TIBCO iProcess® Conductor

Concepts

Software Release 11.2.1
August 2012

two-second advantage™

TIBCO®
The Power of Now®

Important Information

Contents

Preface

The objective of this guide is to describe the concepts behind implementing the fulfillment of orders using TIBCO iProcess Conductor. You must have a thorough understanding of these concepts before you create execution plans and submit orders using the TIBCO iProcess Conductor user interface.

Topics

Changes from the Previous Release of This Guide

This section itemizes the major changes from the previous release of this guide.

There are no changes from the previous release of this guide.

Related Documentation

This section lists documentation resources you may find useful.

TIBCO iProcess Conductor Documentation

The following documents form the TIBCO iProcess Conductor documentation set:

- *TIBCO iProcess Conductor Concepts* Read this manual to gain an understanding of the product that you can apply to the various tasks you may undertake.
- *TIBCO iProcess Conductor Installation* Read this manual for instructions on site preparation and installation.
- *TIBCO iProcess Conductor Implementation* Read this guide for instructions on how to design, plan, and implement the fulfillment of orders.
- *TIBCO iProcess Conductor User's Guide* Read this guide for instructions on using the TIBCO iProcess Conductor user interface to orchestrate execution plans, create process components, amend orders, and so on.
- *TIBCO iProcess Conductor Administrator's Guide* Read this guide for instructions on common administrative tasks, such as archiving completed execution plans, managing users, and deploying the TIBCO iProcess Decisions rule sets.
- *TIBCO iProcess Conductor Release Notes* Read the release notes for a list of new and changed features. This document also contains lists of known issues and closed issues for this release.
- *TIBCO iProcess Conductor Utility Framework User's Guide* Read this guide for details of the Utility Framework and the applications used to export and import plans and process components.
- *TIBCO iProcess Conductor AOPD Message-Driven API Developer's Guide* Read this guide for details of the TIBCO iProcess Conductor's execution plan interfaces, and the facilities for automatic execution plan development.

Other TIBCO Product Documentation

You may find it useful to refer to the documentation set for TIBCO iProcess® Suite, which is used with TIBCO iProcess Conductor.

Third-party Documentation

You may find it useful to read the documentation for the following third-party products:

- Oracle® Database
- Oracle® WebLogic Server
- JBoss® Application Server

Typographical Conventions

The following typographical conventions are used in this manual.

Table 1 General Typographical Conventions

Convention	Use
`code font`	Code font identifies commands, code examples, filenames, pathnames, and output displayed in a command window. For example: Use `MyCommand` to start the foo process.
`bold code font`	Bold code font is used in the following ways: In procedures, to indicate what a user types. For example: Type **`admin`**. In large code samples, to indicate the parts of the sample that are of particular interest. In command syntax, to indicate the default parameter for a command. For example, if no parameter is specified, `MyCommand` is enabled: `MyCommand [`**`enable`**` | disable]`
italic font	Italic font is used in the following ways: To indicate a document title. For example: See *TIBCO ActiveMatrix BusinessWorks Concepts*. To introduce new terms For example: A portal page may contain several portlets. *Portlets* are mini-applications that run in a portal. To indicate a variable in a command or code syntax that you must replace. For example: `MyCommand` *PathName*
Key combinations	Key name separated by a plus sign indicate keys pressed simultaneously. For example: Ctrl+C. Key names separated by a comma and space indicate keys pressed one after the other. For example: Esc, Ctrl+Q.
📝	The note icon indicates information that is of special interest or importance, for example, an additional action required only in certain circumstances.
💡	The tip icon indicates an idea that could be useful, for example, a way to apply the information provided in the current section to achieve a specific result.
⚠	The warning icon indicates the potential for a damaging situation, for example, data loss or corruption if certain steps are taken or not taken.

Table 2 Syntax Typographical Conventions

Convention	Use
`[ ]`	An optional item in a command or code syntax. For example: MyCommand [optional_parameter] required_parameter
|	A logical `OR` that separates multiple items of which only one may be chosen. For example, you can select only one of the following parameters: MyCommand para1 | param2 | param3
`{ }`	A logical group of items in a command. Other syntax notations may appear within each logical group. For example, the following command requires two parameters, which can be either the pair `param1` and `param2`, or the pair `param3` and `param4`. MyCommand {param1 param2} | {param3 param4} In the next example, the command requires two parameters. The first parameter can be either `param1` or `param2` and the second can be either `param3` or `param4`: MyCommand {param1 | param2} {param3 | param4} In the next example, the command can accept either two or three parameters. The first parameter must be `param1`. You can optionally include `param2` as the second parameter. And the last parameter is either `param3` or `param4`. MyCommand param1 [param2] {param3 | param4}

Connecting with TIBCO Resources

How to Join TIBCOmmunity

TIBCOmmunity is an online destination for TIBCO customers, partners, and resident experts. It is a place to share and access the collective experience of the TIBCO community. TIBCOmmunity offers forums, blogs, and access to a variety of resources. To register, go to http://www.tibcommunity.com.

How to Access TIBCO Documentation

You can access TIBCO documentation here:

http://docs.tibco.com

How to Contact TIBCO Support

For comments or problems with this manual or the software it addresses, contact TIBCO Support as follows:

- For an overview of TIBCO Support, and information about getting started with TIBCO Support, visit this site:

  http://www.tibco.com/services/support

- If you already have a valid maintenance or support contract, visit this site:

  https://support.tibco.com

  Entry to this site requires a user name and password. If you do not have a user name, you can request one.

Chapter 1 TIBCO iProcess Conductor Overview

This chapter provides an overview of TIBCO iProcess Conductor. It describes how TIBCO iProcess Conductor addresses the problems inherent in fulfilling complex orders. Finally, it explains the TIBCO iProcess Conductor architecture.

Topics

Introduction to TIBCO iProcess Conductor

TIBCO iProcess Conductor supports Business Process Management (BPM) by enabling you to build and deploy complex business processes rapidly and dynamically. It has been designed to provide extensive capabilities to manage requests to achieve a set of goals - which are termed *orders*:

- These orders are managed by TIBCO iProcess Conductor Order Plug-in.
- The business processes responsible for fulfilling these orders are called *execution plans*; they are managed by TIBCO iProcess Conductor Orchestration Plug-in.

These plug-ins are delivered and installed as follows:

- TIBCO iProcess Conductor Order Server Plug-in and TIBCO iProcess Conductor Orchestration Server Plug-in are installed as part of TIBCO iProcess Technology Plug-ins.
- TIBCO iProcess Conductor Order Client Plug-in and TIBCO iProcess Conductor Orchestration Client Plug-in are installed as part of TIBCO iProcess Workspace Plug-ins.

TIBCO iProcess Conductor also uses the following TIBCO products:

- TIBCO iProcess® Decisions Plug-in
- TIBCO iProcess® Engine

TIBCO iProcess Conductor coordinates business processes that are executed in TIBCO iProcess Engine. It packages these iProcess Engine processes as independent process components, combining them and establishing dependencies between them to create a complete business process. This encourages goal-oriented approaches to BPM, because each process component can be viewed as achieving a sub-goal that contributes to reaching the overall goal of the whole process. iProcess Conductor enables you to build and deploy your processes - including resource requirements, timing dependencies, and jeopardy management conditions - immediately before runtime, thus maximizing flexibility and saving time. Process components can be reused, coordinated, individually monitored, and dynamically modified during runtime.

There are other TIBCO software products that must be installed to use TIBCO iProcess Conductor. For more information, see the installation guide for iProcess Conductor.

Execution Plans

TIBCO iProcess Conductor handles orders by creating execution plans from reusable iProcess Engine process components, sub-goals and deadlines. Execution plans for frequently-repeated business processes can be selected from existing execution plan templates; other execution plans can be created when needed, using existing process components. You can also import and export execution plans and process components, using utilities provided. These utilities are described in *TIBCO iProcess Conductor Utility Framework User's Guide*.

The execution plan is displayed by TIBCO iProcess Conductor in a style like a project plan, and managing an execution plan is very similar to managing a project.

Dependencies

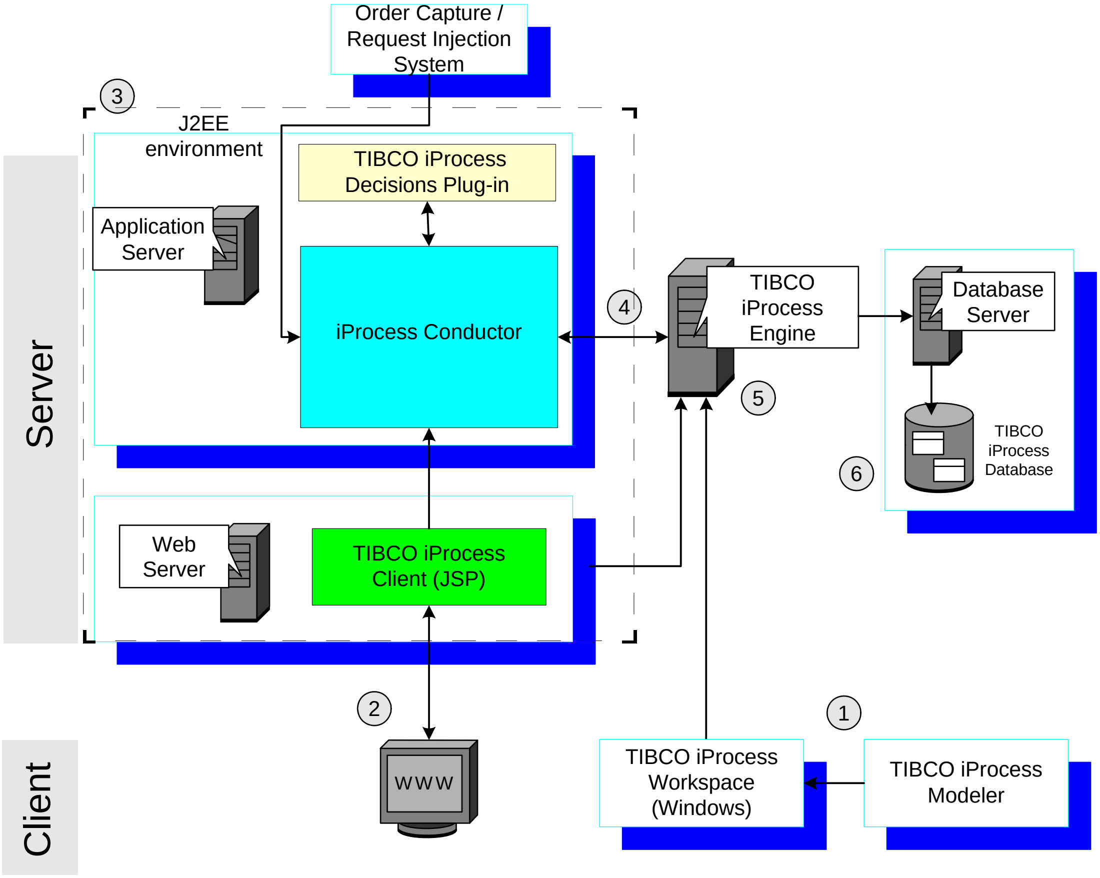
TIBCO iProcess Conductor maps dependencies between process components. This is the key to the easy creation of complex processes and execution plans. Dependencies can be specified either when creating a template, or just before the plan is to be executed. Temporal dependencies can be set, including the "must start on" and "start no earlier than" dependencies.

Jeopardy Management

When you create an execution plan you can establish various *thresholds*, indicating whether a process component is in jeopardy of not starting, or of not being completed, on time. At run time, if it is predicted that a threshold will be passed, or if it actually has been passed, a *jeopardy condition* occurs. Critical path analysis will be performed based on the dependencies that you have set, and action can be taken to avoid or lessen the current or predicted jeopardy condition.

TIBCO iProcess Conductor Architecture

The following diagram illustrates the technical architecture of TIBCO iProcess Conductor.



1. TIBCO iProcess Modeler and TIBCO iProcess Workspace (Windows) enable the process definer to create the iProcess procedures necessary for order fulfillment (see iProcess Procedures on page 57).

2. The Client Browser enables the client to connect to TIBCO iProcess Client (JSP) on the Web server and access the iProcess Conductor User Interface. Execution plans can be created using the TIBCO iProcess Conductor User Interface on the Application Server (for more information see *TIBCO iProcess Conductor User's Guide*).

3. Both the Web Server and the Application Server are implemented using either BEA WebLogic or the JBoss J2EE Application Server.
4. iProcess Conductor uses TIBCO iProcess Decisions Plug-in to determine the appropriate fulfillment procedure for use in TIBCO iProcess Engine, and, if possible, the execution plan template to use.

For configuring iProcess Conductor not to use iProcess Decisions, see *TIBCO iProcess Conductor Administrator's Guide*.

5. TIBCO iProcess Engine executes the fulfillment process, typically carrying out any initial order validation and execution plan development/selection before notifying TIBCO iProcess Conductor which instantiates and executes the designated execution plan.
6. The data associated with the order and the execution plan are stored in the TIBCO iProcess Conductor database.

Summary

The approach taken by TIBCO iProcess Conductor addresses the key requirements of modern, complex, business process management, namely:

- The flexibility to produce the best fit for your business
- The easy reuse of process components.

The next chapter looks at order fulfillment in greater details.

Chapter 2 Understanding Order Fulfillment

This chapter describes how TIBCO iProcess Conductor works to fulfill order requests.

TIBCO iProcess Conductor maps requests to achieve a set of goals on to the business processes necessary to achieve those goals. These goal fulfillment requests are managed by the order management part of TIBCO iProcess Conductor as *orders*. The business processes responsible for fulfilling these orders are called *execution plans*; they are managed by the orchestration component of TIBCO iProcess Conductor.

Topics

Overview of the Order Fulfillment Process

At the heart of TIBCO iProcess Conductor are *process components*. These represent individual activities or collections of activities that make up the tasks required to fulfill an order. The process components form a reusable library that you can call upon to fulfill different types of customer orders. The actual work represented by these process components is carried out by iProcess procedures.

When an order is received, an execution plan is instantiated by TIBCO iProcess Conductor enabling the required process components to be assembled and the associated iProcess procedures to be executed according to the nature of the order.

This section describes three main steps that an order goes through:

- Order submission
- Fulfillment procedure selection/execution
- Template selection/execution plan instantiation

This section provides a broad overview of the fulfillment process. There is, however, much more happening in this process, including status changes, sub procedure calls, orchestration steps, and so on. These topics are covered elsewhere in this manual and in other TIBCO iProcess Conductor documentation.

Order Submission

A customer order is received in TIBCO iProcess Conductor from an external order capture or request injection system, for example, a CRM system or a Business-to-Business (B2B) gateway. The order must be in XML, must conform to the order schema, and is received via a Java Message Service (JMS) message. For more information about the order schema, see *TIBCO iProcess Conductor Implementation*.

You can also submit an order using the iProcess Conductor Order Management user interface. However, this is not the primary method for order submission and it is intended for diagnostic or support purposes only.

What makes up an order?

An order must conform to the order schema discussed in *TIBCO iProcess Conductor Implementation*. It consists of an order header and item lines (see Overview of an Order on page 76).

What happens to an order when it is submitted?

TIBCO iProcess Conductor creates an order request and stores information about the order in the database. It then sends the order request to TIBCO iProcess Decisions Plug-in to identify what fulfillment process should be used (see the next section).

What happens to an order if it is rejected?

When an order does not conform to the order schema, it does not enter TIBCO iProcess Conductor and is sent to a JMS queue, ready to be returned to the order capture system that submitted it. An exception message is entered in the `FFTrace.log` file (in the TIBCO iProcess Conductor deployment directory), but not in the TIBCO iProcess Conductor audit log.

How can I change an order once it has been submitted?

You can change an order by submitting an amendment. An amendment takes exactly the same form as any other order, except that it has the same Order Reference as the existing order that you want to amend. For more information, see Amending Orders on page 79.

Fulfillment Process Selection/Execution

After an order has been submitted, TIBCO iProcess Conductor must decide how best to fulfill the order. The actual fulfillment may be carried out either:

- using iProcess procedures, or
- by sending a JMS message to a queue for use by another application

How is the fulfillment process selected?

The fulfillment process may be selected for each execution plan by evaluating rules. Alternatively, if you do not use rules for this purpose, it can be determined by the value assigned to a configuration parameter in the `AppConfig.xml` configuration file. (See *TIBCO iProcess Conductor Administrator's Guide* for information about editing parameters in the `AppConfig` file.)

By default, iProcess Conductor uses rules that have been created for the TIBCO iProcess Decisions Server. Alternatively, you can configure iProcess Conductor to work with a set of statically defined rules, in which case iProcess Decisions is not required.

For example, a simple rule could be created that examines the Order ID and selects a fulfillment procedure on that basis. The way that you categorize orders determines the fulfillment procedure required and therefore the rules that you set up. For example, wholesale and retail orders may have different validation requirements, and your rules would be designed to recognize each type of order and select the appropriate fulfillment procedures.

You can use either rules or the `AppConfig` file to specify the conditions under which JMS messaging should be used for the fulfillment process. See "Setting the Fulfillment Type" in *TIBCO iProcess Conductor Administrator's Guide*.

What does the fulfillment procedure do?

The fulfillment procedure is specific to your business processes, but it usually has three components:

- **Order Feasibility** - ensures that the order can progress. For example, a credit check on the customer or a resource inventory.
- **Execution Plan Development** - identifies an execution plan template and from this an execution plan (see the following section).
- **Execution Plan Processing** - executes the plan, including the iProcess processes required to complete the execution plan (see the following section).

For more information, see *TIBCO iProcess Conductor Implementation*.

Template Selection/Execution Plan Instantiation

An execution plan template acts as a blueprint for execution of the customer order. Execution plan templates follow standard patterns, for example, for ordering a Digital Subscriber Line (DSL) in a telecommunications provider's order management system. Normally you might have several templates to reflect the different types of order that you typically receive. TIBCO iProcess Decisions Plug-in selects the template according to the rules you have set up.

TIBCO iProcess Conductor instantiates an execution plan based on the template that has been selected. The execution plan runs, assembling the required process components and executing any iProcess procedures that must be run.

Because TIBCO iProcess Conductor uses TIBCO iProcess Engine, you can integrate with other systems during the order fulfillment process. For example, you can pause an order during order feasibility to satisfy an order's inventory requirement. It is also possible to instruct network provisioning systems to configure or change subscriber parameters whilst the order is in the process of being actioned.

What happens if a template cannot be identified?

If TIBCO iProcess Decisions (or the static rules) cannot identify a template, you must progress the order manually (see *TIBCO iProcess Conductor User's Guide*).

What other iProcess procedures are run?

If iProcess is used for the fulfillment process, then as well as the overall fulfillment procedure, there is an iProcess procedure for each version of a process component associated with the tasks in an execution plan. A process component is an individual activity or a collection of activities that make up an individual task within an execution plan. For more information, see About Process Components on page 20.

