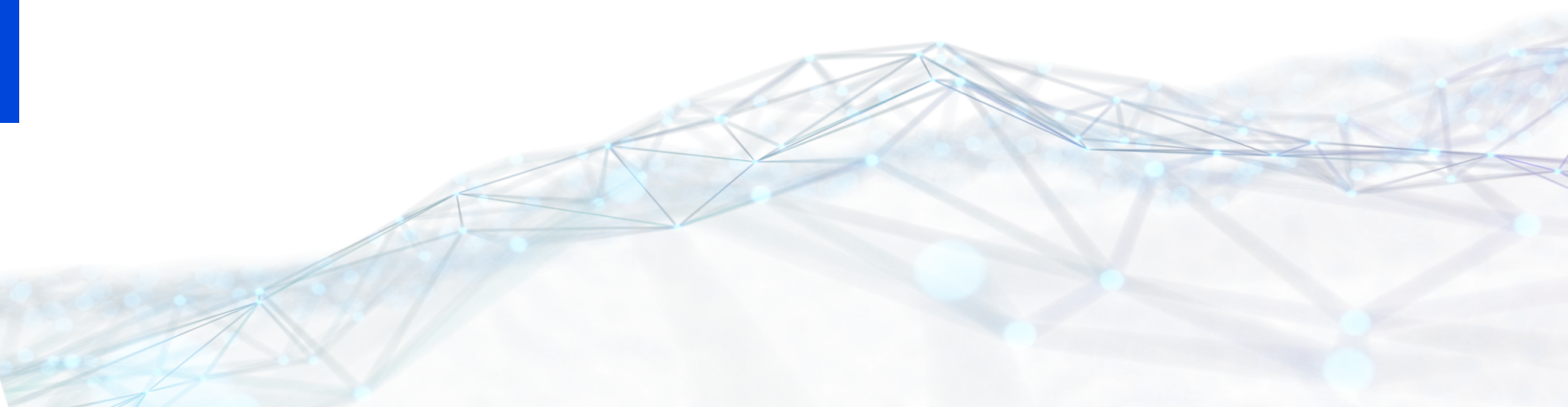




TIBCO® BPM Enterprise

Concepts Guide

Version 5.6.0 | November 2024



Contents

Contents	2
Introduction to TIBCO® BPM Enterprise	6
TIBCO BPM Enterprise Applications	6
TIBCO Business Studio™ - BPM Edition	7
TIBCO BPM Enterprise	7
TIBCO BPM Enterprise Architecture	7
TIBCO BPM Enterprise Reference Architecture	8
High Availability and Fault Tolerance - Clustered Operation	8
Developer Server	8
Shared Resources	9
TIBCO BPM Enterprise REST APIs	11
Client Application Development	11
Workflow Patterns Support	11
Model Driven Architecture and Development	12
Role-Based User Interfaces	13
A Single Design Time Environment	13
Flexible Runtime Clients	13
System Calendars	15
Base and Overlay Calendars	16
Time Zones	16
Exclusions in Overlay Calendars	17
Calendar References	17
Work Views	17
Work View Permissions	18
Creating a Work View for an Organizational Entity	18
Process Management	19

Standards-Based Process Notations	19
Process Design - Business Process Model and Notation (BPMN)	19
Data Type Support	19
Workflow Process and Data Pattern Support	20
Pageflow Processes	21
Business Services	21
Process Instance Migration	23
Example of Process Instance Migration	24
Organization Management	26
Modeling the Workforce	26
Organization Model	26
Organizations Organization Units and Positions	27
Groups	28
Resources (Users)	28
Privileges	29
Capabilities	29
Linking the Organization to the Organization Model	30
Use of LDAP and Dynamic Organization Models	32
Organization Model Partitioning	33
Organization Model Versioning	34
Example - Phase 1 Deploying an Initial Organization Model	37
Example - Phase 2 Making an Additive Update to the Model	40
Example - Phase 3 Making a Destructive Update to the Model	43
Example - Phase 4 Extending the Organization Model	46
Example - Phase 4 An Alternative Implementation	51
Work Management	55
Distributing Work to Users	55
Offering and Allocating Work	56
Participants	59
Dynamic Organization Participants	59

Push and Pull Distribution Models	61
Presenting Work to Users	61
TIBCO BPM Enterprise Client Applications	61
Work Lists	62
Work Items	62
Pageflows	63
Case Management	65
Creating Case Management Applications in TIBCO Business Studio - BPM Edition	67
Case Data Model	68
Interacting with Cases in Business Processes	68
Case Actions	69
Case Data Store	70
Case Manager	70
Administration of TIBCO BPM Enterprise	71
Platform Administration	71
Application Management	71
Organization Model and Resource Management	71
The tibco-admin User	72
System Calendar Management	72
Logging and Auditing	74
Managed Objects and Events	75
Measures	76
Logging	76
Logging Levels	76
Auditing	77
Correlation and Sequencing of Audit Data	77
Viewing Audit Trail	78
Security Features Provided by TIBCO BPM Enterprise	79
Authentication	79

Privilege-Based Authorization	80
System Actions	80
Scope of System Actions	82
System Actions and Organization Model Versions	83
Workflow Patterns Reference	86
Workflow Resource Patterns Support	86
Workflow Process Patterns Support	91
Workflow Data Patterns Support	98
TIBCO Documentation and Support Services	104
Legal and Third-Party Notices	106

Introduction to TIBCO® BPM Enterprise

This section provides a high-level overview of TIBCO BPM Enterprise concepts.

TIBCO BPM Enterprise Applications

TIBCO BPM Enterprise can be used to develop, deploy, execute, and manage *business process management applications*.

The term is used in this context to encompass applications that primarily automate the following processes:

- business processes (sequences of tasks) to be executed by human users, and
- technical processes (the flow of data) to be executed by enterprise applications

The TIBCO Business Process Management (BPM) comprises of:

- **TIBCO Business Studio™ - BPM Edition:** The design time to create and develop your BPM applications
- **TIBCO® BPM Enterprise:** The runtime to test and execute your BPM applications.

BPM applications include business processes, user-interaction forms, and processes that create and update data as per your requirements. A BPM application comprises of the following main components:

- **Business Processes:** Refer to a flow of tasks that can be performed in a defined sequence. Each task performs a specific automated action (service tasks), or allows a user to interact with the process and data (user tasks).
- **Process Instance:** Is the runtime instance of a business process definition with its own data scope.
- **Business Services:** Refer to a flow of user interaction tasks that is initiated by a user and defines a set of form pages that a user must work through and submit to perform some action. For example, starting a business process is a business service.
- **Case Data:** Is the description of an object that is of interest across entire TIBCO BPM

Enterprise. Some examples of a case data are an invoice, an expense claim, and a loss-adjustment. Cases are scoped externally to individual processes.

- **Case Actions:** Are similar to the business services, but are designed to create and update case data.
- **Work Items:** Is the runtime equivalent of a user task in a business process. When a user task is executed within a process instance, a work item is created and made available to the users who need to complete the task in their work list.

TIBCO Business Studio™ - BPM Edition

TIBCO Business Studio - BPM Edition provides a common modeling, implementation, and deployment environment for different types of applications.

Business analysts and solution designers can use the Eclipse-based design environment provided by TIBCO Business Studio - BPM Edition to perform the following tasks:

- Business analysts can capture, design and model all aspects of a business process, including the organization and data models that underpin it.
- Solution designers can implement the process as an executable application, then deploy the application to the TIBCO BPM Enterprise runtime for execution.

TIBCO BPM Enterprise

TIBCO BPM Enterprise is a Kubernetes-based product application that provides the runtime execution environment for business process management applications.

TIBCO BPM Enterprise Architecture

TIBCO BPM Enterprise has a multilayer architecture which provides flexibility and enables your business to adopt the configuration that suits its needs.

TIBCO BPM Enterprise Reference Architecture

TIBCO BPM Enterprise, like most enterprise software, is intended to be used in conjunction with a standard set of software and hardware infrastructure. It is not a software appliance, TIBCO® BPM Enterprise Concepts Guide 19 | Introduction to TIBCO® BPM Enterprise and therefore, cannot be installed on a machine and used in a safe and secure manner. Such an approach can be suitable for some development and "proof of concept" purposes. However, using TIBCO BPM Enterprise in this way in a production environment is not recommended. For this reason, a reference architecture is proposed that defines the kind of deployment that is expected for TIBCO BPM Enterprise in a production environment. For more information, see *TIBCO® BPM Enterprise Getting Started*.

High Availability and Fault Tolerance - Clustered Operation

TIBCO BPM Enterprise can be deployed to provide a high-availability, fault-tolerant configuration, using active-active clustering.

Developer Server

A Developer Server configuration of TIBCO BPM Enterprise consists of three docker containers that are managed using Docker Compose. It is intended for quickly performing development and testing on a developer's machine of a BPM application, which has been designed in TIBCO Business Studio - BPM Edition.



Warning: The Developer Server configuration is intended only for rapid development and testing purposes. It is not intended for use in a production environment and TIBCO recommends that you do not use it in a production environment.

The configuration is a simple one in which containers using the following components are all installed on the same machine:

- TIBCO BPM Enterprise
- Database Server

- LDAP Directory

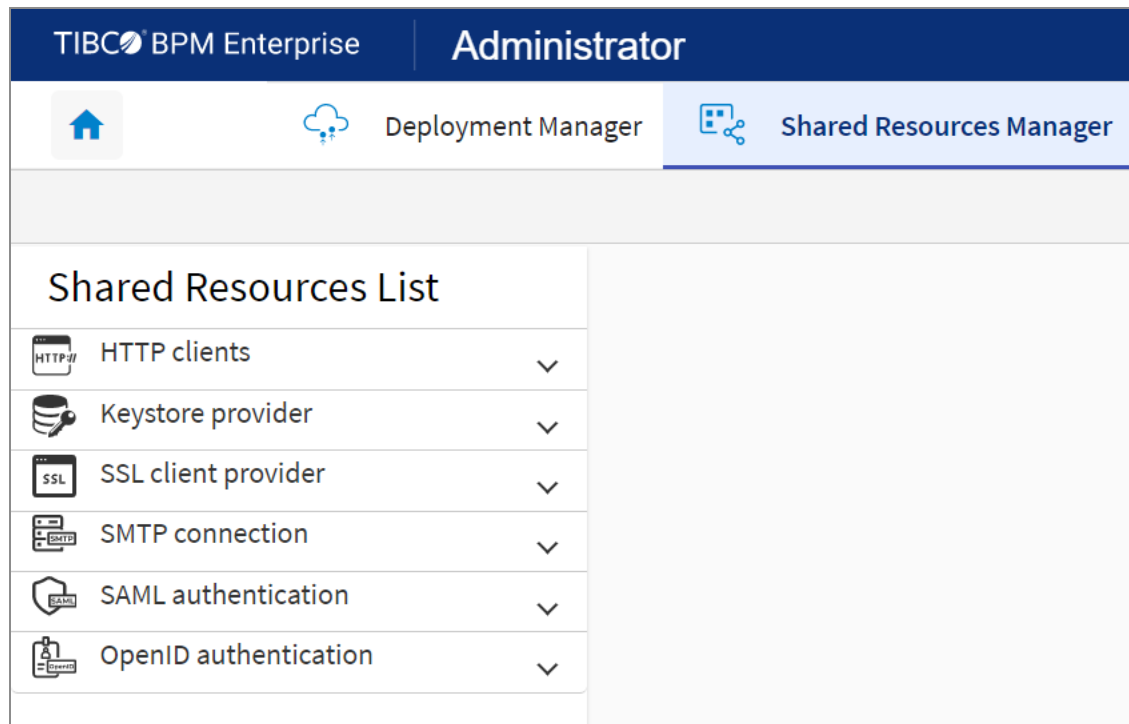
The Developer Server configuration has been designed to be easy to install on a developer's desktop or laptop machine, with minimal system requirements. As a result, the configuration has some limitations:

- All TIBCO software must be installed on the same machine because it cannot be accessed remotely in this configuration.
- Only *one* instance of Developer Server can be installed on a machine.
- Developer Server cannot be reconfigured after it has been installed.
- Developer Server cannot be used to provide a high-availability, fault-tolerant system, nor to provide specialization and scalability.
- An existing Developer Server cannot be upgraded, and hotfixes cannot be applied to it. To use a later version of Developer Server, you must uninstall the installed version, and then install the later version.

Shared Resources

Shared resources contain connection details for physical resources. Shared resources provide a way to use an identifier to reference the configuration of a resource, allowing multiple applications to use an instance of the shared resource without having to configure the resource multiple times.

There are several types of shared resource. Commonly-used ones include:



- HTTP Clients to provide connection to a REST service.
- Keystore providers provide a reference to a keystore that contains the keypair required for encryption.
- SSL client provider shared resources maintain credentials needed by SSL Clients.
- SMTP resources to provide connection to an SMTP mail server.
- SAML authentication resources are used for SAML Web Profile authentication, which allows users of your application to log in using a username and password issued by an Identity Provider (IdP) that supports SAML Web Profile.
- OpenID Authentication resources are used for OpenID Connect authentication, which allows users of your application to log in using a username and password issued by an Identity Provider (IdP) that supports OpenID Connect.

Additionally, there are two more shared resources, JDBC and LDAP connection shared resources. Both these need to be configured in the docker config files.

- JDBC resource instances for connection to an external database. These need to be configured as part of docker configuration.
- LDAP connections define the connection details of the LDAP directory you intend to use. These need to be configured as part of docker configuration.

TIBCO BPM Enterprise REST APIs

TIBCO BPM Enterprise exposes its functionality through comprehensive case management, work management and process management APIs, which are exposed as REST APIs.

For more information on REST APIs, see API Explorer in TIBCO BPM Enterprise.

Client Application Development

TIBCO BPM Enterprise provides Application Development to create, develop, and test custom client applications hosted in TIBCO BPM Enterprise.

Upload application files to Application Development, and then edit, test, and verify changes. For example, keep the service logic of the worklist, but completely change the appearance of the layout.

You can customize applications by adding a company logo, and incorporating the company's color scheme. See the "Customizing your Application" topic in the *TIBCO BPM Enterprise Administration Guide*. The custom application is available to users immediately after it is published.

You can delete the applications and download the application content in a .zip file.

Workflow Patterns Support

TIBCO BPM Enterprise implements many of the workflow patterns defined by the **Workflow Patterns initiative**, giving it the capability to handle a wide range of possible scenarios for business process modeling and execution.

The workflow patterns are grouped into the following perspectives that offer different ways of analyzing the workflow:

- **Resource:** Resource patterns capture the various ways in which resources are represented and utilized in workflows. For more information about how TIBCO BPM Enterprise supports these patterns, see Workflow Resource Patterns Support.
- **Control-flow:** Control-flow (or process) patterns capture the various ways in which activities are represented and controlled in workflows. For more information about how TIBCO BPM Enterprise supports these patterns, see Workflow Process Patterns

Support.

- **Data:** Data patterns capture the various ways in which data is represented and utilized in workflows. For more information about how TIBCO BPM Enterprise supports these patterns, see [Workflow Data Patterns Support](#).

For more information, see about the [Workflow Patterns initiative](#).

Model Driven Architecture and Development

TIBCO Business Studio - BPM Edition provides a model-driven development environment to create applications.

The following types of model are used:

Process model

the formal representation of a business process designed to be run by TIBCO BPM Enterprise. For example, a claims management, recruitment or car hire process.

Organization model

the formal representation of the organization against which the process will be run. For example, the EasyAs Insurance company.

Business object model

the formal representation of the business domain data that will be used by the process. For example, a customer record or an order line.

(optionally) user interface

the formal representation of how a particular user task in a process is to be displayed at runtime. For example, a TIBCO Form or a pageflow process.

(optionally) business service

a specific version of a pageflow process, used to provide an interactive "process starter" mechanism for users.

Role-Based User Interfaces

TIBCO BPM Enterprise provides tools that enable both non-technical and technical users to interact with the product in a manner and language that is natural to them and the role that they perform.

A Single Design Time Environment

TIBCO Business Studio - BPM Edition provides different perspectives for the business analyst, whose role is to identify and capture a business process as an abstract model, and the solution designer, who elaborates the process captured by the business analyst so that it can be executed in the TIBCO BPM Enterprise runtime.

Each perspective provides the appropriate language, tools and level of detail for each user, and hides those aspects with which the user is not concerned. For example, the business analyst's perspective allows them to define a task, but hides all details about how that task is to be implemented.

However, each perspective is not a separate design entity, which has to be integrated with the other. TIBCO BPM Enterprise uses (internally) a common model of the business process, and each perspective provides a view onto that model that is appropriate for the associated user role.

The use of a common model and different views or perspectives has the following benefits:

- It facilitates consistency. Changes in one perspective are automatically reflected in the other. Different parts of TIBCO BPM Enterprise do not work in isolation, therefore, do not require cumbersome crossover points.
- It provides familiarity when working with different parts of the system.
- It provides an improved "end-to-end" user experience.

Flexible Runtime Clients

TIBCO BPM Enterprise provides client applications that meet different end-user requirements. They provide administrative and management interfaces - see Administration of TIBCO BPM Enterprise.

Work Manager

Work Manager allows you to:

- Access and start business processes.
- Create a list of the work assigned to you and your team and view specific work items.
- View a history of business process activity.
- Create and access contributed apps.

Case Manager

Case Manager allows you to manage and update your case data. A case manager user interface helps case workers to view details, perform case actions, view case related work items, and upload case documents. A case data is structured data that is centrally managed by TIBCO BPM Enterprise and can be accessed and updated by multiple TIBCO BPM Enterprise process applications. Case data contain business related information that is required by an organization, like invoice, expense, claim, or policy.

Calendar

Calendar is a time-management tool that helps to schedule work, and calculate deadline based on information added, like working days, working times, holidays, lunches, and meetings.

The following information can be maintained in calendars:

- Working days and times: These are the number of working hours and days.
- Exceptions: These are exceptions to your normal working days and times. For example, a one-off exclusion like a company lunch, or exclusions that are repeated over a defined period, like a regular company meeting.
- Available working hours: These are defined as your working hours minus exceptions. For example, if your normal working hours are seven hours a day, but you have a two-hour company meeting scheduled on a particular day then, on that day, you have five available working hours.

Organization Browser

The Organization Browser can be used to:

- Browse organization models.
- Create LDAP containers that hold potential resources
- Map resources to groups and positions in the organization model
- Edit various organizational entity information

Administrator

Administrator is used to manage business processes and integrate custom apps into the environment.

- Upload and deploy business processes, organization models, data models, forms, and many others with **Deployment Manager**
- Set up shared resources for authentication, security, and HTTP clients with **Shared Resource Manager**
- Initiate, finish, and continue selected business processes with **Process Manager**
- Integrate custom applications into your Work Manager view with **Integrate Your App**
- Set up properties for your application with **Configuration Management**

Application Development

Application development enables users to upload web applications with static resources. You can add and delete the artifacts and applications. It provides appropriate editors to edit the resources. Application development also helps to manage different versions of applications and artifacts.

System Calendars

TIBCO BPM Enterprise supports calendars that maintain information about working times. You can have multiple calendars to define working hours and calculate deadlines for locations in different time zones.

Base and Overlay Calendars

Base calendars maintain basic information regarding working times (for example, hours in the standard working week), and *overlay calendars* define non-working time *exclusions* (for example, public holidays) which can be applied to a base calendar at runtime. For more information on base and overlay calendars, see *TIBCO BPM Enterprise Client User's Guide*.

Time Zones

In a multinational organization, scheduling needs to be done taking into account different timezones. In TIBCO BPM Enterprise, a base calendar is associated with a timezone.

The default base calendar uses the timezone set by the TIBCO BPM Enterprise server's operating system, but you can create other base calendars that use other timezones. Overlay calendars are not associated with timezones, but set their timezone according to whichever base calendar they are associated with. For more information, see *TIBCO BPM Enterprise Client User Guide*.

Deadlines and Localization

When calculating deadlines, start times are specified as actual UTC times. An examination of the organization model determines which base and which (if any) overlay calendar to use. Then, the available working hours are calculated from the information in the calendar entries. The combination of the start time, the duration of the work item, and the available working hours is used to calculate the earliest date and time at which the given work item can be completed.

Where more than one timezone is involved, the timezone to use for deadlines is determined locally. For example, assume that TIBCO BPM Enterprise offers a work item to the users in an organization unit, and that the calendar for that organization unit is based upon the London timezone. However, the work item being offered resides on a TIBCO BPM Enterprise node based in New York. The deadline calculation is based on the timezone and the working times of the London organization unit, and not on the New York timezone of the BPM node.

i Note: Deadlines of less than one hour are calculated without using calendars or invoking the Calendar services components. TIBCO BPM Enterprise simply adds the work item's duration to its start time.

For more information, see *TIBCO BPM Enterprise Client User's Guide*.

Exclusions in Overlay Calendars

Overlay calendars define exclusions to the normal working hours defined in the base calendar. While some exclusions are one-off, many will repeat over a defined period of time. For more information, see *TIBCO BPM Enterprise Client User's Guide*.

Calendar References

In TIBCO Business Studio - BPM Edition, you can assign a calendar reference to a business process, or a timer event within a process.

