GENESYS®
AN ALCATEL-LUCENT COMPANY

intelligent Workload Distribution

Deployment Guide

Copyright

Disclaimer

Feedback

Genesys endeavors to provide accurate and useful documentation for all Genesys products. To achieve this goal, the documentation group welcomes your comments and suggestions regarding any aspect of Genesys user documentation. Send your comments by e-mail to: techpubs.webadmin@genesyslab.com.

Trademarks and Registered Trademarks

Revision History

Version	Issued	Description
1.0	18.06.2007	Initial GTL release 1.0
1.1	01.10.2007	GTL release 1.1
1.2	16.11.2007	GTL release 1.2
1.3	22.04.2008	GTL release 1.3
1.4	28.07.2008	GTL release 1.4
1.6.6	16.03.2009	GTL release 1.6.6
7.6.1.1	14.04.2009	iWD release 7.6.1

Document Version: 76iwd_dep_07-2009_v7.6.101.00

Contents

Preface

Welcome to the *intelligent Workload Distribution Deployment Guide*. This document describes how to install and configure *Genesys intelligent Workload Distribution (iWD)*.

Intended Audience

This guide has two primary audiences:

- IT staff responsible for the iWD installation and configuration.
- Business analysts responsible for process and contract configuration.

Related Resources

See the following additional resources for more information:

- *iWD Overview*
- *iWD Manager Guide*

Document Conventions

This document uses the following stylistic and typographical conventions, which serve to identify specific types of information:

Type Styles

Italic

In this document, italic text denotes emphasis, document titles, definitions of (or first references to) unfamiliar terms, and mathematical variables.

For example:

- Please consult the *intelligent Workload Distribution Manager User Guide* for more information.
- *Do not use* this value for this option.
- The formula, $x + 1 = 7$ where x stands for . . .

Monospace Font

A monospace font, which resembles `teletype` or `typewriter` text, is used for all programming identifiers and graphical user interface (GUI) elements.

This convention includes the names of directories, files, folders, configuration objects, paths, scripts, dialog boxes, options, fields, text and list boxes, operational modes, all buttons (including radio buttons), check boxes, commands, tabs, CTI events, and error messages; the values of options; logical arguments and command syntax; and code samples.

For example:

- Select the `Default` check box.
- Click the `Edit` button.
- In the `Properties` dialog box, enter the value for the host server in your environment.
- Click `OK` to exit the `Properties` dialog box.

Monospace is also used for any text that users must manually enter during a configuration or installation procedure, or on a command line:

For example:

- Enter `exit` on the command line.

Deployment Overview

iWD deployment consists of three phases:

- Installation: Setup of iWD applications on application servers and preparation of databases
- System Configuration: Configuration of iWD tenants, solutions, and services
- Business Logic Configuration: Configuration of iWD contracts, processes, rules, and other business-logic related configuration

Installation Overview

Installation is the initial iWD deployment phase that results into a fully functional iWD Manager application and prepared iWD runtime nodes. iWD Manager will be used for the rest of the deployment configuration, while runtime nodes are containers in which the iWD runtime services will run.

The installation phase requires knowledge of overall system infrastructure in which iWD is being deployed, such Java application server and database configuration. This phase should typically be implemented by IT personnel.

iWD installation consists of two steps:

- Application Installation
- Database Preparation

For a detailed installation reference, see Installation.

Application Installation

When the iWD operational databases are prepared, iWD Manager application and iWD runtime nodes can be installed. Both of these components run on a Java application server.

Note: the current release of iWD supports Apache Tomcat 5.5 application server

A basic iWD deployment utilizes a single instance of iWD Manager, as well as single instance of iWD runtime node. In a more complex scenarios, such as multi-tenant, high-volume or high-availability deployments, multiple iWD runtime node instances can be installed. Such deployment provides a more controlled resource partitioning and allows load distribution across multiple physical servers.

Installation of iWD runtime node is as simple as copying runtime node application folder to application server. The rest of runtime node configuration, such as which services will run on this node, is performed via iWD Manager.

For iWD Manager application, besides copying it over to the application server, two things need to be configured:

- Connection to configuration database
- Authentication mechanism

Note: the current release of iWD supports Active Directory authentication mechanism.

This configuration is stored in a configuration files (contained in the iWD Manager application package) and can be updated via a basic text editor, such as Notepad.

Database Preparation

The first step of iWD installation is the database preparation. A basic iWD deployment utilizes two operational databases:

- Configuration: Stores iWD system and business-logic configuration such as services, processes, and rules
- Runtime: Stores iWD tasks

In more complex scenarios, such as multi-tenant deployment, iWD might utilize multiple runtime databases (typically, one per tenant). This allows implementation of a physical separation of data among tenants, as well as load distribution across multiple databases. Optionally, separate archiving of databases can also be set up to optimize performance of primary runtime databases.

Note: the current iWD release supports MySQL 5.0 database engine

Preparation of the database requires two actions to be performed manually via the database's administrative interface:

- Creation of a database
- Creation of a database user account or accounts

The rest of the database setup, such as creation of tables and indexes, is performed automatically by iWD.

System Configuration Overview

iWD system configuration results in a fully set-up iWD system infrastructure that consists of tenants, solutions, and services that enable iWD functionality. All system configuration is performed via the iWD Manager application.

The iWDsystem-configuration phase requires knowledge of overall system infrastructure in which iWD is being deployed, such as Java application server and database configuration. This phase should typically be implemented by IT personnel.

The following topics provide an overview of different system-configuration aspects:

- Tenants
- Solutions

- Core iWD Services
- Capture
- Distribution
- Reporting

For a detailed system-configuration reference, see General Configuration, Modules and Components and Solutions and Services.

Tenants

iWD configuration supports a hierarchical multi-tenancy. iWD automatically creates a root tenant that is named "System" by default. Although it is possible to configure iWD solutions, services, and business logic directly at the "System" tenant level, it is recommended that a subtenant be created for that. In iWD terminology, such a subtenant is called a managed tenant.

For a multi-tenant environment, multiple managed tenants should be created. In more complex scenarios, managed tenants can be created not just at the root level ("System" tenant), but also under another managed tenant.

The tenant concept in iWD has the following primary purposes:

- Service and data separation.
- Security management: Who can do what is specified per tenant.
- Module management: Each tenant can have access to different iWD modules (modules represent iWD functionality).

iWD supports the following tenant-separation levels:

	Services	**Data**
Configuration	Shared or Physical	Logical
Runtime	Logical or Physical	Logical or Physical
Reporting	Logical or Physical	Logical or Physical

Note: Although both Logical and Physical separations are supported for runtime and reporting services and data, it is recommended to always implement physical separation; in other words, services run on different application server instances, and data is stored in separate databases.

A user configuring the system via iWD Manager can have access to one or more tenants. The access is defined by the security policy that is configured per tenant. The policy allows definition of arbitrary numbers of user roles, where each role is mapped to a single directory group and has a number of associated iWD permissions. The actual permissions that a user has then are determined, based on the groups in the directory to which the user belongs.

The modules that are accessible to the tenant are managed by the parent. Module by itself represents a specific iWD functionality, such as "XML Capture" or "Archiving." The specific functionality is implemented as a collection of components that can be services or business-logic templates (such as rules templates).

Before creating any managed tenants and configuring solutions and services, the system configuration must be imported at the system tenant level. This configuration sets up all of the available iWD modules, so that they can be assigned to managed tenants and used within solutions.

Modules and Components

Modules and components define iWD functionality:

- Component is an atomic object that provides specific iWD function. The following components types are used in iWD:
  - Service Template: This component represents an iWD service implementing specific functionality. Service templates are preconfigured and thus are not changeable in iWD Manager. Service instances that are based on service templates, however, are configurable.
  - Rule Template
- Module is a group of components.

Solutions

A solution in iWD represents a runtime environment, which is composed of:

- Runtime nodes: iWD runtime application instances within the Java application server in which services are being run.
- Runtime database. The database in which task data and audit will be stored.
- Services: Services that enable iWD functionality, such as classification, prioritization, capture, and distribution points.
- Business-logic configuration: iWD contacts, process, rules, and more.

Multiple solution instances can be configured per tenant, if necessary (for example, "Primary" and "Archive").

Runtime nodes and runtime database require simple preparation during installation, as described in Installation Overview.

Runtime Nodes

Runtime nodes are containers for iWD services that enable the necessary service-management infrastructure. Physically, runtime nodes are instances of the iWD runtime application and are run within a Java Application Server. Services that run within a runtime node are configured in (and managed through) the iWD Manager application. Internally, the runtime nodes are accessed via the Webservice interface, which is located by the runtime node context URL, as configured in iWD Manager.

Services

Core iWD Services

Core iWD services are mandatory pieces of any iWD solution and enable the core iWD functionality:

- iWD Database: Connection to runtime database.
- Audit Service: Task auditing.
- Broker Service: Brokers communication between various iWD services.
- Rules Service: A mediator between iWD services and rule-engine implementation; allows iWD services to execute various rules.
- Classification Service: Assigns a task to contract/process, and performs calculation of various task attributes (such as business value and due time).
- Prioritization Service: Calculates/recalculates task priority.
- Archiving Service: Exports expired tasks, and deletes them from the runtime database.

Capture Points

Capture points represent an interface for feeding tasks to iWD. A number of different capture points are available:

- Webservice Capture Point: Provides ability for third-party applications to submit tasks to iWD via Webservice. It also provides interface for managing submitted tasks, such as holding, restarting, and more.
- Database Capture Point: Captures tasks directly from database table(s). Capture configuration for this is based on SQL.
- XML File Capture Point: Captures tasks from XML files. The files should comply with the iWD Task XML schema.

At least one capture point must be configured in a iWD solution. Each capture-point type can locations). also have multiple instances within a solution (for example, to capture XML files from multiple).

Distribution Points

Distribution points represent a iWD interface to task routing systems (specifically, to Genesys CIM platform). The Genesys Distribution Point enables blending of tasks with voice calls and routing to agents in a common manner. The distribution point is supported by a number of helper services, which enable connections to various Genesys servers:

- Genesys Configuration Server Connector
- Genesys Interaction Server Connector
- Genesys TServer Connector

The last one is optional and is not directly used by a distribution point; however, it is required for Genesys Statistics Adapter (see the Reporting section that follows).

The tasks are submitted to Genesys as Genesys Open Media interactions. Each distribution point is configured to submit tasks to a particular Genesys Business Process queue.

Any attribute that is attached to a task in iWD can be attached to a Genesys interaction, as well as any attribute that is attached to an interaction in Genesys can be attached back to a iWD task. What attributes are actually attached is configured via wildcard-based filters per each distribution point.

Reporting

Extended Statistics Service allows performance of custom aggregations on data and sends the resulting statistics via Genesys Statistics Adapter to Genesys Statistics Server. Statistics then can be viewed in CCPulse+. Genesys Statistics Adapter is dependent on iWD TServer Connector and uses a User Event mechanism to deliver statistics to Genesys Stat Server.