TIBCO iProcess Conductor enables you to use different versions of process components that apply at different dates and times by defining an effective date range. When a process component is used, TIBCO iProcess Conductor automatically selects the version of a process component that applies at the current date and time.

How do the iProcess procedures get the data they need?

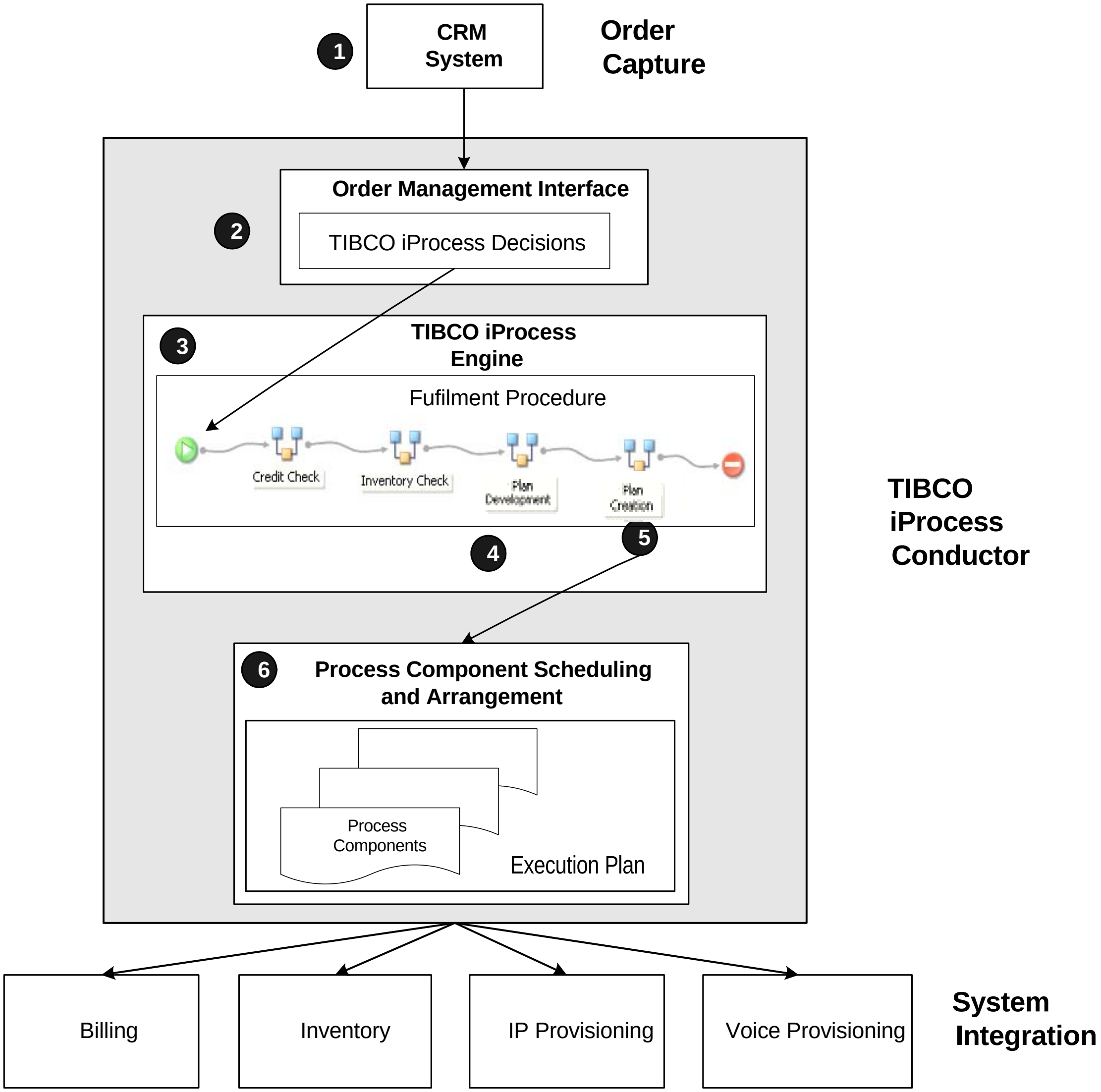
You can pass data to and from iProcess procedures at pre-defined milestones using EAI Orchestration steps. By externalizing the input and output data passed to iProcess procedures, it is easier to define and manage the behavior of procedures. For more information, see Input and Output Data on page 20.

How do I know that the Order has completed?

You can view the status of the order in the TIBCO iProcess Conductor user interface. For more information about the valid status transitions supported by iProcess Conductor Order Management, see Chapter 7, Understanding the Order Lifecycle.

Role of TIBCO iProcess Conductor

The following diagram he following diagram illustrates the relationship between an existing order capture or request injection system, TIBCO iProcess Engine, and TIBCO iProcess Conductor.



Each of the following numbered steps corresponds to the same number in the previous diagram.

1. The order is received from the order capture or request injection system. TIBCO iProcess Conductor determines if the order is an amendment or a new request (see About Amendments on page 79 for information about amending orders).
2. By using iProcess Decisions (or static rules), TIBCO iProcess Conductor identifies the fulfillment procedure required to fulfill the order, for example, the wholesale fulfillment procedure. Order fulfillment may be carried out by using iProcess procedures, or by JMS messages.
3. If iProcess procedures are used, TIBCO iProcess Engine starts a case of the fulfillment procedure and carries out the initial order validation. In this particular example, this involves performing a credit check on the customer and checking the resources.
4. Once the initial order validation has been carried out, the Execution Plan Development sub-procedure attempts to identify an execution plan template using iProcess Decisions. Failing to identify a template results in a manual planning phase wherein you manually select, create or import an execution plan or an execution plan template.
5. The Execution Plan Development sub-procedure returns an Execution Plan ID (identified by TIBCO iProcess Decisions Plug-in) back to the parent fulfillment procedure. The Execution Plan Orchestration sub-procedure passes the Execution Plan ID to iProcess Conductor for execution.
6. iProcess Conductor instantiates an execution plan from the designated template, determining the active process versions for each process component referred to in the execution plan. Each process version is then initiated as a new sub-process. iProcess Conductor manages these sub-processes and their dependencies, triggering subsequent phases of the plan until the execution of the order is complete.

Chapter 3 Managed Objects

This chapter describes the concepts relating to managed objects (orders, execution plans, and process components) in TIBCO iProcess Conductor.

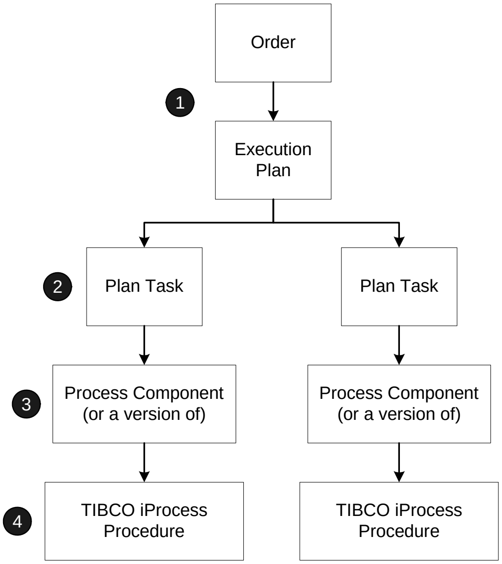
Topics

Overview

This chapter assumes that process fulfillment is being carried out by starting an iProcess case (see Fulfillment Process Selection/Execution on page 9).

TIBCO iProcess Conductor enables you to use the iProcess procedures that you have defined on your TIBCO iProcess Engine to create process components. You can then organize these process components into execution plan templates. When an order is received, TIBCO iProcess Conductor processes execution plans based on the execution plan templates (or else creates a manual execution plan).

The diagram below illustrates the relationships between the elements that are used by TIBCO iProcess Conductor and TIBCO iProcess Engine:



1. Each order has one execution plan that consists of one or more plan tasks.
2. Each plan task is associated with a process component. A process component can have more than one version.
3. Each process component version is associated with an iProcess procedure. All referenced iProcess procedures must have the same signature. See *TIBCO iProcess Conductor Implementation* for more information about process signatures.

About Orders

An order is a request to fulfill a particular set of goals, usually received in TIBCO iProcess Conductor from an external system.

Orders in TIBCO iProcess Conductor must conform to the order schema described in *TIBCO iProcess Conductor Implementation*, otherwise they are rejected. A customer order is either received in TIBCO iProcess Conductor from an external order capture system, such as a CRM system or a B2B gateway, or else can be created by iProcess Conductor Order Management. For more information about orders, see Chapter 6, Working with Orders.

An order request in TIBCO iProcess Conductor never changes. If you need to change an order, you can submit an amendment (which is sent as a separate order request). The view of an order and any amendments that have been received is brought together in a single view called the consolidated view (see Amending Orders on page 79).

About Execution Plans and Templates

An execution plan is a collection of the activities that need to be completed to fulfill a customer order. Execution plans are usually instantiated from execution plan templates that specify how the process components should be arranged to fulfill the order. If no suitable template is available to meet a particular order, an execution plan can be created manually. An execution plan or execution plan template consists of the following:

- Plan tasks and their associated process components
- Actions
- Dependencies

Plan Tasks and Their Associated Process Components

Each plan task is associated with a process component. When you create an execution plan, you add the plan tasks that are required to fulfill the customer order by selecting from the available process components configured in the system.

You can also group plan tasks into groups. This allows flexibility in creating the execution plan. For example, you can create dependencies on groups of tasks that all must be completed before the next task is started. Sets of process components can also be imported into a plan as a summary group.

Actions

Each plan task has an action associated with it. There are three possible actions you can select for each plan task:

- Provide
- Cease
- Update

A plan task manages a particular item, for example, **Provide Wiring**. Each action defines what needs to be done for a particular item. For example, if the plan task is to achieve the sub-goal of **Provide Wiring**, you would select **Provide** as the action. This action is passed to TIBCO iProcess Engine. An action serves as an annotation to make the execution plan more comprehensible.

Dependencies

When you add your plan tasks, you need to add any dependencies that exist between the plan tasks or summary groups. Dependencies are set up between milestones in a task. These milestones correspond to the EAI Orchestrator steps defined in your iProcess procedures that are associated with your process components.

- See *TIBCO iProcess Conductor Implementation* for more information about using EAI Orchestrator steps with TIBCO iProcess Conductor.
- See Understanding Dependencies on page 42 for more information about the different types of dependencies that you can create.

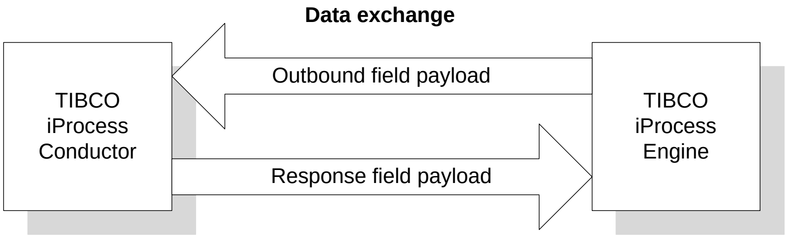
About Process Components

A process component is associated with an iProcess procedure. It consists of:

- input and output data
- milestones
- resource requirements

Input and Output Data

When you create your iProcess procedures, you specify 'field payloads' that contain the input and output data consumed or provided by each EAI Orchestration step in a process component. You specify a field payload for outbound messages (from TIBCO iProcess Engine to TIBCO iProcess Conductor), and for certain message types, you specify a field payload that specifies the expected response from TIBCO iProcess Conductor:



See *TIBCO iProcess Conductor Implementation* for more information.

Milestones and Dependencies

A milestone in a process component is a point in the underlying process which can be dependent on another event or another milestone in some way. For example, suppose you have two process components:

- Supply Chain Management (SCM)
- Customer Premises Equipment (CPE)

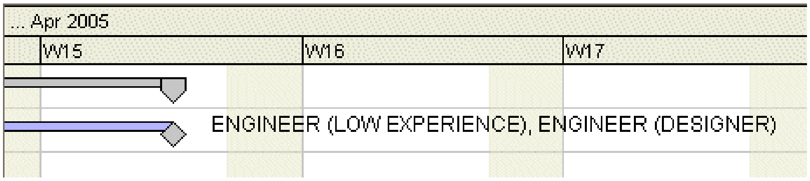
The `CPE` process component could be dependent on a milestone in the `SCM` process component to provide a delivery date.

The milestones in each process component come from the EAI Orchestration steps that you define in your iProcess procedure. When you create a process component in TIBCO iProcess Conductor, TIBCO iProcess Conductor automatically identifies the EAI Orchestration steps in the iProcess procedure that you have associated with the process component.

TIBCO iProcess Conductor validates that each subsequent iProcess procedure that you associate with a process component has the same EAI Orchestration steps and input and output data as the original procedure. This combination of EAI Orchestrator steps and input and output data is known as the "process signature." For more information, see *TIBCO iProcess Conductor Implementation*.

Resource Requirements

When you define a process component, you can specify the resource requirements for that process component (which resources are required and how long they are required for). These resource requirements can be displayed in the TIBCO iProcess Conductor user interface. For example, a plan task may require two engineers:



You can specify resource requirements both for the entire process component and also for the *sections* of a process component. A section is the interval between two milestones; a simple process component that has only start and end milestones will consist only of one section, but more complex components will be made up of several sections.

For more information, see *TIBCO iProcess Conductor User's Guide*.

Understanding Jeopardy Management

The jeopardy management facilities in TIBCO iProcess Conductor enable you to manage the risk associated with plan tasks that fall behind schedule, and to prevent them from jeopardizing the timely fulfillment of the order. This section describes:

- Overview of Jeopardy Management on page 22
- About Critical Paths on page 23
- Jeopardy Management for Process Components on page 27
- Jeopardy Management for Execution Plans on page 32
- About Execution Plan Thresholds on page 32
- Jeopardy Conditions on page 35
- About Jeopardy Consequential Actions on page 37

Overview of Jeopardy Management

To help manage the risk associated with plan tasks that are behind schedule, TIBCO iProcess Conductor can:

- calculate several critical paths through an execution plan. Two of these critical paths correspond to the typical and maximum durations of process components. The third type of critical path is based on the actual duration to date, once the execution plan has started processing. The critical paths are used to project the completion date and time of the execution plan. By viewing the critical paths in TIBCO iProcess Conductor, you can determine whether your execution plan is progressing normally or if it is in danger of over-running (in jeopardy). See About Critical Paths on page 23.
- monitor jeopardy conditions at each of the following levels:
  - plan task (process component)
  - execution plan

  For example, if a plan task has a Must Start On dependency, TIBCO iProcess Conductor enables you to configure a jeopardy condition on that plan task so that if it is in danger of not starting on time, an event can be generated to make you aware of this. See Jeopardy Management for Process Components on page 27 and Jeopardy Management for Execution Plans on page 32.
- enable you to configure thresholds and jeopardy states for your execution plans. When the end date and time of the plan is predicted to pass, or actually

has passed, a particular threshold, an event is generated to make you aware of this. See About Execution Plan Thresholds on page 32.

- perform consequential actions at each of the following levels:
  - plan task (process component)
  - milestone
  - execution plan

  At each of these levels, you can set up TIBCO iProcess Conductor so that, if a jeopardy condition occurs, it will perform one or more of the following consequential actions:

  - suspend the plan task or execution plan.
  - generate an iProcess work item to be sent, for example, to an Administrator's work queue to notify them that the plan is in jeopardy.
  - generate a JMS message that can be sent to a topic or queue that warns when a task or a plan is jeopardy.
  - if preferred, take no action.

  See About Jeopardy Consequential Actions on page 37.

About Critical Paths

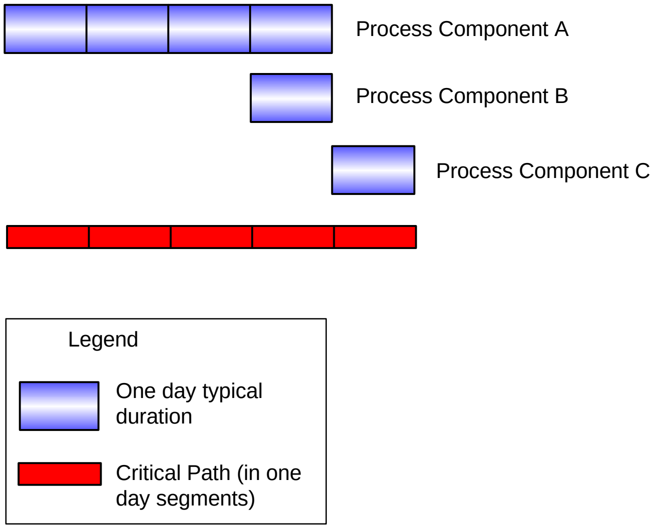
There are three types of critical path used in TIBCO iProcess Conductor:

- Typical duration - A critical path that corresponds to the typical duration of the plan.
- Maximum allowed duration - A critical path that corresponds to the maximum allowed duration of the plan.
- Execution - Once the plan starts, you can also view the critical path represented by the actual duration of completed tasks combined with the typical duration of the tasks yet to complete.

You can choose to view one of the critical paths through the execution plan, either the typical duration or the maximum allowed duration path, highlighted in red in the iProcess Conductor user interface. For more information about viewing critical paths and summary groups, see *TIBCO iProcess Conductor User's Guide*.

Critical Path Example - Typical Duration

The typical durations that you enter for process components are used to calculate a critical path that represents the "best case scenario" for order fulfillment. For example:

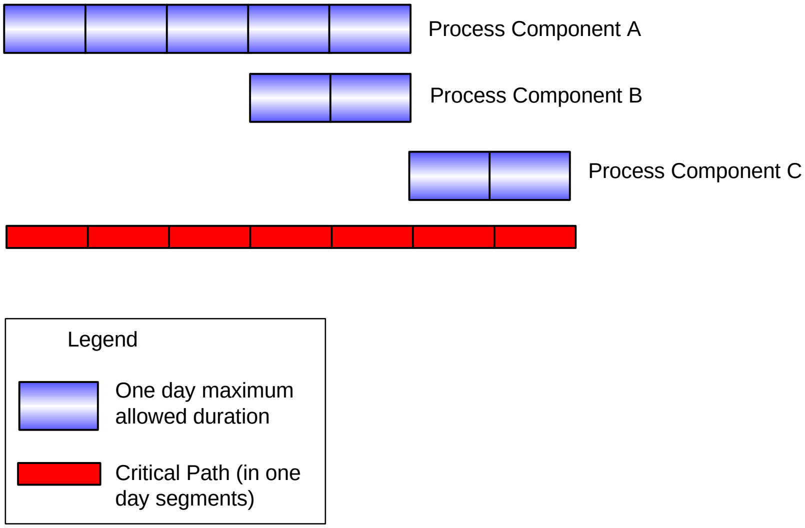


- We would like Process Component A to complete in four days.
- Process Component B takes one day, but it can overlap with Process Component A.
- Process Component C takes one day but cannot start until process components A and B are complete.