Calendar references (also called *calendar aliases*) are used to map calendars to organizational entities. For more information, see *TIBCO BPM Enterprise Client User's Guide*.

Work Views

Work views allow you to display your work items according to your requirements. You can set filters so that only work items that match specific criteria are displayed.

For example, you may only want to view work items of a certain priority or that have arrived today. You can also sort work items so that they are displayed in the order you want them to be displayed. Work views are created in Work Manager.

Public Work Views

You can specify whether or not a work view is public when you create it. If a work view is public every user can view the work view.

Note that:

- Users cannot automatically edit a public view unless they are the Owner or Author.
- If a public view is for a different organizational entity (in other words, not your work list), then you must be assigned the **View Work List** privilege for that organizational entity to load the view.

Temporary Work Views

When using the search feature in the Work View gadget, a temporary view is created for each search.

For example, searching for **id=123** would create a temporary view called **id=123** to allow the search results to be viewed. Temporary views are automatically deleted once you log out. You cannot make a temporary view permanent. You must create it using the Create new view wizard.

Work View Permissions

You can assign permissions to your work view when you create it using the work view wizard.

You can make your work views available to either of the following:

- individual users
- specific organizational entities. If you assign an organizational recipient permission to a work view, all the child organizational recipients in the parent recipient inherit that permission. However, the individual user must be assigned the **View Work List** system action for the child organizational entities.

Creating a Work View for an Organizational Entity

You can create work views that display work items offered to an individual resource or a particular entity in an organization model.

This is useful, if you are a supervisor and want to create work views of all the users that you supervise. However, you must be assigned the **View Work List** system action for the organization entity you require. You can create work views for the following roles:

- an individual resource
- an organizational entity such as an organization unit, group or position.

Process Management

This section describes the process management capabilities provided by TIBCO BPM Enterprise Process Manager.

Standards-Based Process Notations

TIBCO BPM Enterprise uses different standards-based notations to describe processes, depending on where they are being used.

Process Design - Business Process Model and Notation (BPMN)

BPMN is the de facto standard graphical notation for business process modeling. BPMN is designed for use by business users and is intended to be completely runtime-platform-independent.

This release of TIBCO Business Studio - BPM Edition (the TIBCO BPM Enterprise design environment) is based on the BPMN Version 2.0 specification (with some modifications).

For more information about BPMN Version 2.0, see the [Object Management Group/Business Process Model and Notation](#) website.

Data Type Support

TIBCO BPM Enterprise supports two main types of data (basic data or business data) in processes.

- **Basic data** - simple data types such as text, number, boolean, date and time.
- **Business data** - structured data that contains information about real-world entities that an organization deals with, for example, Customer, Order, Orderline. Each of

these entities will have a number of attributes for example, name, address, date. These objects are also connected to each other in different relationships and with different multiplicities.

Business data can also include derived types, which are derived from basic types. For example, an index type that can only contain positive integers in the range 10 to 20.

- **Case data** - business data that is centrally managed and can therefore be accessed and updated by multiple BPM process applications. Case data is modeled at design-time as case classes in a case data model, then represented at runtime as case objects, which can be referenced by corresponding case references.

Arrays of basic and business types are also supported.

At design time, these data types can be used in the process design. Example:

- fields and parameters in processes and forms
- in scripts
- input or output parameters on service tasks.

i Note: Business data and derived data types must first be defined (as business object model classes) using the TIBCO Business Studio - BPM Edition Business Object Modeler. A business object is an instantiation of a type defined in a business object model.

See the *TIBCO Business Studio - BPM Edition Modeling Guide* for more information.

At runtime, business data objects are managed and stored locally by TIBCO BPM Enterprise. (Business data objects are propagated through the system as copies - they are not passed by reference.)

Workflow Process and Data Pattern Support

Control-flow (or process) patterns capture the various ways in which activities are represented and controlled in process workflows, ranging from basic control patterns to advanced multi instance patterns.

Data patterns capture the various ways in which data is represented and utilized in workflows.

TIBCO BPM Enterprise provides in-built support for many of these patterns - see Workflow Patterns Reference - giving it the capability to handle the widest range of possible scenarios for modeling and executing processes and process data.

Pageflow Processes

A pageflow process is a special type of business process that can be used to provide an animated user interface for a single work item to the same user. For example, an animated user interface would include a sequence of forms instead of a single form.

See [Pageflows](#) for more information about how pageflows can be used to provide user interfaces.

A pageflow process can also include other activities - such as service or scripts and conditional logic - which can be used to drive the interaction with the user.

i Note: Unlike a normal business process, a pageflow process is *stateless* and *non-transactional*. If the process is not completed, any data set earlier in the process is lost. If a pageflow process performs a *stateful* action - something that cannot be reversed, such as writing to a database as part of REST API call or starting a process instance - it should be the final action performed by the pageflow process.

Business Services

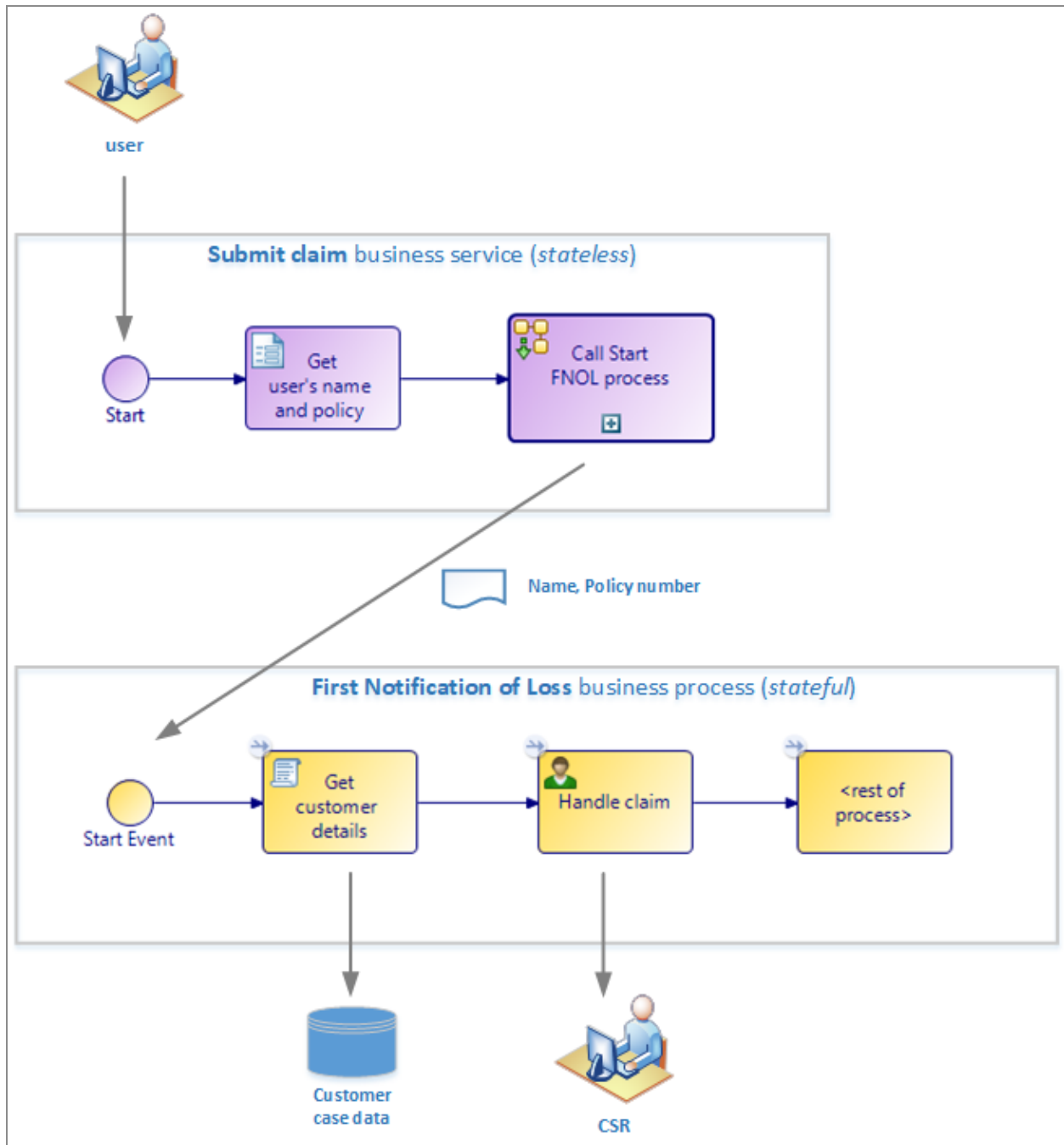
A pageflow process can be published as a business service.

A business service provides a user with direct access to a set of actions that accomplishes some sort of business function. A business service can (but does not have to) be used as a "process starter" mechanism to trigger an instance of a stateful business process.

The following Starting a Business Process example describes how business services can be used.

Example - Starting a Business Process

This example shows how a business service can be used to gather data and start an instance of a business process.



Procedure

1. A user selects and clicks the **Submit claim** business service. This starts the **Submit claim** business service.
2. The user immediately sees the **Get user's name and policy number** form, enters their name and policy number, then clicks **Submit**.
3. The **Start FNOL process** sends task starts an instance of the **First Notification of Loss** business process, passing the name and policy number as inputs to the process. The **Submit Claim** business service terminates.
4. The **First Notification of Loss** business process uses the received name and policy number to obtain the customer's full details from the customer database, via an API call, then generates a **Handle claim** work item for a Customer Service Representative, as the first step in processing the claim.

i Note: During the design phase of the **First Notification of Loss** business process, the business analyst or solution designer can automatically generate the **Submit Claim** business process - there is no requirement to design and implement it from scratch.

Process Instance Migration

Process instance migration is the ability to migrate a long running process instance to a different version of the process template from which it was generated. Process instance migration is controlled by the use of *migration points* and *migration rules*.

- A *migration point* is a task in the process template at which a process instance can be migrated to a different version. Not all tasks are valid migration points - for example, tasks that have a parallel path, or tasks that may have parallel executions due to multiple tokens flowing on a single path. Valid migration points are automatically identified by TIBCO Business Studio - BPM Edition at design time and are denoted by an icon next to the task in the Process Modeler.
- A *migration rule* defines when, how and to what version a process instance will migrate. A migration rule identifies the migration point in the process template from which a process instance will migrate, and the process template version to which it will migrate.

See "Design Considerations for Process Migration" in the *TIBCO Business Studio - BPM Edition Modeling Guide*.

At runtime, TIBCO BPM Enterprise checks if a migration rule is set when a task executes a process instance that is defined as a migration point. If a migration rule is set, TIBCO BPM Enterprise:

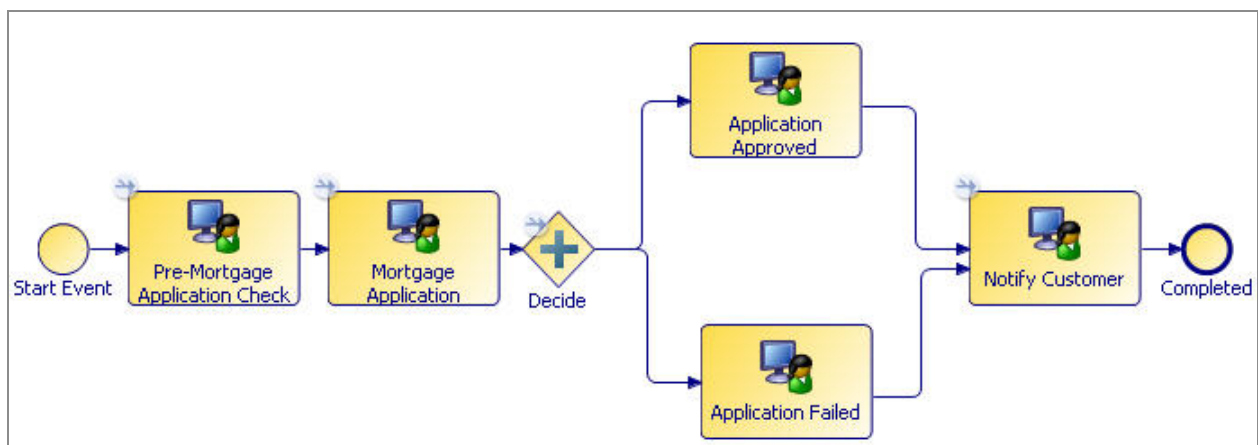
- migrates the process instance to the new version of the process template defined in the migration rule.
- continues execution of the task using the migrated to version of the process instance.

Process instance migration can be performed in the following ways:


- using Process Templates. For more information, see "Working with the Process Templates Gadget" in the *TIBCO BPM Enterprise Client User's Guide*.
- using the TIBCO BPM Enterprise REST API. For more information, see "Process Migration" in the *TIBCO BPM Enterprise Developer's Guide*.

Example of Process Instance Migration

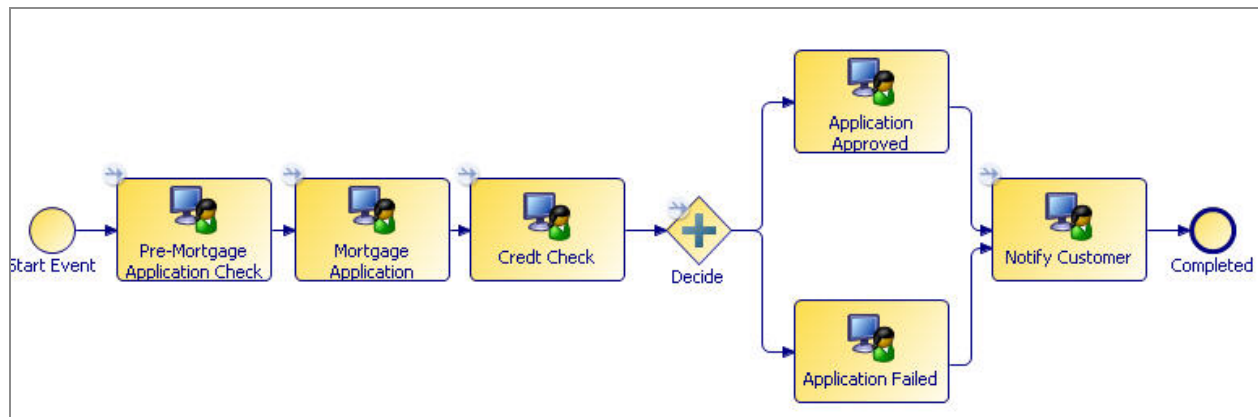
The following figure shows version 1 of a process.



In this process:

- an icon  next to a task shows that it is a valid migration point.
- **Pre-Mortgage Application Check**, **Mortgage Application Submitted** and **Decide** tasks are valid migration points.
- The tasks **Application Approved** and **Application Failed** following the **Decide** gateway are not valid migration points (as more than one task at a time could be active).

The following figure shows version 2 of the same process. A new task called **Credit Check** has been added to the process.



To migrate the process instances from Version 1 to Version 2, a migration rule is created that specifies the **Mortgage Application Submitted** task as the migration point.

i Note: A migration point must exist in both the source and destination versions of the process template. **Credit Check** is therefore not a valid migration point for migration between these two versions.

Note that:

- Process instances that are running against version 1 will migrate to Version 2 when they have finished executing the **Pre-Mortgage Application** task but not yet started to execute the **Mortgage Application Submitted** Task.
- Process instances that are already running the **Mortgage Application Submitted** task will continue with Version 1 of the process.

This means that all process instances that have started executing against Version 1 of the process template will migrate to Version 2 unless they have reached the **Mortgage Application Submitted** task.

Organization Management

This section describes the organization management capabilities provided by TIBCO BPM Enterprise Work Manager.

Modeling the Workforce

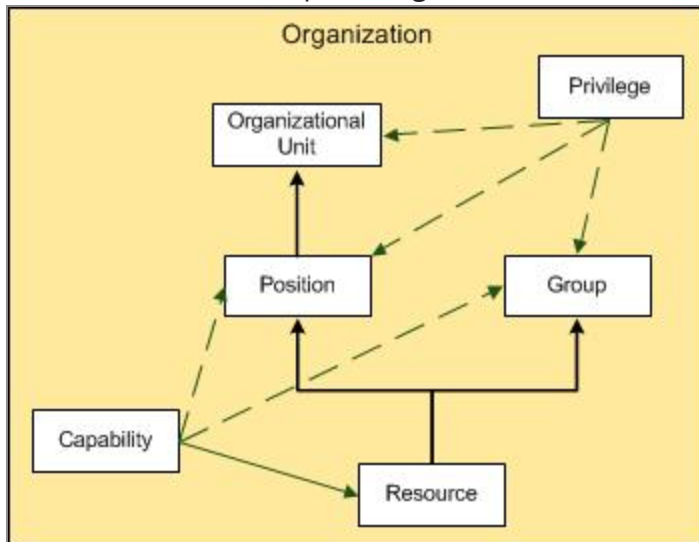
Every enterprise uses an organization model. This is typically a hierarchical structure that identifies the resources - people - who work for the organization, their *structural* relationships to one another and (possibly) other data about the resources. The model is typically maintained in one or more enterprise directories.

TIBCO BPM Enterprise uses the enterprise's own organization model as the basis for work distribution. TIBCO BPM Enterprise maintains its own model of the enterprise's organizational structure. This organization model is based on the enterprise directory, augmented with additional business process management-related information required to manage work for users, such as, an employee's role (their job title or function), technical capabilities and so on.

Organization Model

The main elements used in the organization model maintained by TIBCO BPM Enterprise are: Organization Unit, Position, Resource, Capability, Group and Privilege.

Main TIBCO BPM Enterprise Organization Model Elements



Organizations Organization Units and Positions

Organizations, organization units and positions are *structural* elements that define the organizational structure of the enterprise:

- *Organizations* denote the overall container for an organizational hierarchy. Typically this means a company.
- *Organizational units* represent structural associations of people in the context of the organization. Organizational units can represent traditional, hierarchical entities such as a division, department or team. They can also represent functional or ad-hoc groupings, such as committee, a task force, a project management organization, a class (for education) and so on.
- *Positions* define the membership roles of an organization unit. For example, a Customer Services department may define the positions of manager, team leader and customer services representative. A position can be filled by any number of human resources.

Organizations, organization units and positions are defined in TIBCO Business Studio - BPM Edition's Organization Modeler. Once defined, they can be used as process participants to define who a user task should be distributed to.

Groups

Groups define the work that a particular group of users are capable of doing. They provide a *functional* view of the organization, which is separate from the enterprise's formal and structural organization.

For example, groups can define:

- a job title or function, such as Customer Service Representative or Loss Adjuster.
- a technical skill set, such as Java software architects.

Groups can also be related in a hierarchical, tree-like structure (like structural elements) that refines the nature of the group as it deepens. In other words, a sub-group can be created from a parent group, or both co-exist. All members of a sub-group are members of the parent group. For example, an insurance company could use a general Customer Services Representatives group, with sub-groups for those CSRs who specialize in motor or travel insurance. Specialization may be on the basis of location, skills or the ability to speak a particular language.

Groups are defined in TIBCO Business Studio - BPM Edition's Organization Modeler. Once defined, they can be used as process participants to define who a user task should be distributed to.

Resources (Users)

Resources represent users who are explicitly named individuals among whom work items can be distributed. Information on users is held in the LDAP-compliant corporate directories used by the enterprise.

Unlike other elements in the organization model, users are not defined in TIBCO Business Studio - BPM Edition's Organization Modeler. Instead, users are added to TIBCO BPM Enterprise at runtime from the enterprise's LDAP-compliant corporate directories. They can then be assigned to positions and/or groups. A user can be assigned to many groups/positions, and a group/position can be associated with many users.

Once added to the system, users can access TIBCO BPM Enterprise by logging in to a TIBCO BPM Enterprise client application. They can access their own work list, which contains all the work items assigned to them. The privileges that they hold determine what parts of the system they can see and what they can do.

Privileges

Privileges are authorizations allocated to a user with respect to applications or functionalities within TIBCO BPM Enterprise.

For example, a user can be assigned privileges to perform the following actions:

- approve expense claims
- start process instances or business services
- see the work lists of other users

Privileges can be further qualified - for example, to be authorized to approve expense claims up to a limit of \$1000.

Using TIBCO Business Studio - BPM Edition's Organization Modeler, privileges can be defined and then assigned to groups, organization units and positions.

i Note: Users are not assigned privileges directly - instead, they inherit privileges based on their membership of groups, organization units and positions.

Privileges can be used to control what a user can do in the following ways:

- They can be assigned to system actions. These are tasks, such as re-allocating or skipping work-items, that a user may want to perform that might need to be authorized in some way. Only users who hold that privilege are then allowed to execute that system action. See System Actions for more information.
- They can be assigned to user access sets, which are used to control access to different components of the user interface.
- They can be used as process participants to define the user to whom the task should be distributed to. For example, a work item to issue a payment claim could be distributed only to users who have the privilege to sign off final payments.

Capabilities

Capabilities refer to abilities, skills, or aptitudes of a user. For example, a user might have any of the following capabilities:

- speak a foreign language

- hold professional qualifications in a particular area of business expertise
- hold a driving license

Capabilities can be further qualified - for example, to be able to speak French, German or both.