The historical reporting is enabled by a number of ETL jobs that transform and load iWD runtime data into a separate reporting database that also is called iWD Datamart. The datamart essentially is a set of star schemas that contain historical iWD data that is optimized for reporting. The ETL jobs are set up as scheduled services in iWD:

Job	Description	Default Schedule
GTL DM Initialize	Initializes datamart and loads static dimensions	Run once
GTL DM Load Config	Loads configuration data into dimensions	15 min
GTL DM Load Intraday	Loads runtime data into intraday fact tables and updates dimensions	15 min
GTL DM Aggr Intraday	Aggregates intraday facts	15 min
GTL DM Load Historical	Loads data from intradat fact tables into historical fact tables	24 h
GTL DM Aggr Historical	Aggregates historical facts	24 h
GTL DM Maintain	Purges expired historical facts	24 h

Business Logic Configuration Overview

The iWD business-logic configuration phase is where iWD business context is introduced. This includes definition of contracts, process, business rules, and business calendars. After this phase, iWD is fully functional and can start processing tasks.

The iWD system-configuration phase requires knowledge of business context for tasks that wi ll be handled by iWD. This includes business processes, service-level agreements, and other factors that influence task-handling logic.

The following topics provide an overview of different business-configuration aspects:

- Contracts and Processes
- Rules
- Task Classification
- Task Prioritization
- Business Calendars

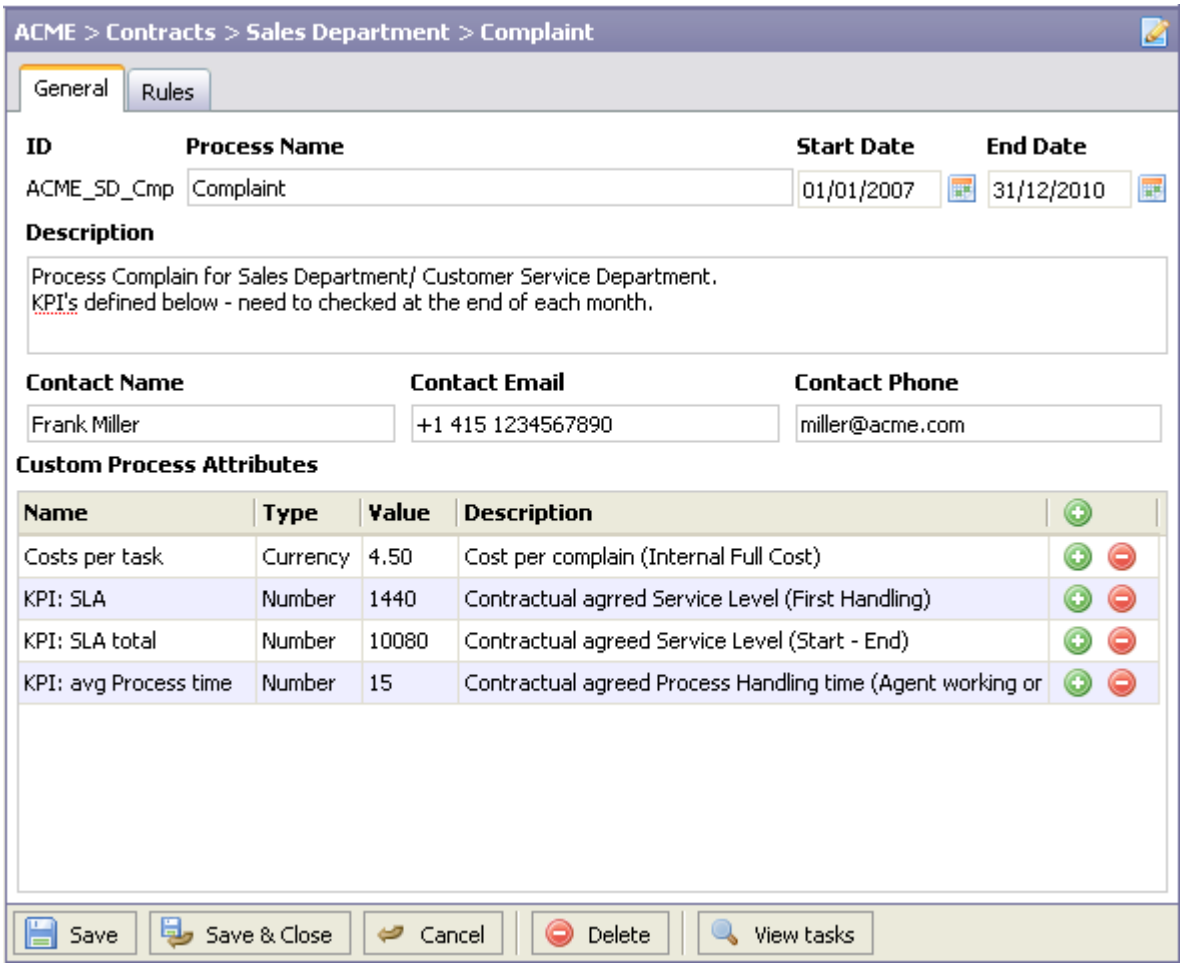
For a detailed business configuration reference, see Contracts and Processes.

Contracts & Processes

iWD contracts and processes represent the primary means of task classification in iWD. The iWD process allows mapping of tasks from multiple systems, each having its own process definitions for a set of normalized business processes in iWD. iWD contract represents a logical grouping of processes, such as an enterprise's department.

Contracts and processes allow definition of task-handling business rules that are specific to a context. For more information on rules, see both Classification Rules and Prioritization Rules.

Each contract and process also allows definition of an arbitrary number of custom attributes that provide additional enterprise-specific context for reporting purposes.



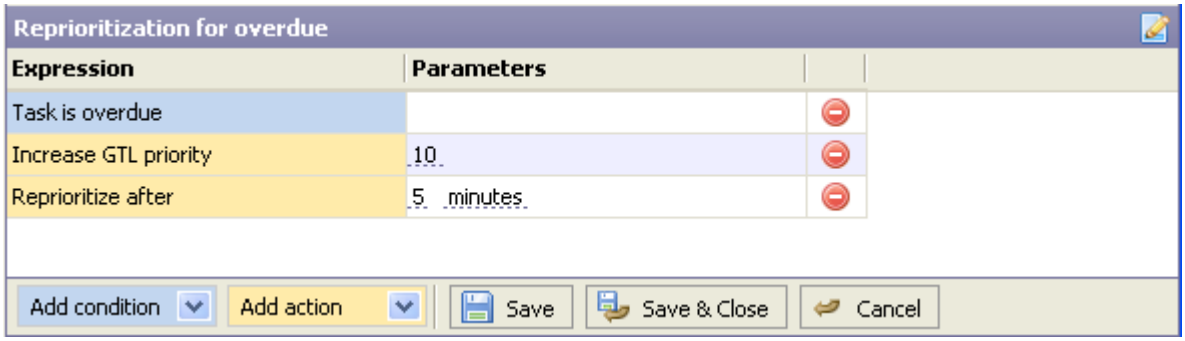
Example Process

Rules

While contracts and processes define task-handling business context, rules define task-handling business logic. Generally, a rule is represented by one or more conditions and one or more actions. If all of the conditions are true, all of the actions will be executed. If any condition is not true, none of the actions will be executed.

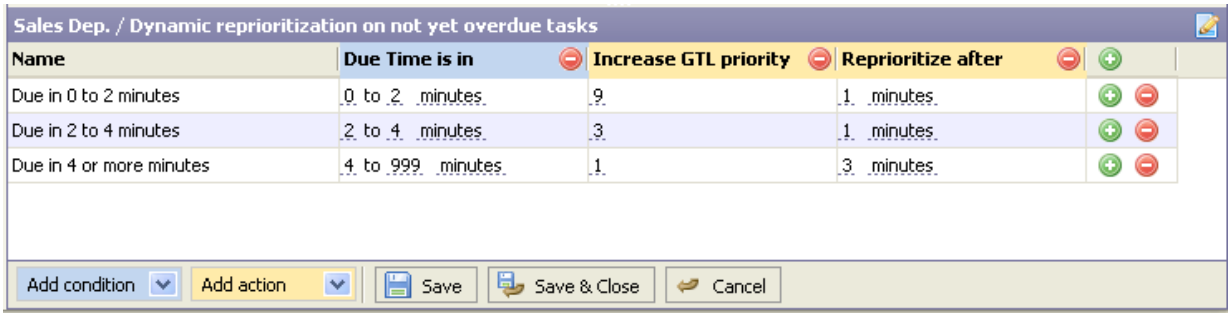
An important aspect of rules in iWD is separation of implementation details from business-level logic. This allows expression of rules in an easy-to-understand human language, such as, "If task is due in 10 or more minutes, increase priority by 10." The implementation details are hidden in rule templates, and users who configure business logic deal only with high-level logical expressions.

Rules can be defined in one of two ways: as a linear rule or as a decision table. Linear rules are intended for complex rules that have many conditions. Each condition or action is represented by a single line in the rule.



Linear Rule Example

Decision tables represent a more compact form of rule representation; however, they might not be as as well suited to complex rules. In a decision table, multiple rules are grouped together, so that each each condition or action is represented by a column in a table, and each rule represents a row.



Decision Table Example

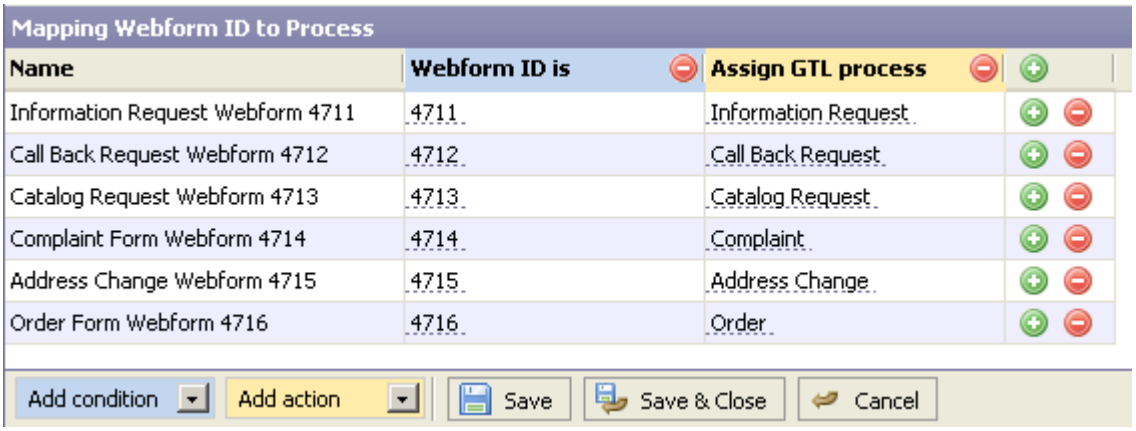
Task Classification

The primary purpose of task classification is to associate a task with a business context, such as an iWD process. Additionally, classification can calculate different task attributes, such as business value and due time.