Therefore the critical path based upon the typical duration is five days. This can be calculated before the plan starts and represents the amount of time in which you would like the execution plan to be fulfilled.

Critical Path Example - Maximum Duration

The following diagram shows how the critical path corresponding to the maximum allowed duration is calculated:

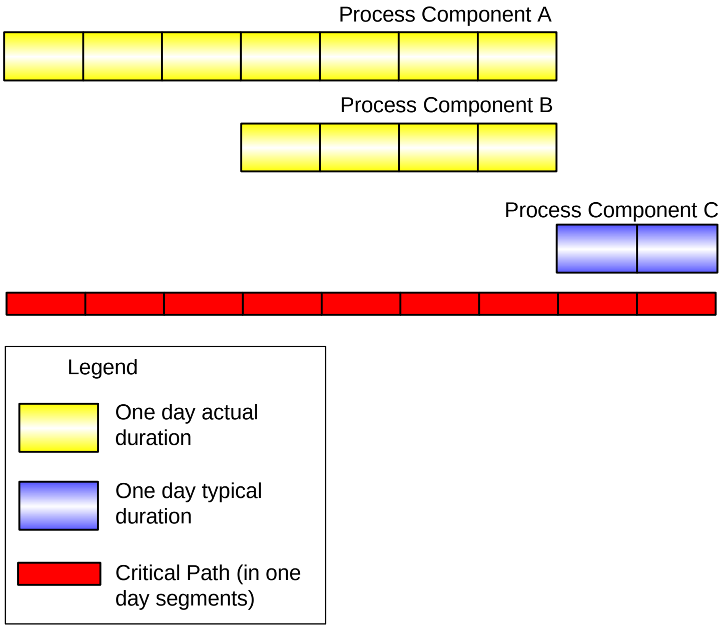

Process Component A
Process Component B
Process Component C
Legend
One day maximum allowed duration
Critical Path (in one day segments)

- Process Component A must complete in five days.
- Process Component B takes two days, but it can overlap with Process Component A.
- Process Component C takes two days but cannot start until process components A and B are complete.

Therefore the maximum duration critical path is seven days. This can be calculated before the plan starts, based on the maximum durations that you entered, and represents the maximum amount of time you are prepared to accept for the execution plan to be fulfilled.

Critical Path Example - Viewing Plan Progress

When the execution plan starts and progresses, the critical path can be calculated from the actual duration of completed tasks combined with the typical duration of the tasks yet to complete. This gives you an idea of how the plan is actually progressing. For example:



- Process Component A has actually taken seven days rather than the expected five.
- Process Component B has also overrun.

TIBCO iProcess Conductor updates the critical path to nine days to reflect this. Recall that we had hoped this order would be fulfilled in five days and were prepared to tolerate it taking up to seven days. The plan currently is projected to complete two days past the maximum, so the plan is considered "in jeopardy."

This example shows a relatively simple execution plan. In fact complex execution plans have multiple, parallel critical paths through a plan. The critical path is monitored by TIBCO iProcess Conductor, and updated to provide you with a projected date/time for fulfillment of the order.

TIBCO iProcess Conductor automatically monitors the progress of the execution plan in order to determine whether it is in jeopardy. In addition, you can view the critical path at any point during the processing of an execution plan, to see its jeopardy status. For more information, see *TIBCO iProcess Conductor User's Guide*.

Dependencies and Critical Paths

TIBCO iProcess Conductor also takes into account dependencies when updating the critical path (such as "Must Start On" and "Start No Earlier Than" dependencies). For example, suppose Process Component C has a dependency that is configured so that it "must start on the first of February" (the original end date of Process Components A and B). Because the actual durations of Process Components A and B have exceeded this date, a jeopardy condition exists ("Must Start On Dependency Predicted Infeasible"; see Jeopardy Conditions on page 35). TIBCO iProcess Conductor takes action, such as sending a case start for an iProcess process, which could generate a work item to a queue for manual intervention (see About Jeopardy Consequential Actions on page 37), and the plan is considered to be in jeopardy. Depending on how TIBCO iProcess Conductor is configured, the execution plan continues or is suspended.

Jeopardy Management for Process Components

TIBCO iProcess Conductor enables you to configure jeopardy conditions for *sections* of a process component. A section is the interval between two milestones; a simple process component that has only start and end milestones will consist only of one section, but more complex components will be made up of several sections.

At the level of process component sections, you can configure jeopardy conditions that enable you to detect both if the task has taken longer to complete than it should have, and also to detect if a task that is under way or has not yet started is *predicted* to take longer to complete than scheduled.

You can monitor the following durations:

- typical duration - the time the activity represented by the process component is *expected* to take. You can specify this value when you create a process component, or when you define the plan task that uses the component in an execution plan.
- maximum allowed duration - the *maximum* amount of time the activity represented by the task can take before it is considered to have overrun. You can specify this value when you create a process component, or when you define the plan task that uses the component in an execution plan.

Both the typical duration and the maximum allowed duration can be overridden in the user interface, at the plan task level (see *TIBCO iProcess Conductor User's Guide*).

There are critical paths identified in execution plans, constructed using the typical and maximum durations of the plan tasks included in those plans. See About Critical Paths on page 23 for more information.

The length of the typical duration and maximum allowed duration are specified for sections of a process component, not for the process component itself. For more information, see *TIBCO iProcess Conductor User's Guide*.

If one of the process component sections being monitored has not completed before its defined typical duration, TIBCO iProcess Conductor triggers a monitor event; and it triggers another monitor event if the section has not completed before the end of its maximum allowed duration. If the task being monitored has still not completed after a defined monitoring interval, the monitor event is triggered again. You can specify that the monitor event should be triggered a maximum number of times, or that it should continue to fire indefinitely, after each monitoring interval, until the task has completed.

The monitoring interval and the number of times that the monitor event should be used are defined in the `AppConfig.xml` file. TIBCO iProcess Conductor is delivered with the monitoring interval set to a default value of 60 seconds. You can change this value by editing the file. See *TIBCO iProcess Conductor Administrator's Guide* for information about editing parameters in the `AppConfig` file.

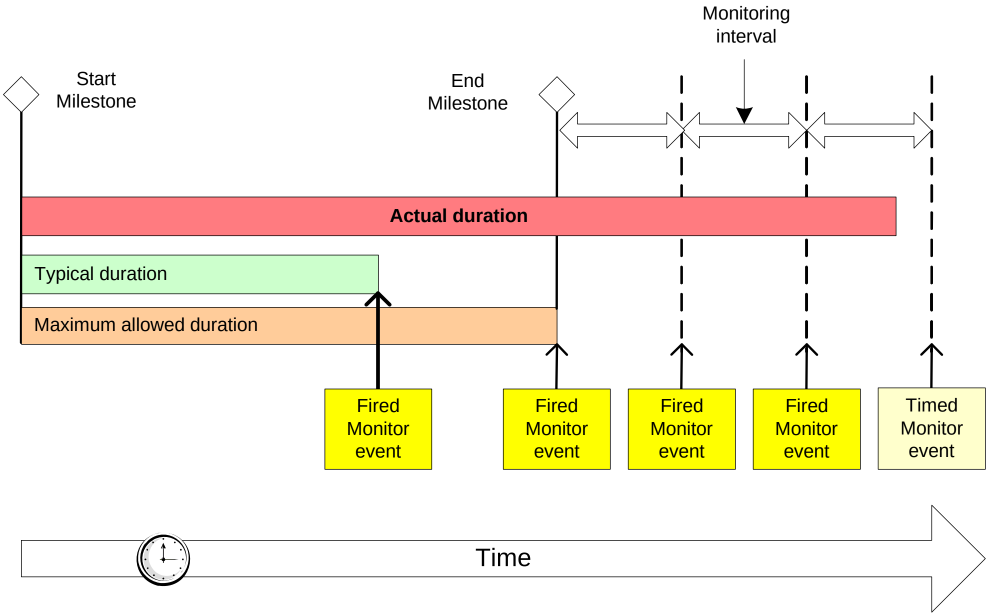
Whatever value you select for the monitoring interval applies globally. You cannot change it for individual execution plans.

Example

In the following example, a plan task has taken longer than it is predicted to. The plan task has exceeded its typical duration, its maximum duration, and two subsequent monitoring intervals. A monitor event has been fired at each stage to notify you of this:

- At the end of the typical duration
- At the end of the maximum allowed duration
- At the end of each of the next two monitoring intervals.

Another monitor event would have been fired at the end of the third monitoring interval, but since the task is completed before then, this event is disabled.



Must Start On Dependencies

The "must start on" dependency is a powerful tool that indicates that an activity must start at a specific point in time. You can apply these dependencies to milestones that denote the start of such activities; in normal circumstances, when the execution plan is running on schedule, iProcess Conductor uses them to schedule these activities at the right time, by releasing the relevant milestones. However, if iProcess Conductor forecasts that it will not be able to release a milestone at the scheduled time, or if that time is reached and the milestone still cannot be released, it recognizes a jeopardy condition.

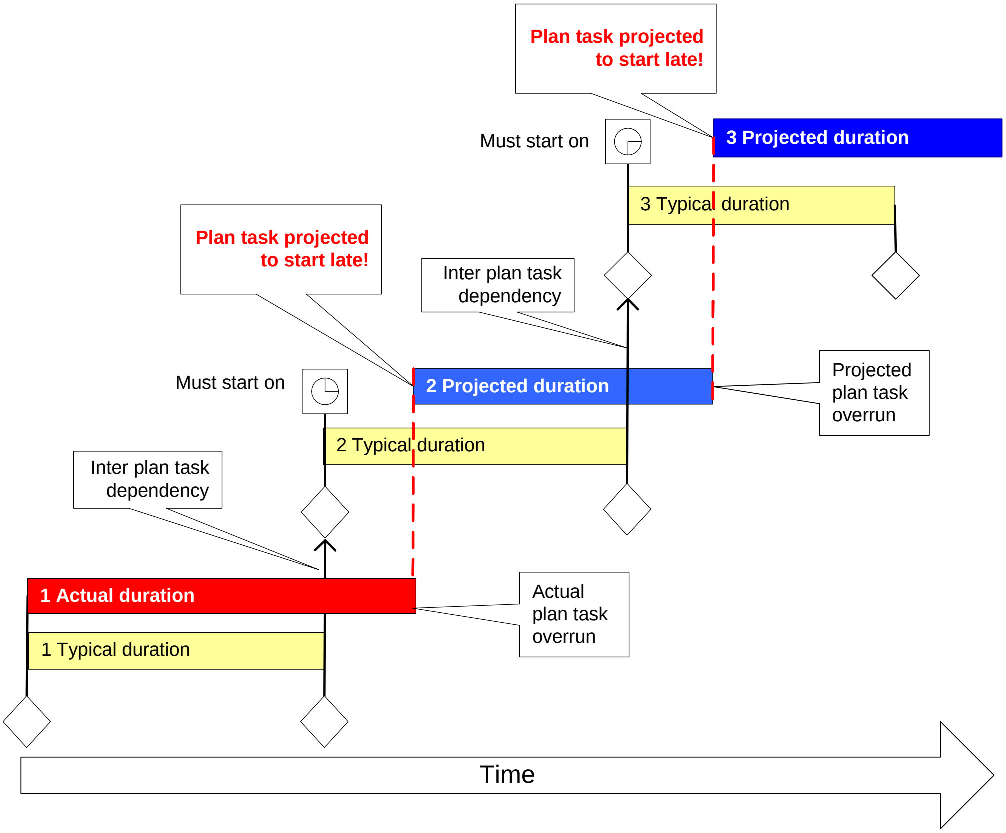
iProcess Conductor predicts when plan tasks subject to "must start on" dependencies are likely to be late in starting. It re-calculates this forecast whenever a plan task is completed.

See Understanding Dependencies on page 42 for more information about dependencies.

iProcess Conductor also recalculates its predictions whenever a monitor event is triggered because a plan task has exceeded either its typical duration or its maximum allowed duration, and after each subsequent monitoring period for as long as the task is still uncompleted. It then uses the duration of the task so far, plus the forecast duration of the remaining tasks in the plan, as the basis of its calculations.

Example

In the following example, the first plan task has taken longer than its maximum duration to complete, and this means that the second and third tasks, connected by "must start on" dependencies, are now predicted to start late.



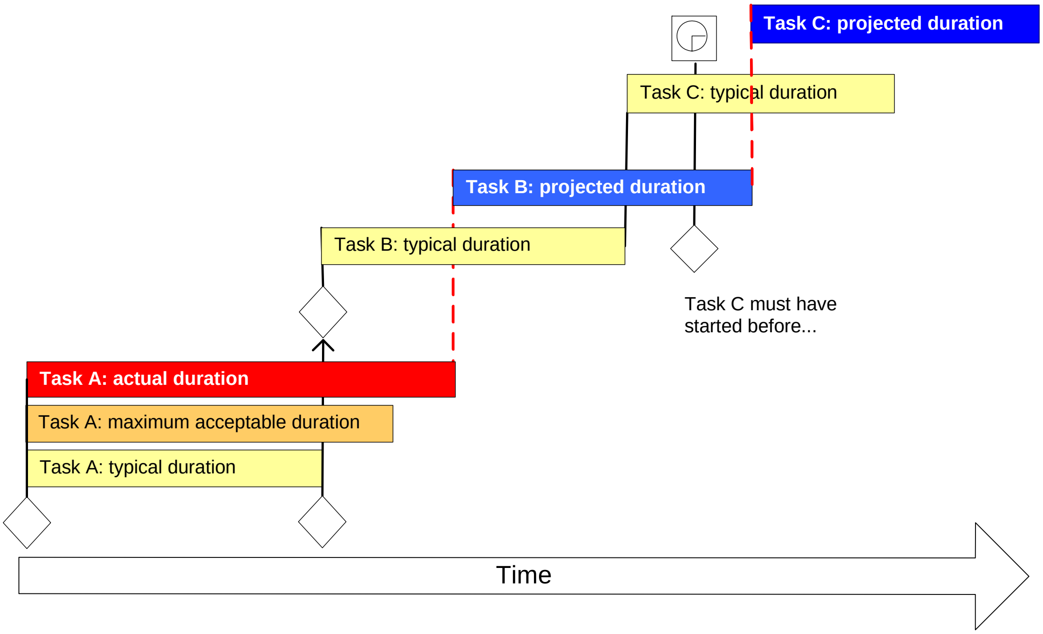
Must Have Started Before

The "must have started before" milestone works in a very similar way to the "must start on" dependency. It is used to indicate that an activity must have started before the date and time specified. TIBCO iProcess Conductor predicts when plan tasks subject to "must have started before" milestones are likely to be late in starting.

If iProcess Conductor forecasts that it will not be able to release a milestone at or before the scheduled time, it recognizes a jeopardy condition. If that time is reached and the milestone still cannot be released, it recognizes another jeopardy condition.

Example

In the following example, TIBCO iProcess Conductor predicts that Task C is not going to start by its required start time.


Task C: projected duration
Task C: typical duration
Task B: projected duration
Task B: typical duration
Task C must have started before...
Task A: actual duration
Task A: maximum acceptable duration
Task A: typical duration
Time

Plan task A has taken longer than its maximum acceptable duration to complete, leading to both tasks B and C being projected to start late. The effect of this delay on Plan Task C is that it is projected to start late by the difference between Plan Task A's typical duration and its actual duration. A "must have started before" dependency exists on plan task C. This does not affect the scheduling of Plan Task C, but monitors the point at Plan Task C is started.

Jeopardy Management for Execution Plans

The fact that a given plan task or milestone is in jeopardy, as described in the previous sections, may or may not indicate that the overall execution plan is in jeopardy. TIBCO iProcess Conductor therefore also provides facilities for monitoring whether the whole execution plan is running on time, or is on the contrary taking longer to complete than it is predicted to. iProcess Conductor does this by monitoring the forecast end date and time of the plan and comparing it against several threshold dates.

If you use start date scheduling or end date scheduling for your execution plans, you can set a different set of jeopardy conditions at plan level. See Scheduling Execution Plans According to Start Dates and End Dates on page 40 for these scheduling techniques.

About Execution Plan Thresholds

To determine whether the execution plan as a whole is in danger of not being completed within an acceptable time, the projected end date and time of the plan is monitored, and is then compared against threshold dates which indicate whether or not the plan is in jeopardy, and if so, how seriously. An execution plan can be monitored against the following four thresholds:

- Earliest Expected Completion Date - this is the end of the typical duration critical path for the plan.
- Risk Region Reporting Threshold - this is a point between the end of the typical duration and the end of the maximum allowed duration for the plan. It is defined as a percentage of the difference between these two dates - for example, half-way between the earliest and latest dates, or 30% of the difference. This percentage is defined globally for all execution plans, but you can override it for a particular plan in the Jeopardy Action Preferences window of the user interface. See *TIBCO iProcess Conductor User's Guide* for details.
- Latest Expected Completion Date - this is the end of the maximum allowed duration critical path for the plan.
- Latest Acceptable Completion Date or Out of Specification Threshold - this threshold marks the end of the contingency period allotted for the execution plan, and thus the point at which the plan must be considered out of specification. A percentage of the maximum duration allowed critical path can be allowed for contingency - for example, if you allow 15% contingency on a 10-hour critical path, the out of specification threshold will be 90 minutes after the Latest Expected Completion Date. This percentage is defined globally for all execution plans, but you can override it for a particular plan in the Jeopardy Action Preferences window of the user interface. See *TIBCO iProcess Conductor User's Guide* for details.

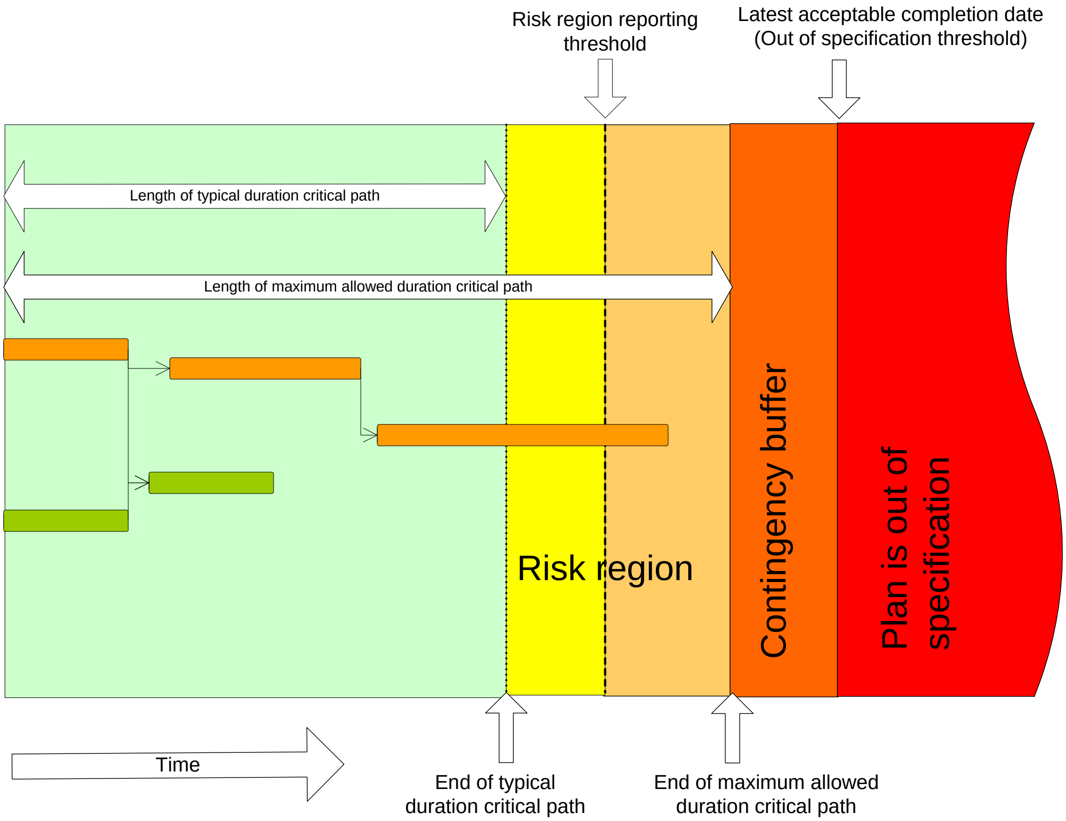
If an execution plan passes any of the thresholds, in either direction, it enters a specific jeopardy management state. Each jeopardy management state indicates a different level of risk, and there is a color associated with each one for easy recognition.