Using TIBCO Business Studio - BPM Edition's Organization Modeler, capabilities can be defined and then assigned to groups and positions.

i Note: Users do not inherit capabilities based on their membership of groups and positions. Instead, capabilities must be assigned directly to users at runtime using the Organization Browser.

Capabilities can be used in the following ways:

- They can define the entry criteria for a group or position. At runtime, an administrator can ensure that they only add users to particular groups or positions who have the necessary skills. For example, this can include the capability to add users who hold a particular qualification to a group of specialist motor claims handlers.

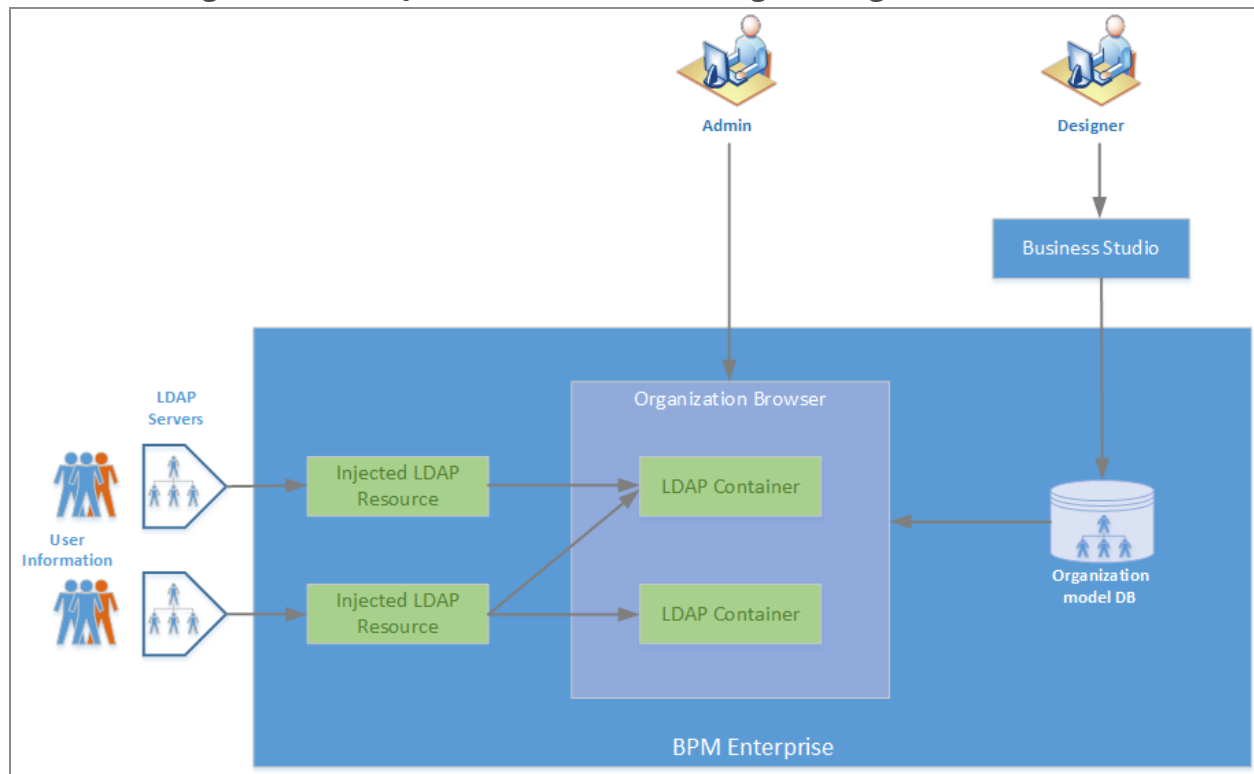
i Note: TIBCO BPM Enterprise does not enforce any defined entry criteria. The administrator can, if they want, add users who do not have the necessary skills to a group or position.

- They can be used as process participants to define whom a user task should be distributed to. For example, in an insurance claims process, a work item to capture initial claim details could be distributed only to call handlers who speak the language requested by the person reporting the claim.

Linking the Organization to the Organization Model

This section covers the steps involved in linking an organization to the organization model.

Diagrammatic Representation for Creating the Organization Model



Procedure

1. At design-time, the business analyst and solution designer create an organization Linking the Organization to the Organization Model and Linking the Organization to the Organization Model it to TIBCO BPM Enterprise.

i Note: The design-time model is an abstract view that does not include resources (users).

2. At runtime, an administrator uses the Organization Browser to perform the following tasks:
 - a. Create *LDAP containers*. An LDAP container defines a collection of *LDAP sources*, which are aliases for connections to LDAP-compliant corporate directories. These directories contain details of potential resources (users) who may need to use or participate in TIBCO BPM Enterprise applications. For details, see **Configure the LDAP Directory Server** in *TIBCO BPM Enterprise Installation*
 - b. Add resources (users) to TIBCO BPM Enterprise from the LDAP containers.

- c. Assign those resources to positions and/or groups in the deployed organization model.

For more detailed information about these steps, see the following references.

Topic	Reference
Creating and deploying an organization model	<ul style="list-style-type: none"> • <i>TIBCO Business Studio - BPM Edition Modeling Guide</i> • <i>TIBCO Business Studio - BPM Edition Application Designer's Guide</i>
Creating LDAP Connection and LDAP Authentication shared resources/ applications	<ul style="list-style-type: none"> • <i>TIBCO BPM Enterprise Installation</i> • <i>TIBCO BPM Enterprise Administration</i>
Add resources and assign them to positions and groups	<ul style="list-style-type: none"> • <i>Client User's Guide</i>

Use of LDAP and Dynamic Organization Models

From an Organization Unit extension point (places in a static organization model where you want the Dynamic Organizations to appear) you need to perform an LDAP query.

The extension point is the holder of the LDAP Query configuration, and the point within the model when Dynamic Organization Model instances resulting from that LDAP Query will be located and assigned. The LDAP Query is the search filter, used to identify LDAP entries by their attribute values. An attribute commonly used within these queries is `objectClass`: for example, where `objectClass = organizationalUnit`.

The LDAP Alias (LDAP Connection Shared Resource), Base-DN and Search Scope properties determine the start position and depth of the search within the LDAP Directory.

There is a named LDAP attribute from which the name of the Dynamic Organization Model instances will be derived.

You need to map the Dynamic Organization Identifier fields to named LDAP attributes. The values of the named LDAP attributes, taken from the LDAP entry from which each instance is generated, will be used, by a process task, to uniquely identify the instance as a

Participant of that task. It is, therefore, important that these named LDAP attributes hold values unique to each instance.

Any Base-DN, specified in the extension point configuration, will be appended to any Base-DN applied to the URL of the selected LDAP Connection Shared Resource.

From each LDAP entry, returned by the LDAP Query, the values of the named LDAP Attribute will be read; and, for each value, a new instance of the Dynamic Organization Model will be created. Also, for each new Dynamic Organization Model instance, the values of the LDAP Attributes mapped to the Dynamic Organization Identifier fields will be read from the same LDAP entry and recorded against that instance.

See [Dynamic Organization Participants](#) for a graphic showing an Organization Unit extension point where you want the Dynamic Organizations to appear in an Organization Model.

i Note: In TIBCO Business Studio - BPM Edition you usually allocate work to a position. With a Dynamic Organization all that exists is the template. The Dynamic Organization Identifier is used to map LDAP attributes (Attribute/LDAP attribute). This is likely to be the same LDAP Attribute from which the Dynamic Organization Model instance derives its name. See "Dynamic Organization Identifier Mapping" in the *TIBCO Business Studio - BPM Edition Modeling Guide*.

Organization Model Partitioning

TIBCO BPM Enterprise's approach to organization modeling provides a powerful and flexible way of modeling an organization.

Using TIBCO Business Studio - BPM Edition, analysts and developers can produce and use organization models that provide a view of the organization tailored to the needs of their specific application. An organization model can encompass multiple organizations, the whole organization or just specific parts of it, as required.

An organization model can contain multiple organizations. In this case, it may be desirable for privacy reasons to ensure that users can only browse, edit and allocate work within the confines of the organization or organizations to which they have been given access.

An LDAP container can be associated with zero, one or more organizations. When an LDAP container is associated with an organization, a user derived from that LDAP container can only access (or be assigned to) organizations that are either associated with the same LDAP

container, or not associated with any LDAP container. (The user cannot access organizations that are associated only with other LDAP containers.)

Organization Model Versioning

Versioning is used to control the interaction of different organization models. When an organization model is deployed, TIBCO BPM Enterprise manages all updates (additions, deletions and changes) and resolves any conflicts caused by those updates.

Design Time

At design time, in TIBCO Business Studio - BPM Edition:

- A project containing an organization model is given a version number, of the form *major.minor.micro.qualifier*.
- If a process participant is defined as an external reference to that organization model, the process definition records (internally) the *major* version number of the referenced organization model. All references within a project must be to the same major version of the organization model.

i Note: If a project contains only additions to an existing runtime organization model, the version number may be changed as required.

If a project contains a *destructive change* to an existing runtime organization model, the project's major version number *must* be incremented. A destructive change is one that changes, or is intended to remove, an existing organization model entity. For example:

- deleting a position (as opposed to simply not including it in the project - which is a *partial deployment*)
- changing the name of an organization unit

If a destructive change is made and the major version number is *not* incremented:

- If an organization model entity has been changed, when the project is deployed TIBCO BPM Enterprise treats this as a *destructive change* to the referenced *major* version of the organization model and raises an error. The deployment is rejected and no changes are made to the organization model.
- If an organization model entity has been deleted, when the project is deployed TIBCO BPM Enterprise treats this as an *additive update* to the referenced *major* version of the organization model. The deployment is accepted, but assumed to be only a partial deployment, and the entity is *not* deleted from the organization model.

Deployment

When an application that contains an organization model is deployed, TIBCO BPM Enterprise either creates a new runtime organization model version or modifies an existing one:

- An organization model with a new major version number is treated as a complete and separate runtime organization model.
- An organization model with a major version number that already exists is treated as an *additive update* to the existing runtime organization model with that major version number.

i Note: An additive update does not need to contain all the entities defined in the original organization model of the same major version. (This is called *partial deployment*.) Any organization model entities not defined in the additive update remain in the model, and are not deleted.

- By default, the *qualifier* part of the version number, becomes a timestamp that records the date and time when the application was deployed.

When you *undeploy* an organization model, the model's organization entities are removed immediately if the same organization entities are also not a part of another organization model deployment within the same major version. This happens even if another organization model, of the same major version, was deployed after that organization model.

For example, deploy three organization models to the same major version: organization models v1.2.0 and v1.1.0 are deployed in that order. V1.2.0 contains entities 'a', 'b' and 'c', and v1.1.0 contains entities 'a', 'b' and 'd'. If v1.2.0 is undeployed, only entity 'c' would be removed, as 'a' and 'b' are also in v1.1.0. Minor version numbers do not affect the order of deployment or undeployment.

Runtime

At runtime:

- When an application containing an organization model is deployed, TIBCO BPM Enterprise treats it as an upgrade if its name matches that of an existing application. (The platform ignores the project's version number when determining whether the application is new or an upgrade.)

The platform deploys the new version and deletes the existing version of the application. (This has implications for existing process instances.)

- TIBCO BPM Enterprise manages deployed organization model artifacts, and combines them to build up the runtime organization model.

User Application Dependencies

At runtime:

- A process application has a dependency on the application that contains the referenced organization model. The dependency can be resolved by a version of that

application that has the same major version number, and a later minor or micro version number.

- A user task in a process runs against the major version of the runtime organization model that is referenced in the process definition.

These dependencies must be considered for the following issues:

- naming conventions and version numbering schemes to be used for organization models.
- the impact of upgrading an organization model - both on dependent process applications and on the runtime organization model.
- the impact of deleting an organization model - either directly, or indirectly as a result of an application upgrade - both on dependent process applications and on the runtime organization model.

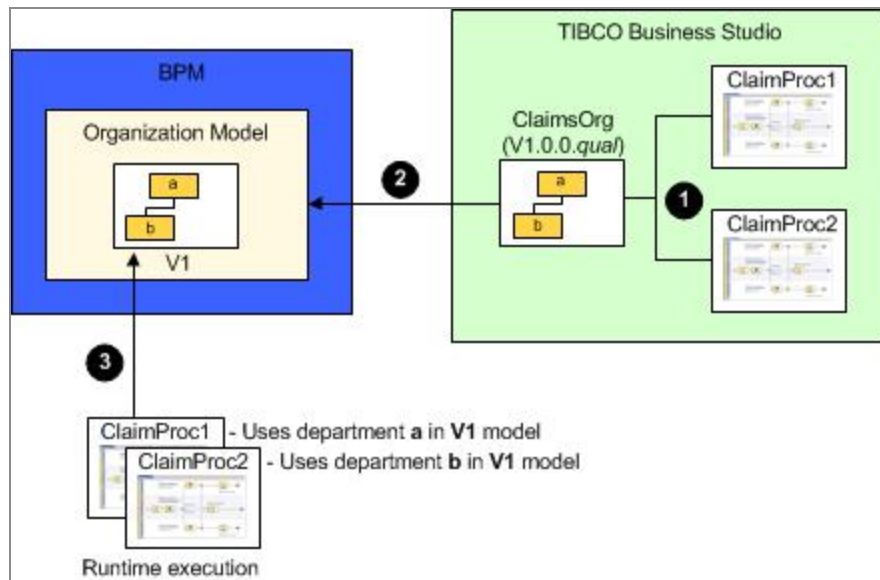


Note: To upgrade an organization model to a new major version, you should use the same application name only if the previous version is no longer needed.

If the previous organization model version is (or may be) still required, deploy the new version as a new application (that is, using a different application name). Both versions of the runtime organization model will then continue to be available. Deployed processes can then be either left to run against the old version or upgraded to use the new one as required.

Example - Phase 1 Deploying an Initial Organization Model

EasyAs Insurance is rolling out an implementation of a new application, starting with the Customer Services division.



The initial implementation defines two applications:

- **ClaimProc1** involves department **a**.
- **ClaimProc2** involves department **b**.

Design Time

The solution designer:

- (1) Produces **V1.0.0.qual** of the **ClaimsOrg** organization model (in its own project).
- (2) Adds participants from department **a** to **ClaimProc1**, and from department **b** to **ClaimProc2**. Each participant is defined as an external reference to the **ClaimsOrg** organization model.

ClaimsOrg's major version number (1) is recorded (internally) in the **ClaimProc1** and **ClaimProc2** process definitions as part of the participant definitions.

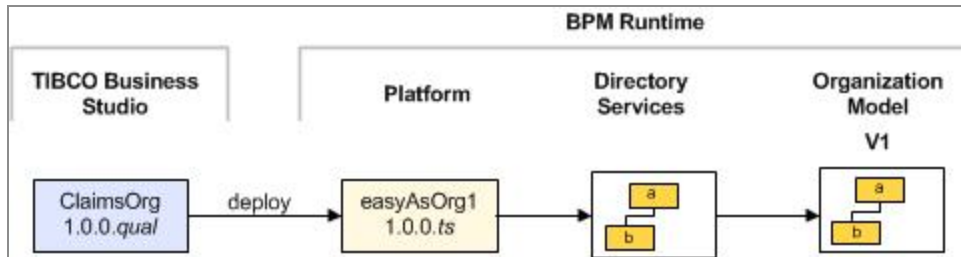
Deployment

The solution designer:

- (1) Deploys the **ClaimsOrg** organization model as the **easyAsOrg1** application, version **1.0.0.ts**.
- (2) Deploys the **ClaimProc1** and **ClaimProc2** projects as the correspondingly named applications, version **1.0.0.ts**.

Runtime

As the deployed ClaimsOrg organization model has a new major version number, TIBCO BPM Enterprise creates it as a new runtime organization model.



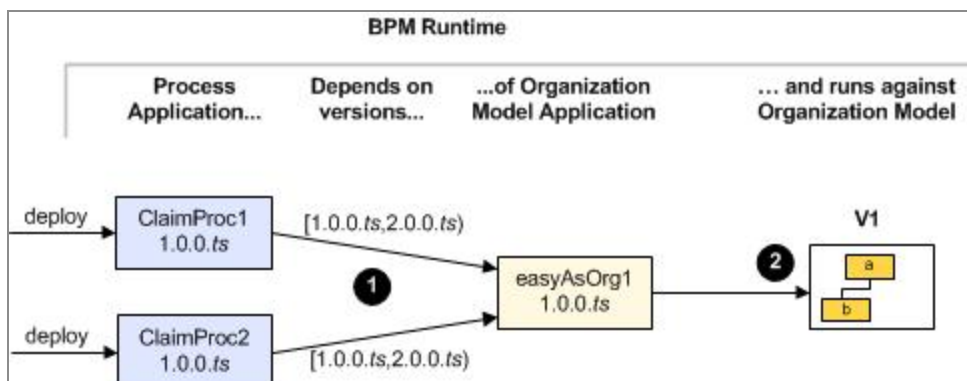
Directory Services uses the deployment artifact from the **1.0.0.ts** version of the **easyAsOrg1** application to create a new **V1** organization model.

User Application Dependencies

i Note: In the following diagram, application dependencies are shown using the convention $[Min, Max)$:

- *Min* is the version referenced when the process application is initially deployed. The square bracket denotes that the value is inclusive.
- *Max* is the next major version. The round bracket denotes that the value is exclusive.

For example, $[1.0.0.ts, 2.0.0.ts)$ denotes any version from 1.0.0.ts (inclusive) to 2.0.0.ts (exclusive). (*ts* is not used in the comparison.)

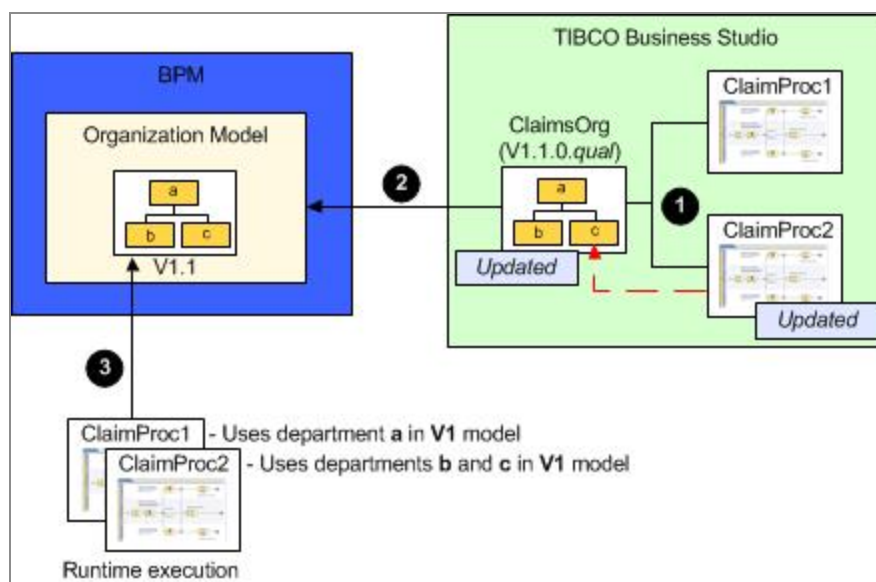


ClaimProc1 and ClaimProc2 both have a dependency on the **easyAsOrg1** application, version 1.0.0.ts (inclusive) to version 2.0.0.ts (exclusive).

Also, ClaimProc1 and ClaimProc2 both execute against V1 of the runtime organization model.

Example - Phase 2 Making an Additive Update to the Model

Following some user testing, EasyAs decide that they need to change one of the processes to involve an additional department, **c**.



Design Time

The solution designer:

- (1) Adds department **c** to the ClaimsOrg organization model.
- (2) Changes the ClaimsOrg project's version number to 1.1.0.ts. This is an extension to the existing organization model, therefore, the version number can be changed on the *minor* or *micro* level.)
- (3) Modifies ClaimProc2 to use participants from departments **b** and **c**.
- (4) Leaves ClaimProc1 unchanged, as it still just uses participants from department **a**.

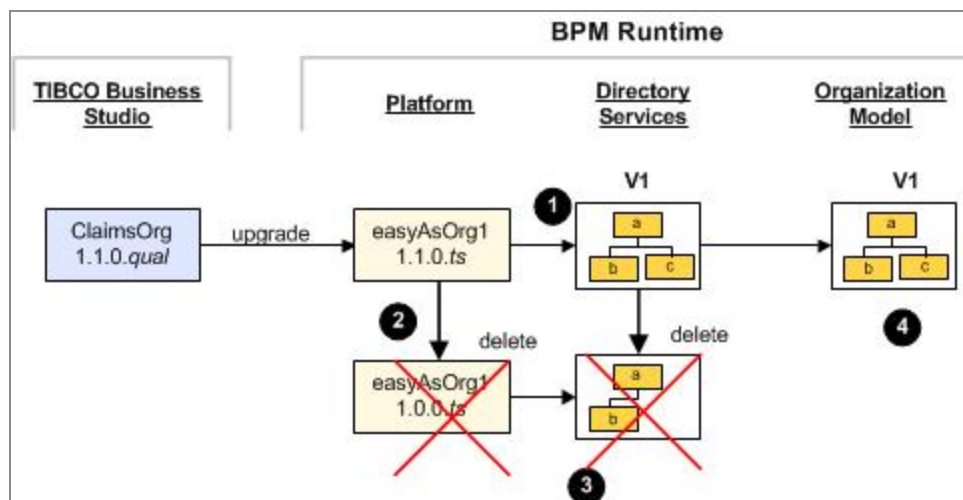
Deployment

The solution designer:

- (1) Deploys the ClaimsOrg organization model as an upgrade to the easyAsOrg1 application, as version 1.1.0.ts.
- (2) Upgrades the ClaimProc2 application to version 1.1.0.ts.