Task-classification logic is expressed via business rules that can be defined for three different contexts:

- Capture point
- Contract
- Process

If rules are defined for more than one context, they are applied in sequence, as previously listed. The task-process selection is a mandatory classification action that needs to be configured for capture-point context.

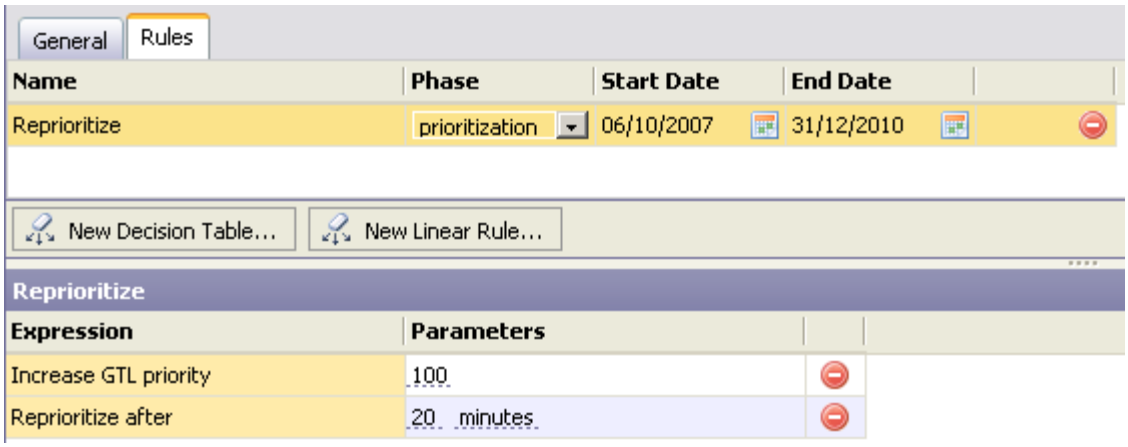


After a process context has been selected for a task, additional rules are applied that have been defined for the selected process and its associated contract.

Task Prioritization

The primary purpose of task prioritization is dynamic priority calculation, where dynamic means that that task priority can be recalculated multiple times during the task life cycle. Similarly to task classification, prioritization logic is expressed via rules.

Prioritization rules are initially applied immediately after contract-/process-level classificationrules, and then reapplied after a specified reprioritization period. The reprioritization period is expressed in the same way as any other rule action, as illustrated in the following example:



If reprioritization period is not applied to a task during the prioritization phase, the task does not undergo further reprioritization.

Business Calendars

A business calendar is a set of rules that define working days and hours, as well as holidays that apply to the business. In its simplest form, the business calendar would consist of definitions for both working week and working hours that apply to all working days. A definition of a working

week from Monday to Friday, where each day starts at 9:00 AM and ends at 5:00 PM, is a classic example of a simple business calendar. If necessary, exceptions from the usual working schedule (public holidays, business-specific holidays, nonstandard working hours, and so on) can be added to the business calendar. Business calendars can be used in iWD rules to perform date and time calculations that take into account the working schedule of the business.

Installation

Preparing for Installation

Installation Prerequisites

Before you start the installation, ensure that the environment meets the following prerequisites:

- One of the following Java application servers is installed on the computer(s) where iWD will be installed:
  - Apache Tomcat 5.5
- One of the following database servers is available:
  - MySQL 5.0
- One of the following LDAP providers is available:
  - Active Directory (Windows 2000 or newer)
- Java SDK 1.5.0.09 or newer is installed.

Installation Package Structure

iWD software is packaged as ZIP archives that have the following structure:

- `gtl-<version>.zip`
  - `webapps`
    - `evo.cmc.web`: iWD Manager application
    - `evo.runtime.web`: iWD runtime node
  - `config`
    - `gtl.xml`: iWD core configuration
    - `gtl_reporting.xml`: iWD Statistics and iWD Datamart ETL services and iWD reports
- `gtl-examples-<version>.zip`: iWD examples
  - `webapps`
    - evo.acme: example PHP web application that submits tasks to iWD
  - `acme`: example tenant

- `data`: example data files for XML capture
- `acme_gen.xml`: example iWD tenant configuration with Genesys connections
- `acme_sim.xml`: example iWD tenant configuration with simulated distribution
- `captureExample.php`: example script that submits tasks viaiWD Webservice Capture Point
- `readme.txt`: configuration instructions for Genesys environoment
- `ccpulse_readme.txt`: configuration instructions for Genesys CCPulse+ application
- `acme.stg`: example Genesys CCPulse+ templates

- `gtl-doc-<version.zip`: iWD documentation
  - `guidesiWD`
    - `Overview.pdf`
    - `iWD Deployment Guide.pdf`
  - `api_ref`
    - `gtl_webservice_capture_api`: iWD Webservice Capture Point API reference guide

iWD Manager Installation

iWD Manager Database Preparation

The iWD Manager application utilizes a database in which all of the configuration data is stored. This database must be created before the iWD Manager application can be installed and accessed.

Note: The current release of iWD Manager supports MySQL 5.0 database

To create a database:

1) Ensure the database server is running.

2) Logon to database server's administrative interface (such as MySQL Administrator).

3) Create a new database user account (for example, "`iwdmanageruser`"). This account will be used by iWD Manager to access the database.

4) Create a new database (for example, "`iwdmanagerdb`"). This database will be used by iWD Manager to store configuration data.

5) Ensure that the user who was created in Step 3 has full access to the database, including CREATE TABLE and CREATE INDEX permissions.

iWD Manager Application Installation

The iWD Manager application must be installed on a Java application server before it can be accessed.

Note: The current release of iWD Manager supports the Apache Tomcat 5.5 application server.

To install the iWD Manager application:

1) Ensure that the application server is stopped.

2) Ensure that the database that was prepared earlier is accessible.

3) Ensure that the directory service is accessible.

Note: The current release of iWD Manager supports Active Directory (Windows 2000 or newer).

4) Copy the evo.cmc.web application from the installation package to the server application directory(for example: in Tomcat, this directory is called "`webapps`"). Optionally, the application directory can be renamed, which changes the URL by which the application will be accessible.

5) Open the `evo.cmc.web\WEB-INF\classes\META-INF\evo.cmcDatabase.properties` file in a text editor, such as Notepad, and modify the database connection parameters, so that they point to the configuration database that was prepared earlier. Also, set the evo.host parameter to a unique instance identifier string (required by scheduled deployment). For example:

```
evo.cmcDatabase.url=jdbc:mysql://dbserver:3306/iwdmanagerdb
evo.cmcDatabase.username=iwdmanageruser
evo.cmcDatabase.password=iwdmanager456
evo.cmcDatabase.driverClassName=com.mysql.jdbc.Driver
evo.cmcDatabase.hibernateDialect=org.hibernate.dialect.MySQL5Dialect
evo.host=iwdmanager_host_01
```

6) Open the `evo.cmc.web\WEB-INF\classes\META-INF\evo.cmcevo.securityService.service.xml`file, and modify the directory connection parameters. Only the following section of the file has to be updated:

```
<property name="url" value="ldap://directory_server:389"/>
<property name="base">
    <list>
        <value>DC=company,DC=com</value>
    </list>
</property>
<property name="username" value="CN=ldap,CN=Users,DC=company,DC=com"/>
<property name="password" value="password123"/>
```

7) Start the application server, and wait for initialization to complete.

8) iWD Manager is now accessible via the `http://applicationserver:8080/evo.cmc.web/` URL, where application server is the name of the server in which iWD Manager is installed; 8080 is the application server port; and evo.cmc.web is the name of the application folder, as set in Step 4. Log in to iWD Manager by using the built-in system-administrator account (user name: system; password: evo).

Note: The system-administrator account password can be changed in the `evo.cmc.web\WEBINF\ classes\META-INF\localUsers.properties` file.

9) iWD Manager is packaged with database-creation/-upgrade scripts that ease both initialization and upgrade of the configuration database. Any required operation is detected automatically during login, and the user is presented with a database-creation/-upgrade dialog box.

10) Open Import/Export in the General section of System tenant, and import the `gtl.xml` (and, optionally `ic.xml`) file.

iWD Manager is now set up and ready for the configuration of tenants and solutions.

iWD Runtime Installation

Runtime Database Preparation

iWD runtime utilizes a runtime database in which all of the task data is stored. Technically, a single database can be used for multiple tenants and solutions; however, it is recommended always to have a dedicated database for each solution.

Note: The current release of iWD runtime supports the MySQL 5.0 database.

To create a database:

1) Ensure that the database server is running.

2) Log on to the database-server administrative interface, such as MySQL Administrator.

3) Create a new database user account (for example, "`gtluser01`"). This account will be used by iWD runtime to access the database.

4) Create a new database (for example, "`gtldb01`"). This database will be used by iWD runtime to to store task data.

5) Ensure that the user who was created in Step 3 has full access to the database, including CREATE TABLE and CREATE INDEX permissions.

Runtime Node Installation

The iWD runtime node must be installed on a Java application before it can run iWD services. Technically, a single runtime node can be used to run services for multiple tenants and solutions; however, it is recommended always to have one or more dedicated runtime nodes for each

solution. Multiple runtime nodes can be set up to distribute load by running different services on different physical computers.

Note: The current release of iWD runtime supports the Apache Tomcat 5.5 application server.

To install the iWD runtime node:

1) Ensure that the application server is stopped.

2) Copy the `evo.runtime.web` application from the installation package to the server application directory (for example: in Tomcat, this directory is called "`webapps`"). Optionally, the application directory can be renamed, which changes the URL by which the runtime node will be accessible.

3) Start the application server, and wait for initialization to complete.

The iWD runtime node is now set up and ready to run services. To do so, the runtime node must be registered under the solution; also, solution configuration must be deployed. See Solutions and Services for more information.

Configuration

iWD Manager Overview

iWD Manager is a thin-client web application for configuring and managing the intelligent Workload Distribution. It can be accessed via Firefox (version 2.0) or Internet Explorer (version 6.0 SP2 or 7.0) web browsers. The application URL is installation-specific (see iWD Manager Installation). The following is an example of a typical iWD Manager URL:

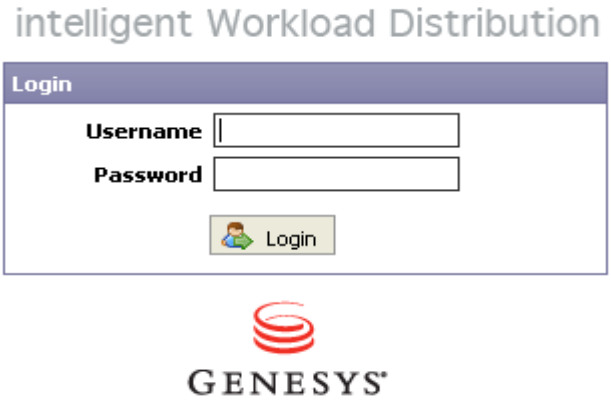
http://server:8080/evo.cmc.web/

Note: Firefox 2.0 is currently recommended browser for best performance and usability.