These jeopardy management states are as follows:

- Execution Plan Projected to Complete Within the Defined Typical Duration. This means that the plan is expected to complete by the Earliest Expected Completion Date. While the plan is within this threshold, it is not considered to be in jeopardy. The color associated with this state is green.
- Execution Plan Projected to Complete Within Acceptable Slippage. The plan has slipped past the Earliest Expected Completion Date but is still projected to complete before it reaches the Risk Region Reporting Threshold. While the plan is within this threshold, it is not considered to be in jeopardy. The color associated with this state is yellow.
- Execution Plan Projected to Complete Within Maximum Allowed Duration for Plan. The plan has slipped beyond the Risk Region Reporting Threshold but is still projected to complete before it reaches the Latest Expected Completion Date. While the plan is within this threshold, it is considered to be in jeopardy. The color associated with this state is amber.
- Execution Plan Projected to Complete Using Contingency Buffer. The plan has slipped beyond the Latest Expected Completion Date but is still projected to complete before it reaches the Out of Specification Threshold. While the plan is within this threshold, it is considered to be in jeopardy. The color associated with this state is red/amber.
- Execution Plan Projected to Complete Out of Specification. The plan has slipped beyond the Latest Acceptable Completion Date (Out of Specification Threshold). If the plan has passed this threshold, it is considered to be in extreme jeopardy. The color associated with this state is red.

Example

In this example, the projected end of the execution plan has passed the Risk Region Reporting Threshold. The plan is currently forecast to complete before the Latest Expected Completion Date. It is therefore in a jeopardy management state of Execution Plan Projected to Complete Within Maximum Allowed Duration for Plan - amber.


Risk region reporting threshold
Latest acceptable completion date (Out of specification threshold)
Length of typical duration critical path
Length of maximum allowed duration critical path
Risk region
Contingency buffer
Plan is out of specification
Time
End of typical duration critical path
End of maximum allowed duration critical path

Jeopardy Conditions

To meet the considerations set out in the preceding sections, TIBCO iProcess Conductor uses the following list of jeopardy conditions:

Monitored Item	Jeopardy Condition	Description	Lowest Level
Execution Plan End Date	Predicted Infeasible	For end date scheduling	Execution Plan
	Predicted Feasible	For end date scheduling	Execution Plan
	Passed	For end date scheduling	Execution Plan
Typical Duration Critical Path End Point	Threshold Passed Predicted Increasing		Execution Plan
	Threshold Passed Predicted Decreasing		Execution Plan
	Threshold Crossed		Execution Plan
Reporting Threshold	Threshold Passed Predicted Increasing		Execution Plan
	Threshold Passed Predicted Decreasing		Execution Plan
	Threshold Crossed		Execution Plan
Maximum Allowed Duration Critical Path End Point	Threshold Passed Predicted Increasing		Execution Plan
	Threshold Passed Predicted Decreasing		Execution Plan
	Threshold Crossed		Execution Plan
Out of Specification Threshold	Threshold Passed Predicted Increasing		Execution Plan
	Threshold Passed Predicted Decreasing		Execution Plan
	Threshold Crossed		Execution Plan

Monitored Item	Jeopardy Condition	Description	Lowest Level
Process Component	Typical Duration Exceeded		Section
	Maximum Allowed Duration Exceeded		Section
	Monitor Period Duration Exceeded		Section
Must Start On dependency	Predicted Infeasible		Milestone
	Predicted Feasible		Milestone
	Time Passed	The milestone could not be released	Milestone
Must Have Started Before dependency	Predicted Infeasible		Milestone
	Predicted Feasible		Milestone
	Time Passed	The milestone was not released	Milestone

For this table:

- In the Jeopardy Condition column, "Increasing" means that the execution plan’s forecast duration has increased, and therefore its forecast end date has passed the specified threshold. "Decreasing" means the opposite; the plan is now forecast to take less time than the previous prediction, and therefore its predicted end date has passed back to a date sooner than the specified threshold.
- For more information on end date scheduling, see Scheduling Execution Plans According to Start Dates and End Dates on page 40.

You can define the consequential actions that will take place as a result of each of these jeopardy conditions.See "Configuring a Jeopardy Condition for an Execution Plan or Process Component" in *TIBCO iProcess Conductor User’s Guide* for details of how to set this up. The default values that you define for these consequential actions can be overridden if you define specific actions for a particular execution plan, plan task, or milestone, as follows:

- You can define global consequential actions for all jeopardy conditions. These actions will always be taken unless overridden by a more specific action.
- For all jeopardy conditions, you can define consequential actions at execution plan level. These override the global settings for that particular plan.

- For those jeopardy conditions where the Lowest level in the table is defined as Section, you can define consequential actions at the level of a process component section. For that particular process component section, these override both the global settings and any plan-level settings.
- For those jeopardy conditions where the Lowest level in the table is defined as Milestone, you can define consequential actions at milestone level. For that particular milestone, these override both the global settings and any plan-level settings.

About Jeopardy Consequential Actions

Once an execution plan or a plan task has been identified as being in jeopardy, iProcess Conductor can take action in response to it, as a consequence of the jeopardy condition. For each of the listed jeopardy conditions, you can configure iProcess Conductor to take any of the following three possible consequential actions:

- suspend the managed object (the plan task or the execution plan).
- generate an iProcess procedure.
- generate an XML-based JMS message through a topic or a queue. This could be used, for example, to generate an event to another system, allowing easy integration with event correlation systems such as TIBCO BusinessEvents® or business activity monitoring systems such as TIBCO BusinessFactor®.

You can define any combination of these actions or can specify that iProcess Conductor should take no action at all, depending on your requirements for that particular jeopardy condition.

The information input to an iProcess procedure and the JMS message will have the same XML payload. The XML payload defines the jeopardy condition and also uniquely identifies the item in which the jeopardy condition occurred.

You can define jeopardy actions at the following levels:

- globally
- execution plan
- plan task (process component)

If you set jeopardy actions at each of the levels, they are overridden as follows:

- A jeopardy action set at execution plan level overrides a global jeopardy action.
- A jeopardy action set at plan task level overrides both a jeopardy action set at execution plan level and a global jeopardy action.

Configuring Jeopardy Management

TIBCO iProcess Conductor enables you to configure a range of jeopardy management settings to suit your business needs. iProcess Conductor is installed with default values for these settings defined in the `AppConfig.xml` file. You can edit that file to replace these delivered defaults with your own values, which will then apply globally; see *TIBCO iProcess Conductor Administrator's Guide* for details of the file and of how to edit it. For many of these parameters, you can also define values which apply to particular execution plans or process components and which will override your global settings; see *TIBCO iProcess Conductor User's Guide* for details of how to do this.

You can set the following values:

- Turn jeopardy management on or off. By default it is enabled, but if the jeopardy monitoring capability of TIBCO iProcess Conductor is not required you can disable it.
- Specify which items should be monitored. For example it is possible to monitor when execution plans actually cross the defined thresholds, but to turn off forecasting of whether they are likely to exceed these thresholds. Similarly, you could choose not to monitor when process component sections are forecast as likely to exceed their typical and maximum durations - or even choose not to monitor process component sections at all. For example, the following file entries set thresholds. You can specify where the risk reporting threshold and the out of specification threshold for execution plans are to be set. See About Execution Plan Thresholds on page 32 for details. These settings are defined globally for all execution plans, but you can override them for a particular plan; see *TIBCO iProcess Conductor User's Guide* for details.
- Set the monitoring interval. You can specify the interval between monitor events for process component sections. The delivered default is 60 seconds; you can set a new global value, but you cannot override the global value for specific execution plans.
- You can also specify how many times a monitor event should be triggered when a process component overruns its defined duration. The delivered default is to trigger monitor events indefinitely until the section being monitored completes, but you can change this to specify a maximum number. This value also applies globally, and cannot be overridden.
- Set jeopardy consequential actions. You can specify what consequential actions should be associated with a particular jeopardy condition. These global settings can be overridden for a particular execution plan or process component section, as described in About Jeopardy Consequential Actions on page 37.

You can set global consequential actions for:

- When an execution plan passes a threshold.
- When an execution plan is predicted to exceed a threshold by an *increasing* margin.
- When an execution plan is predicted to exceed a threshold by a *decreasing* margin.
- When a process component section exceeds a defined duration.
- When a monitor event is triggered by a process component section.

By default, these are all set to generate an iProcess procedure, but, as described in About Jeopardy Consequential Actions on page 37, you can define other actions to take place instead, or additionally.

- If any consequential action is defined as being an iProcess work item, you must specify the name for that procedure, and may optionally specify a description for the iProcess case started by the procedure, and the username under which it will be run. These settings apply globally, to all work items in the iProcess Conductor's jeopardy management.

You must specify a procedure name before you begin to operate TIBCO iProcess Conductor, because the default value as delivered (ERROR001) is not suitable for actual use.

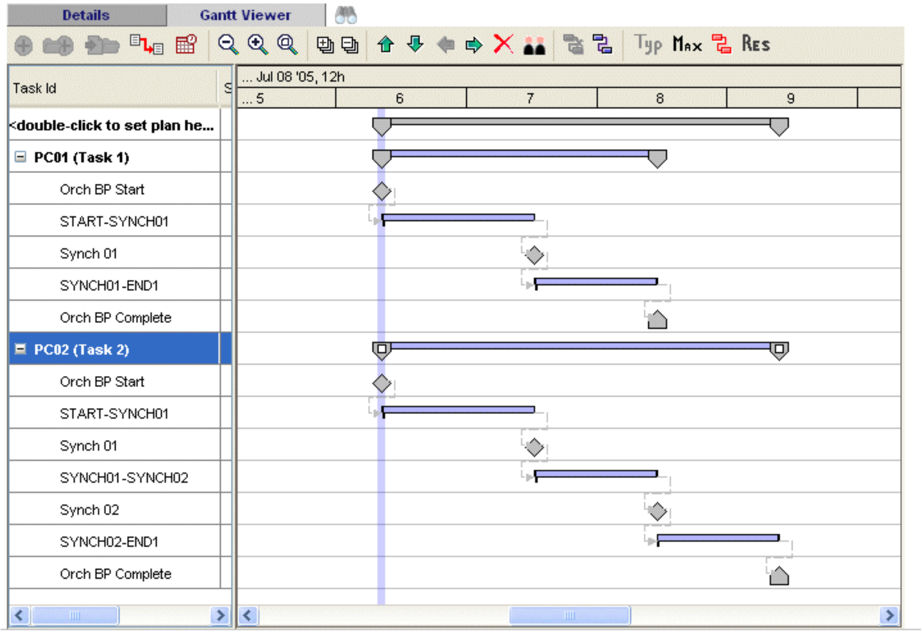
- If any consequential action is defined as being a JMS message, you need to define the destination queue or topic to which the messages will be sent. This applies globally, to all JMS messages in the iProcess Conductor's jeopardy management.

Scheduling Execution Plans According to Start Dates and End Dates

It is possible to schedule execution plans depending on the required start date/time or the required end date/time of execution plans.

Scheduling Execution Plans According to the Required Start Date/Time

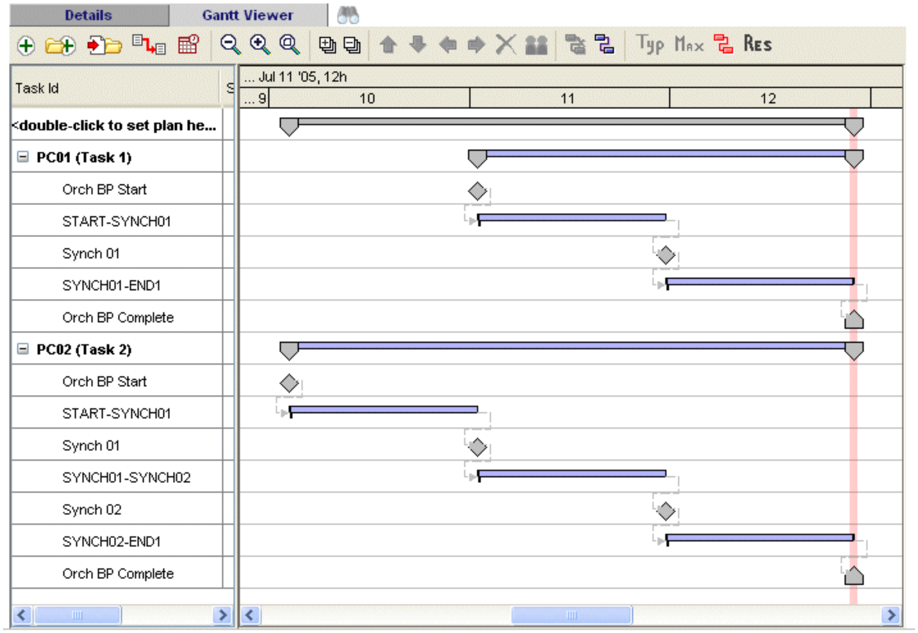
The following diagram shows how the execution plan is scheduled according to the required start date/time:



In this example, both plan tasks start on the same date at the same time. Plan Task 01 is predicted to finish before Plan Task 02 because its typical duration is less than Plan Task 02.

Scheduling Execution Plans According to the Required End Date/Time

The following diagram shows how the execution plan is scheduled according to the required end date/time:



In this example, Plan Task 01 starts later than Plan Task 02 but finishes on the same date and time as Plan Task 02. This is because the typical duration for Plan Task 01 is less than for Plan Task 02.

See "Scheduling an Execution Plan on Required End Date/Time" in *TIBCO iProcess Conductor User's Guide* for more information.

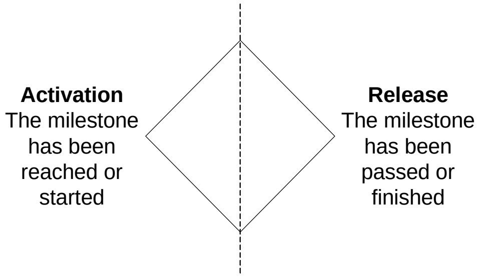
Understanding Dependencies

Dependencies can be created between:

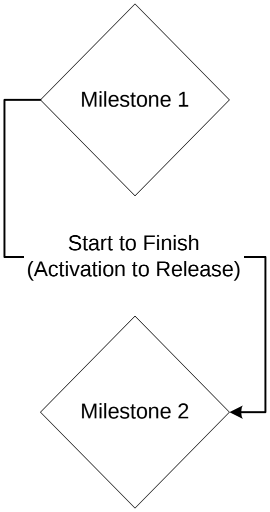
- plan task milestones
- summary groups (or between plan task milestones and summary groups)
- execution plans

Milestone Dependencies

The duration of a milestone may be so short as to appear instantaneous, but in reality every milestone has two parts:

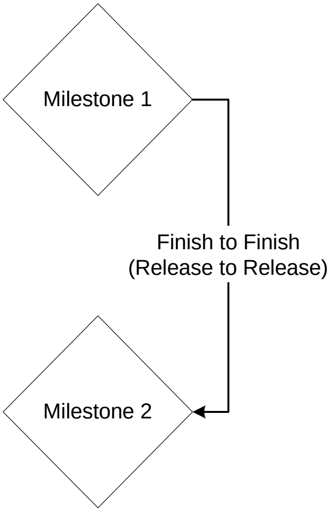

Activation
The milestone has been reached or started
Release
The milestone has been passed or finished

When you create a dependency on a milestone, you can either depend on the milestone being activated (starting) or releasing (finishing). This is shown by a line coming from either the left (start) side of the milestone, or the right (finish) side. The following example shows a Start to Finish Dependency:



This means that Milestone 2 cannot Release (Finish) until Milestone 1 is Activated (Starts).

In contrast, a Finish to Finish Dependency is represented like this:



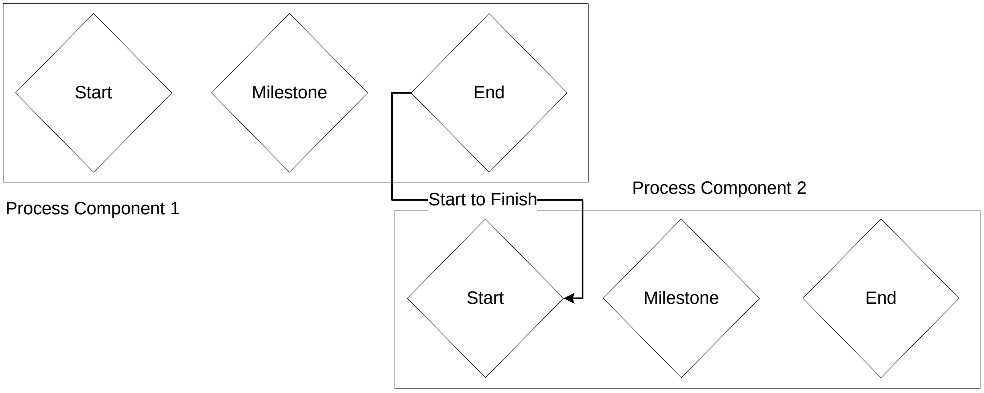
This means that Milestone 2 can only be released when Milestone 1 is released.