Runtime

TIBCO BPM Enterprise merges the changes from the deployed ClaimsOrg organization model - the addition of department c - into the existing V1 runtime organization model.



- (1) Directory Services adds the deployment artifact from the **1.1.0.ts** version of the **easyAsOrg1** application to the **V1** organization model.
- (2) The platform deletes the **1.0.0.ts** version of the **easyAsOrg1** application.

i Note: The deployed application has the same name as an existing application. The version number is irrelevant.

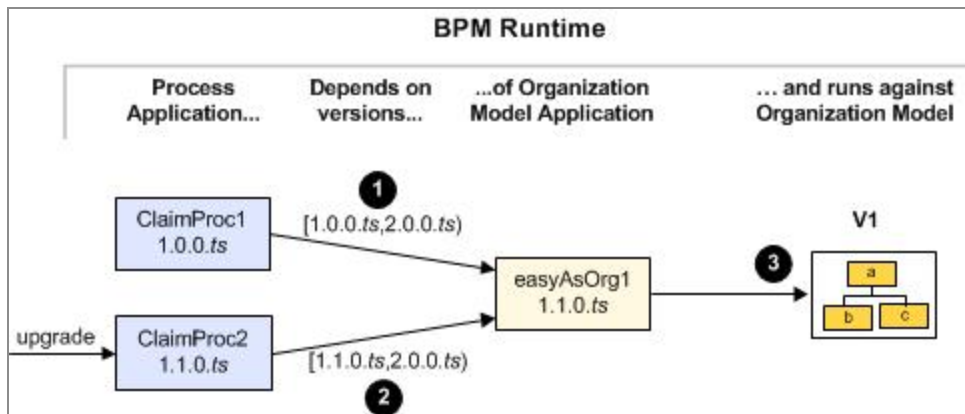
- (3) Directory Services deletes the deployment artifact from the **1.0.0.ts** version of the easyAsOrg1 application.
- (4) Directory Services uses the deployment artifact from the **1.1.0.ts** version of the easyAsOrg1 application as the V1 organization model.

User Application Dependencies

i Note: In the following diagram, application dependencies are shown using the convention $[Min, Max)$:

- *Min* is the version referenced when the process application is initially deployed. The square bracket denotes that the value is inclusive.
- *Max* is the next major version. The round bracket denotes that the value is exclusive.

For example, $[1.0.0.ts, 2.0.0.ts)$ denotes any version from 1.0.0.ts (inclusive) to 2.0.0.ts (exclusive). (The *ts* is not used in the comparison.)



(1) ClaimProc1 is not upgraded and still depends has a dependency on the **easyAsOrg1** application, version 1.0.0.ts (inclusive) to version 2.0.0.ts (exclusive).

(2) ClaimProc2 has a dependency on the **easyAsOrg1** application, version 1.1.0.ts (inclusive) to version 2.0.0.ts (exclusive).

(3) ClaimProc1 and ClaimProc2 both execute against V1 of the runtime organization model (which now includes departments **a**, **b** and **c**.)

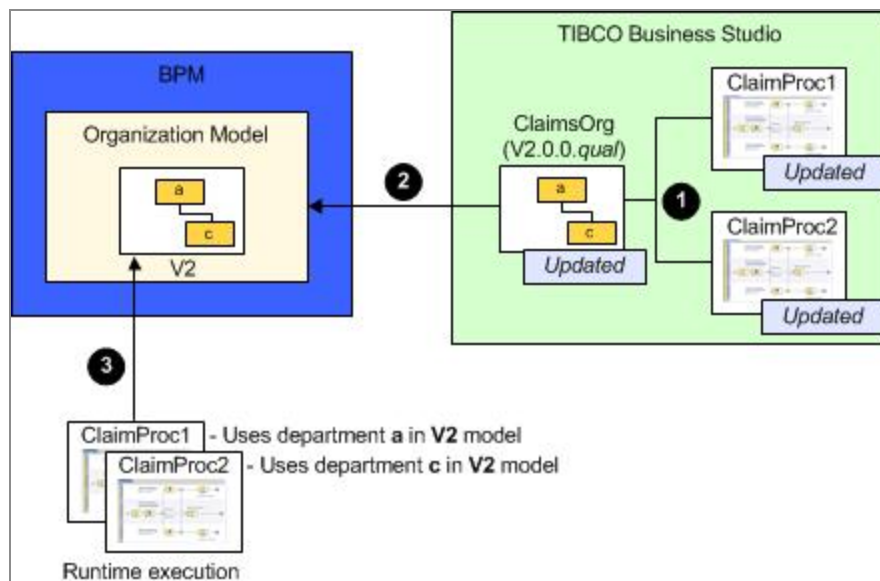
i Note: If the organization model 1.0.0.ts had more entities than the one deployed in 1.1.0.ts, the un-deployment of 1.1.0.ts removes those additional entities.

So, if 1.0.0.ts contained **a**, **b** and **d**, and 1.1.0.ts contained **a**, **b** and **c**, entity **d** would be removed after 1.1.0.ts is deployed (causing 1.0.0.ts to be un-deployed).

To avoid this, change the name of the application when deploying the updated model (and deploy as a new application and not as an upgrade).

Example - Phase 3 Making a Destructive Update to the Model

A company reorganization now occurs which results in department **b** being broken up.




Design Time

The solution designer:

- (1) Removes department **b** from the ClaimsOrg organization model.
- (2) Changes the ClaimsOrg project's version number to 2.0.0.qual. (As this is a destructive change to the existing organization model, the *major* part of the version number **must** be changed.)


- (3) Modifies ClaimProc2 to remove participant references to department **b**.
- (4) Updates ClaimProc1 so that ClaimsOrg's updated major version number (**2**) is reflected (internally in the process definition) in its participant references to department **a**.

 **Note:** If ClaimProc1 is not updated, at runtime it will continue to try and use participants defined in V1 of the organization model.

Deployment

The solution designer:

- (1) Deploys the ClaimsOrg organization model as an upgrade to the easyAsOrg1 application, as version 2.0.0.ts.

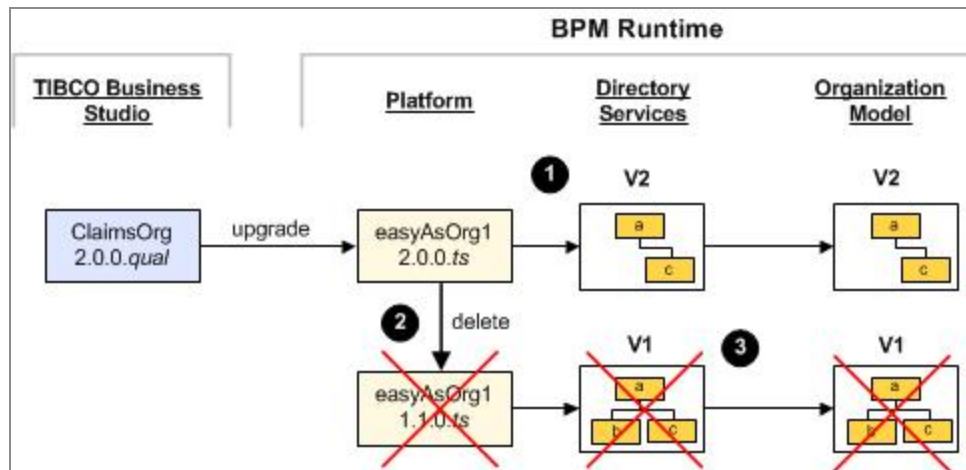
 **Note:** Using the same application name (that is, upgrading) undeploys the V1 runtime organization model. (This is why ClaimProc1 must also be updated as described above - see User Application Dependencies .)

- (2) Upgrades the ClaimProc1 and ClaimProc2 applications to, respectively, version 1.1.0.ts and version 2.0.0.ts.

Runtime

TIBCO BPM Enterprise:

- (1) creates a V2 runtime organization model.
- (2) deletes the V1 runtime organization model.



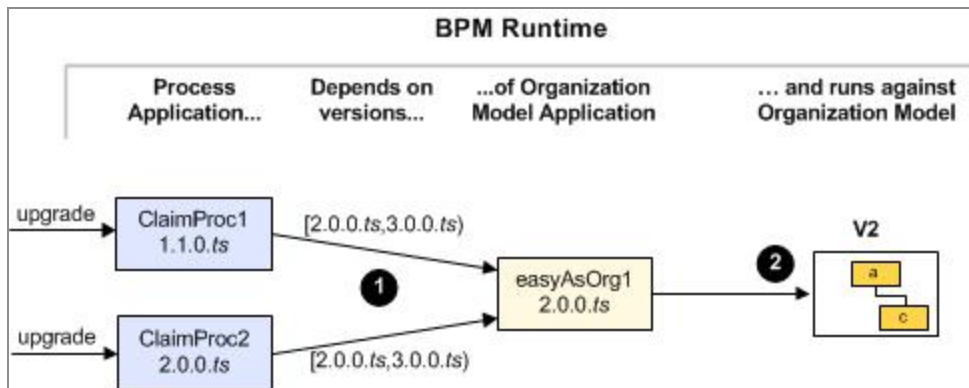
- (1) Directory Services uses the deployment artifact from the **2.0.0.ts** version of the **easyAsOrg1** application to create a new **V2** organization model.
- (2) The platform deletes the **1.1.0.ts** version of the **easyAsOrg1** application (because it has the same name as the deployed application).
- (3) Directory Services deletes the deployment artifact from the **1.1.0.ts** version of the **easyAsOrg1** application. As there are no longer any deployment artifacts contributing to the V1 organization model, it is deleted.

User Application Dependencies

i Note: In the following diagram, application dependencies are shown using the convention $[Min, Max)$:

- Min* is the version referenced when the process application is initially deployed. The square bracket denotes that the value is inclusive.
- Max* is the next major version. The round bracket denotes that the value is exclusive.

For example, $[1.0.0.ts, 2.0.0.ts)$ denotes any version from 1.0.0.ts (inclusive) to 2.0.0.ts (exclusive). (The *ts* is not used in the comparison.)



ClaimProc1 and ClaimProc2 both:

(1) have a dependency on the **easyAsOrg1** application, version 2.0.0.ts (inclusive) to version 3.0.0.ts (exclusive).

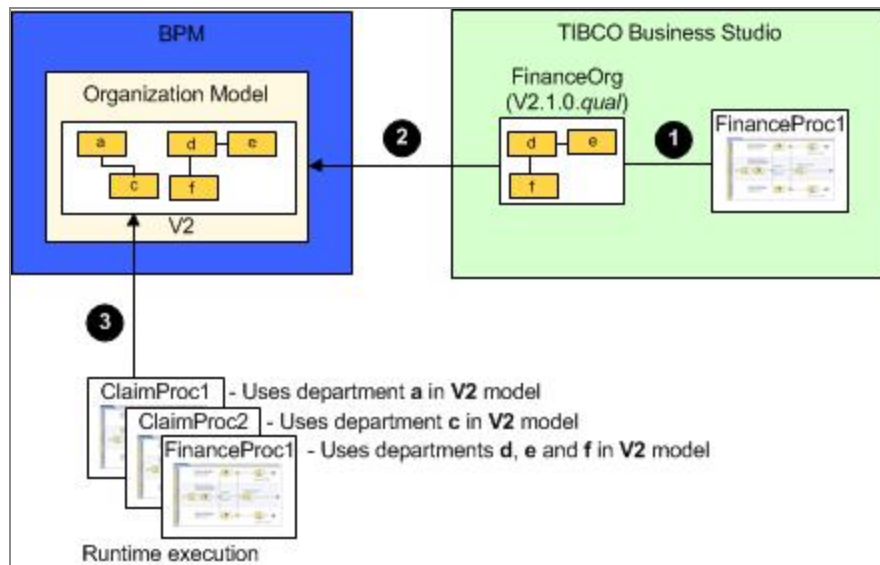
i Note: Although ClaimProc1 does not reference department **b**, its participant references (to department **a**) must be updated to use the new version. If this is not done, ClaimProc1 would still have a dependency on the **easyAsOrg1** application, version 1.0.0.ts (or later 1.x.x.ts).

As the 1.1.0.ts application has now been deleted, this dependency cannot be resolved and the ClaimProc1 application would enter a "Waiting for dependencies" state.

(2) execute against V2 of the runtime organization model (which now includes only departments **a** and **c**.)

Example - Phase 4 Extending the Organization Model

Finally, the project is rolled out to include other parts of the company.



The initial Finance division implementation defines a single application, FinanceProc1, which involves departments **d**, **e** and **f**.

Design Time

The solution designer:

- (1) Creates a FinanceOrg organization model with a version number of V2.1.0.ts.
- (2) Adds participants from departments **d**, **e** and **f** to FinanceProc1. Each participant is defined as an external reference to the FinanceOrg organization model.

FinanceOrg's major version number (2) is recorded (internally) in the FinanceProc1 process definition as part of the participant definitions.

i Note: The FinanceOrg organization model does not need to include departments **a** and **c** - it simply provides a view of those parts of the organization that are relevant to the Finance department and the FinanceProj1 application.

Because this will be an extension to the existing runtime organization model (which does not impact departments **a** and **c**) the same major version number can be used with an update to the minor version number.

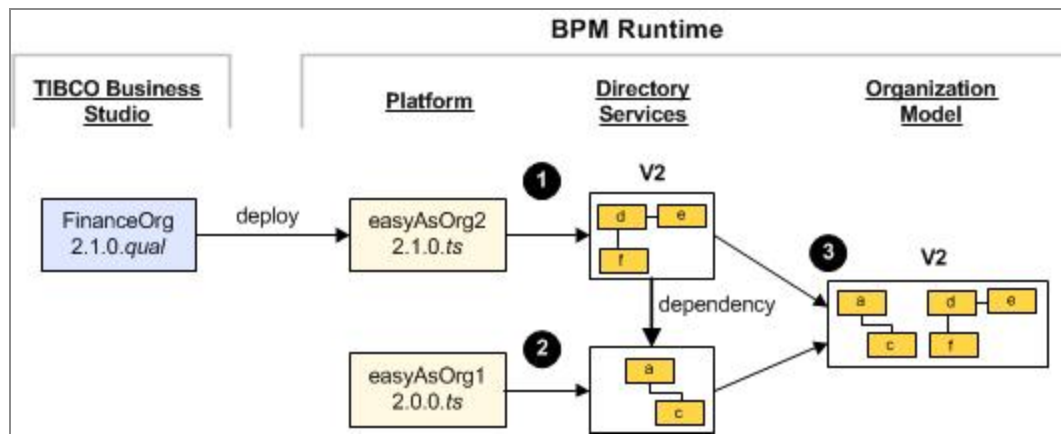
Deployment

The solution designer:

- (1) Deploys the FinanceOrg organization model as the **easyAsOrg2** application, version 2.1.0.ts.
- (2) Deploys the FinanceOrg project as the FinanceProc1 application, version 1.0.0.ts.

Runtime

TIBCO BPM Enterprise merges the changes from the deployed FinanceOrg organization model - the addition of departments **d**, **e** and **f** - into the existing V2 runtime organization model.



- (1) Directory Services adds the deployment artifact from the **2.1.0.ts** version of the **easyAsOrg2** application to the **V2** organization model.

This establishes a dependency on the deployment artifact from the **2.0.0.ts** version of the **easyAsOrg1** application.

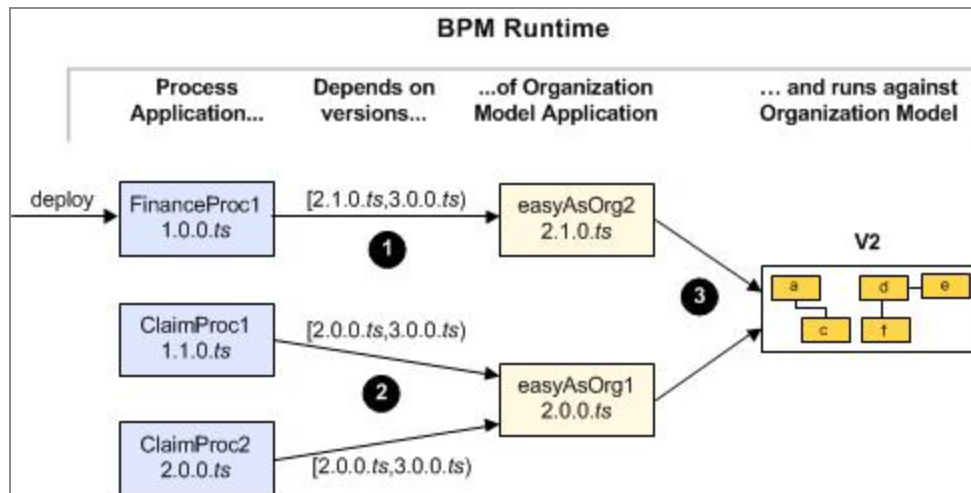
- (2) The platform does not delete the **2.0.0.ts** version of the **easyAsOrg1** application (because it has a different name from the deployed application).
- (3) The **V2** organization model is the combination of both deployment artifacts.

User Application Dependencies

i Note: In the following diagram, application dependencies are shown using the convention $[Min, Max)$:

- *Min* is the version referenced when the process application is initially deployed. The square bracket denotes that the value is inclusive.
- *Max* is the next major version. The round bracket denotes that the value is exclusive.

For example, $[1.0.0.ts, 2.0.0.ts)$ denotes any version from 1.0.0.ts (inclusive) to 2.0.0.ts (exclusive). (The *ts* is not used in the comparison.)



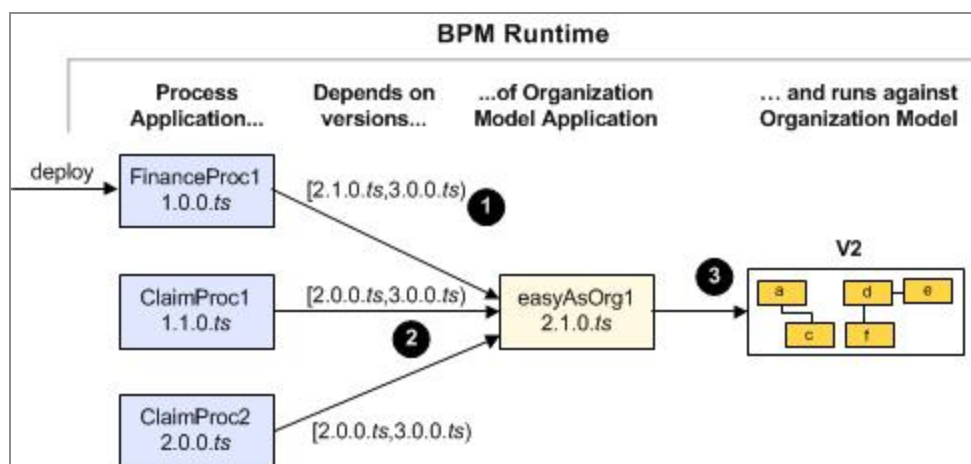
- (1) FinanceProc1 has a dependency on the **easyAsOrg2** application, version 2.1.0.ts (inclusive) to version 3.0.0.ts (exclusive).
- (2) ClaimProc1 and ClaimProc2 have a dependency on the **easyAsOrg1** application, version 2.0.0.ts (inclusive) to version 3.0.0.ts (exclusive).
- (3) ClaimProc1, ClaimProc2 and FinanceProc1 all execute against V2 of the runtime organization model (which includes departments **a**, **c**, **d**, **e** and **f**).

i Note: If the FinanceOrg organization model was instead deployed as version 3.0.0.ts of the easyAsOrg1 application:

- Version 2.0.0.ts of the easyAsOrg1 application would be deleted.
- This would undeploy the V2 runtime organization model. The ClaimProc1 and ClaimProc2 applications would then enter a "Waiting for Dependencies" state, as they would be unable to resolve their references to departments **a** and **c**.

Alternative Scenario

Instead of using a new application name, the **FinanceOrg** project could have been deployed as version 2.1.0.ts of the **easyAsOrg1** application. This would have resulted in the scenario shown here.



In this case:

- (1) FinanceProc1 has a dependency on the **easyAsOrg1** application, version 2.1.0.ts (inclusive) to version 3.0.0.ts (exclusive).
- (2) ClaimProc1 and ClaimProc2 have a dependency on the **easyAsOrg1** application, version 2.0.0.ts (inclusive) to version 3.0.0.ts (exclusive).