Note: Browser cookies must be enabled for iWD Manager application to function correctly.

iWD Manager Login

Before accessing iWD Manager functionality, the user is prompted for user name and password via the Login screen:



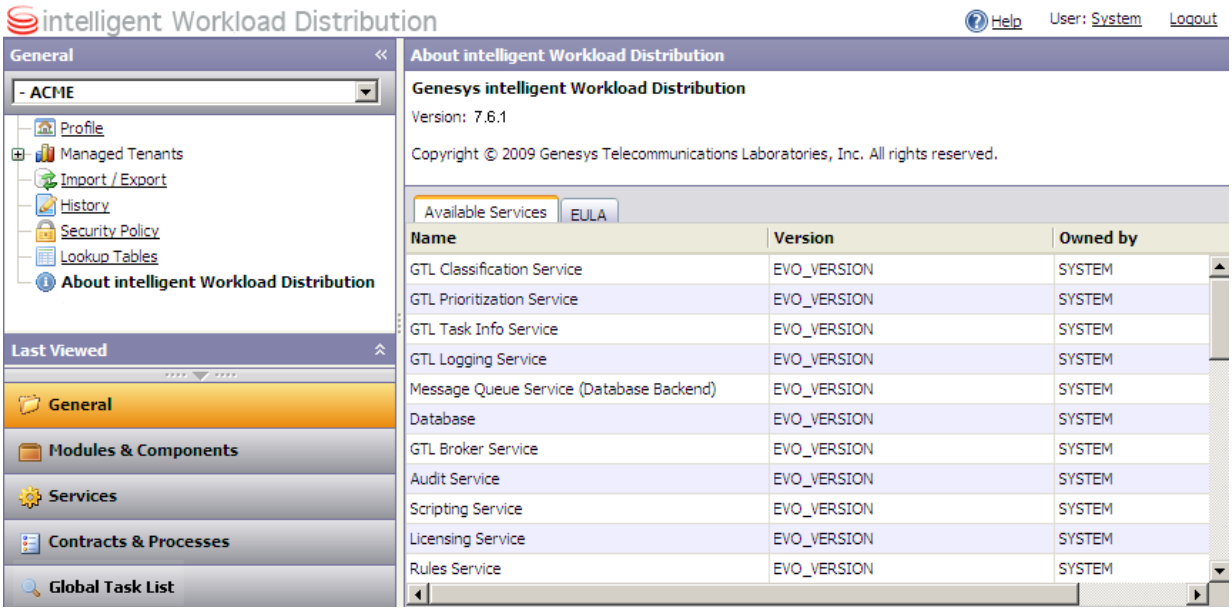
The authentication mechanism (in other words, where the user names and passwords are configured) depends on deployment configuration; for example, it could be configured to

authenticate users against Active Directory (this is described in more detail in iWD Manager Installation).

iWD Manager User Interface

The iWD Manager UI is composed of three main application areas:

- Header bar: Located on top, it shows general information and actions, such as logged-in user name and logout action.
- Navigation panel: Located on the left-hand side, its purpose is as its name implies: navigation among the various configuration/management objects accessible via iWD Manager.
- Details view: The largest area that is on the screen, it is where all of the details about a selected object can be viewed and modified.



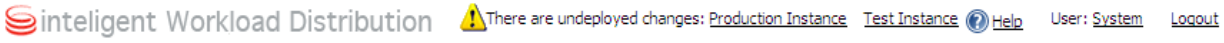
Header Bar

The iWD Manager Header bar provides access to the following information and actions (left to right):

- Application logo.
- Undeployed-changes notification: This is shown only when there are changes in the current tenant's configuration that have not been deployed to runtime; such changes are not

active. The notification also includes direct links to a deployment screen for each affected solution.

- Help: Opens context-sensitive help.
- User: Displays the user name of the currently logged-in user.
- Logout: Logs the user out of the iWD Manager application.

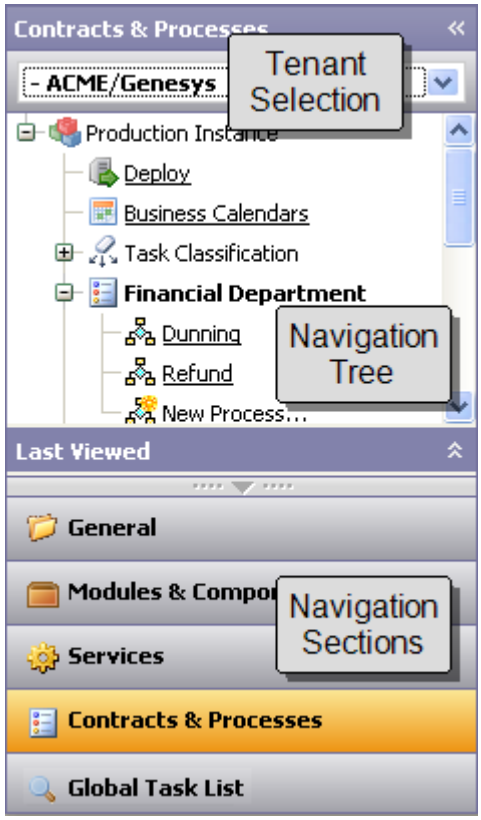


Navigation Panel

The iWD Manager Navigation panel consists of the following UI objects:

- Tenant selection: On top of the navigation area; allows switching among iWD tenants
- Navigation sections: At bottom of the navigation area; allows switching among top-level navigation sections
- Navigation tree: In the middle, provides access to configuration objects for the selected tenant navigation section
- Last Viewed: Between the navigation tree and its sections, provides quick access to the last-viewed configuration objects

Navigation area can be temporary hidden by clicking hide icon on the top-right corner; a hidden area can be made visible again by clicking show icon.



Tenant Selection

iWD configuration supports a hierarchical multi-tenancy, in which each iWD Manager user can have access to one or more tenants. If more than one tenant is accessible to the logged-in user, tenant selection allows switching among them.

All configurations shown in different Navigation sections always are specific to the selected tenant.

Navigation Sections

iWD Manager provides the following navigation sections:

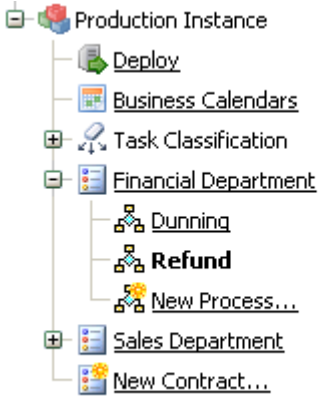
- General: General tenant configuration and actions, such us security policy and configuration import/export. For a detailed description, see General Configuration.
- Modules & Components: Modules and components that are owned by or accessible to tenant. For a detailed description, see Modules & Components.
- Services: Solutions and services that are configured for a tenant. For a detailed description, see Solutions & Services.
- Contracts & Processes: iWD business configuration, such as contracts, processes, and rules. For a detailed description, see Solutions & Services.
- Global Task List: iWD management views, dashboards, and reports. For a detailed description, see iWD Manager Guide.

Depending on the application permissions of a logged-in user, only some of these sections might be visible. By default, upon logging in to iWD Manager, the General section is selected. To switch to another section, just click it. The selected section is also displayed on the top of the Navigation panel (just above the tenant selection).

Navigation can also be switched to an alternative, compact mode by clicking the collapse icon (on top). In compact mode, each navigation section will be represented by an icon.

To restore normal navigation mode again, click the expand icon (on top).

Navigation Tree

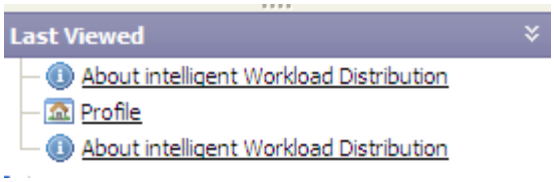


The navigation tree displays iWD configuration and management objects for the selected tenant and navigation section in a tree structure. Each selectable object is underlined and can be selected by clicking it. When it is selected, the object is marked in bold, and the corresponding details are displayed in the iWD Manager Details area.

For those objects that support creation of new instances, a "New..." action also is present in the tree (such as "New Process..." and "New Contract...", in the example at right). When a "New..." action is selected, details for the new object instance can be entered and saved in the iWD Manager Details area.

Last Viewed

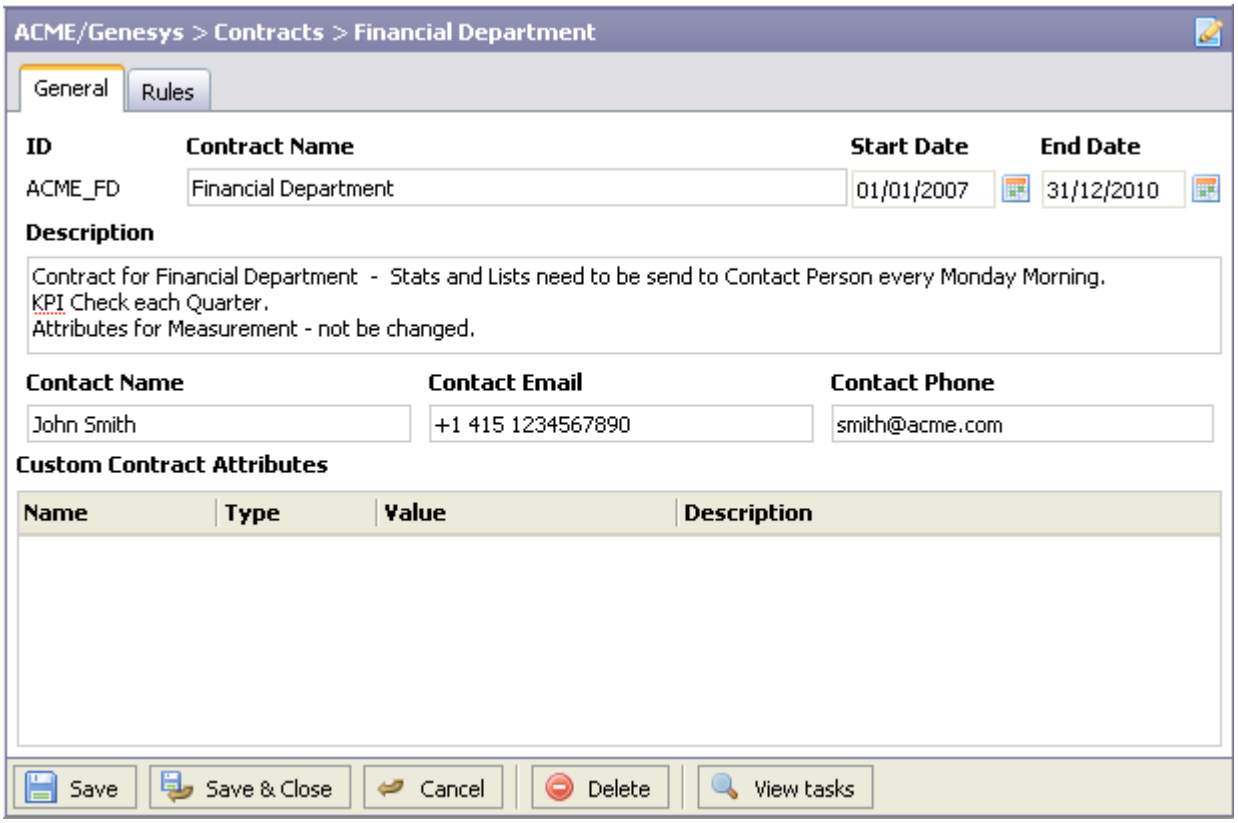
By default, the Last Viewed list is hidden. It can be displayed and hidden again, similarly to the whole Navigation section.