End Milestones

An end milestone is a special case in that:

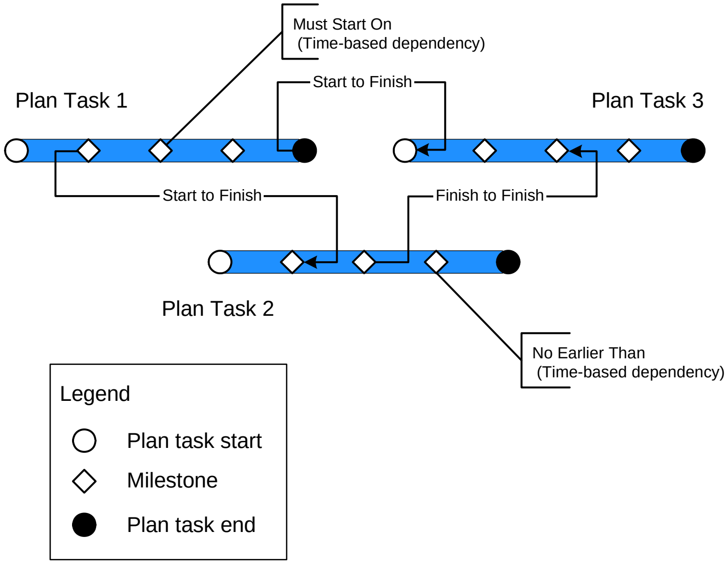
- another milestone cannot depend on the finish of an end milestone because there is no Finish (Release) on a Task Complete message
- the end milestone cannot be dependent on any other milestones.

The following example shows a Start to Finish dependency on an end milestone:



This example shows that the start of Process Component 2 is dependent on the end milestone of Process Component 1 (technically the activation of the end milestone).

The following example shows some dependencies between execution plan task milestones:



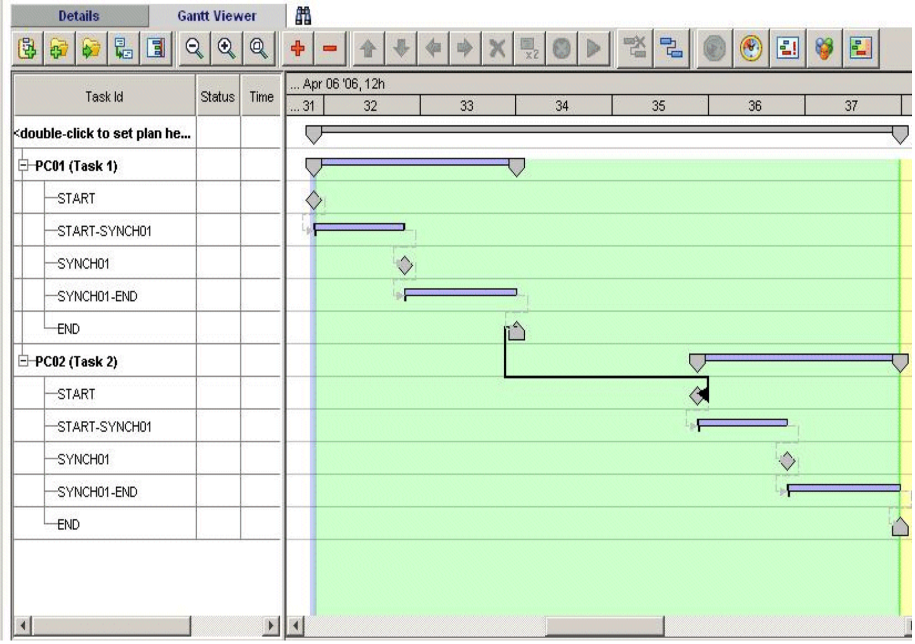
The following table lists the types of dependencies that can be set up between plan task milestones.

Dependency	Effect
Must Start On	The milestone must start on the date/time specified. If it cannot for some reason (for example, because a previous plan task is late), a jeopardy condition is triggered; see Must Start On Dependencies on page 29 for more details.
Start No Earlier Than	The summary group must not start before the date/time specified. If the date/time is reached and other dependencies to the milestone have not been satisfied, the milestone remains pending.
Finish to Finish	One milestone is able to be released when the other milestone is released.
Start to Finish	One milestone can release only when the other begins.

Delayed Milestones

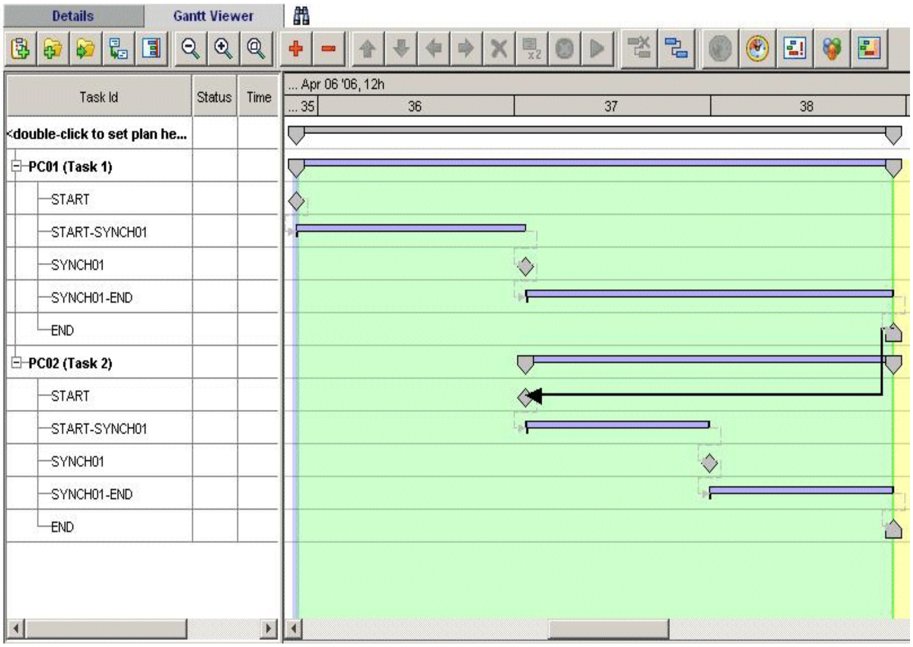
For all types of plan task dependency, you can delay the release of a milestone even though its dependencies have been satisfied. You do this by adding a time lag to the dependency. Time lags can be positive and negative:

- **Positive**. The following example demonstrates a positive time delay.



In this example, process component 02 cannot start until 2 minutes after process component 01 has completed.

- **Negative**. The following example demonstrates a negative time delay.



In this example, process component 02 cannot start until 2 minutes before process component 01 is predicted to have finished.

Note:

- When using negative time values, you cannot pass field data; there can be no field data to pass, because the milestone cannot start or release until a point in time before the milestone that is the source of the dependency is predicted to release.

- iProcess Conductor evaluates the time delays based on completed sections. This means that if a section in a path that can affect a dependency with a negative time lag completes later than it should have but before the projected release time for a milestone, then the negative time delta can be re-evaluated and this slippage taken into account. However, if the section does not complete before the projected release time for a milestone but is taking longer than expected, iProcess Conductor will not take any specific action for the negative time lag. This means it will release the negative time dependency at the time it last calculated and not take into account the slippage.

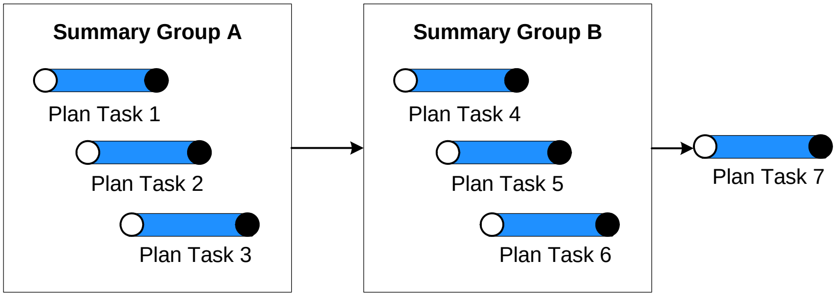
- The time at which a milestone subject to a dependency with a negative time lag releases, is dependent on the actual durations of sections in any related paths through the execution plan. This means that the activation/release time of the source milestone can change as the plan is executing. This is because the source milestone is dependent on the actual times for completed sections of other plan tasks that are part of any related paths of which the source milestone is also a part.

- If, for any reason, it is impossible to release a dependency with the required negative time lag (in other words, the source milestone for the dependency is projected to release in less time into the future than is required by the dependency), then the milestone is started as soon as it can.

Summary Group Dependencies

As well as having individual tasks in a plan, you can also create logical collections of plan tasks known as summary groups. This allows flexibility in creating the execution plan. For example, you can create dependencies on summary groups that all must be completed before the next task is started.

The following example shows dependencies between summary groups and plan tasks:



All of the tasks in Summary Group A must complete before the tasks in Group B can begin. All of the tasks in Summary Group B must complete before Task 7 can begin.

The following table lists the types of dependencies that can be set up between summary groups.

Dependency	Effect
Must Start On	The summary group must start on the date/time specified. If it cannot for some reason (for example, because a previous group completes late), a jeopardy condition is triggered; see Must Start On Dependencies, page 29 for more details.
Start No Earlier Than	The summary group must not start before the date/time specified. If the date/time is reached and other dependencies to the milestone have not been satisfied, the milestone remains pending.

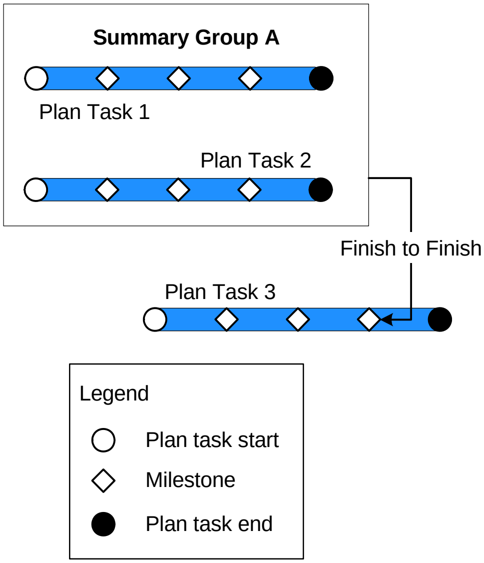
Dependency	Effect
Finish to Start	One summary group can start only when all tasks in the other have finished.

Milestone and Summary Group Dependencies

The dependencies that you can create between milestones and summary groups are summarized in the following sections:

From a Summary Group to a Milestone (except for End)

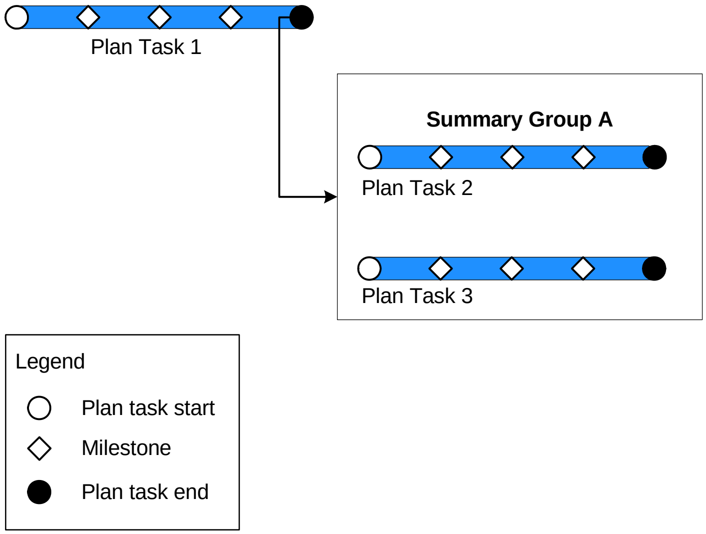
You can create a Finish to Finish dependency from a Summary Group to a milestone in a plan task (except for the End milestone):



This indicates that the milestone in Plan Task 3 is dependent on Summary Group A finishing. In other words, Plan Task 3 cannot finish until all the plan tasks in Summary Group A finish.

From an End Milestone to a Summary Group

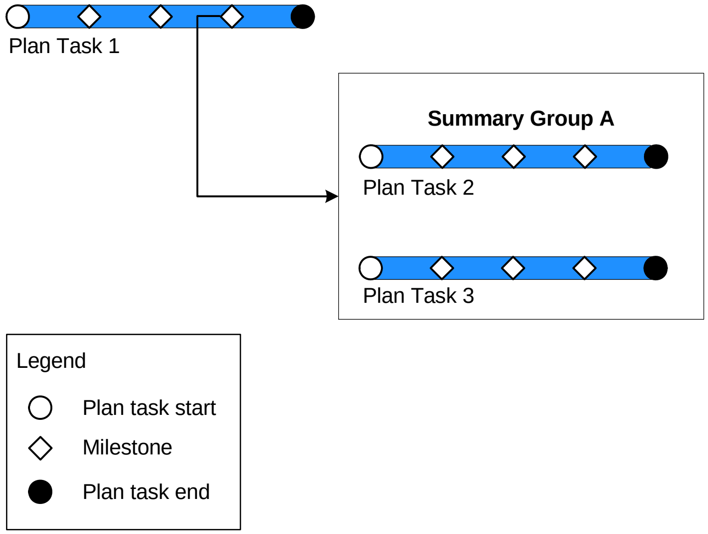
You can create a Start to Start milestone from an End milestone to a Summary Group:



This means that Summary Group A is dependent on the start of the end milestone in Plan Task 1. In other words, the tasks in Summary Group A cannot start until the end milestone in Plan Task 1 is reached (logically the end of Plan Task 1).

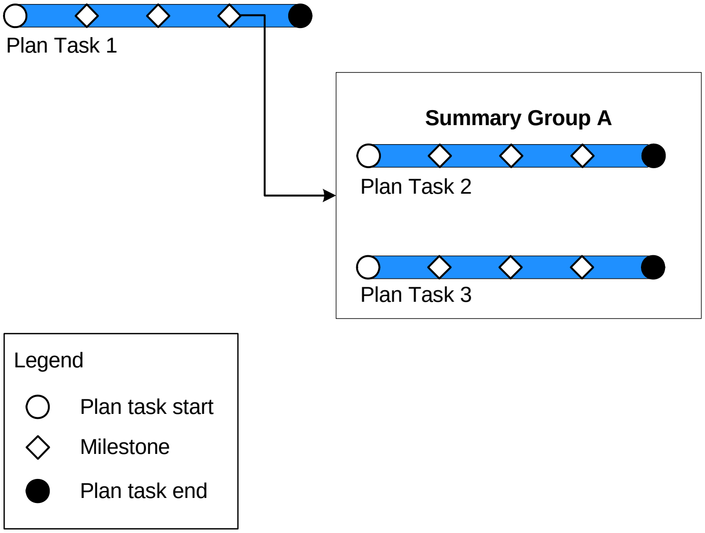
From a Milestone (except End) to a Summary Group

You can create either a Start to Start or Finish to Start dependency from a milestone to a summary group. The following example shows a Start to Start dependency:



This means that Summary Group A is dependent on the start of the milestone in Plan Task 1. In other words, the tasks in Summary Group A cannot start until the milestone in Plan Task 1 is reached.

The following example shows a Finish to Start dependency:

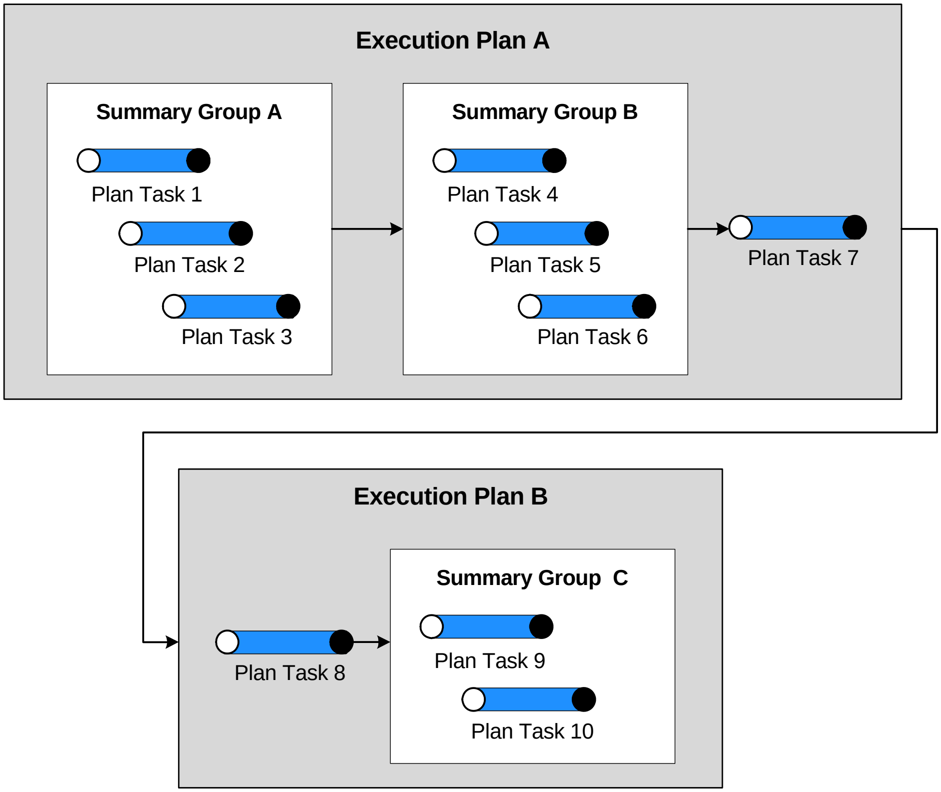


This means that the start of Summary Group A is dependent on the finish of the milestone in Plan Task 1. In other words, the tasks in Summary Group A cannot start until the milestone in Plan Task 1 has finished (released).

Execution Plan Dependencies

Another powerful dependency that you can create is a finish to start dependency between execution plans.

The following example shows a finish to start dependency between Execution Plan A and Execution Plan B:



Execution Plan A must complete before Execution Plan B can begin.

Summary

The following table summarizes the dependencies that you can set up (excluding time-based dependencies such as "Must Start On.")

		TO			
		Milestone (except End)	End milestone	Summary group	Execution plan
FROM	Milestone (except End)	Start to Finish/ Finish to Finish		Start to Start/ Finish to Start	
	End milestone	Start to Finish		Start to Start	
	Summary group	Finish to Finish		Finish to Start	
	Execution plan				Finish to Start

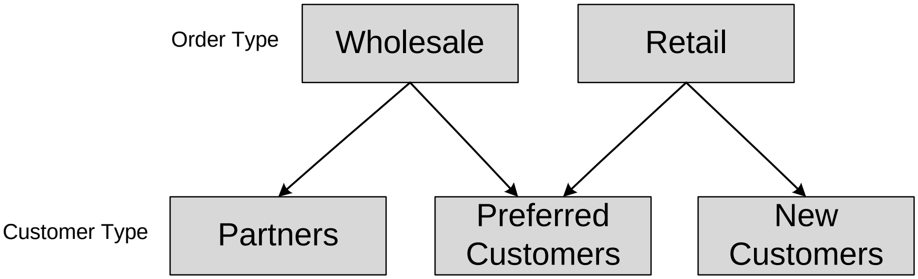
Classifying Objects

TIBCO iProcess Conductor allows you to create named groups of managed objects (orders, execution plans, and process components). This feature is called classification and it allows you to group objects in ways that are meaningful to your business processes (for example, by region, department or by lines of business), and to view objects based on their classification.