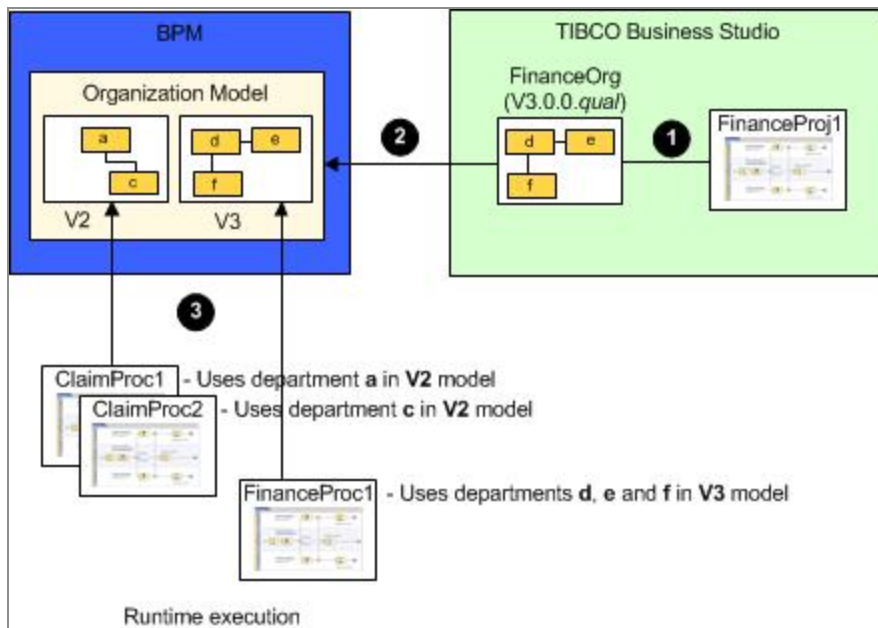
i Note: Although version 2.0.0.ts of this application has been deleted, the dependency is still resolved by version 2.1.0.ts.

- (3) ClaimProc1, ClaimProc2 and FinanceProc1 execute against V2 of the runtime organization model (which now includes only departments **a**, **c**, **d**, **e** and **f**).

Example - Phase 4 An Alternative Implementation

The preceding example, [Example - Phase 4 Extending the Organization Model](#), showed how the separate ClaimsOrg and FinanceOrg organization models could be merged into a single runtime model, by using the same major version number.

However, the Finance division could have kept their organization model completely separate by using a different major version number.



Design Time

The solution designer:

- (1) Creates a FinanceOrg organization model with a version number of **V3.0.0.qual**.
- (2) Adds participants from departments **d**, **e** and **f** to FinanceProc1. Each participant is defined as an external reference to the FinanceOrg organization model.

FinanceOrg's major version number (3) is recorded (internally) in the FinanceProc1 process definition as part of the participant definitions.

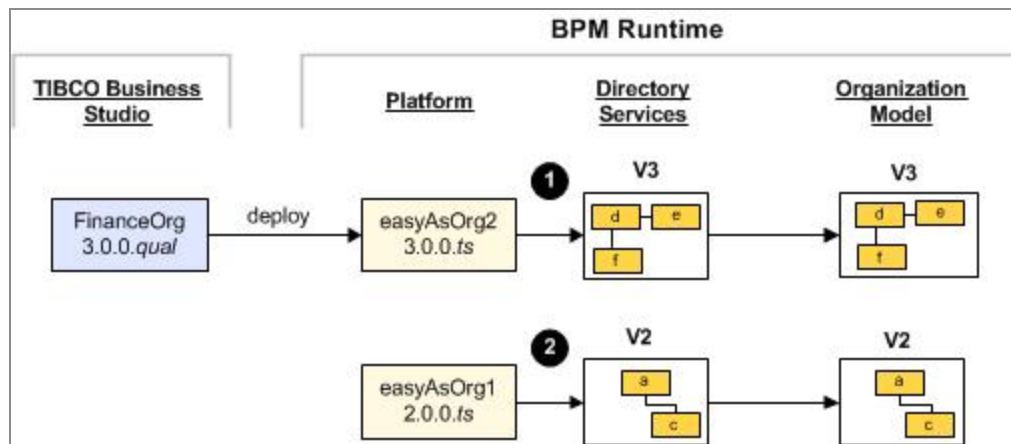
Deployment

The solution designer:

- (1) Deploys the FinanceOrg organization model as the easyAsOrg2 application, version 3.0.0.ts.
- (2) Deploys the FinanceOrg project as the FinanceProc1 application, version 1.0.0.ts.

Runtime

TIBCO BPM Enterprise creates the deployed FinanceOrg organization model as a new V3 runtime organization model.



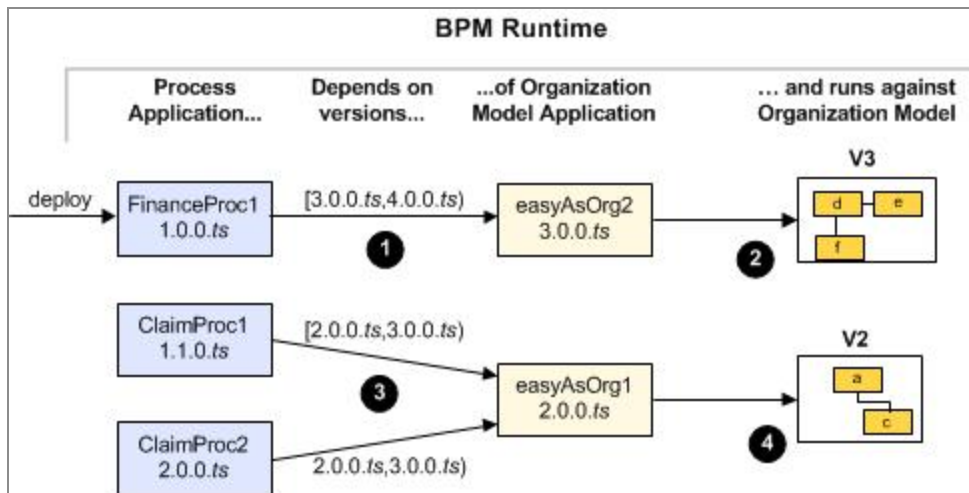
- (1) Directory Services uses the deployment artifact from the **3.0.0.ts** version of the **easyAsOrg2** application to create a new **V3** organization model.
- (2) The platform does not delete the **2.0.0.ts** version of the **easyAsOrg1** application (because it has a different name from the deployed application).
- (3) Both the **V2** and **V3** organization models are available.

User Application Dependencies

i Note: In the following diagram, application dependencies are shown using the convention $[Min, Max)$:

- *Min* is the version referenced when the process application is initially deployed. The square bracket denotes that the value is inclusive.
- *Max* is the next major version. The round bracket denotes that the value is exclusive.

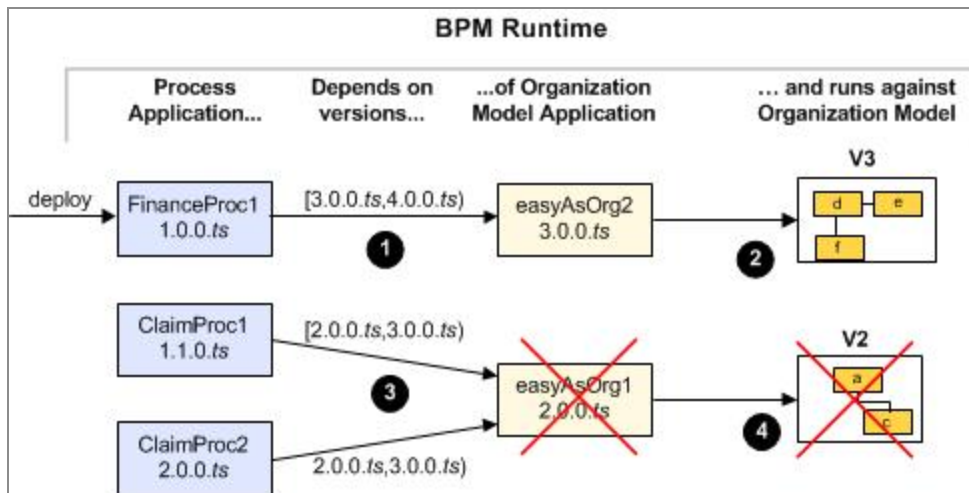
For example, $[1.0.0.ts, 2.0.0.ts)$ denotes any version from 1.0.0.ts (inclusive) to 2.0.0.ts (exclusive). (The *ts* is not used in the comparison.)



- (1) FinanceProc1 has a dependency on the **easyAsOrg2** application, version 3.0.0.ts (inclusive) to version 4.0.0.ts (exclusive).
- (2) FinanceProc1 executes against the V3 runtime organization model.
- (3) ClaimProc1 and ClaimProc2 have a dependency on the **easyAsOrg1** application, version 2.0.0.ts (inclusive) to version 3.0.0.ts (exclusive).
- (4) ClaimProc1 and ClaimProc2 continue to execute against V2 of the runtime organization model (which includes departments **a** and **c**).

Alternative Scenario

If the **FinanceOrg** project was deployed as a new major version of the **easyAsOrg1** application - version 3.0.0.ts - the resulting scenario would be as shown here.



In this case:

- (1) FinanceProc1 has a dependency on the **easyAsOrg1** application, version 3.0.0.ts (inclusive) to version 4.0.0.ts (exclusive).
- (2) FinanceProc1 executes against V3 of the runtime organization model (which now includes only departments **d**, **e** and **f**).
- (3) ClaimProc1 and ClaimProc2 have a dependency on the **easyAsOrg1** application, version 2.0.0.ts (inclusive) to version 3.0.0.ts (exclusive).
- (4) ClaimProc1 and ClaimProc2 still reference V2 of the runtime organization model, which has been deleted.

Work Management

Work management can be defined as the distribution and control of work that needs to be performed by human resources. Some of the work items which human resources execute are listed here:

- distributing the right tasks to the right people at the right time.
- managing what happens when work doesn't go as planned.
- presenting a user with an appropriate user interface to do a piece of work.
- integrating work with business processes.

Work items are managed separately from the process that created them.

Distributing Work to Users

When an item of work is ready to be processed, TIBCO BPM Enterprise has to determine which user(s) to distribute it to.

The initial stage of assigning a work item is done in the design time, that is in TIBCO Business Studio - BPM Edition. When process designers create a user task, they must identify who performs that task at runtime, by specifying organization model entities as participants to the user task.

At runtime, the design-time work distribution is resolved by TIBCO BPM Enterprise, which resolves the specified organization model entities into a user or set of users who receive the work item.

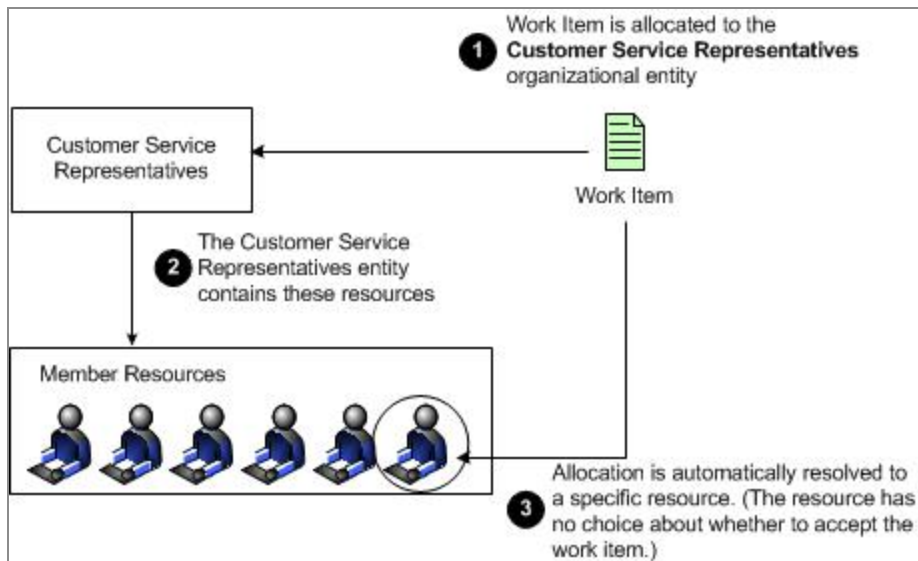
i Note: Numerous workforce management patterns also impact the way in which work is distributed to users. See [Workflow Resource Patterns Support](#) for more information.

Offering and Allocating Work

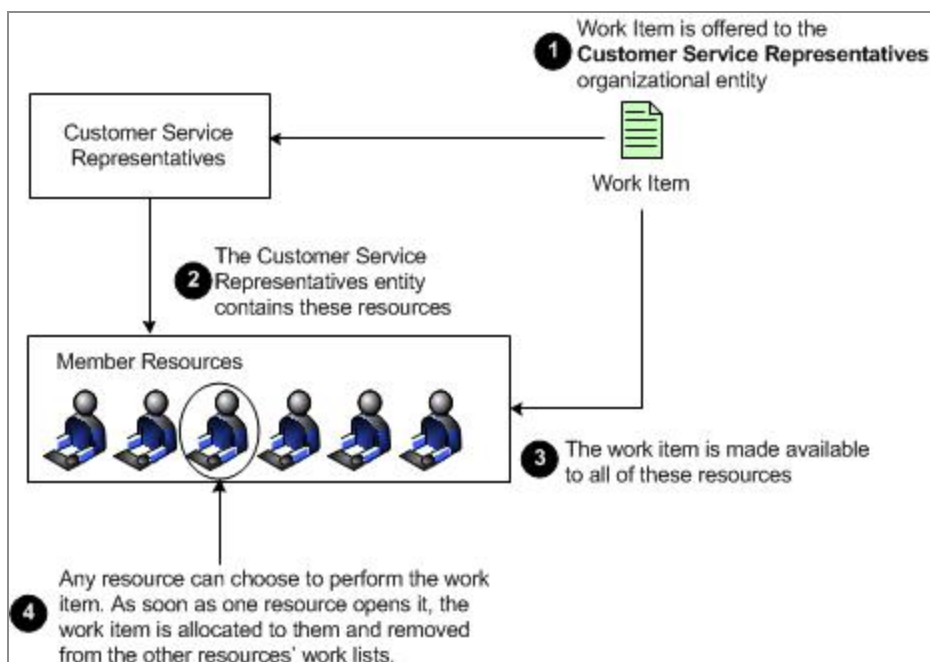
Work items can be either *allocated* to users, or *offered* to them. **Allocated work** is specifically given to a single user to perform. **Offered work** is made available to a single user or to a group of users.

The following diagrams show the difference between these methods.

Allocating a work item to a specific user



Offering a work item to many users



Distribution Strategies and Offer Sets

At runtime, TIBCO BPM Enterprise determines who a work item should be distributed to. It also decides whether the work item should be allocated or offered. These decisions are made based on how the user task that generated the work item was defined at design-time.

Offer Sets

An offer set is the set of valid resources that can execute a user task.

At design-time, the user task's participant definition defines the offer set.

At run-time, TIBCO BPM Enterprise determines which users belong to that offer set.

Distribution Strategy

A distribution strategy determines how a work item should be distributed to the users who make up the offer set.

At design-time, one of the following user task's **distribution strategy** must be defined:

- **Offer to all:** At run-time, TIBCO BPM Enterprise offers the work item to all users who are members of the offer set.
- **Allocate to one:** At run-time, TIBCO BPM Enterprise allocates the work item to a single user who is a member of the offer set. It determines which user to allocate the work item to by selecting one of the following allocation method.
 - **Round-robin:** Work items are allocated to members in strict rotational order.
 - **Random:** Work items are allocated to members in random order.

Allocation methods can be assigned to organizational entities using TIBCO Business Studio - BPM Edition's Organization Modeler. TIBCO BPM Enterprise uses the allocation method assigned to the requisite organizational entity. If that entity does not have an allocation method, it uses random allocation instead.

- **Allocate to offer-set member:** A **Performer Field** must also be specified with this option. The process must populate this field with the GUID of a specific member of the offer set. (For example, the user who started the process.)

At run-time, TIBCO BPM Enterprise allocates the work item to the user identified by

the value of the Performer Field. The user should also be a member of the offer set. If that user is *not* a valid member of the offer set, the work item is then offered to the remaining members of the offer set, *as if* the **Offer to all** distribution strategy had been used instead.

The **Allocate to offer-set member** strategy allows you to support, for example, a case handler or account manager pattern, so that although the work item is originally allocated to a member of a team, the team manager can still perform the following actions:

- see all items that were originally offered to the team.
- re-allocate the work item to another member if required - for example, if the user who started the case is off work due to sickness.
- provide a report on a work item from the team's perspective.



Note: The **Allocate to offer-set member** distribution strategy cannot be used with the Chained Execution, Separation of Duties or Retain Familiar workflow patterns.

Resource Query Language

Resource Query Language (RQL) is a textual query syntax. It is used to identify resources within the BPM destination environment that meet a defined set of criteria. An RQL query returns a set of resources that match the criteria expressed in the query. Work can then either be allocated to one of those resources, or offered to multiple individual resources.

RQL is dynamic, and is evaluated when the work item is created and whenever it changes. This means that if the items referred by the RQL change in some way, (for example if the resources mapped to an organizational position are changed) then this will be reflected in the set of resources associated with the work item.

For details about how RQL is used, see the *TIBCO Business Studio - BPM Edition Modeling User's Guide*.

You also have the option to use either RQL or Dynamic Organization Participants. For more information, see [Dynamic Organization Participants](#).

Participants

When a process designer creates a user task, they can define the participant(s) who will perform the task at runtime.

Participants are defined in the following ways:

- **statically**, by specifying one or more organizational entities - groups, positions, and organization units or organizations.
- **dynamically**, by using runtime data to identify the required organizational entities.
- **using expressions**, by building a query that interrogates the organization model to identify the required organizational entities.

This flexibility allows a process designer to handle both simple and complex distribution scenarios without impacting the overall process design. For example:

- offer a user task to all Customer Service Representatives.
- allocate a user task to an accountant if the value to be signed off is less than \$5000, but allocate it to an Accounts Manager if the value is \$5000 or more.
- allocate a user task to a single loss adjuster who holds at least level 2 motor insurance certification and is based in the Chicago office.

Dynamic Organization Participants

With Dynamic Organization Participants, you can create references to dynamic organization entities, that can only be completed when the organization model is deployed.

The User Task component passes a reference (Organization Entity Reference) to the Organization Participant for work allocation.

Organization Entity References consist of the following components:

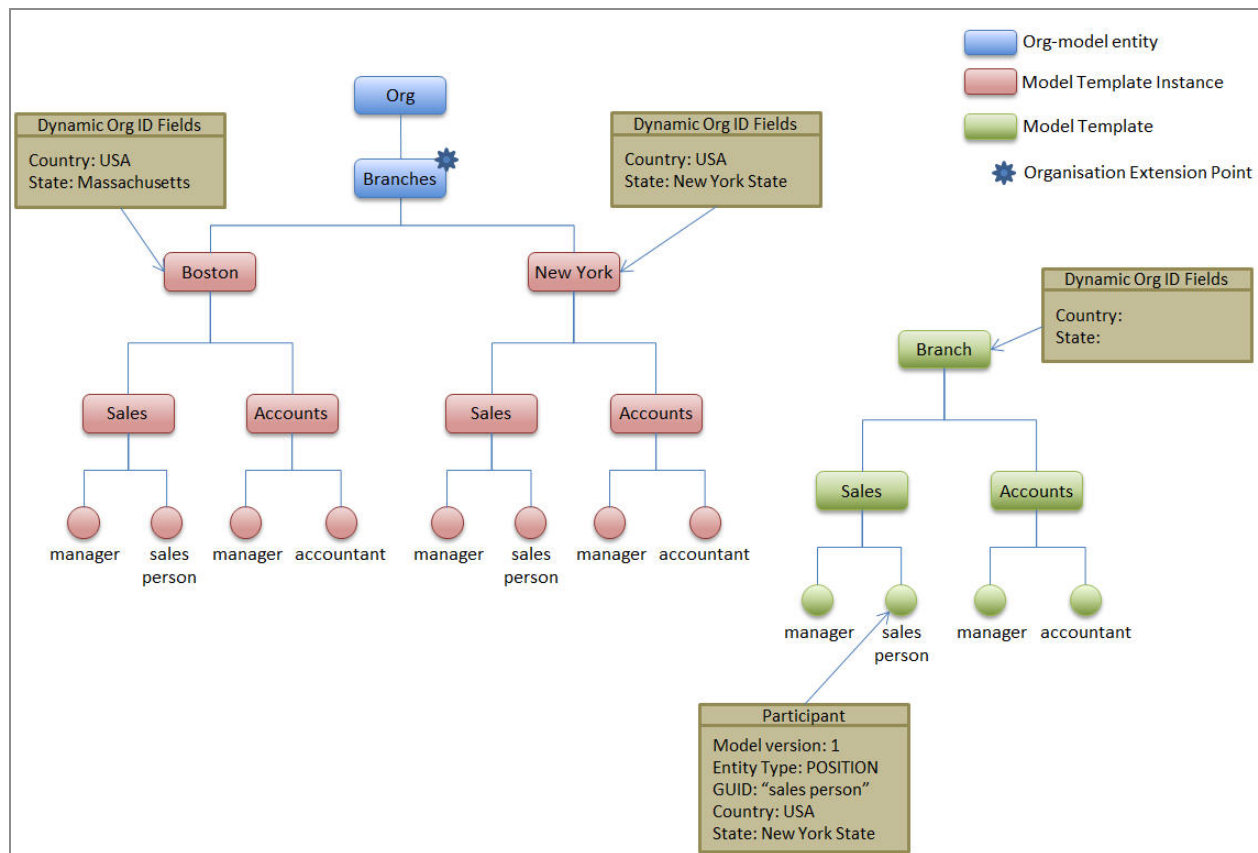
- an OrganizationModel Major Version - the major version of the Organization Model in which the entity resides.
- an Entity Type - the type of Organization Entity expected; for example, GROUP, POSITION, ORGANISATION_UNIT.
- a GUID - the unique identifier of the Organization Entity (only unique within a given major version of the Organization Model).