When it is visible, Last Viewed lists the five last-viewed configuration objects. The details for the listed objects can be opened directly from the Last Viewed list by clicking the corresponding item.

Details View

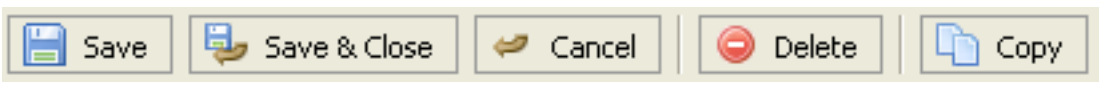
Details view shows and (for some objects) allows modification of information that is related to the object that is selected in Navigation. The following picture illustrates an example Details view:



Object Actions

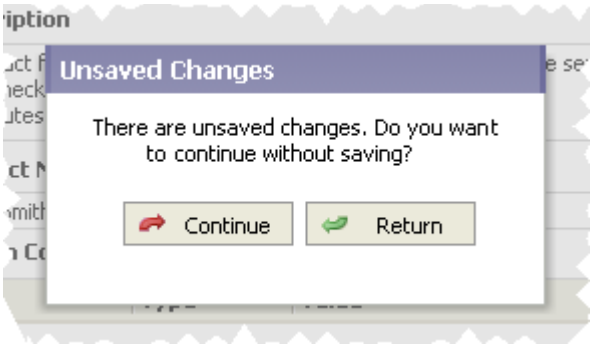
Most of the iWD Manager objects can be modified, in which case the Save, Save & Close, Cancel, and Delete actions are available at the bottom of the Details views:

- Save: Saves the current object, and leaves it selected.
- Save & Close: Saves and closes the current object.
- Cancel: Discards any modifications, and closes the current object.
- Delete: Deletes the current object.
- Copy: iWD Manager has built-in copy-and-paste functionality for some configuration objects: Services, Contracts, Processes, Rules, Business Calendars. Services, Rules, and Business Calendars are pasted as new objects; the user must save them. Contracts and Processes perform cascaded copy; child rules also are copied. Clicking Copy on the edit page of an object puts it in the clipboard (only one object at a time is supported). When the clipboard contains an object, depending on the object type, additional tree nodes or buttons will be visible/enabled that allow pasting the clipboard object to a new instance.



Note: The ability to modify an object also depends on the user's permissions. If the user does not have permission to modify a particular object, the previously described object actions will not be accessible.

Unsaved Changes



iWD Manager keeps track of any changes that have been made to the current selected object by a user.

If an object is being closed, but unsaved changes are present, iWD Manager will display the Unsaved Changes dialog box, which has the following actions:

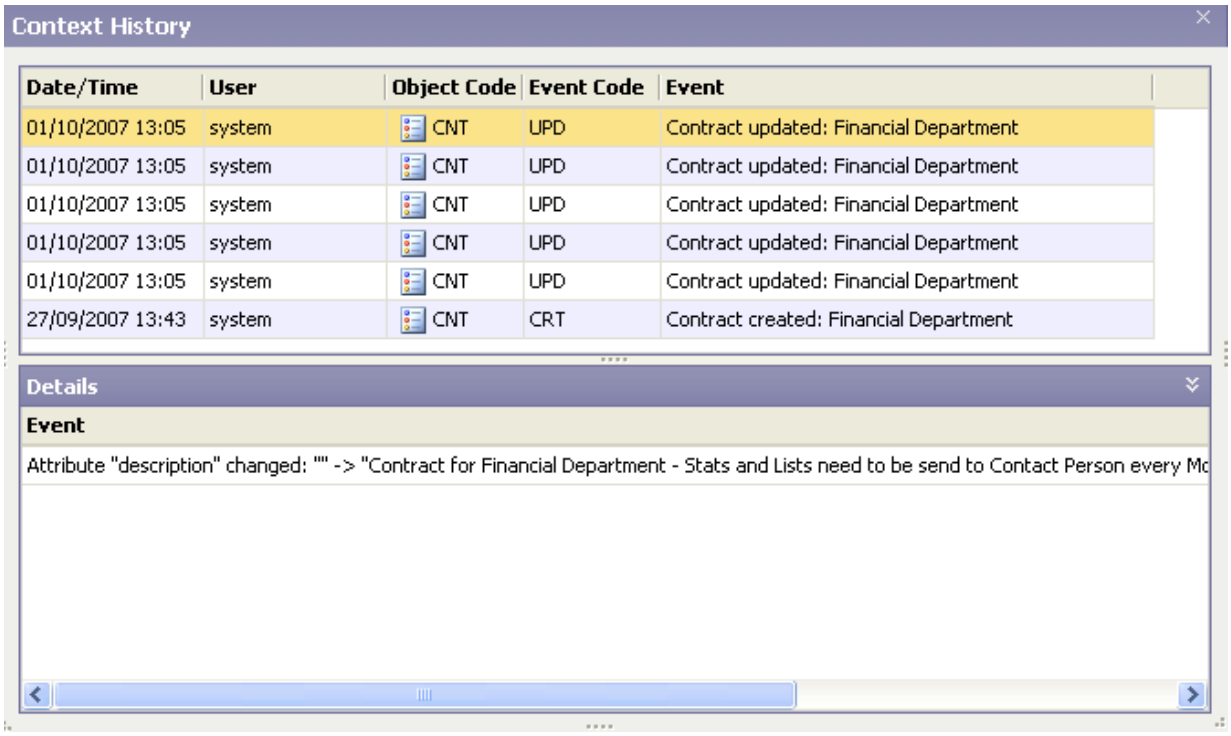
- Continue: Discards the modifications and closes the object
- Return: Does not discard the modifications, but returns to the Details view of the object

Context History

Besides warning about unsaved changes, iWD Manager also records all saved changes into the object's Context history. Context history can be viewed by clicking the Context History icon  in the top-right corner of the object's Details view.

The Context History appears in a new window and displays detailed information about all of the changes that have been made to the current object:

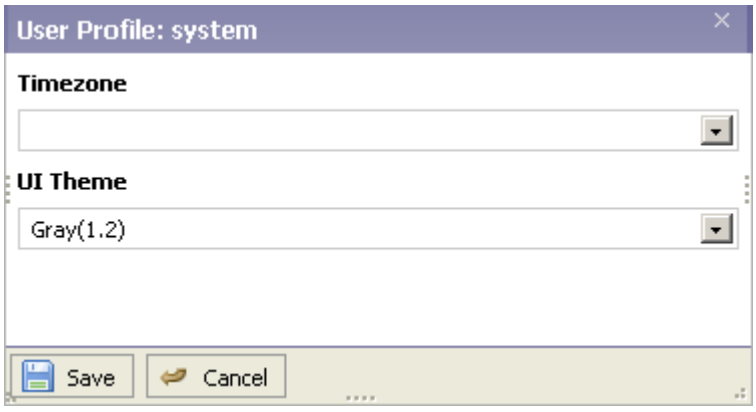
- Date/Time: When the modification was made
- User: Who made the modification
- Object Code: Describes the object type (such as CNT, which means "CoNTract")
- Event Code: Describes the action type (such as UPD, which means "UPDate")
- Event: Formatted description of the change



Additionally, for UPD (update) events, the Context History provides details on each attribute of an object that has been changed. These details can be viewed by clicking a particular UPD event in the upper table. They are displayed in the following image.

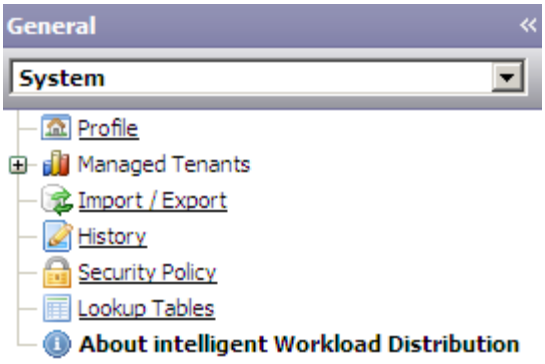
User Profile

The user can customize date/time conversions and iWD Manager appearance in the User Profile dialog box, which is available by clicking the user name in the application header. If Timezone is not specified, all timestamps will be shown by using the UTC time zone.