There are three steps for creating a classification:

1. Create the classification group (for example, order type or customer type).
2. Create the classification values (for example, wholesale, retail, partners, and so on).
3. Associate a managed object with a classification group.

For example, the following classification groups and values could be created for execution plan templates:



The goal of the classification in this example is to use different types of templates for different types of orders and customers. There can be multiple classification values, so an execution plan template for a wholesale partner could be classified differently from an execution plan template for a wholesale preferred customer.

You can specify that a classification must be mandatory. To continue the example in the previous diagram, this would mean that when an execution plan template is created or updated, you must classify it as Wholesale or Retail.

In the case of templates, classification groups and values can be inherited. For example, you can create an execution plan template with the desired classifications so that execution plans instantiated from the template will inherit the classifications.

You can also use TIBCO iProcess Decisions Plug-in to create rules that determine the classification of received order requests or amendments. There are two ways of ensuring an object is created with mandatory classification values:

- At design time, you can specify classification data in an EAI Orchestration step with a message type of Start Plan.
- In TIBCO iProcess Decisions Plug-in, you can create rules that supply the values at run time in the form of an XML payload.

For more information about specifying classification data, see *TIBCO iProcess Conductor Implementation*.

- If you use mandatory classifications and TIBCO iProcess Decisions Plug-in cannot determine the mandatory classifications, the order request or amendment is rejected.
- You cannot create classifications that invalidate currently active managed objects. For example, you cannot specify that orders must have mandatory classification values if there are active orders that do not contain those classification values.
- You cannot delete a classification value if any managed objects, even inactive ones, use that classification. This prevents you losing valuable audit information from the database.
- If an execution plan template has an association with a classification group that has subsequently been withdrawn, execution plans can still be instantiated from that template without generating an error. However, the instantiated plans will not have a reference to the withdrawn classification group.

Chapter 4 iProcess Procedures

This chapter describes iProcess procedures that you create for use with TIBCO iProcess Conductor.

Topics

Overview

The two main types of iProcess procedures required for use with TIBCO iProcess Conductor:

- An overarching fulfillment procedure is required to manage the fulfillment of the order.
- An iProcess procedure is required for each version of each process component to be defined in TIBCO iProcess Conductor.

You must also plan for order amendments. This can be done as part of the fulfillment procedure or by creating a separate procedure.

EAI Steps

To manage TIBCO iProcess Conductor data and the exchange of data between TIBCO iProcess Engine and TIBCO iProcess Conductor, you use EAI steps, specifically EAI Orchestrator, EAI Order and EAI Transform steps.

Using EAI Orchestrator Steps

The EAI Orchestrator step enables you to send iProcess data to iProcess Conductor and iProcess Conductor process the data before sending the resulting data back to TIBCO iProcess Engine.

When an EAI Orchestrator step is reached in an iProcess procedure, it sends a message to iProcess Conductor indicating that it has been reached. Depending on the message type, it can then wait for a message from iProcess Conductor before it releases. This enables iProcess Conductor to make sure that any tasks that have defined dependencies must have those dependencies satisfied (including the transfer of input and output data) before being allowed to proceed.

EAI Orchestrator steps used in procedures must be made public.

Using EAI Order Steps

The EAI Order step has several functions:

- to provide an interface that enables data from a specific order request or order amendment to be used in iProcess fields and vice versa. It is also possible to extract order details as XML. Tools are provided with TIBCO iProcess Conductor that enable you to manipulate the data, depending on your requirements. For more information, see *TIBCO iProcess Conductor Implementation*.
- to progress the status of an order within TIBCO iProcess Conductor based on the current stage in the fulfillment process.
- to pass the name of the event step to TIBCO iProcess Conductor so that it knows which step in the process to notify when an order amendment is received.

Using EAI Transform Steps

EAI Transform steps provide the facility to transform XML to/from iProcess field data and/or a designated URL. For example, the EAI Transform step could be used to take XML data from an iProcess memo field, apply a transformation to the data, and pass the result to another memo field. Equally, XML data can be parsed and mapped onto discrete iProcess fields.

The EAI Transform step enables you to:

- populate iProcess fields with XML
- map input iProcess fields to XML
- extract XML content and structure
- map XML to output iProcess fields.

The data that is passed in and out of iProcess is called an XML payload.

Fulfillment Procedures

The fulfillment procedure should handle order feasibility, execution plan development and execution plan processing. The goals are to identify the execution plan template that the execution plan should be generated from, and notify TIBCO iProcess Conductor which iProcess Suite processes to orchestrate to fulfill the order.

Order feasibility, execution plan development and execution plan processing can be accomplished in the fulfillment procedure itself, or by using sub-procedures. This section assumes you are using sub-procedures.

The fulfillment procedure has the following primary objectives:

- **Order Feasibility**

  Order feasibility carries out any validation that is required so that the order can proceed. This is defined according to your requirements and can consist of any validation steps that are required by your business process. This could be, for example, a credit check on the customer or a resource inventory.

- **Execution Plan Development**

  The Execution Plan Development sub-procedure fetches the order details from the parent fulfillment procedure and passes them to TIBCO iProcess Decisions Plug-in. TIBCO iProcess Decisions Plug-in then identifies an execution plan template and returns an execution plan ID.

  The Execution Plan Development sub-procedure then passes the execution plan ID to iProcess Conductor which instantiates an execution plan based on the execution plan template.

  In the case of amendments, depending on the status of the execution plan that is fulfilling the order when an amendment is received, TIBCO iProcess Conductor needs to either initiate manual execution plan development or initiate the execution of a new execution plan.

  If TIBCO iProcess Decisions Plug-in cannot identify an execution plan template from the order details, then it does not return an execution plan ID. At this point, the Execution Plan Development sub-procedure sends a work item to a user queue to initiate a manual execution plan.

- **Execution Plan Processing**

  The Execution Plan Processing sub-procedure directs iProcess Conductor to start executing the plan. iProcess Conductor then starts and orchestrates the iProcess Suite processes required to complete the execution plan.

Procedures for Process Components

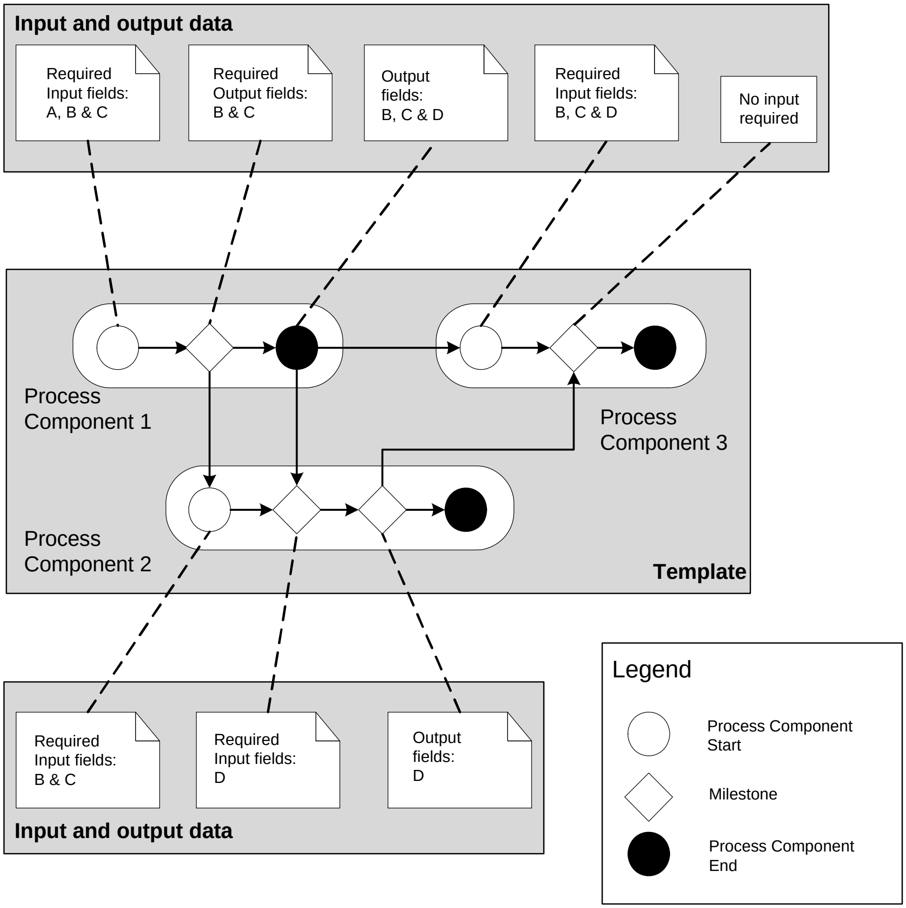
An iProcess procedure is required for each version of each process component to be defined in TIBCO iProcess Conductor. A procedure that is to be associated with a process component must contain EAI Orchestration steps. The EAI Orchestration steps provide the interfaces that enable you to direct TIBCO iProcess Conductor to execute particular tasks in the execution plan and pass any data that is required to and from TIBCO iProcess Conductor. The procedure should also contain an ad-hoc event step that handles cancellation.

This ad-hoc event step must be made public. It must also use the same template as the projected graft step.

EAI Orchestration steps enable you to define messages that specify how TIBCO iProcess Conductor should process the tasks in the execution plan. For example, if some steps are dependent on information from other sources or are dependent on another task completing before they can start, you can define your EAI Orchestration step to have an Activate Sink message type. This means that the EAI Orchestration step will wait for either a message and/or some data from TIBCO iProcess Conductor before releasing. See *TIBCO iProcess Conductor Implementation* for more information.

Example Execution Plan Template

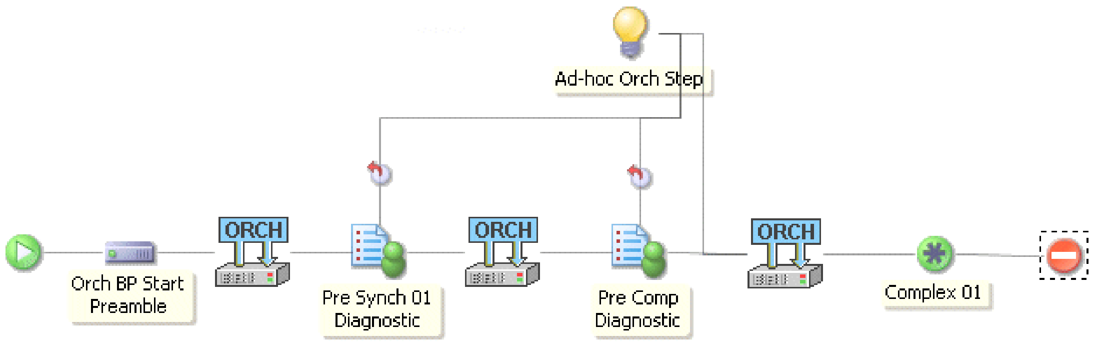
The following diagram shows each process component in the execution plan, the milestones and any required data:



The following table describes how the milestones in the process components match the EAI Orchestration steps in the iProcess procedures.

Process Component	Description	iProcess Procedure
○	Start point of the process component. **Note:** The entry point is the first EAI Orchestration step in the process component. There may be some preamble steps before that, depending on your business process.	An EAI Orchestration step with a message type of Activate Sink for the start point.
◇	Milestone.	An EAI Orchestration step with a message type of Activate Sink for the milestone.
●	Exit point of the process component.	An EAI Orchestration step with a message type of Task Complete.

As mentioned earlier, each process component corresponds to an iProcess procedure. A simplified example of a skeleton iProcess procedure for a process component is illustrated below:



For more information about designing iProcess procedures for use with TIBCO iProcess Conductor, see *TIBCO iProcess Conductor Implementation*.

Chapter 5 Understanding the Execution Plan Lifecycle

This chapter describes the lifecycle of an execution plan in TIBCO iProcess Conductor.

Topics

Overview

When executing an execution plan, if an aspect of an order has changed, you can suspend a plan that is currently running, amend the execution plan and re-activate it again.

Both execution plans and plan tasks within execution plans have a status. The status of an execution plan affects the status of the plan task and vice versa.

Execution Plan Status

An execution plan can have the following statuses that you can view from the user interface:

Status	Description
Draft	The execution plan is being developed.
Pending	The execution plan has been created and is waiting for processing.
Template	An execution plan template cannot be executed directly but is used to instantiate a new execution plan at run-time. The execution plan is validated when you change the status from Draft to Template. You cannot change the status of an execution plan from Template to Draft once an execution plan has a status of Template.
Activating	The execution plan is waiting for an Activate Sink message from each of the grafted tasks.
Started	The execution plan is ready for orchestration (the Activate Sink messages have been received). Typically a plan is in this state waiting for inter-plan dependencies to be satisfied, or it may be waiting as late as possible before becoming orchestrating.
Active	The execution plan is currently being orchestrated and is executing (inter-plan dependencies have been satisfied).
Cancel	The execution plan is currently being orchestrated but the user has requested that the execution plan is cancelled. The cancel operation is currently in progress.
Cancelled	The cancel operation performed on the execution plan is complete.
Suspended	The execution plan is currently being processed but the user has requested that the execution plan is suspended. The suspend operation on the execution plan is complete.

Status	Description
Withdrawn	The execution plan template has been withdrawn and is no longer available.
Completed	The execution plan has been orchestrated and completed.

Plan Task Status

A plan task can have the following statuses:

Status	Description
Started	The start milestone is active. The plan task has received an Activate Sink message from the iProcess process that is implementing the task.
Active	The start milestone has fired. The plan task is currently being executed.
Cancel	The plan task is currently being orchestrated but the user has requested that the containing execution plan is being cancelled. The cancel operation on this plan task is currently in progress.
Cancelled	The cancel operation performed on the plan task is complete.
Suspended	The plan task is currently being executed but the user has requested that the containing execution plan is suspended. The suspend operation on the plan task is complete.
Pending	The plan task is waiting for an Activate Sink message from the iProcess process that is implementing the task.
Completed	The plan task has been orchestrated and completed.

Summary Group Status

A summary group can have the following statuses:

Status	Description
Started	All of the plan tasks in the summary group have the status of Started.
Active	At least one of the tasks contained in the summary group is Active, the remaining plan tasks have at least Started.

Status	Description
Cancel	A plan task contained in the summary group is currently being orchestrated but the user has requested that the containing execution plan is being cancelled. The cancel operation on this plan task is currently in progress.
Cancelled	The cancel operation performed on a plan task contained in the summary group is complete.
Suspended	A plan task in the summary group is currently being executed but the user has requested that the containing execution plan is suspended. The suspend operation on the plan task is complete.
Pending	A plan task contained in the summary group is waiting for an Activate Sink message from the iProcess process that is implementing the task.
Completed	The plan tasks contained in the summary group have been orchestrated and completed.

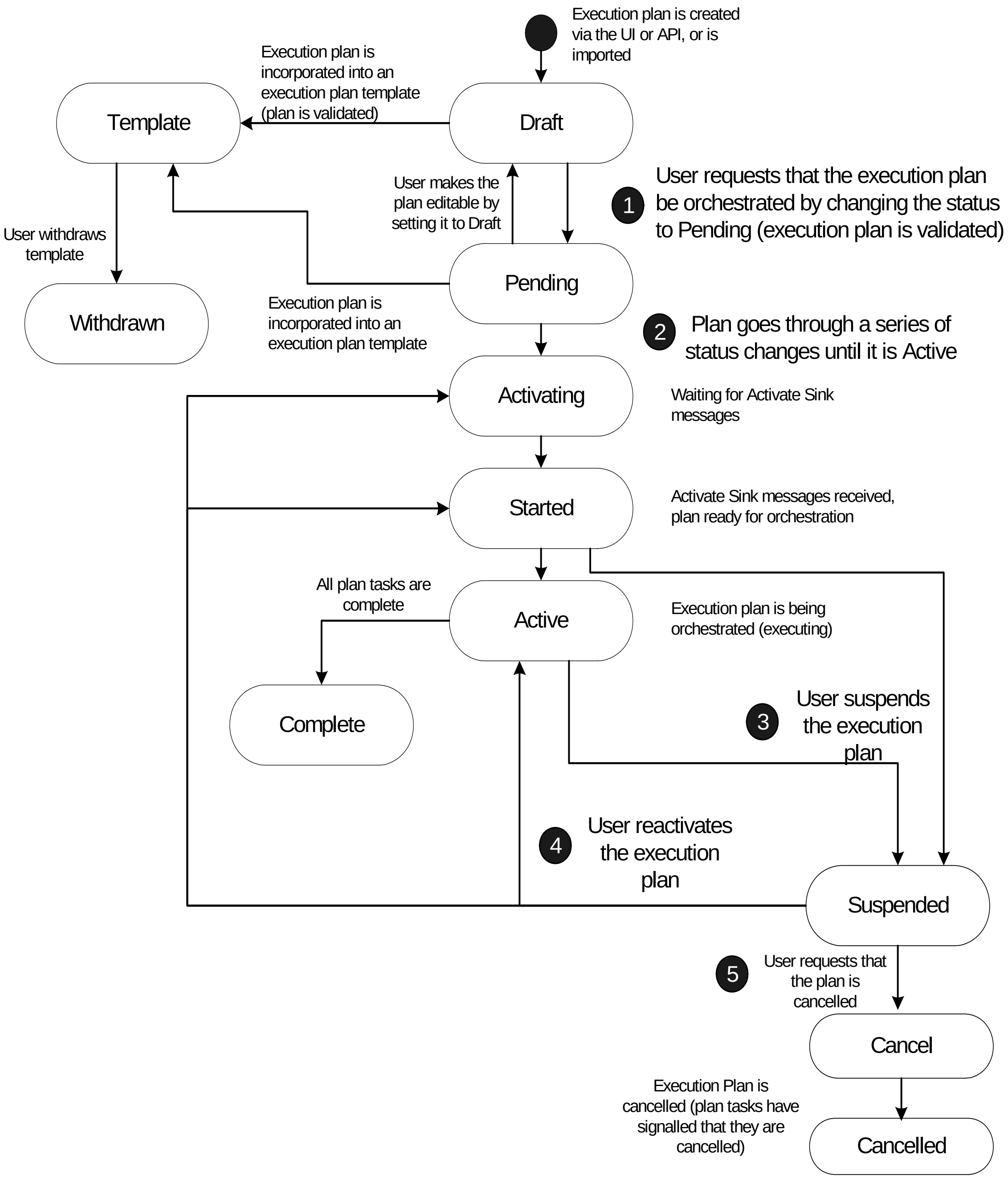
Milestone Status

A milestone can have the following types of statuses:

Status	Description
Pending	The EAI Orchestration step has not yet been reached.
Active	The EAI Orchestration step has sent out its Activate Sink message and is waiting for TIBCO iProcess Conductor to send a message back to tell it to release.
Fired	The EAI Orchestration step has received a message from TIBCO iProcess Conductor telling it to release and it has released.