- The value of the **Dynamic Organization Identifier** fields.

The Organization Entity Reference is used to locate the Dynamic Organization Model, and use the Dynamic Organization Identifier field values to identify the Dynamic Organization Model instance, under which it will locate the corresponding entity.

As an example; the figure below shows an Organization Model with a Dynamic Organization Model, the root node of which is named **Branch**. **Branches** is the name of the extension point. There are two Dynamic Organization Model instances, named **Boston**, and **New York**. A Dynamic Organization Participant references the **sales person** Position within the Dynamic Organization Model (the GUID would actually be a generated value, but the name has been used here for clarity).

At runtime, the values of Dynamic Organization Identifiers **Country** and **State** will identify the particular instance of the Dynamic Organization Model as the **New York** instance. It is from that instance that the Position **sales person** will be selected for the work allocation.



Push and Pull Distribution Models

Work can either be pulled by users or pushed to them.

The usual model is pulled distribution. In this model, when a work item is generated it is added to a user's work list. When the user is ready to work on the work item, they open - or pull - it from their work list. A user needs to log in to TIBCO BPM Enterprise to access their work list.

TIBCO BPM Enterprise also supports pushed distribution. In this model, when a work item is generated it is sent to a user as an email. The email contains the URL of the work item, which the user can click to open and process the work item. The users need to authenticate themselves before they can open the work item.

You can enable receiving pushed work items for specific organizational entities and users.

Pushed distribution is useful for occasional TIBCO BPM Enterprise users - for example, managers who only need to become involved in a process when some form of higher level approval is required. These users will typically not be logged into TIBCO BPM Enterprise all the time and so could otherwise miss the arrival of high priority work items.

Presenting Work to Users

TIBCO BPM Enterprise uses a number of different methods, tools, and artifacts to allocate or offer work to users.

TIBCO BPM Enterprise Client Applications

TIBCO BPM Enterprise provides a default client application to give a user interface to work.

Users can utilize these clients to access and perform the functions they need to complete their daily tasks. For example, they can:

- log in to TIBCO BPM Enterprise.
- see the contents of their work list.
- open work items and complete and submit the forms that are displayed.

- apply filter and/or sort parameters so that only the desired work items are listed in the desired order in their work list.
- start business services.

TIBCO BPM Enterprise client applications can be easily configured and customized to suit the requirements of a particular enterprise - see [Flexible Runtime Clients](#).

Users' access to individual client application functions can be controlled, based on the privileges that they hold.

Work Lists

When a user logs into a TIBCO BPM Enterprise client application, they are presented with a single view of their work - the **work list** - that allows them to view and process all the work items currently assigned to them (whether offered or allocated, and however generated).

The work list is dynamically created and is maintained by TIBCO BPM Enterprise. The work list is unique to each user, and can be retrieved using the TIBCO BPM Enterprise REST API. See TIBCO BPM Enterprise REST APIs for more information.

An offered work item appears in the work list of every qualifying user. When a user first opens it, the work item is allocated to them and removed from all other users' work lists. An allocated work item appears only in the work list of the user that it has been allocated to.

Tools are provided to allow users to sort and filter the contents of their work list according to different criteria, and to configure the work list display according to their preferences.

Users who have the privileges to perform specific system actions can also, for example, view the work list for a particular organization unit or position.

Work Items

Work items are the individual pieces of work that a user needs to perform.

When a user opens a work item they are presented with a user interface - typically a form - that allows them to process and complete the task. As soon as the user edits any information in the form, the work item becomes allocated to them.

Once a work item has been allocated to a user, that user must complete it (although in some cases they can re-offer or reallocate it).

Pageflows

A pageflow is a specialized version of a normal business process that can be used to provide an animated user interface - a sequence of forms rather than just a single form - for a single work item to the same user.

A pageflow process can also include other activities - such as scripts and conditional logic - which can be used to drive the interaction with the user. Pageflows can also be chained together.

For example, an insurance company process includes a user task to capture claim details from a customer. Instead of a form, the process designer creates a pageflow and associates it with the user task. The pageflow process:

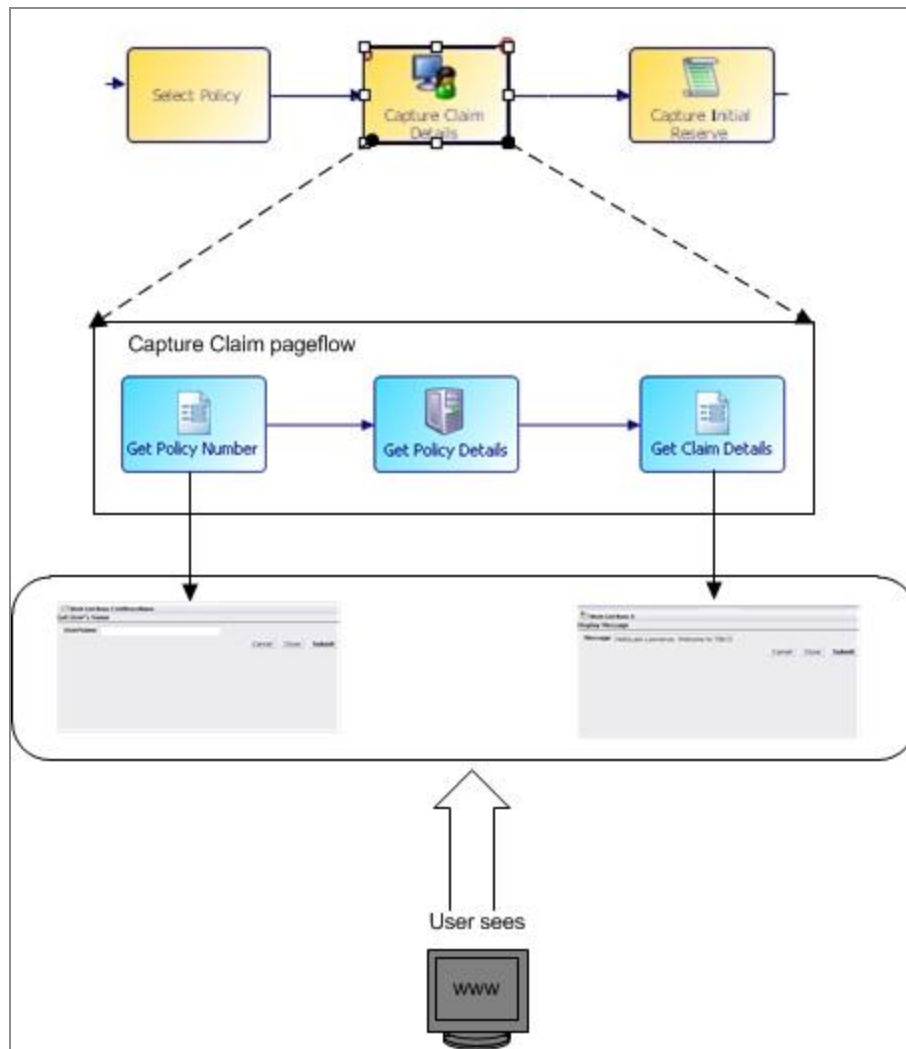
- presents an initial form which allows the user to enter the claimant's policy number.
- calls a policy database, via an API call, to obtain the policy details.
- presents a second form to the user which displays the policy details and allows them to continue with the remaining claim information. (The pageflow would most likely contain an extended sequence of forms, rather than just the two used in this simplified example.)



Note: The process designer defines forms for the Get Policy Number and Get Claim Details user tasks in the pageflow process, in the same way as they would define a form for a business process.

At runtime, when a user opens the Capture Claim Details work item, TIBCO BPM Enterprise runs the Capture Claim pageflow process. The user sees the Get Policy Number and Get Claim Details forms as a continuous dialogue - the second form is displayed as soon as they submit the first one. They do not have to open separate work items from their work list, and there is no possibility of the forms being handled by different users.

When the user submits the Get Claim Details work item, the pageflow process completes and control returns to the main business process, which then proceeds to the Capture Initial Reserve script step.



Note: Unlike a normal business process, a pageflow process is *stateless*. If the process is not completed in full, any data set earlier in the process is lost. In the example above, if the user chose to cancel the Get Claim Details form, the data entered into the Get Policy Number form and retrieved from the Get Policy Details task would be lost.

Using Sticky Sessions with Pageflows

Case Management

From a conceptual standpoint, a case is the "subject of a process." Examples are "Order", "Exception", "Customer", or "Claim". That is, you can have an "Order" case that is created when someone places an order for a product that you sell. The type of cases you use depends on the type of business you are in.

All cases have a *life-cycle* that includes a number of *states*. You can think of case states as logical phases in the life-cycle of a case. A case has at least two states in its life-cycle: created and completed. Typically, a case will have other states. For example, an Order case could have states of Created, Fulfilled, Packaged, Delivered, and Completed.

Central to a case is *case data*. Case data is business data that is collected together as a business object. This set of case data that is collected together, including its state, is often referred to as "the case", or "the case object". Case data is centrally managed by TIBCO BPM Enterprise and can be accessed and updated by multiple BPM process applications. For example, for an Order case, the case data could include information such as order number, order date, item ordered, and delivery address. Other applications, such as a billing application, can access and possibly modify this case data.

Another key aspect of cases are *case actions*. These are business processes that are defined in a case management application to provide the end-user with the ability to apply an action on a case. For example, for an Order case, there may be actions such as "Cancel" or "Change Delivery Address." Case actions are operations that may be initiated by the end-user as needed; they are not mandatory.

The association between case states and case actions is such that your application can be configured to apply a specific case action only when the case is in certain states. For example, you can ensure that the case allows the 'Change Delivery Address' case action in all states except in the 'Delivered' state.

The following diagram illustrates the possible life-cycle, states, and case actions of a simple Order case:

|

In this example, after the case is started (created), a work item is generated to inform the appropriate user(s) that the order needs to be fulfilled. After the user fulfills the order and submits the work item, the case state changes to Fulfilled, then the appropriate user(s) are informed that the order needs to be packaged, and so on, through the entire life-cycle. At

any of the allowed states, a user has the option to initiate one of the available case actions to perform an operation on the case, such as canceling the order or changing the delivery address.

Occasionally, special activities can be made available to users progressing the case by an executing business process related to the case. These activities can either be mandatory or optional. These activities are typically much more dynamic in nature as they are made available dynamically due to the circumstances determined by the process, and the users interacting with the process, and not just simply by the state of the case and the predetermined actions designed into the supporting BPM application. They will only apply in special cases and, therefore, will not be appropriate for all cases, and will not be seen or performed for every case. These special activities are called ad-hoc activities in BPM Enterprise. Examples of this for an Order management case might be custom build options to an item in the order, foreseen product customization prior to dispatch, special packaging, or delivery requirements that are bespoke for this order.

BPM applications are increasingly seen as business applications that require an application-type-specific user interface, rather than having many BPM applications being served by a single general-purpose user interface that can be used with any BPM application. This is especially true for case management BPM applications. In the above example, the user interacts with their order management system, irrespective of whether or not it is implemented using general-purpose case management capabilities layered on top of general-purpose work management and business process execution capabilities. For this application to comprise a usable system, the user interface must be specific to order management, and also include customized versions of such general purpose BPM and case management capabilities, all made available in a form that is customized for order management. BPM Enterprise supports this by providing customizable user interface components that provide a user interface for the different aspects of case management and BPM that can easily be combined into custom applications. Sample user interface applications that can be easily customized that are built out of these components are also provided, enabling the production of such custom user interfaces in a simple and quick way, requiring the minimum amount of technical skill to produce.

In a case management application, users can view lists of currently active cases, list the work items associated with a particular case, as well as view all events that have taken place for a case.

Industry Standard Classifications of Case Management

BPM Enterprise supports aspects of both types of industry standard classifications of case management: **Production Case Management (PCM)** and **Adaptive Case Management**

(ACM):

- **PCM** - Production Case Management is used where cases have some limited unpredictability in the way they are processed. PCM applications provide a fixed repertoire of operations that the knowledge worker can decide to use, or not, depending on the specific circumstances of the case being progressed. The set of appropriate operations varies, depending on the role of the knowledge worker and the current state of the case itself.

Key features of PCM, case states and case actions, are provided in BPM Enterprise.

- **ACM** - Adaptive Case Management handles cases where the processing is unpredictable and the end state may not be well defined. They often require input from other stakeholders in an unpredictable manner, reacting to events and circumstances that may not be clear at the outset of the case. However, event-adaptive cases work within a given context, which is known at the outset (for example an "order" does not turn into an "insurance claim").

Key features of ACM, ad-hoc activities and case linkage capabilities, are provided in BPM Enterprise.

The primary difference between the two classifications is that PCM case management applications are more predictable (like conventional business processes) and ACM case management applications are more dynamic and unpredictable. Most case management applications fall in the spectrum of these classifications, rather than at one polar extreme or another. BPM Enterprise provides the features needed to develop applications that include aspects of both classifications.

Creating Case Management Applications in TIBCO Business Studio - BPM Edition

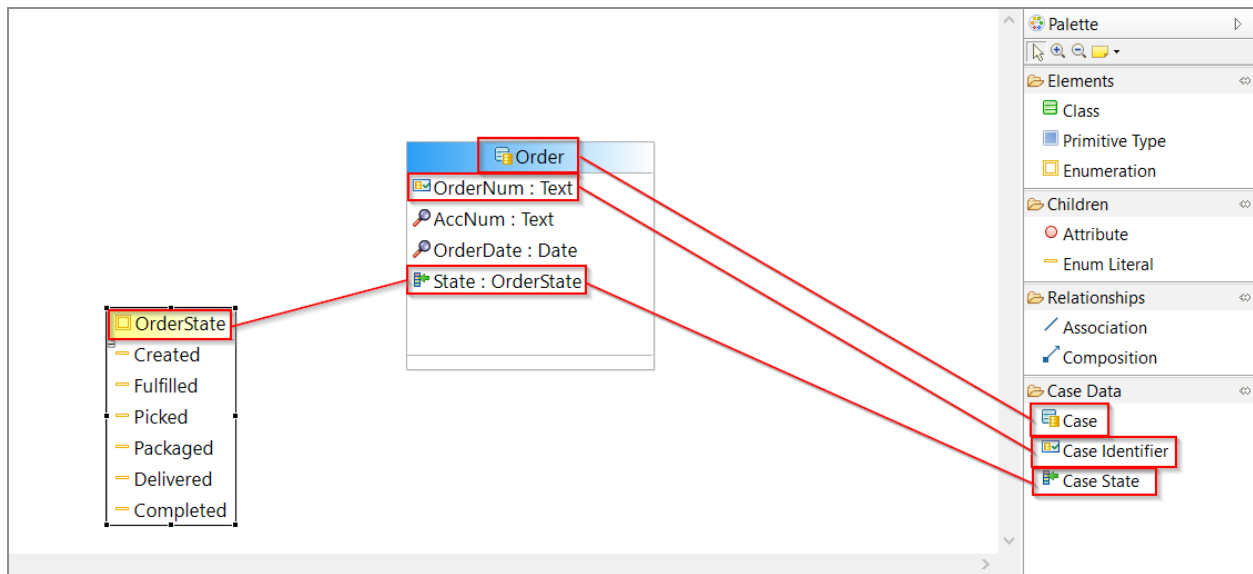
TIBCO Business Studio - BPM Edition is used to create all of the elements used in a case management application; case data model, business processes that use case data, and case action processes.

- Case Data Model
- Interacting with Cases in Business Processes
- Case Actions

Case Data Model

Case data is modeled as a *case data model* (or *case model*), consisting of one or more global BOMs, in a Business Data Project in TIBCO Business Studio - BPM Edition .

The following diagram illustrates an example case model that could be used in a simple Order case:



The case (**Order**) is modeled as a case class. It contains information relevant to the case, such as order number, account number, and the date of the order.

Every case must have a *Case Identifier* (CID), which is a special type of attribute that can be used to uniquely identify an instance of a case *class* and type *text*. It can be used in processes, scripts or API calls to create or to find a particular case. The case identifier for the **Order** case class in this example is the order number (**OrderNum**).

A case class can also have a *case state*, which can be used to uniquely identify a set of business-specific states that a case can be in. Case states can be used to control the availability of case actions to users. In this example, the **Order** case class uses the **OrderState** enumeration to define the available states for the case.

Interacting with Cases in Business Processes

Within a business process created in TIBCO Business Studio - BPM Edition , you can create, read, update and delete case objects created from the case classes defined in a referenced case data model.

This is done using the "Case Data Operations" service task.

The screenshot shows the 'Task' configuration window in TIBCO Business Studio. The 'General' tab is selected, showing the following details:

- Label:** Persist Order
- Description:** Name: PersistOrder
- Activity Markers:** ☐ Standard Loop ☐ Multiple Instance Loop ☐ Ad-Hoc
- Participants:** - not set - (with a 'Clear' button)
- Activity Type:** Service Task

The 'Service Type' is set to 'Case Data Operations'. The 'Create Case Object(s)...' option is selected. The configuration for this operation is as follows:

- Case Class:** Order (Order)
- Operation:** Create Case Object(s) From Local Data
- Local Data Value(s):** dfOrder
- Return Case Reference(s):** dfOrderRef

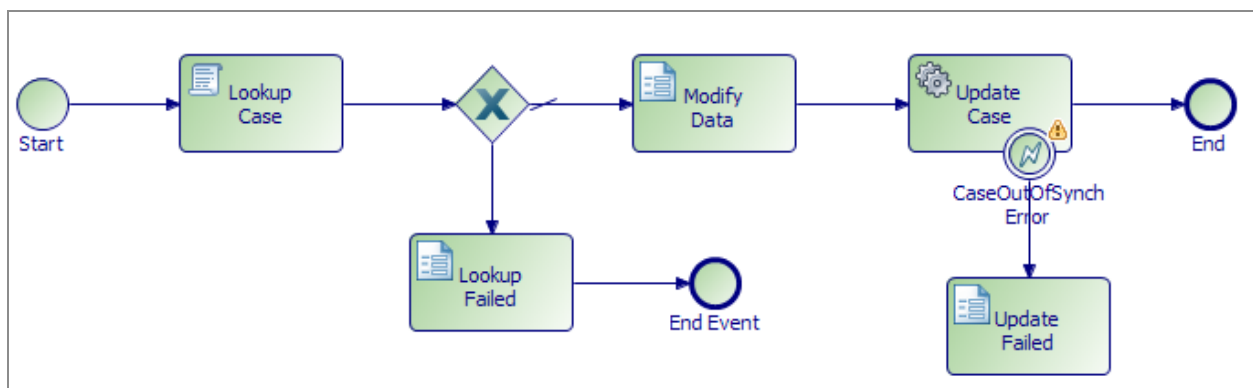
A description for the operation states: 'Create new case object(s) from local data value(s). All mandatory content in the source data must be assigned.'

In this example, an instance of the Order case class is created by the Case Data Operations service task.

Case Actions

A case action is a type of business service that is associated with a case class. A case action defines an action a user can perform that is related to a case. The availability of case actions can be restricted based on the current case state or the user's permissions.

From TIBCO Business Studio - BPM Edition, you can either create a new case action or generate one directly from a case class. Template case action processes are provided that allow you to view or update the contents of a case object:



This is an example of the update case template created from a case class. It contains the Case Data Operations service task (update case), as described in [Interacting with Cases in Business Processes](#). You can modify the template to provide the functionality you require for a particular case class. Business processes can also be invoked from within the case action, if required.

At runtime, case actions can be invoked by a user if the user has the required permissions, and the case is in the appropriate state for the action.

For more information, see "Creating a Case Action Process" in the *TIBCO Business Studio - BPM Edition Modeling Guide*.

Case Data Store

The case data store is the database in which case data models are stored, and from which processes can access items of case data (case objects).

The case data store is created as part of the TIBCO BPM Enterprise installation process and is a part of the BPM database.

Case Manager

Case Manager is used to manage and update case data.

Using Case Manager, you can perform the following operations:

- Search cases, including performing ad-hoc searches.
- Perform case actions.
- View the work items associated with cases.
- View process instances associated with cases.
- View events that have taken place for a selected case.

