(depth) = '/DataArea/CreditMemo' +
                      '/CreditMemoRequest/Header';
   @@orsq = 1+ @@orsq;

   // hard code values to ORHOF
   Orhofds.xhCONO = 01;
   orhofDS.xhrtst = 'Y';
   orhofDS.xhortp = 'R';

   Exec SQL
     INSERT Into orhof (xhcono, xhcsno, XHORTP, XHRTST)
           Values(:orhofDS.xhcono, :orhofds.xhcsno,
```

```
                                  :orhofDS.xhortp, :orhofDS.xhrtst);
          @@orsq = 0;

        // end of order detail
        when xPath(depth) =  '/DataArea/CreditMemo' +
                                '/CreditMemoRequest/LineItem';
          // hard code values to ORDOF
          ordofDS.XDRTRC = 'MS';
          ordofDS.XDRTST = 'Y';
          Exec SQL
            INSERT Into ordof (XDITNO, XDASLP, XDQTOR)
                  Values(:ordofDS.xditno,:ordofds.xdaslp,:ordofds.xdqtor);

        endsl;

        depth -= 1;

      when event = *XML_CHARS
        or event = *XML_PREDEF_REF
        or event = *XML_ATTR_CHARS
        or event = *XML_ATTR_PREDEF_REF;
        xPathVal(depth) += %subst(value:1:stringLen);

          when event = *XML_UCS2_REF
                    or event = *XML_ATTR_UCS2_REF;
                    xPathVal(depth) += %char(%subst( ucs2val : 1
                                              : %div(stringLen:2) ));
      endsl;


      return 0;
   /end-free
 P xmlHandler      E
   /free
```