Lifecycle of an Execution Plan

The following diagram illustrates the lifecycle of an execution plan. It is important to note that the path the execution plan follows depends on its current status. For example, an execution plan with a status of Draft cannot have its status set to Cancelled.


Execution plan is created via the UI or API, or is imported
Template
Draft
Execution plan is incorporated into an execution plan template (plan is validated)
User makes the plan editable by setting it to Draft
1
User requests that the execution plan be orchestrated by changing the status to Pending (execution plan is validated)
User withdraws template
Withdrawn
Pending
Execution plan is incorporated into an execution plan template
2
Plan goes through a series of status changes until it is Active
Activating
Waiting for Activate Sink messages
Started
Activate Sink messages received, plan ready for orchestration
All plan tasks are complete
Active
Execution plan is being orchestrated (executing)
Complete
3
User suspends the execution plan
4
User reactivates the execution plan
Suspended
5
User requests that the plan is cancelled
Cancel
Execution Plan is cancelled (plan tasks have signalled that they are cancelled)
Cancelled

Each number in the following headings corresponds to the same number in the previous diagram.

1 - Validating an Execution Plan

When you design an execution plan in TIBCO iProcess Conductor, you are either:

- designing an execution plan template on which to base future orders, or
- designing a one-off, manual execution plan in response to a specific customer order.

When you create an execution plan in TIBCO iProcess Conductor, its status is initially Draft.

For the execution plan to be processed by TIBCO iProcess Conductor, you need to change the status of the execution plan from Draft to either:

- Pending, if this is a manual execution plan.
- Template, if this is an execution plan template.

When you change the status of an execution plan to Pending or Template, TIBCO iProcess Conductor validates the execution plan at that point. If there are any problems identified with the execution plan, they will be reported to you by TIBCO iProcess Conductor. It is possible to change the status of an execution plan whose status is Pending back to Draft.

Execution plans imported into iProcess Conductor using the Import utility can be at either Draft, Pending or Template status.

Amending an Execution Plan Template

Once an execution plan has a status of Template, it is not possible to change its status back to Draft, for example, if you want to amend the execution plan. To amend an execution plan template, create a new execution plan. You can either create a completely new execution plan or copy the existing execution plan template and use that as a basis for modification.

2 - Processing an Execution Plan

When an execution plan is ready to be processed, its status is Pending. An execution plan must have a status of Pending before TIBCO iProcess Conductor can start to process it.

The way an execution plan reaches the Pending status depends on whether it is a manual execution plan or an execution plan that is based on an execution plan template:

- If it is a manual execution plan, it is Pending because you have set its status to Pending when you create the execution plan.
- If it is based on an execution plan template, TIBCO iProcess Conductor has automatically instantiated an execution plan based on the execution plan template due to a received order and set its status to Pending.

Once TIBCO iProcess Conductor has started to process an execution plan, it goes through the following states, finally reaching the Active state:

- Activating - the plan is waiting for Activate Sink messages.
- Started - the Activate Sink messages have been received and the plan is ready for orchestration.
- Active - the plan is being orchestrated.

This means TIBCO iProcess Conductor starts a case of each of the procedures associated with each process component and plan task. The plan tasks start processing through until they reach any dependencies. The status of the plan tasks changes depending on where they are in the progress of the execution plan.

Once an execution plan has a status of Active, it can either

- process through until it is complete, or,
- it can be suspended. You can suspend an execution plan from the TIBCO iProcess Conductor user interface.

3 - Suspending an Execution Plan

When you suspend an execution plan from TIBCO iProcess Conductor, TIBCO iProcess Conductor sets its status to Suspended. It also suspends the currently running processes that are grafted to this execution plan. Once you have suspended an execution plan, you need to refresh it to get the latest status.

Once an execution plan has a status of Suspended, it can become either:

- re-activated. Its status is set to Started or Active, depending on the original status of the execution plan. If any new plan tasks were added to the execution plan while it was suspended, its status is set to Activating.
- cancelled. Its status is set to Cancelled.

You can re-activate or cancel an execution plan from the TIBCO iProcess Conductor user interface.

The status of the plan tasks in a suspended plan depends on the stage the plan tasks have reached:

- If a plan task has already completed, its status is Complete.
- If a plan task is currently running, its status is Suspended.
- If a plan task has not started executing, its status is Pending.

Amending an Execution Plan

To amend an execution plan, you need to amend the execution plan's individual plan tasks. You can amend plan tasks providing the source of the dependency has not yet become active or fired.

An execution plan must be suspended before you can do this. You can:

- amend plan tasks whose status are Suspended or Pending.
- cancel plan tasks whose status are Suspended or Pending.
- add new plan tasks.
- create new dependencies on plan tasks whose status is Completed.
- create new dependencies on plan tasks whose status is Pending.
- delete dependencies on plan tasks whose status is Pending.

Once you have amended an execution plan, you need to re-activate it so that the changes take effect.

4 - Re-activating an Execution Plan

When you re-activate an execution plan from the TIBCO iProcess Conductor user interface, TIBCO iProcess Conductor sets the status of the execution plan to Active.

Once you have re-activated an execution plan, you need to refresh it to get the latest status.

5 - Cancelling an Execution Plan

An execution plan's status must be Suspended before you can cancel it.

When you cancel an execution plan from TIBCO iProcess Conductor, TIBCO iProcess Conductor sets the status of the execution plan to Cancel. TIBCO iProcess Conductor then automatically cancels all of the processes that are being orchestrated for the execution plan and sets the status of those plan tasks to Cancelled. Once all of the processes have stopped, TIBCO iProcess Conductor sets the status of the execution plan to Cancelled. Once you have cancelled an execution plan, you need to refresh it to get the latest status.

An order cannot be reactivated once its status is set to Cancelled.

Chapter 6 Working with Orders

This chapter is an overview of an order in TIBCO iProcess Conductor.

Topics

Overview of an Order

An order is managed by iProcess Conductor Order Management by processing order requests and order amendments. Order requests and amendments are submitted to iProcess Conductor Order Management in the form of an XML document, either as a JMS message or via the TIBCO iProcess Conductor user interface.

Order requests and amendments consist of an order header and one or more order items. Order requests must conform to the XML schema. It is possible to extend the schema. For more information, see *TIBCO iProcess Conductor Implementation*.

When an order request is submitted to iProcess Conductor Order Management, it creates the order and stores it in the database. It also maintains a working view of the order. This is a consolidated view of the order request and any order amendments that have been applied to an order. See About the Consolidated View on page 85 for more information.

About Order Headers

The table below lists the information that can be contained in an order header:

Type	Description
Order Reference	(Mandatory) A unique identifier supplied by the system that submits the order. iProcess Conductor Order Management uses the Order Reference to determine whether the order is a new request or an amendment. It does this by checking if an order with the same Order Reference is already stored in the database. If the order already exists, the order is an amendment. If there is not, then it is a new order request.
Customer Reference	(Mandatory) A reference that enables iProcess Conductor Order Management to retrieve the current customer profile and to identify the customer to other systems interested in the order.
Originator	(Mandatory) The identity of the system that submits the order. For example, the name of the customer service representative or a peer system name.
Required by Date	(Optional) The date and time the order is required.
Invoice Address	(Optional) The address to invoice for the order, if different from the customer address.

Type	Description
Delivery Address	(Optional) The address to deliver the order, if different from the customer address.
Order Service Level Agreements	(Optional) List of the identifiers of any service level agreement that applies to this order.
Notes	(Optional) Any additional text that may be supplied by the submitter or submitting system.

About Order Items

The table below lists the information that can be contained in an order item:

Type	Description
Order Item Identifier	(Mandatory) A unique identifier that identifies the item with the order.
Product Specification Identifier	(Mandatory) The identifier of the specification of the product to be supplied.
Quantity	(Mandatory) The amount of the product required.
Unit of Measure	(Mandatory) The unit of measure of the product required, for example, kg, m, unit, minutes.
Required by Date	(Optional) The date and time the order is required.
Delivery Address	(Optional) The address to deliver the order, if different from the customer address.
Notes	(Optional) Any additional text that my be supplied by the submitter or submitting system.
Product Characteristics	(Optional) A list of product characteristics that are supplied as input parameters to the order to provide additional information to the product specification. For example, this may be the color of a mobile telephone or a particular customer's off-peak tariff.
Order Line Service Level Agreements	(Optional) A list of the identifiers of any service level agreements that apply to this order line.
Customer Item Identifier	(Optional) The identity of the existing image item to be updated. This is only applicable for cease or update actions.

Type	Description
Order Line Action	(Mandatory) The action required for the specific product referred to in the order line. You can enter one of the following actions: Provide - The customer has requested a new service. Cease - The customer has requested that an existing service should cease. Update - The customer has requested that an existing service be updated in some way. Cancel - The customer has cancelled the product request.
Status	(Mandatory) The current status of the order. This is automatically filled in and you cannot amend it. The status changes with the order item's lifecycle.
Status Changed	(Mandatory) The date and time that the order item's status last changed. This is automatically filled in and you cannot amend it. It initially shows the date and time the order item was created, and is updated to reflect later status changes.
User Defined Data Fields	(Optional) A list of name value pairs that you can supply as additional input parameters to the order.

Amending Orders

This section describes the concepts behind amending orders in TIBCO iProcess Conductor.

About Amendments

An order amendment has the same structure as an order request. See Overview of an Order on page 76.

iProcess Conductor Order Management uses the Order Reference to determine whether the order is an order request or an order amendment. It does this by checking to see if it already has an order with the same Order Reference stored in the database. If there is an order with that reference already in the database then it is an order amendment. If there is not, then it is an order request.

When amending orders, you can:

- add new order lines
- change existing order lines

You cannot:

- delete order lines.
- blank field names in the order. For example, if there are any fields in the amendment order header that are empty this does not cause the equivalent fields in the consolidated view to be cleared. This is because iProcess Conductor Order Management interprets a blank field to mean that no changes need to be made.

You can make as many amendments as you like to an order during its lifecycle. Each amendment to an order is stored in the database building up a hierarchical tree in the sequence the amendments were submitted. Each amendment is also applied to a consolidated view of the order.

Order Amendment Process

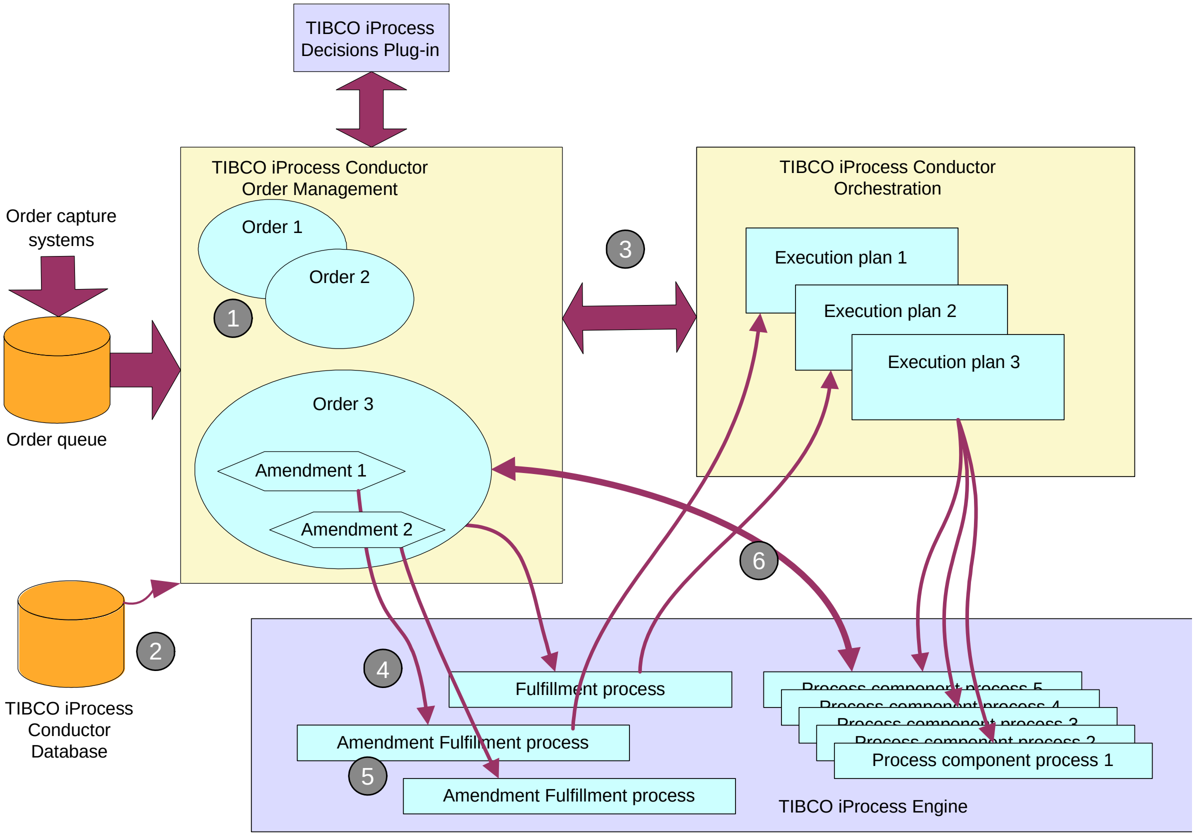
You can amend an order in iProcess Conductor whether or not the execution plan fulfilling the order has started processing. However, the process that iProcess Conductor Order Management follows is different in each case. This section describes:

- Amending an Order Whose Execution Plan Is Orchestrating on page 81
- Amending an Order Whose Execution Plan Has Not Started Processing on page 83.

The following sections refer to "amendment fulfillment process." Depending on how you have implemented TIBCO iProcess Conductor, this may be a case of your fulfillment procedure or a different procedure designed specifically to handle amendments. For more information, see *TIBCO iProcess Conductor Implementation*.

Amending an Order Whose Execution Plan Is Orchestrating

The following diagram illustrates what typically happens when an order amendment is submitted to iProcess Conductor Order Management when the execution plan fulfilling the order is currently orchestrating:


TIBCO iProcess Decisions Plug-in
TIBCO iProcess Conductor Order Management
TIBCO iProcess Conductor Orchestration
Order capture systems
Order queue
Order 1
Order 2
Order 3
Amendment 1
Amendment 2
1
2
3
4
5
6
Execution plan 1
Execution plan 2
Execution plan 3
TIBCO iProcess Conductor Database
Fulfillment process
Amendment Fulfillment process
Amendment Fulfillment process
Process component process 5
Process component process 4
Process component process 3
Process component process 2
Process component process 1
TIBCO iProcess Engine

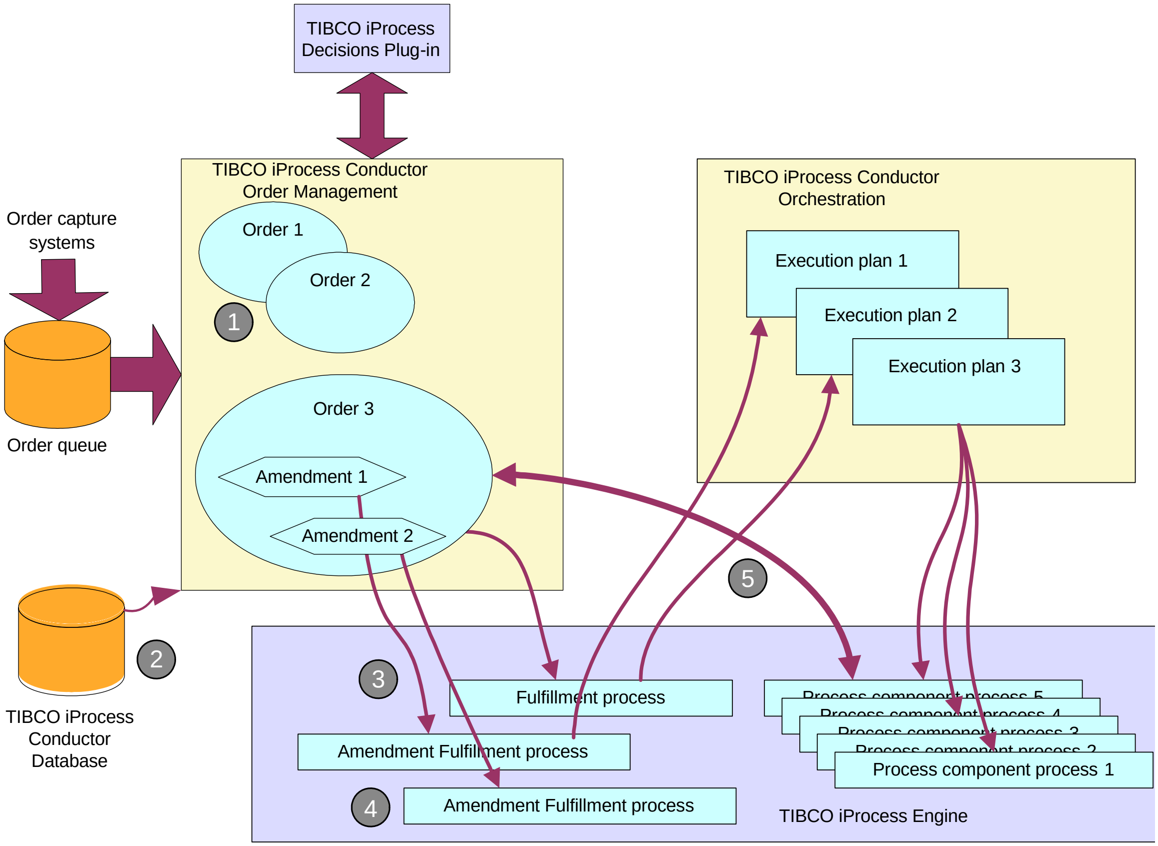
Each of the following numbered steps corresponds to the same number in the previous diagram.

1. iProcess Conductor Order Management determines that it has received an amendment to an existing order. It also determines that the execution plan fulfilling this order is currently processing.
2. iProcess Conductor Order Management updates the consolidated order view of the order in the database to reflect the amendment.

3. iProcess Conductor Order Management instructs iProcess Conductor Orchestration via a JMS Queue to suspend the execution plan associated with the order. It then submits the order amendment to TIBCO iProcess Decisions Plug-in. TIBCO iProcess Decisions Plug-in notifies iProcess Conductor Order Management of the fulfillment process and event step to be used and iProcess Conductor Order Management starts a case of the amendment fulfillment process in TIBCO iProcess Engine. See *TIBCO iProcess Conductor Implementation* for more information about fulfillment processes.
4. iProcess Conductor Order Management signals all the existing fulfillment processes associated with the order that a new order amendment has been received. The existing fulfillment processes associated with the order are expected to withdraw all actions and terminate, whether or not cases of the fulfillment processes have started. However, if the processes that are to fulfill the execution plan are currently being grafted on the graft step, they will not withdraw.
5. The amendment fulfillment process performs any feasibility checking defined in the process and triggers a manual execution plan development. This means that a work item is sent to the `COMUsers` work queue. When the work item is opened, the TIBCO iProcess Conductor user interface displays the order and the existing execution plan.
6. The amendment fulfillment process sends a message to reactivate the plan to iProcess Conductor via an EAI Orchestrator step. iProcess Conductor re-activates the updated execution plan.