You can also manage case documents from the Case Manager. You can store and retrieve documents related to a case. These documents are stored in "case folders", which are related to the case data, in an inbuilt case document store. One case folder is assigned for each case. The folder is automatically created when the case is created. When a case is in scope in a case management application user interface, the case folder is also available, so that not only do you get to see the details of the "order", you also have access to the documents associated with the order (for example, order-update email, purchase order document, custom product specification, and so on).

Administration of TIBCO BPM Enterprise

This section describes the administration and management features of TIBCO BPM Enterprise.

Platform Administration

Platform administration involves the administration of TIBCO BPM Enterprise itself, and of the underlying platform elements that TIBCO BPM Enterprise makes use of.

For example:

- managing additional configuration such as logging, threads or memory usage.
- creating and managing shared resources (resource templates and resource instances) that are used by TIBCO BPM Enterprise, such as LDAP, SMTP and HTTP server connections.

Application Management

Application management involves the management of deployed business process management applications. Browse to **Administrator** to deploy and undeploy an application, and to manage business processes.

Organization Model and Resource Management

Organization model and resource management involves the management of users who need to interact with business process management applications.

Organization model and resource management tasks are performed using the Organization Browser.

At design-time, the business analyst and solution designer create an organization model view and deploy it to TIBCO BPM Enterprise - see [Creating the Organization Model](#). The design-time model is an abstract view that does not include resources (users).

An administrator must manage the deployed organization model, using the Organization Browser to:

- add resources (users) to TIBCO BPM Enterprise from LDAP containers. LDAP containers are a collection of one or more LDAP sources, which contain candidate resources for use with TIBCO BPM Enterprise.
- assign resources obtained from the LDAP sources to groups and/or positions in the deployed organization model.

The tibco-admin User

TIBCO BPM Enterprise is configured with a single internal user, called **tibco-admin**. This is an alias to a user in an LDAP source.

This is the only user who is authorized to login until another user is configured (by using the Organization Browser to create LDAP containers and create and map resources).

i Note: The name of the internal user can be changed if desired, by adding (or changing) the **AdminLdapName** property in the `de.properties` file. See the *TIBCO BPM Enterprise Administration* guide for more information.

System Calendar Management

System calendar management tasks are performed using Calendar .

An administrator can define the following for the system calendar:

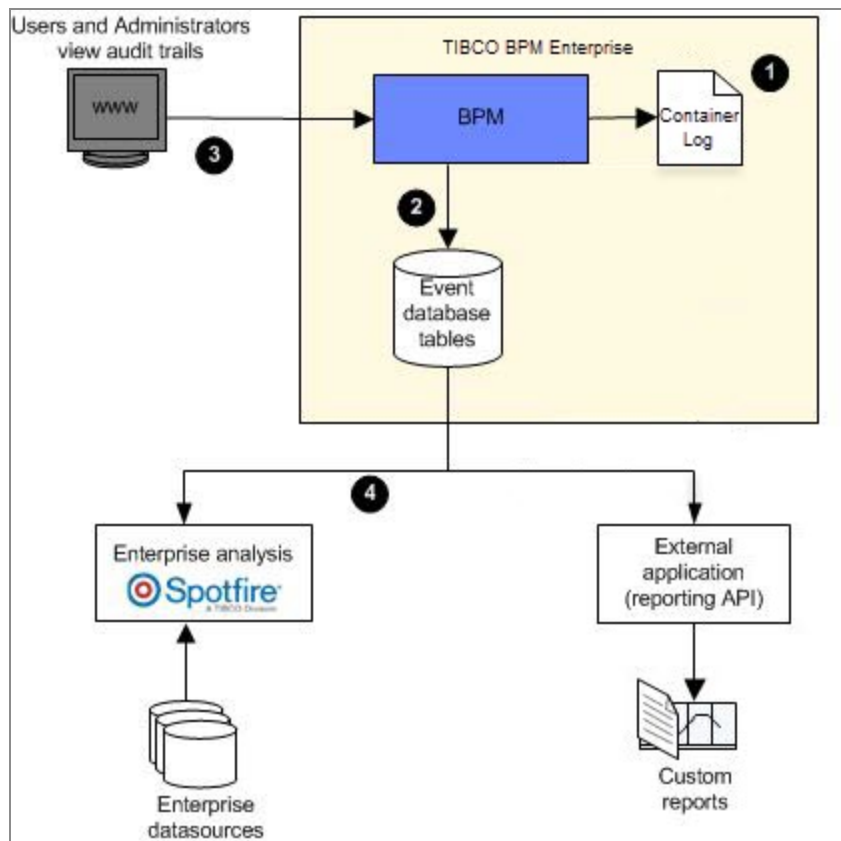
- the working week and associated daily working times.
- working week exceptions - such as public holidays - as date-based working and non-working times.

i Note: The system calendar is intended to model working weeks and, for example, public holidays. It cannot model free/busy periods for individual resources (for example, meetings). See System Calendars for more information about calendars.

Logging and Auditing

This section describes the auditing and logging facilities provided by TIBCO BPM Enterprise.

TIBCO BPM Enterprise Logging and Auditing



(1) TIBCO BPM Enterprise generates events recording its activity, which it logs to local log files.

i Note: Increasingly, security concerns and legal compliance requirements must be considered when determining where and how systems store and use data. TIBCO BPM Enterprise allows you to control what information is logged to ensure that sensitive or confidential information is excluded. See "Controlling What Information is Logged" in *TIBCO BPM Enterprise Administration* for more information.

(2) Selected events are forwarded to and stored in central event database tables.

(3) TIBCO BPM Enterprise client application users can access the event database tables to display audit information for managed objects - processes, work items and so on.

(4) External products and utilities can access TIBCO BPM Enterprise's event database tables. For example:

- Products such as TIBCO Spotfire™ can use the data to perform enterprise analysis tasks by directly accessing the event collection tables of the database. For details of the database schema for the event collection tables, see the *TIBCO BPM Enterprise Concepts Guide*.
- External applications can use the reporting API to generate custom reports.

Managed Objects and Events

Managed objects are tracked by TIBCO BPM Enterprise in the runtime environment. Managed objects include process templates, process instances, work items, organization model entities, resources and TIBCO BPM Enterprise components.

Events happen to managed objects and must be recorded for logging and auditing purposes. For example:

- when a process template is created, deleted or updated.
- when a process instance starts, suspends, resumes or completes.
- when a work item is opened, closed, suspended, submitted or completed.
- when a resource is created, mapped to a group, updated or deleted.

An event records what happens to a managed object, when it happened, and the context that the event happened in.

The event format used by BPM is the TIBCO BPM Enterprise Event format.

i Note: You can configure whether TIBCO BPM Enterprise purges all process instance data when the instance completes, or whether it stores this information. See "Purging Processes through the Command Line Interface" in *TIBCO BPM Enterprise Administrator's Guide* for more information.

Measures

TIBCO BPM Enterprise provides various measures that can be queried to obtain statistical information about process instances and work items.

For example:

- total number of process instances by their current status.
- total number of work items for a process instance by work item status.
- average duration of a process instance or work item

Measures can be:

- queried for all process templates, for an individual process template or for a group of process templates, according to requirements.
- categorized by hours, days, weeks, months or years, over a specified period of time.

For more information about measures and how to access them using TIBCO BPM Enterprise client, see the Audit UI.

Logging

Logging refers to the recording of all events generated by TIBCO BPM Enterprise.

Logging data can be used for numerous purposes, ranging from debugging within a system, through to storage for non-repudiation logs, and all messaging in between.

Logging Levels

Every logged event is categorized with one of a number of severity levels.

- DEBUG or TRACE events provide low-level diagnostic information about the system, which can be used to assist in diagnosing a process or system that is not behaving as expected. DEBUG or TRACE events can generate high volumes of low level output, so are typically turned on and off as required.
- INFO and AUDIT events provide information about what is happening on a normally running system. AUDIT events are those which you can audit centrally, whereas INFO

events are not expected to require central auditing.

- SERVICE events refer to start and end messages from particular services.
- WARN, ERROR or FATAL events provide warnings or errors about the system that need to be relayed to system administrators and users.

Auditing

TIBCO BPM Enterprise auditing involves the collection, correlation and central storage of selected logged events. Administrators can configure which events are forwarded to the central event database tables.

Audit data can be used for numerous purposes - for example, displaying audit trail information to users, or statistical analysis by using external tools.

You can use the Event Collector component to collect and store events that occur during runtime. The component controls how and when the audit data is purged.

The event collector is designed to automatically purge old audit data in order to keep the size of the audit table under control and avoid the need of purging the data manually. The event collector only purges data that is completed, for example, it only purges process audit where an entire process tree is completed or where a case has been deleted. You can also configure the retention period for the completed audit data. For more details, see the [Event Collector](#) topic in the *TIBCO® BPM Enterprise Administration* guide.

Correlation and Sequencing of Audit Data

TIBCO BPM Enterprise uses a distributed, component-based application architecture. Related events can therefore be generated by different sources and at different times.

To deal with events being generated by different sources and at different times, events are logged with:

- a timestamp that includes a standard UTC offset.
- a correlation identifier, which is shared by all events that are part of the same service call.
- correlatable data, such as a managed object ID.

TIBCO BPM Enterprise uses this information to correlate and sequence the events in the event database, so that the data can be meaningfully interpreted.

Viewing Audit Trail

Users and system administrators can view the audit trail for different managed objects. For example, to see how a particular process instance is progressing.

TIBCO BPM Enterprise provides a query API (Event Management Services API) that external systems can use to access the data in the central event database tables.

The API provides methods to execute ad-hoc and stored queries against the central event database tables. For more details about the Event Management Services API, refer to the API Explorer in the TIBCO BPM Enterprise User Interface.

Security Features Provided by TIBCO BPM Enterprise

This section describes the security features provided by TIBCO BPM Enterprise.

Authentication

All access to TIBCO BPM Enterprise requires an authenticated user, whether that access is through run-time user interfaces, web service APIs, deployment or other supported access mechanisms.

Users must be registered with TIBCO BPM Enterprise via the Organization Browser - see Organization Model and Resource Management.

TIBCO BPM Enterprise supports the following methods to authenticate users:

- **Basic authentication** - Basic authentication requires the calling application to provide valid TIBCO BPM Enterprise login credentials when calling a TIBCO BPM Enterprise service. This is the default authentication method used by TIBCO BPM Enterprise.

The type of Basic authentication to use depends on the type of interface you are using:

- **REST API**

An API call to the REST API must include a UsernameToken in the header, which specifies the username and password of the user on whose behalf the call is being made. This uses Security UsernameToken Profile 1.0.

A TIBCO BPM Enterprise LDAP authentication provider resource instance (for example, `amx.bpm.auth.easyAs`) is also required, which validates:

- the supplied username against the BPM organization model.
- the supplied password against the LDAP entity represented by that BPM

user.

i Note: For a secure communication TIBCO BPM Enterprise needs to be front ended with a load balancer or proxy with HTTPS enabled.

The sample client applications provided with TIBCO BPM Enterprise implement direct authentication using a UsernameToken.

- **Single sign-on (SSO) authentication** - With SSO authentication, a user who already has a login session with the client application does not need to provide login credentials again when calling a TIBCO BPM Enterprise service (provided that their credentials are also valid for logging in to TIBCO BPM Enterprise).

Different types of SSO authentication can be used, depending on the API or client you are using:

- SAML Web Profile
- OpenID Connect

For additional information, as well as the APIs and clients that support each of these SSO types, see [Introduction to Single Sign-On Authentication](#).

For additional information, as well as the APIs and clients that support each of these SSO types, see *TIBCO BPM Enterprise Administrator's* guide.

Privilege-Based Authorization

TIBCO BPM Enterprise uses privilege-based authorization.

Privileges represent authorities - what a user is authorized to do, either with respect to TIBCO BPM Enterprise functionality or with respect to an application. See [Privileges](#) for more information.

System Actions

System actions are predefined system tasks that users may want to perform, but which an organization may want to control access to.

System actions provide access to a wide range of functions - for example, work list and work item management, process management and user administration. The following list contains a small sample of the available system actions.

User Admin

Administer users (resources) using the Organization Browser.

View Work List

View another user's work list.

Open Other Resources' Items

Open work items that are currently allocated to other users.

Open Work Item Audit Trail

Open the audit trail for a work item.

Each system action has a system-wide default value, which is either:

- Allowed - The system action can be performed by any user without authorization.
- Denied - The system action cannot be performed by any user unless they have the correct authorization.

This default value can be overridden for individual users by using privileges. In TIBCO Business Studio - BPM Edition , an analyst can assign privileges required to execute a particular system action against specific entities in the organization model. At runtime, only users who hold the required privileges will be able to perform that system action.

i Note: See System Actions and Organization Model Versions for the effect of changing the assignment of privileges between versions of the organization model.

If a set of required privileges for a given system action are assigned to an organizational entity, that setting will apply to all users. Any users not holding those required privileges will be denied access to that system action, for that organizational entity. The default will not be applied.

Scope of System Actions

The scope of a system action is defined by where the privilege required to perform the system action is assigned in the organization model:

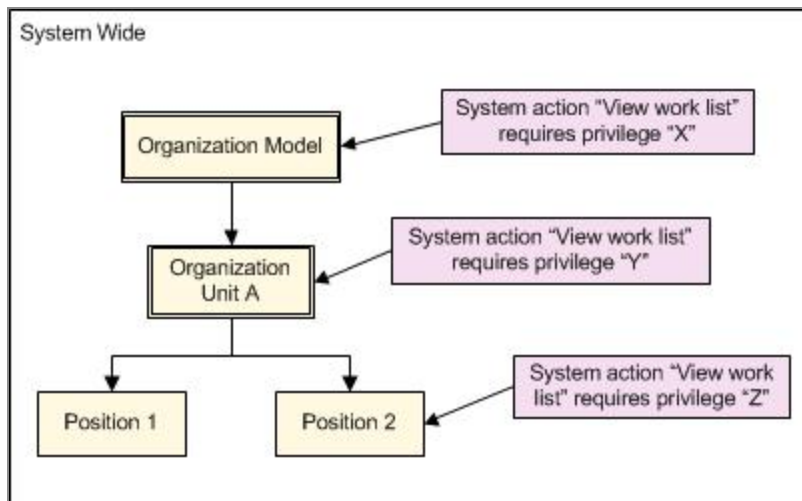
- The default scope of a system action is system-wide. (Either any user or no user can perform the system action.)
- Privileges can be assigned to any system action at the organization model level.
- Privileges can also be assigned to some system actions at the level of the organization unit, position or group.
- A user can always perform certain actions (for example - View Work List and Set Resource Order Filter Criteria) if they are themselves the explicit scope of that action - that is, if they are not just related by position or group.

For example, the following table shows the default value and the possible scope for the system actions mentioned in the preceding section.

System Action	Permitted to all users by default?	Privilege to perform system action:	
		Organization Model?	Organization Unit, Position or Group?
User Admin	Yes	Yes	No
View Work List	No	Yes	Yes
Open Other Resources' Items	No	Yes	Yes
Open Work Item Audit Trail	Yes	Yes	No

If different privileges are assigned to the same system action at different levels in the organization model, each level is checked when determining whether a user has the necessary privilege to be able to perform a particular system action. If the required privileges are not assigned to the lowest level, the parent entity is checked. The same process is followed along the organization model hierarchy up to the default and system-wide value.

For example, in the following diagram the "View work list" system action has been associated with three different privileges, "X", "Y" and "Z", at three different levels - the organization model, organization unit A, and position 2.



This means that a user must hold privilege "X", "Y" or "Z" to view the work list of a user who holds Position 2.

If a user wants to view the work list of a user who holds Position 1, they must hold privilege "X" or "Y". This is because no privilege has been associated with Position 1, so any privileges associated with the parent entity are used instead. If privilege "Y" had not been associated with Organization Unit A, the user would then need privilege "X", defined in the parent Organization Model.

As well as assigning different privileges at different levels, as shown above, qualifiers on the same privilege can be used to refine how access to a particular system action is controlled. (When comparing a required privilege to a held privilege, if either side is not qualified the comparison is positive. If both sides are qualified, the qualifications must match for the comparison to be positive.)

Controlling access to system actions by the application of (user-defined) privileges within the organization model provides an organization with a powerful and completely flexible way to customize and tailor users' access to system functions.

System Actions and Organization Model Versions

When testing whether a user has the authorization to perform a system action, which is that the user holds the required privileges, all major versions of the organization model are considered.

i Note: See [Organization Model Versioning](#) for information on versions.

The privileges required to perform a system action are applied on a per-major-version basis. That is, the same system action may require a different set of privileges in different major versions of the organization model, and each set of required privileges is tested independently. Similarly, it is possible that a position to which a user is mapped may be granted different privileges in different versions of the organization model.

To use a system action, a user must be mapped to a position that has been granted *all* of the privileges that are required in *any* major version of the organization model.

To test this, TIBCO BPM Enterprise examines each major version of the organization model in turn. For each major version, TIBCO BPM Enterprise gathers the required privileges defined in that version for the system action. Then:

- If no required privileges have been defined in a given major version, that version is ignored.
- If required privileges are found in a version, and the user *does not* hold all those privileges, TIBCO BPM Enterprise proceeds to test other major versions.
- If any required privileges are found in a version, and the user holds all those privileges in that version, access to the system action is **granted** and the search stops, no further major versions of the organization model are checked.

If TIBCO BPM Enterprise has checked all the major versions of the organization model that exist, then:

- If a required privilege is defined in any major version, but the user does not qualify for access (see third bullet above), then access to the system action is **denied**.
- If there are no required privileges for the system action in *any* major version, access is granted or denied using the default access for that system action. Some system actions are open to all users by default unless any required privileges have been defined to override this default, while other system actions are denied by default; see "System Actions Reference" in the *TIBCO Business Studio™ - BPM Edition - Application Designer's Guide*.

Different Organization Models with the Same Major Version

All organization models of the same major version - for instance, versions 2.0, 2.1, 2.2, 2.2.1, and 2.3—are merged, and any required privileges set against any system action in any such version are similarly merged. Therefore, to use a system action, a user must hold

all the required privileges that are defined in all organization models of the same major version.

Example of using System Actions to Control Users' Access to System Functions, continued

See: Example of using System Actions to Control Users' Access to System Functions.

In the organization described in the example, changes in the business lead to the introduction of a new version of the organization model, Version 2.0, and the system action "View Work List" no longer requires the Manage Work privilege.

Carol Watts tries to view her colleague Phil Gregg's worklist. In the current version of the organization model, no required privileges are in place to prevent this. Therefore:

- TIBCO BPM Enterprise examines each major version of the organization model in turn. It starts with the current Version 2.0. No required privileges have been defined in this major version, so that version is ignored.
- Testing Version 1.0. however, TIBCO BPM Enterprise finds that a required privilege has been defined, the Manage Work privilege. In that same version, Carol Watts *does not* hold this privilege.
- TIBCO BPM Enterprise therefore does not grant Carol access, but proceeds to look for other major versions to test. Finding none, it refuses Carol access to the "View Work List" system action, even though there is no restriction in the *latest* version of the organization model to prevent her.

Workflow Patterns Reference

This section describes the workflow patterns (Workflow Resource Patterns, Workflow Process Patterns, and Workflow Data Patterns) that are supported in this release of TIBCO BPM Enterprise.



Note: Sample projects demonstrating the use of some of these patterns with TIBCO BPM Enterprise are available on the [TIBCO Access Point](#) website.

Workflow Resource Patterns Support

Workflow resource patterns capture the various ways in which resources are represented and utilized in workflows. By implementing these patterns, TIBCO BPM Enterprise can handle the widest range of possible scenarios for modeling and executing business processes.

See [Workflow Patterns Support](#).

The table lists the workflow resource patterns that are supported in this release of TIBCO BPM Enterprise.

The pattern numbers, names and descriptions are those defined by the Workflow Patterns initiative. See:

- <http://www.workflowpatterns.com/patterns/resource/index.php>
- N. Russell, W.M.P. van der Aalst, A.H.M. ter Hofstede, and D. Edmond. Workflow Resource Patterns: Identification, Representation and Tool Support. (PDF, 206 Kb). In Proceedings of the 17th Conference on Advanced Information Systems Engineering (CAiSE'05), volume 3520 of Lecture Notes in Computer Science, pages 216-232. Springer-Verlag, Berlin, 2005.

Supported workflow resource patterns

Pattern Number	Pattern Name	Pattern Description
<i>Creation Patterns</i>		
1	Direct Distribution	The ability to specify at design time the identity of the resource(s) to which instances of this task will be distributed at runtime.
2	Role-Based Allocation	The ability to specify at design time that a task can only be executed by resources which correspond to a given role.
3	Deferred Distribution	The ability to specify at design-time that the identification of the resource(s) to which instances of this task will be distributed will be decided at runtime.
4	Authorization	The ability to specify the range of privileges that a resource possesses in regard to the execution of a process. Typically, these privileges define the range of actions that a resource can initiate when undertaking work items associated with tasks in a process.
5	Separation of Duties	The ability to specify that two tasks must be executed by different resources in a given case.
7	Retain Familiar	The ability to allocate a work item within a given case to the same resource that undertook a preceding work item. This is applicable where several resources are available to undertake a work item.
8	Capability-Based Distribution	The ability to distribute work items to resources based on specific capabilities that they possess.