General Configuration

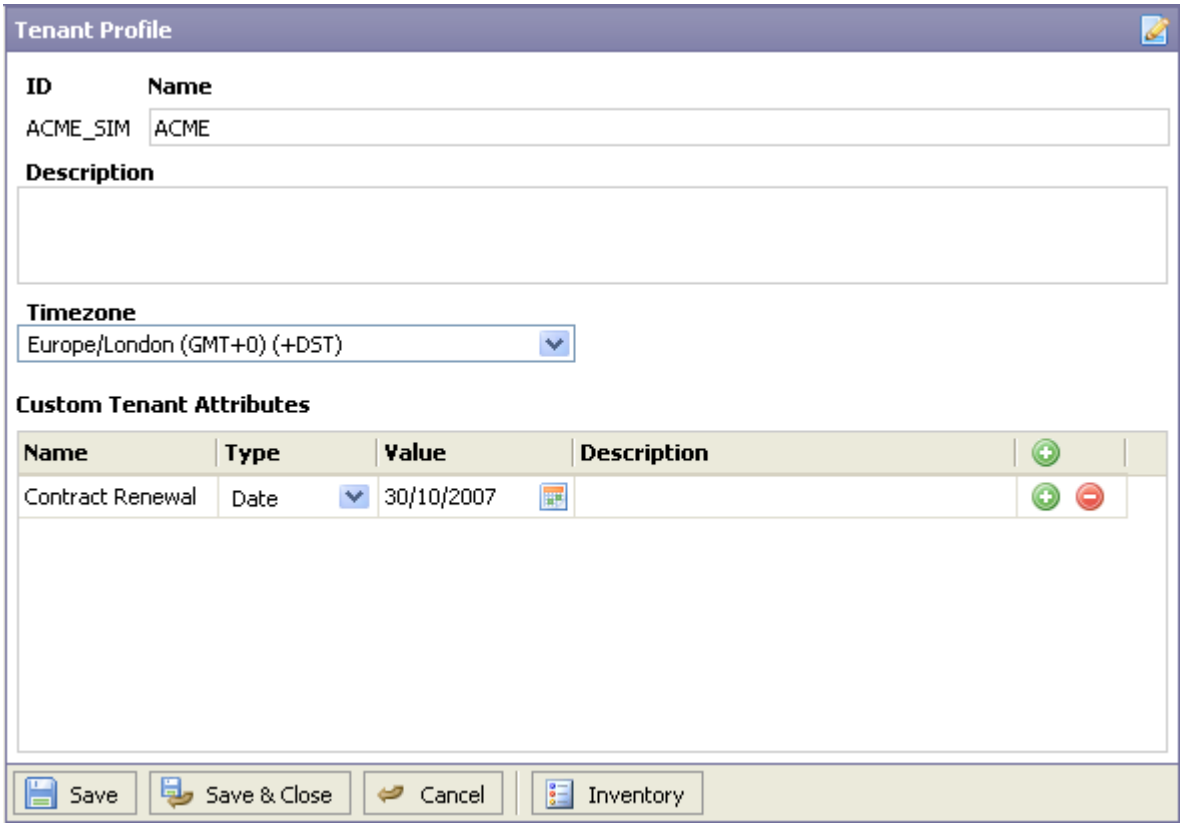
General configuration section allows managing general tenant-level configuration:



- Use Profile to view and modify general tenant attributes
- Use Import and Export view to export and import tenant's configuration to and from XML file
- Use Security Policies view to manage tenant roles and permissions
- Use Managed Tenant details view to create new managed tenants, and view and modify attributes of managed tenants
- Use Lookup Tables view to specify tenant-specific constants as key/label pairs
- Use History view to keep track of changes in tenant configuration

Profile

The Profile view displays and allows modification of general tenant details, such as name, description, and default time zone.



The following attributes and actions are available in the Profile view:

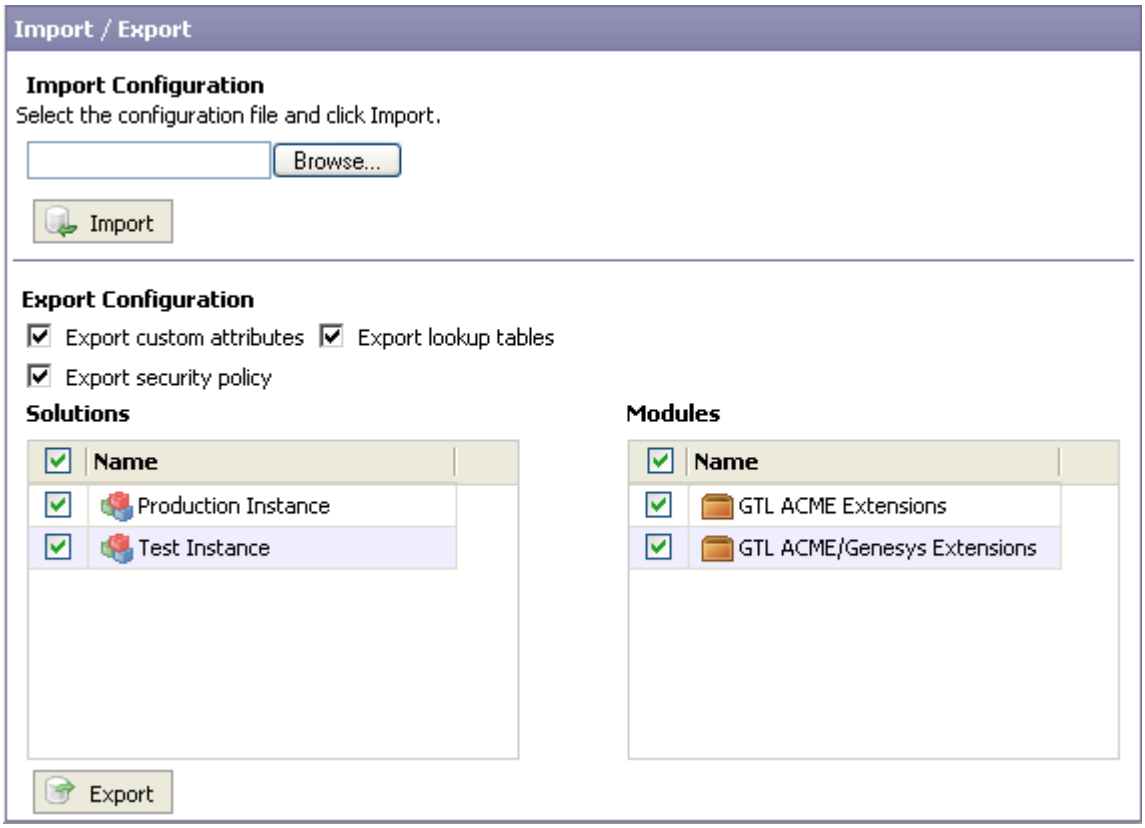
Attribute/Action	Description
ID	Tenant ID; this field is read-only.
Name	Tenant name; this field is editable only by parent tenant's administrators
Description	Tenant description; this field is editable only by parent tenant's administrators.
Time Zone	Tenant default time zone. Default time zone is used when no time zone is specified for a capture or distribution point.
Custom Tenant Attributes	Custom tenant attributes that provide additional information about tenant, for informative and reporting purposes.
Save/Save & Close/Cancel	Standard iWD Manager functions as described in iWD Manager Overview
Inventory	Opens tenant business-process inventory report that contains detailed information about tenant business configuration (such as contracts, processes, and rules).

Import and Export

The Import/Export function enables you to import and export the iWD configuration to and from an XML file. This function allows you to:

- Transfer configurations between environments and tenants.
- Load the configuration from the solution template.

When you import or export the configuration, take into account that the configuration contains only the current tenant configuration. If the tenant has child tenants, you must import or export all of the child-tenant configurations separately.



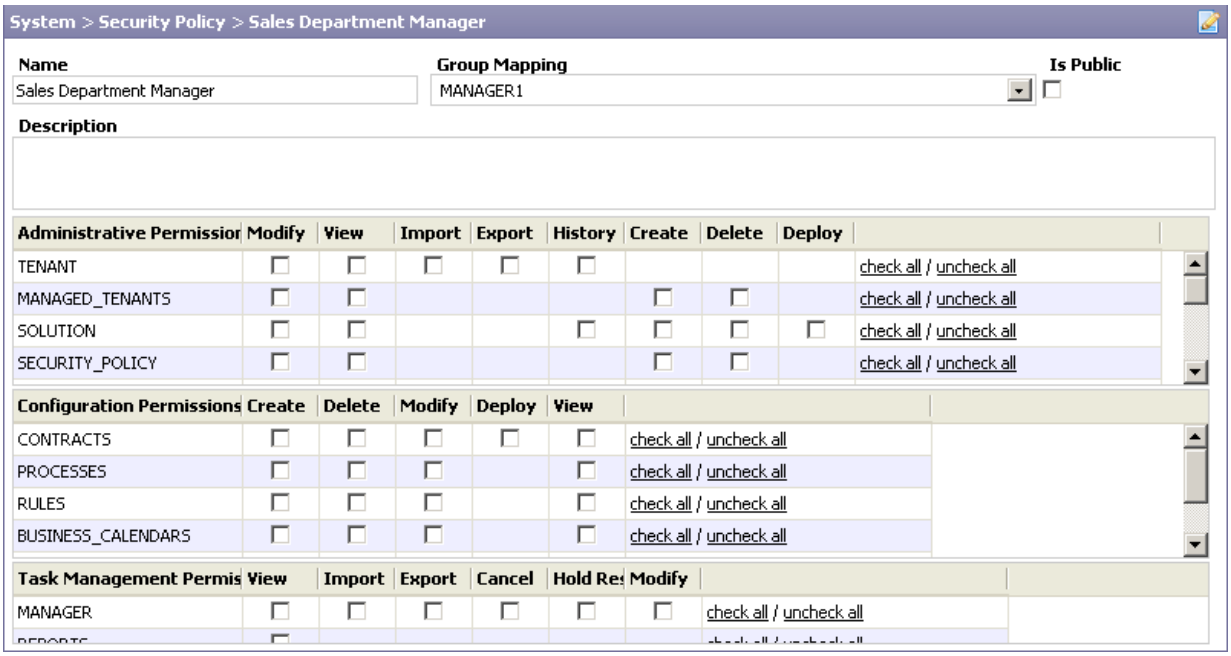
The following properties and actions are available in the Import/Export view:

Property/Action	Description
Configuration file/Browse	Allows selection of a configuration file to import.
Import	Imports the configuration file that is selected in field above.
Export Custom Attributes	Whether to include the tenant's custom attributes in exported configuration file.
Export Security Policy	Whether to include the tenant's security policies in exported configuration file.
Solutions	Which tenant's solutions to include in exported configuration

Property/Action	Description
	file.
Modules	Which tenant's modules to include in exported configuration file.
Export	Export's the configuration object's that are selected above to an XML configuration file.

Security Policies

Security Policy allows a user to create custom security roles for each tenant and map them to an LDAP group. During authentication, a user is granted all of the permissions that are combined from roles that are mapped to the LDAP groups of which the user is a member. A role can contain any combination of permissions from three groups: administrative permissions (technical configuration), configuration permissions (business-user configuration), and task-management permissions (iWD Manager). For them to function properly, most permissions must be added in groups; for example, modify or delete permissions are useless without the view permission.



Key features:

- Multiple roles can be mapped to the same LDAP group.
- The system user automatically has all permissions to all tenants, and does not need an explicit role.
- Automatic LDAP group lookup from (Microsoft Active Directory) server.