Amending an Order Whose Execution Plan Has Not Started Processing

The diagram below illustrates what typically happens when an order amendment is submitted to iProcess Conductor Order Management when the execution plan fulfilling the order is not currently processing:


TIBCO iProcess Decisions Plug-in
TIBCO iProcess Conductor Order Management
Order 1
Order 2
1
Order capture systems
Order queue
Order 3
Amendment 1
Amendment 2
TIBCO iProcess Conductor Orchestration
Execution plan 1
Execution plan 2
Execution plan 3
5
2
TIBCO iProcess Conductor Database
3
Fulfillment process
Amendment Fulfillment process
4
Amendment Fulfillment process
Process component process 5
Process component process 4
Process component process 3
Process component process 2
Process component process 1
TIBCO iProcess Engine

Each of the following numbered steps corresponds to the same number in the diagram.

1. iProcess Conductor Order Management determines that it has received an amendment to an existing order. It also determines that the execution plan fulfilling this order has not yet started processing.
2. iProcess Conductor Order Management updates the consolidated order view of the order in the database to reflect the amendment.

3. iProcess Conductor Order Management submits the order request and information that this is an order amendment to TIBCO iProcess Decisions Plug-in to select a fulfillment process. TIBCO iProcess Decisions Plug-in notifies iProcess Conductor Order Management of the fulfillment process and event step to be used and iProcess Conductor Order Management starts a case of the amendment fulfillment process in TIBCO iProcess Engine.
4. The amendment fulfillment process determines that the execution plan has not started processing yet and processes the order amendment as a new order.
5. iProcess Conductor Order Management signals all the existing fulfillment processes associated with the order that a new order amendment has been received. The existing fulfillment processes are expected to withdraw all actions and terminate.

About the Consolidated View

iProcess Conductor Order Management stores a working version of the initial order request and any amendments applied to the order. This is called the consolidated view and can be viewed and edited from the TIBCO iProcess Conductor user interface. The following steps illustrate the application of an order request and an amendment to the consolidated view.

1. The order request is received:

Order Request
Header
Order Item 1
Order Item 2
Order Item 3

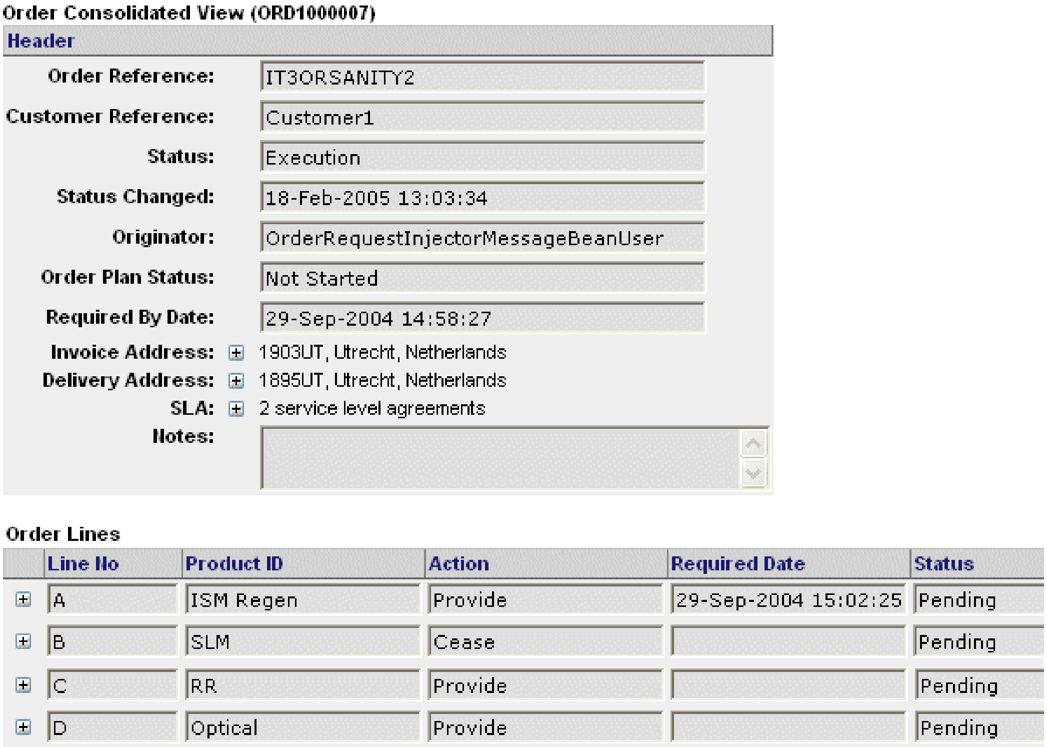
2. An order amendment is received. It contains amendments to Order Item 1 and a new order item (Order Item 4):

Order Amendment 1
Header
Order Item 1^1
Order Item 4

The data submitted as part of Order Item 1 will completely replace the data for Order Item 1 in the consolidated view, and Order Item 4 will be added:

Consolidated View
Header
Order Item 1^1
Order Item 2
Order Item 3
Order Item 4

You can view the consolidated view from the iProcess Conductor Order Management user interface:



Order Consolidated View (ORD1000007)

Header

Order Reference:	IT3ORSANITY2
Customer Reference:	Customer1
Status:	Execution
Status Changed:	18-Feb-2005 13:03:34
Originator:	OrderRequestInjectorMessageBeanUser
Order Plan Status:	Not Started
Required By Date:	29-Sep-2004 14:58:27
Invoice Address:	⊞ 1903UT, Utrecht, Netherlands
Delivery Address:	⊞ 1895UT, Utrecht, Netherlands
SLA:	⊞ 2 service level agreements
Notes:	

Order Lines

	Line No	Product ID	Action	Required Date	Status
⊞	A	ISM Regen	Provide	29-Sep-2004 15:02:25	Pending
⊞	B	SLM	Cease		Pending
⊞	C	RR	Provide		Pending
⊞	D	Optical	Provide		Pending

For more information about viewing execution plans, see *TIBCO iProcess Conductor User's Guide*.

Chapter 7 Understanding the Order Lifecycle

This chapter describes the lifecycle of an order in TIBCO iProcess Conductor.

Topics

Overview

An order goes through a number of state transitions from when it is submitted into the system through to its completion or cancellation. The transitions it goes through can be categorized into two stages:

- Design-time (draft/submit to execution plan development)
- Runtime (execution to completion, including orchestration of the execution plan)

The status of an order is changed:

- automatically by iProcess Conductor Order Management
- using an EAI step which must be defined in your fulfillment process
- by a user in the iProcess Conductor Order Management user interface.

The following sections describe the transitions of the order lifecycle and indicate which transitions are performed by iProcess Conductor Order Management automatically and which are performed by the fulfillment process via an EAI Order step.

At each stage in the lifecycle of an order, iProcess Conductor Order Management sets the order status.

Order Status

An order request can have the following types of statuses:

Status	Description
Draft	The order request is being developed.
Submitted	The order has been created and sent to iProcess Conductor Order Management. The Order Manager has stored the order in the database.
Plan Development	The fulfillment process has notified iProcess Conductor Order Management that it is at the stage where it is waiting for TIBCO iProcess Decisions Plug-in to select an execution plan template on which to base an execution plan.
Feasibility	The fulfillment process has notified iProcess Conductor Order Management that the order is at the stage where it is having its feasibility checked as part of the fulfillment process.
Feasibility Failed	The fulfillment process has notified iProcess Conductor Order Management that the order has failed the feasibility checking phase defined in the fulfillment process.
Execution	The fulfillment process has notified iProcess Conductor Order Management that the order request is currently being orchestrated and is executing.
Cancel	The order is currently being executed but the user has requested that the order is cancelled. The cancel operation is currently in progress.
Complete	The order has completed.
Cancelled	A cancel operation performed on the order is complete.
Order Suspended	The suspend operation on the order is complete.

Order Line Status

An order line can have the following types of statuses:

Status	Description
Cancelled	The cancel operation performed on the order line is complete.
Complete	The associated execution plan task or tasks have completed orchestration and the order line has completed.
Pending	The order line has not yet been grafted.

Execution Plan Status

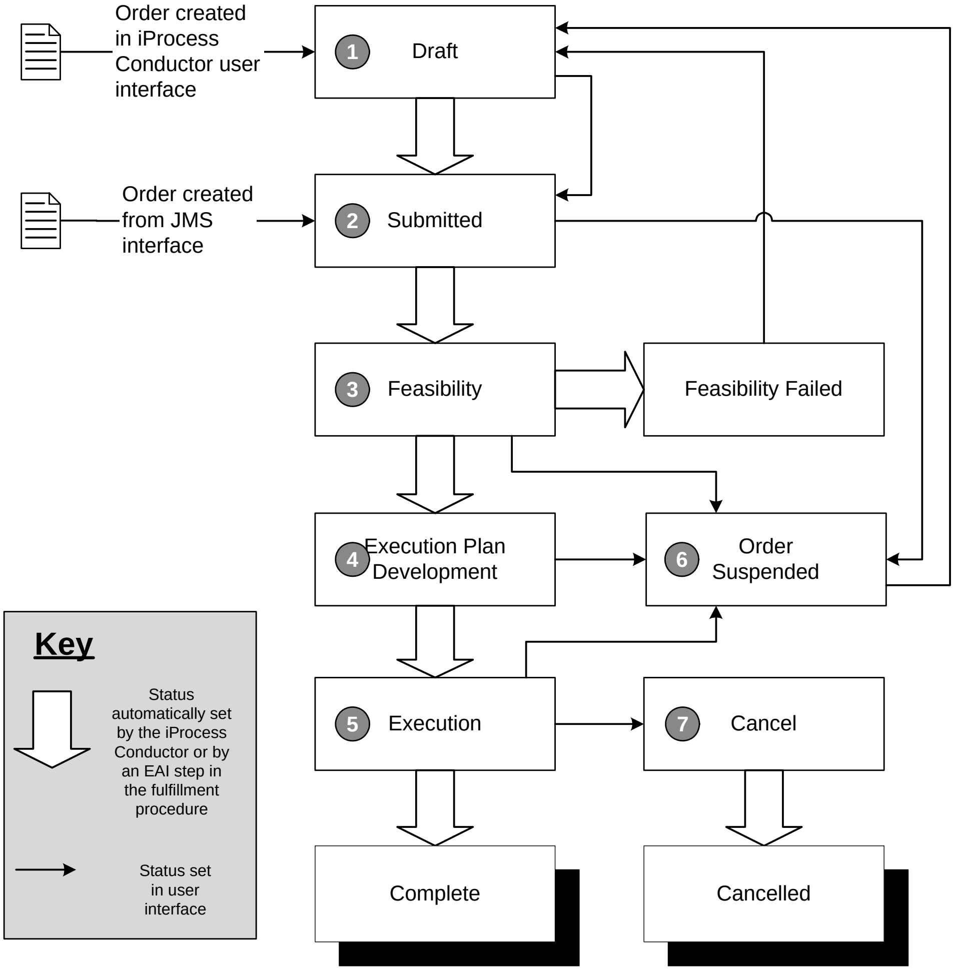
To maintain synchronization between the order lifecycle and the execution plan lifecycle (see Lifecycle of an Execution Plan on page 69) an order has a view of the associated execution plan status. This can be viewed from the TIBCO iProcess Conductor user interface. The table below describes the three additional status types:

Status	Description
Execution Plan Not Started	The execution plan has been created and is awaiting execution.
Execution Plan Active	The execution plan is currently being orchestrated and is executing.
Execution Plan Suspended	The execution plan was executing but the user has requested that the execution plan is suspended. The suspend operation on the execution plan is complete.

Order Lifecycle Example

This section describes a typical example of a lifecycle of an order. However, an order can follow different lifecycles. The diagram below illustrates all the different possible lifecycles an order can have:

Each number in the following diagram corresponds to the same number in the subsequent headings.


Order created in iProcess Conductor user interface
Order created from JMS interface
1 Draft
2 Submitted
3 Feasibility
Feasibility Failed
4 Execution Plan Development
6 Order Suspended
5 Execution
7 Cancel
Complete
Cancelled
Key
Status automatically set by the iProcess Conductor or by an EAI step in the fulfillment procedure
Status set in user interface

1 - Drafting an Order Request

An order request or order amendment can be drafted in iProcess Conductor Order Management. When you create an order in iProcess Conductor Order Management, its initial status is set to Draft. See *TIBCO iProcess Conductor User's Guide* for more information about drafting orders using the iProcess Conductor Order Management user interface.

2 - Submitting an Order Request

An order request or order amendment can be submitted from the iProcess Conductor Order Management user interface or from a CRM system. Once an order request or amendment has been received by iProcess Conductor Order Management, TIBCO iProcess Conductor does the following:

- It creates an order request and a consolidated view of the order and stores them in the database.
- It sets the status of the order to Submitted.
- It sends the order request to TIBCO iProcess Decisions Plug-in to identify what fulfillment process should be used.
- Once TIBCO iProcess Decisions Plug-in has identified the fulfillment process, iProcess Conductor Order Management starts a new case of the selected process.

3 - Feasibility Checking of an Order Request

Once the fulfillment process has started, it notifies iProcess Conductor Order Management that it is performing any feasibility testing that you have designed into your fulfillment procedure. To reflect this, iProcess Conductor Order Management sets the status of the order to Feasibility.

Although feasibility checking is optional, usually the TIBCO fulfillment process defined for use with TIBCO iProcess Conductor should contain some feasibility checking. The steps that need to be carried out as part of the feasibility checking depend on your business processes. For example, it may depend which service the customer has ordered whether or not credit checking is carried out. The fulfillment process must notify iProcess Conductor Order Management via an EAI Order step of the outcome of the feasibility checking.

If the order fails feasibility checking, the fulfillment process sets the status of the order to Feasibility Failed. If the status of the order is set to Feasibility Failed, then a work item is sent to the COMUsers work queue so that the order can be handled manually. See *TIBCO iProcess Conductor Implementation* for more information about defining feasibility checking in your fulfillment process.

4 - Developing an Execution Plan

The fulfillment process notifies iProcess Conductor Order Management that the order has passed the feasibility checking and has been sent to TIBCO iProcess Decisions Plug-in so that the execution plan template can be selected. At this point, iProcess Conductor Order Management sets the status of the order to Plan Development.

iProcess Conductor Order Management creates the execution plan from the template and associates the order with the execution plan. It stores them in the database.

If an execution plan template has not been selected, iProcess Conductor Order Management instigates manual execution plan development.

5 - Executing an Order

Once the fulfillment process has obtained the execution plan ID from the execution plan development sub-procedure, it progresses to the execution phase, where the status of the order is set to Execution and a call is made to the iProcess Conductor to start the plan. iProcess Conductor activates the plan and updates the execution plan status to Execution Plan Active.

6 - Suspending an Order

You can only suspend an order using the iProcess Conductor Order Management user interface. See *TIBCO iProcess Conductor User's Guide* for more information about suspending orders using the TIBCO iProcess Conductor user interface.

Once iProcess Conductor Order Management has received a request to suspend an order, it sets the status of the order to Order Suspended. It also sends a suspend message to all the fulfillment processes associated with the order. The existing fulfillment processes associated with the order are expected to tidy up any outstanding processes and pass control to the new fulfillment process.

7 - Cancelling an Order

An order request can be cancelled in iProcess Conductor Order Management. You can cancel an order as part of the process using an ad-hoc event step or you can cancel an order from iProcess Conductor Order Management. See *TIBCO iProcess Conductor User's Guide* for more information about cancelling orders using the TIBCO iProcess Conductor user interface.

Once iProcess Conductor Order Management has received a request to cancel an order, it sets the status of the order to Order Cancelled. It also sends a cancel message to each of the processes relating to the active tasks in the execution plan. These processes are then expected to withdraw all outstanding actions, rollback as appropriate and terminate.

When all of the processes grafted to an execution plan have stopped, iProcess Conductor Order Management sets the status of the plan to Cancelled.

Glossary

A

actual duration

The actual amount of time that a plan task takes to complete.

B

B2B gateway

Software that makes the products or services of two or more organizations available to each other in a controlled and secure manner.

C

classification

A way of grouping managed objects in the TIBCO iProcess Conductor in a way that is meaningful to your business (for example, by geographic location).

critical path

Represents the maximum length path through an execution plan. There are various types of critical paths that are calculated by the TIBCO iProcess Conductor to determine whether an execution plan is "in jeopardy." See **jeopardy management**.

CRM

Customer Relationship Management. Enterprise-wide software applications that allow companies to manage every aspect of their relationships with customers.

D

dependency

A relationship between milestones in an execution plan in the TIBCO iProcess Conductor. For example, Milestone B cannot start until Milestone A completes. See **milestone**.

E

execution duration

Used to calculate the critical path for an execution plan and obtained from the actual duration of the tasks completed and the typical duration of the tasks yet to complete.

execution plan

A collection of process components and their dependencies. Used by the TIBCO iProcess Conductor to manage the fulfillment of a given order.

execution plan task

This provides the control structure for a single process component with an execution plan.

execution plan task group

Collection of execution plan tasks that can be arranged in a hierarchical structure of dependencies.

execution plan template

This is an example execution plan that is used as the basis for activating execution plan instances.

J

jeopardy condition

A circumstance that causes an execution plan, or part of a plan, not to complete on time; or which puts it at risk of not completing on time. See also jeopardy management.

jeopardy consequential action

An action taken by the TIBCO iProcess Conductor as a result of a jeopardy condition. It could be to send a JMS message, initiate an iProcess Engine case start, or suspend the execution plan.

jeopardy management

The process of managing execution plan task progress and risk of failure. This is done by monitoring the critical paths through execution plans. See **critical path**.

JMS

Java Message Service. API messaging standard that allows application components based on the Java 2 Platform, Enterprise Edition (J2EE) to create, send, receive, and read messages.

M

maximum allowed duration

The maximum period of time that a process component version is allowed to take without jeopardizing the underlying business process to be considered. See also typical duration.

milestone

A significant scheduling point within the process components of an execution plan. Inter-plan task dependencies can be defined from and to Milestones. Milestones also provide points through which meta-data can pass to and from Process Components.

O

order

A request to fulfill a particular action. Typically orders request actions to provide, cease or update products or services for customers.

P

process component

A logical process controlled from a plan task within an execution plan. It maps to a single iProcess procedure at a particular date and time at run time. At design time, a process component may refer to one or more process versions. Each version will, in turn, reference a single iProcess procedure and an effective date/time range.

process component section

The interval between two milestones in a process component. This may be the whole of a simple process component that has only start and end milestones, or may be a part of a larger process component.

T

typical duration

The amount of time in which you would realistically like a process component or an execution plan to complete. See also maximum allowed duration.

Index