Pattern Number	Pattern Name	Pattern Description
		Capabilities (and their associated values) are recorded for individual resources as part of the organization model.
10	Organizational Distribution	The ability to distribute work items to resources based on their position within the organization and their relationship with other resources.
11	Automatic Execution	The ability for an instance of a task to execute without needing to utilize the services of a resource.
<i>Push Patterns</i>		
13	Distribution by Offer – Multiple Resources	The ability to offer a work item to a group of selected resources.
14	Distribution by Allocation - Single Resource	The ability to distribute a work item to a specific resource for execution on a binding basis.
15	Random Allocation	The ability to allocate work items to a selected resource chosen from a group of eligible resources on a random basis.
16	Round Robin Allocation	The ability to allocate a work item to a selected resource chosen from a group of eligible resources on a cyclic basis.
19	Distribution on Enablement	The ability to advertise and allocate work items to resources when they are enabled for execution.
<i>Pull Patterns</i>		

Pattern Number	Pattern Name	Pattern Description
21	Resource-Initiated Allocation	The ability for a resource to commit to undertake a work item without needing to commence working on it immediately.
22	Resource-Initiated Execution - Allocated Work Item	The ability for a resource to commence work on a work item that is allocated to it.
23	Resource-Initiated Execution - Offered Work Item	The ability for a resource to select a work item offered to it and commence work on it immediately.
24	System-Determined Work Queue Content	The ability of the workflow engine to order the content and sequence in which work items are presented to a resource for execution.
25	Resource-Determined Work Queue Content	The ability for resources to specify the format and content of work items listed in the work queue for execution.
26	Selection Autonomy	The ability for resources to select a work item for execution based on its characteristics and their own preferences.
<i>Detour Patterns</i>		
27	Delegation	A resource is able to allocate a work item that was previously allocated to it, but is yet to commence, to another resource.
28	Escalation	The ability of a system to distribute a work item to a resource or group of resources other than those it has previously been distributed to in an attempt to

Pattern Number	Pattern Name	Pattern Description
		expedite the completion of the work item.
29	Deallocation	The ability of a resource (or group of resources) to relinquish a work item which is allocated to it (but not yet commenced) and make it available for distribution to another resource or group of resources.
30	Stateful Reallocation	The ability of a resource to allocate a work item that they are currently executing to another resource without loss of state data.
31	Stateless Reallocation	The ability for a resource to reallocate a work item that it is currently executing to another resource without retention of state.
32	Suspension-Resumption	The ability for a resource to suspend and resume execution of a work item.
33	Skip	The ability for a resource to skip a work item allocated to it and mark the work item as complete.
<i>Auto-Start Patterns</i>		
37	Commencement on Allocation	The ability to commence execution on a work item as soon as it is allocated to a resource.
38	Piled Execution	The ability to initiate the next instance of a task (perhaps in a different case) once the previous one has completed with all associated work items being allocated to the same resource. The transition to Piled Execution mode is at the instigation of an individual resource. Only one resource can be in Piled Execution mode for a given task at any time.

Pattern Number	Pattern Name	Pattern Description
39	Chained Execution	The ability to automatically start the next work item in a case once the previous one has completed. The transition to Chained Execution mode is at the instigation of the resource.
<i>Visibility Patterns</i>		
41	Configurable Allocated Work Item Visibility	The ability to configure the visibility of allocated work items by process participants.
<i>Multiple Resource Patterns</i>		
42	Simultaneous Execution	The ability for a resource to execute more than one work item simultaneously.

Workflow Process Patterns Support

Control-flow patterns capture the various ways in which activities are represented and controlled in workflows. Implementing these patterns gives TIBCO BPM Enterprise the capability to handle the widest range of possible scenarios for modeling and executing processes.

See [Workflow Patterns Support](#).

The table lists the control-flow patterns that are supported in this release of TIBCO BPM Enterprise. The pattern numbers, names and descriptions are those defined by the Workflow Patterns initiative. See:

- <http://www.workflowpatterns.com/patterns/control/index.php>
- N. Russell, A.H.M. ter Hofstede, W.M.P. van der Aalst, and N. Mulyar. Workflow Control-Flow Patterns: A Revised View. BPM Center Report BPM-06-22, BPMcenter.org, 2006.
- W.M.P van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski, and A.P. Barros. Workflow Patterns. Distributed and Parallel Databases, 14(3), pages 5-51, July 2003

Supported control flow patterns

Pattern Number	Pattern Name	Pattern Description
<i>Basic Control Flow Patterns</i>		
1	Sequence	An activity in a workflow process is enabled after the completion of a preceding activity in the same process.
2	Parallel Split	The divergence of a branch into two or more parallel branches, each of which execute concurrently.
3	Synchronization	The convergence of two or more branches into a single subsequent branch such that the thread of control is passed to the subsequent branch when all input branches have been enabled.
4	Exclusive Choice	The divergence of a branch into two or more branches. When the incoming branch is enabled, the thread of control is immediately passed to precisely one of the outgoing branches based on the outcome of a logical expression associated with the branch.
5	Simple Merge	The convergence of two or more branches into a single subsequent branch. Each enablement of an incoming branch results in the thread of control being passed to the subsequent branch.
<i>Advanced Branching and Synchronization Patterns</i>		
6	Multi-Choice	The divergence of a branch into two or more branches. When the incoming branch is enabled, the thread of control is passed to one or more of the outgoing branches based on the outcome of distinct logical expressions associated with each of the branches.

Pattern Number	Pattern Name	Pattern Description
7	Structured Synchronizing Merge	The convergence of two or more branches (which diverged earlier in the process at a uniquely identifiable point) into a single subsequent branch. The thread of control is passed to the subsequent branch when each active incoming branch has been enabled.
8	Multi-Merge	The convergence of two or more branches into a single subsequent branch such that each enablement of an incoming branch results in the thread of control being passed to the subsequent branch.
9	Structured Discriminator	The convergence of two or more branches into a single subsequent branch following a corresponding divergence earlier in the process model. The thread of control is passed to the subsequent branch when the first incoming branch has been enabled. Subsequent enabling of incoming branches does not result in the thread of control being passed on. The discriminator construct resets when all incoming branches have been enabled.
29	Canceling Discriminator	The convergence of two or more branches into a single subsequent branch following one or more corresponding divergences earlier in the process model. The thread of control is passed to the subsequent branch when the first active incoming branch has been enabled. Triggering the Canceling Discriminator also cancels the execution of all of the other incoming branches and resets the construct.
30	Structured Partial Join	The convergence of two or more branches (for example, m) into a single subsequent branch

Pattern Number	Pattern Name	Pattern Description
		<p>following a corresponding divergence earlier in the process model such that the thread of control is passed to the subsequent branch when n of the incoming branches have been enabled where n is less than m. Subsequent enabling of incoming branches do not result in the thread of control being passed on. The join construct resets when all active incoming branches have been enabled. The join occurs in a structured context, that is, there must be a single Parallel Split construct earlier in the process model, with which the join is associated and it must merge all of the branches emanating from the Parallel Split. These branches must either flow from the Parallel Split to the join without any splits or joins, or be structured in form (that is, balanced splits and joins).</p>
32	Canceling Partial Join	<p>The convergence of two or more branches (for example, m) into a single subsequent branch following one or more corresponding divergences earlier in the process model. The thread of control is passed to the subsequent branch when n of the incoming branches have been enabled where n is less than m. Triggering the join also cancels the execution of all of the other incoming branches and resets the construct.</p>
<i>Multiple Instance Patterns</i>		
12	Multiple Instances without Synchronization	<p>Within a given process instance, multiple instances of an activity can be created. These instances are independent of each other and run concurrently. There is no requirement to synchronize them upon completion.</p>

Pattern Number	Pattern Name	Pattern Description
13	Multiple Instances with a priori Design Time Knowledge	Within a given process instance, multiple instances of an activity can be created. The required number of instances is known at design time. These instances are independent of each other and run concurrently. It is necessary to synchronize the activity instances at completion before any subsequent activities can be triggered.
14	Multiple Instances With a priori Run-Time Knowledge	Within a given process instance, multiple instances of an activity can be created. The required number of instances may depend on a number of runtime factors, including state data, resource availability and inter-process communications, but is known before the activity instances must be created. Once initiated, these instances are independent of each other and run concurrently. It is necessary to synchronize the instances at completion before any subsequent activities can be triggered.
15	Multiple Instances without a priori Run-Time Knowledge	Within a given process instance, multiple instances of a task can be created. The required number of instances may depend on a number of runtime factors, including state data, resource availability and inter-process communications and is not known until the final instance has completed. Once initiated, these instances are independent of each other and run concurrently. At any time, whilst instances are running, it is possible for additional instances to be initiated. It is necessary to synchronize the instances at completion before any subsequent tasks can be triggered.

State-based Patterns

Pattern Number	Pattern Name	Pattern Description
16	Deferred Choice	A point in a process where one of several branches is chosen based on interaction with the operating environment. Prior to the decision, all branches represent possible future courses of execution. The decision is made by initiating the first task in one of the branches, that is, there is no explicit choice but rather a race between different branches. After the decision is made, execution alternatives in branches other than the one selected are withdrawn.
18	Milestone	A task is only enabled when the process instance (of which it is part) is in a specific state (typically a parallel branch). The state is assumed to be a specific execution point (also known as a milestone) in the process model. The nominated task can be enabled when the execution point is reached. If the process instance has progressed beyond this state, then the task cannot be enabled now or at any future time (that is, the deadline has expired). Note that the execution does not influence the state itself, that is, unlike normal control-flow dependencies it is a test rather than a trigger.
<i>Cancellation and Force Completion Patterns</i>		
19	Cancel Task	An enabled task is withdrawn prior to it commencing execution. If the task has started, it is disabled and, where possible, the running instance is halted and removed.
20	Cancel Case	A complete process instance is removed. This includes currently executing tasks, those which may execute at some future time and all sub-processes. The process instance is recorded as having

Pattern Number	Pattern Name	Pattern Description
		completed unsuccessfully.
25	Cancel Region	The ability to disable a set of tasks in a process instance. If any of the tasks are already executing (or are currently enabled), then they are withdrawn. The tasks need not be a connected subset of the overall process model.
<i>Iteration Patterns</i>		
10	Arbitrary Cycles	The ability to represent cycles in a process model that have more than one entry or exit point.
21	Structured Loop	The ability to execute a task or sub-process repeatedly. The loop has either a pre-test or post-test condition associated with it that is either evaluated at the beginning or end of the loop to determine whether it should continue. The looping structure has a single entry and exit point.
22	Recursion	The ability of a task to invoke itself during its execution or an ancestor in terms of the overall decomposition structure with which it is associated.
<i>Termination Patterns</i>		
11	Implicit Termination	A given process (or sub-process) instance should terminate when there are no remaining work items that are able to be done either now or at any time in the future.
43	Explicit Termination	A given process (or sub-process) instance should terminate when it reaches a nominated state. Typically this is denoted by a specific end node.

Pattern Number	Pattern Name	Pattern Description
		When this end node is reached, any remaining work in the process instance is canceled and the overall process instance is recorded as having completed successfully, regardless of whether there are any tasks in progress or remaining to be executed.
<i>Trigger Patterns</i>		
23	Transient Trigger	The ability for a task instance to be triggered by a signal from another part of the process or from the external environment. These triggers are transient in nature and are lost if not acted on immediately by the receiving task. A trigger can only be utilized if there is a task instance waiting for it when it is received.
24	Persistent Trigger	The ability for a task to be triggered by a signal from another part of the process or from the external environment. These triggers are persistent in form and are retained by the process until they can be acted upon by the receiving task.

Workflow Data Patterns Support

Data patterns capture the various ways in which data is represented and utilized in workflows. Implementing these patterns gives TIBCO BPM Enterprise the capability to handle the widest range of possible scenarios for modeling and executing data.

See [Workflow Patterns Support](#).

The table lists the data patterns that are supported in this release of TIBCO BPM Enterprise. The pattern numbers, names and descriptions are those defined by the Workflow Patterns initiative. See:

- <http://www.workflowpatterns.com/patterns/data/index.php>
- N. Russell, A.H.M. ter Hofstede, D. Edmond, and W.M.P. van der Aalst. Workflow Data

Patterns: Identification, Representation and Tool Support. (PDF, 281Kb) In Proceedings of the 24th International Conference on Conceptual Modeling (ER 2005), volume 3716 of Lecture Notes in Computer Science, pages 353-368. Springer-Verlag, Berlin, 2005.

Supported Data Patterns

Pattern Number	Pattern Name	Pattern Description
<i>Data Visibility</i>		
1	Task Data	Data elements can be defined by tasks which are accessible only within the context of individual execution instances of that task.
2	Block Data	Block tasks (that is, tasks which can be described in terms of a corresponding subprocess) are able to define data elements which are accessible to each component of the corresponding subprocess.
5	Case Data	Data elements, which are specific to a process instance or case, are supported. They can be accessed by all components of the process when the case is being executed.
7	Workflow Data	Data elements accessible to all components in each and every case of the process and within the context of the process itself are supported.
8	Environment Data	Data elements which exist in the external operating environment can be accessed by the components of processes during execution.
<i>Internal Data Interaction</i>		
9	Task to Task	The ability to communicate data elements between two task instances within the same case. The

Pattern Number	Pattern Name	Pattern Description
		communication of data elements between two tasks is specified in a form that is independent of the task definitions themselves.
10	Block Task to Sub-Workflow Decomposition	The ability to pass data elements from a block task instance to the corresponding subprocess that defines its implementation. Any data elements that are available to a block task are able to be passed to (or be accessed) in the associated subprocess although only a specifically nominated subset of those data elements are actually passed to the subprocess.
11	Sub-Workflow Decomposition to Block Task	The ability to pass data elements from the underlying subprocess back to the corresponding block task. Only nominated data elements defined as part of the subprocess are made available to the (parent) block task.
12	Data Interaction - to Multiple Instance Task	The ability to pass data elements from a preceding task instance to a subsequent task which is able to support multiple execution instances. This may involve passing the data elements to all instances of the multiple instance task or distributing them on a selective basis. The data passing occurs when the multiple instance task is enabled.
13	Data Interaction - from Multiple Instance Task	The ability to pass data elements from a task which supports multiple execution instances to a subsequent task. The data passing occurs at the conclusion of the multiple instance task. It involves aggregating data elements from all instances of the task and passing them to a subsequent task.

Pattern Number	Pattern Name	Pattern Description
14	Data Interaction - Case to Case	The passing of data elements from one case of a process during its execution to another case that is running concurrently.
<i>External Data Interaction</i>		
15	Task to Environment - Push Oriented	The ability of a task to initiate the passing of data elements to a resource or service in the operating environment.
16	Environment to Task - Pull Oriented	The ability of a task to request data elements from resources or services in the operational environment.
19	Data Interaction - Case to Environment - Push-Oriented	<p>The ability of a case to initiate the passing of data elements to a resource or service in the operational environment.</p> <p>Note: Case in this situation means a TIBCO BPM Enterprise process instance.</p>
20	Data Interaction - Environment to Case - Pull-Oriented	<p>The ability of a case to request data from services or resources in the operational environment.</p> <p>Note: Case in this situation means a TIBCO BPM Enterprise process instance.</p>
21	Data Interaction - Environment to Case - Push-Oriented	<p>The ability of a case to accept data elements passed to it from services or resources in the operating environment.</p> <p>Note: Case in this situation means a TIBCO BPM Enterprise process instance.</p>
22	Data Interaction - Case to	The ability of a case to respond to requests for data elements from a service or resource in the operating

Pattern Number	Pattern Name	Pattern Description
	Environment - Pull-Oriented	environment. Note: Case in this situation means a TIBCO BPM Enterprise process instance.
23	Data Interaction - Workflow to Environment - Push-Oriented	The ability of a process environment to pass data elements to resources or services in the operational environment.
24	Data Interaction - Environment to Workflow - Pull-Oriented	The ability of a process environment to request global data elements from external applications.
25	Data Interaction - Environment to Workflow - Push-Oriented	The ability of services or resources in the operating environment to pass global data to a process.
26	Data Interaction - Workflow to Environment - Pull-Oriented	The ability of the process environment to handle requests for global data from external applications.
<i>Data Transfer</i>		
29	Data Transfer - Copy In/Copy Out	The ability of a process component to copy the values of a set of data elements from an external source (either within or outside the process environment) into its address space at the commencement of execution and to copy their final values back at completion.
30	Data Transfer by	The ability to communicate data elements between

Pattern Number	Pattern Name	Pattern Description
	Reference - Unlocked	process components by utilizing a reference to the location of the data element in some mutually accessible location. No concurrency restrictions apply to the shared data element.
32	Data Transformation - Input	The ability to apply a transformation function to a data element prior to it being passed to a process component. The transformation function has access to the same data elements as the receiving process component.
33	Data Transformation - Output	The ability to apply a transformation function to a data element immediately prior to it being passed out of a process component. The transformation function has access to the same data elements as the process component that initiates it.
<i>Data-based Routing</i>		
38	Event-based Task Trigger	An external event is able to initiate a task and to pass data elements over to it.
40	Data-Based Routing	With data-based routing, you can alter the control-flow within a case based on the evaluation of data-based expressions. A data-based routing expression is associated with each outgoing arc of an OR-split or XOR-split. It can be composed of any data-values, expressions and functions available in the process environment providing it can be evaluated at the time the split construct with which it is associated completes. Depending on whether the construct is an XOR-split or OR-split, a mechanism is available to select one or more outgoing arcs to which the thread of control should be passed based on the evaluation of the expressions associated with the arcs.

TIBCO Documentation and Support Services

For information about this product, you can read the documentation, contact TIBCO Support, and join TIBCO Community.

How to Access TIBCO Documentation

Documentation for TIBCO products is available on the [Product Documentation website](#), mainly in HTML and PDF formats.

The [Product Documentation website](#) is updated frequently and is more current than any other documentation included with the product.

Product-Specific Documentation

The documentation for this product is available on the [TIBCO® BPM Enterprise Product Documentation](#) page.

How to Contact Support for TIBCO Products

You can contact the Support team in the following ways:

- To access the Support Knowledge Base and getting personalized content about products you are interested in, visit our [product Support website](#).
- To create a Support case, you must have a valid maintenance or support contract with a Cloud Software Group entity. You also need a username and password to log in to the [product Support website](#). If you do not have a username, you can request one by clicking **Register** on the website.

How to Join TIBCO Community

TIBCO Community is the official channel for TIBCO customers, partners, and employee subject matter experts to share and access their collective experience. TIBCO Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from TIBCO products. In addition, users can submit and vote on feature

requests from within the [TIBCO Ideas Portal](#). For a free registration, go to [TIBCO Community](#).

Legal and Third-Party Notices

SOME CLOUD SOFTWARE GROUP, INC. (“CLOUD SG”) SOFTWARE AND CLOUD SERVICES EMBED, BUNDLE, OR OTHERWISE INCLUDE OTHER SOFTWARE, INCLUDING OTHER CLOUD SG SOFTWARE (COLLECTIVELY, “INCLUDED SOFTWARE”). USE OF INCLUDED SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED CLOUD SG SOFTWARE AND/OR CLOUD SERVICES. THE INCLUDED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER CLOUD SG SOFTWARE AND/OR CLOUD SERVICES OR FOR ANY OTHER PURPOSE.

USE OF CLOUD SG SOFTWARE AND CLOUD SERVICES IS SUBJECT TO THE TERMS AND CONDITIONS OF AN AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER AGREEMENT WHICH IS DISPLAYED WHEN ACCESSING, DOWNLOADING, OR INSTALLING THE SOFTWARE OR CLOUD SERVICES (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH LICENSE AGREEMENT OR CLICKWRAP END USER AGREEMENT, THE LICENSE(S) LOCATED IN THE “LICENSE” FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE SAME TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

This document is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of Cloud Software Group, Inc.

TIBCO, the TIBCO logo, the TIBCO O logo, Business Studio, TIBCO Business Studio, and Spotfire are either registered trademarks or trademarks of Cloud Software Group, Inc. in the United States and/or other countries.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only. You acknowledge that all rights to these third party marks are the exclusive property of their respective owners. Please refer to Cloud SG’s Third Party Trademark Notices (<https://www.cloud.com/legal>) for more information.

This document includes fonts that are licensed under the SIL Open Font License, Version 1.1, which is available at: <https://scripts.sil.org/OFL>

Copyright (c) Paul D. Hunt, with Reserved Font Name Source Sans Pro and Source Code Pro.

Cloud SG software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. See the “readme” file for the availability of a specific version of Cloud SG software on a specific operating system platform.

THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. CLOUD SG MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S), THE PROGRAM(S), AND/OR THE SERVICES DESCRIBED IN THIS DOCUMENT AT ANY TIME WITHOUT NOTICE.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "README" FILES.

This and other products of Cloud SG may be covered by registered patents. For details, please refer to the Virtual Patent Marking document located at <https://www.cloud.com/legal>.

Copyright © 2015-2024. Cloud Software Group, Inc. All Rights Reserved.