The following properties and actions are available in Profile view:

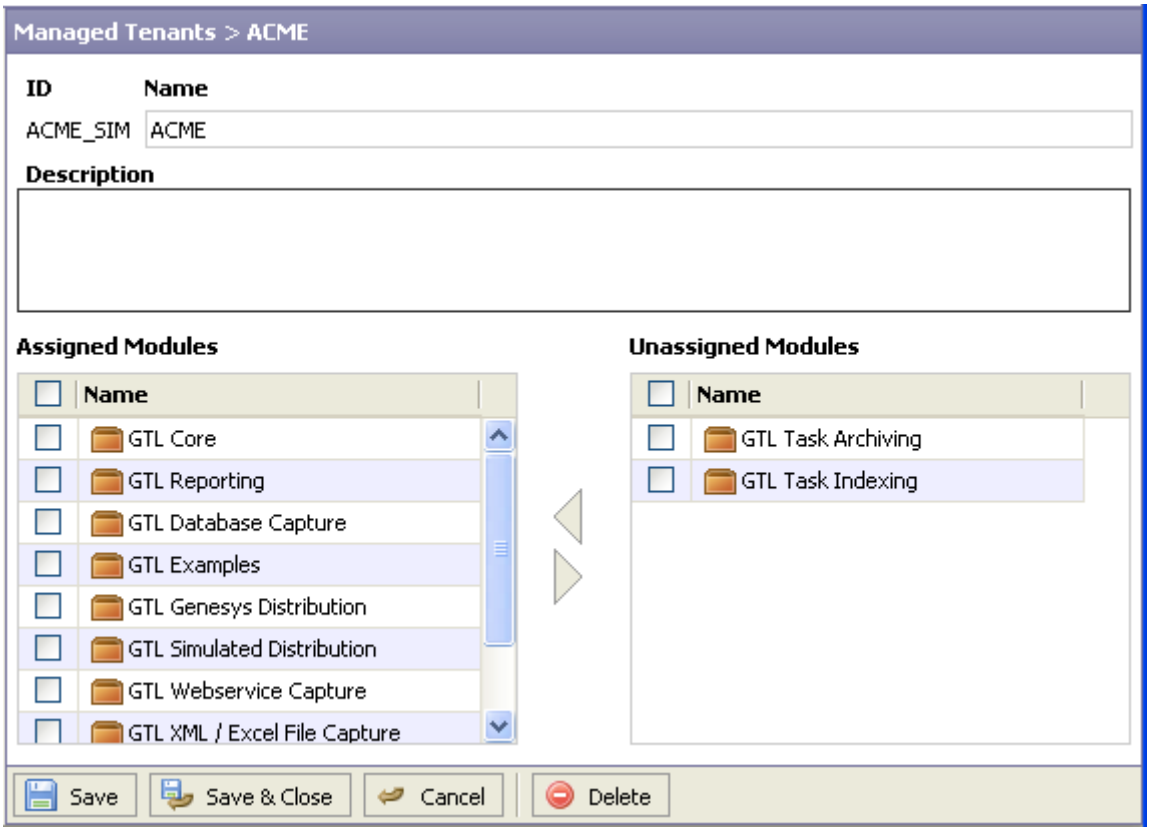
Property	Description

Property	Description
Name	Role name, Mandatory field; must be unique within tenant.
Group Mapping	LDAP group mapping. Mandatory field.
Description	Plain-text description of role (up to 4,000 characters).
Is Public	If role is marked as Public, it can be assigned to child tenants.

Note: Available permissions can vary, depending on available modules/configuration.

Managed Tenants Details

Managed Tenant Details view displays and allows to modify general managed-tenant attributes.



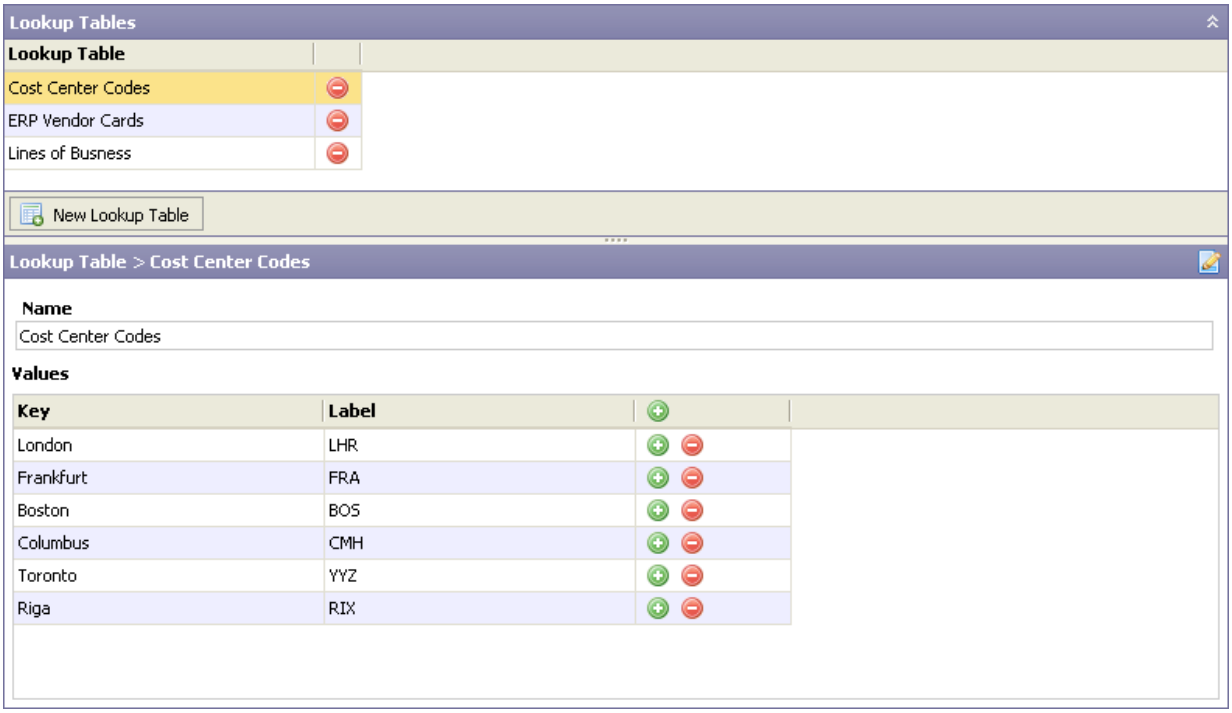
The following attributes and actions are available in Managed Tenant view:

Attribute/Action	Description

Attribute/Action	Description
ID	Tenant ID; this field becomes read-only after a tenant is created.
Name	Tenant name.
Description	Tenant description.
Assigned Modules	Modules (in other words, functionality) that are assigned to the tenant. Such modules are available for use in tenant solutions.
Unassigned Modules	Modules that are not assigned to the tenant. Such modules are not available for use in tenant solutions.
Save/Save & Close/Cancel/Delete	Standard iWD Manager functions as described in iWD Manager Overview

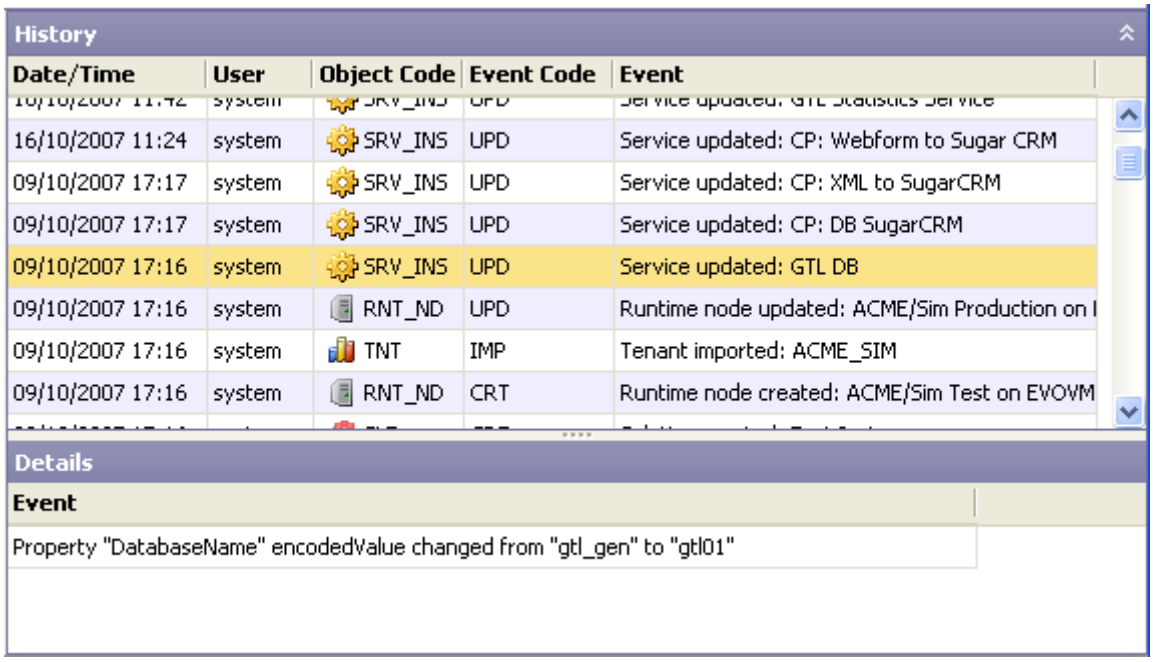
Lookup Tables

Users can specify lookup tables that can be used in rules, custom attributes, and metrics. Lookup tables are simple key/label pairs and are displayed as dropdown controls.



History

The History page provides a detailed log of activities that have been performed in iWD Manager by users. Each activity is represented by an audit event that represents who did what and when.



The following properties and actions are available in History view:

Property/Action	Description
Date/Time	Date/time when the activity was performed.
User	Who performed the activity.
Object Code	Represents the type of object on which the activity was performed: • TNT: Tenant • SLT: Solution • SRV_INS: Service • RNT_ND: Runtime Node • RL_TMP: Rule Template • MDL: Module • PRC: Process • RL: Rule • CNT: Contract • ROLE: Security Policy Role • BC: Business Calendar • REP_TMP: Report Template
Event Code	Represents the type of the activity:

Property/Action	Description
	• CRT: Create • UPD: Update • DLT: Delete • IMP: Import • EXP: Export
Event	Formatted description of the activity.
Details	Display attribute-level changes to the object for the activity that is selected in the main History table. Only update activities (Event Code is UPD) contain these details.

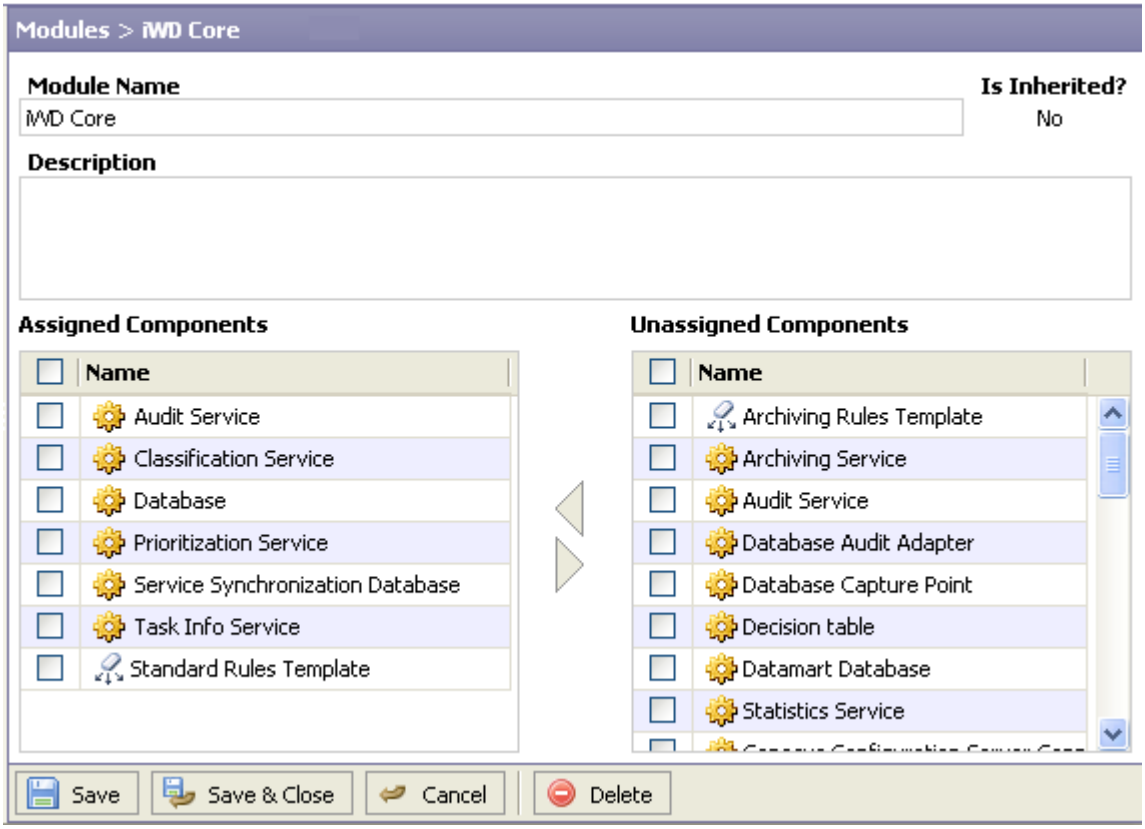
Modules & Components

Modules and components define iWD functionality:

- A component is an atomic object that provides a specific iWD function. The following component types are used in iWD:
  - Service templates: This component represents a iWD service that implements specific functionality. Service templates are preconfigured and, hence, are not changeable in iWD Manager. Service instances that are based on service templates, however, are configurable.
  - Rule template
  - Report templates
  - Metrics templates
- A module is a group of components.

Modules

Modules bundle a set of iWD components into a named functional area, such as "iWD Core" or "Task Archiving."



The following properties and actions are available in Modules view:

Property/Action	Description
Name	Name of module
Is Inherited	Whether a module is inherited from the parent tenant. This means that the module has been assigned to this managed tenant by the parent tenant. If this is true, all fields are read-only.
Description	Description of module.
Assigned Components	Services and rule templates that belong to the module.
Unassigned Components	Services and rule templates that do not belong to the module.
Save/Save &Close/Cancel/Delete	Standard iWD Manager functions as described in iWD Manager Overview

Rule Templates

In the iWD solution, Rule templates are used to set up different business rules for task classification, contracts, and processes. Business rules usually have the following:

- Conditions; that is, when or if expressions
- Actions; that is, then expressions
- Parameters, which are used in the actions and conditions

Business rules within iWD are based on Rule templates that are created by IT personnel. As with templates for Microsoft Office Word, business-rule templates are the foundation for business users to create or configure existing business rules that govern tasks in iWD. Business users are empowered to make the necessary changes, when required, while IT personnel can focus on tasks that are related to IT.

Note: Definition of Rule templates requires a basic knowledge of the Java programming language.

The iWD solution provides a set of predefined, standard rule templates and the possibility to define custom extensions as new rule templates.

The following properties and actions are available in Rule Template view:

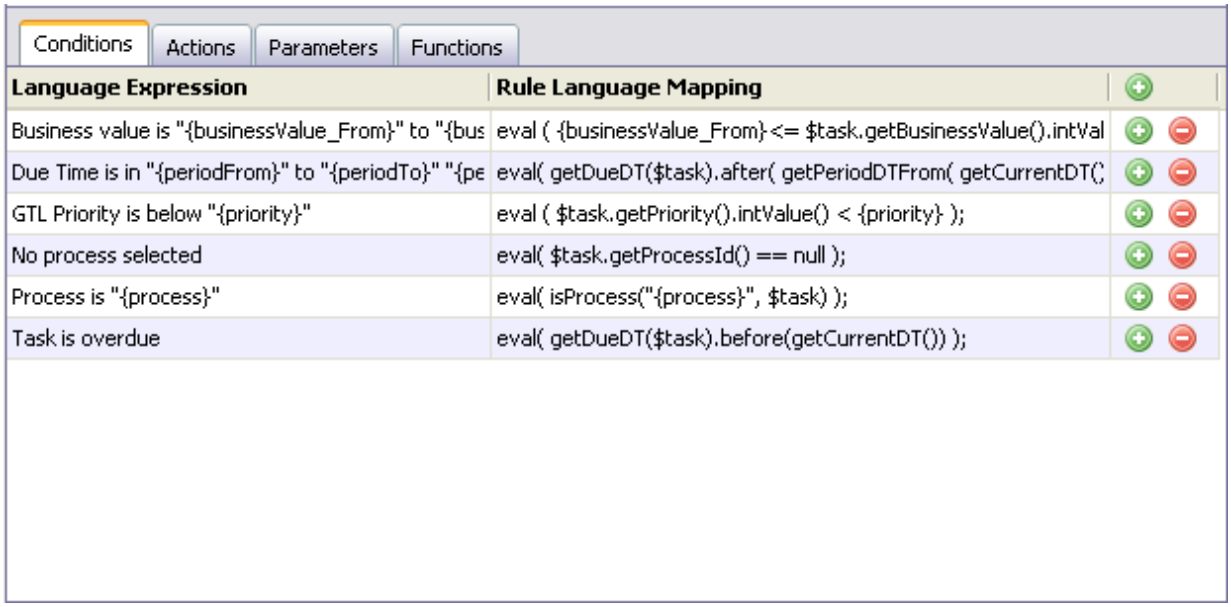
Property	Description
Rule Template Name	The name of the Rule template. This field is mandatory.
Conditions	The rule-condition template; that is, a *when* or *if* expression (for example, *when due time is in x hours*). The Rule template can contain any number of (or no) conditions.
Actions	The rule-action template; that is, a then expression (for example, *then assign priority*). The Rule template can contain any number of actions.
Parameters	Rule parameters that are used in conditions and actions, and describe how parameters are presented to the user (who is creating the rule), how input is validated, and so on.
Functions	Allows definition of more complex functions, which then can be referenced in actions and conditions.
Save / Save & Close / Cancel	Standard iWD Manager functions as described in iWD Manager Overview

Conditions & Actions

Alhough conditions and actions have different purposes, their definition is the same and consists of the following property specifications:

Property	Description
Language Expression	Semantics of the rule language; meant for business people. They are simple enough to be understood by non-Java specialists. Language expressions can contain different parameters.
Rule Language Mapping	The same rule-language expression, in Java code. This is the real code that is executed; it requires an understanding of Java and JBoss rules.

The following example is of conditions for a Rule template:

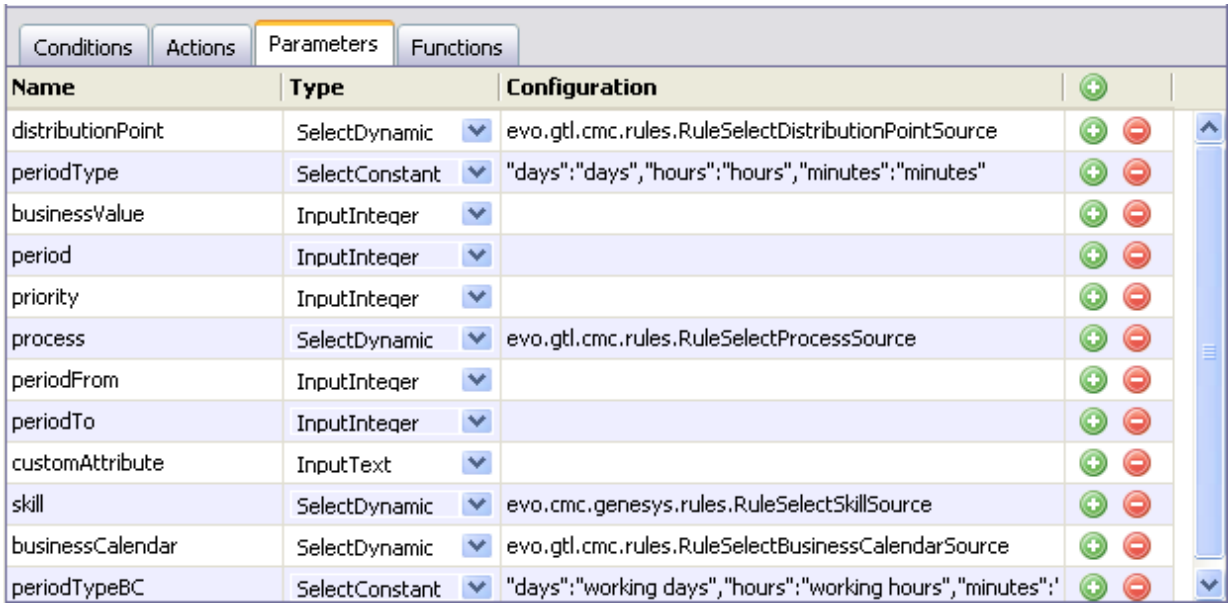


Toward reviewing the third condition, the language expression contains the parameter {priority}, which is mapped to the rule language. This expression is executed for each task to which this rule can apply, as follows: The parameter is replaced with the current parameter value, which is compared to the task priority value, which in turn is retrieved by calling the Task object method getPriority() and converted to an integer value. During the building of rules, only the language expressions are visible to business users, which enables them to build rules that are easy to read and understand.

Parameters

Rule-language expressions and the corresponding language mappings (used in conditions and actions) can contain different parameters. For example, the expression Process is {process} has the parameter process.

The following example is of parameters for a Rule template:



Parameters that are used as simple input controls appear only as specific names in , in action and condition definitions. If a parameter (such as a combo box) is more complex than a simple input control, it must be defined in the Parameters tab. Parameters in the Parameters tab have the following definition properties:

Property	Description
Parameter Name	The name of the parameter. This field is mandatory.
Parameter Type	The type of input control for the parameter. There are six input control types: • SelectConstant; constant combo box with labels and values. • SelectDynamic; dynamic combo box; labels and values are received from the specific Java class that implements the interface evo.rules.cmc.RuleSelectDynamicSource. • InputText; input box that allows to input any text. • InputInteger; input box that allows to input integer values. • InputNumber; input box that allows to input number values. • InputDate; input box that allows to input date values. This field is mandatory.
Configuration	The attributes for SelectConstant and SelectDynamic type parameters. The pattern for SelectConstant is the following: "<value_1>":"<label_1>", "<value_2>":"<label_2>",..., "<value_n>":"<label_n>"

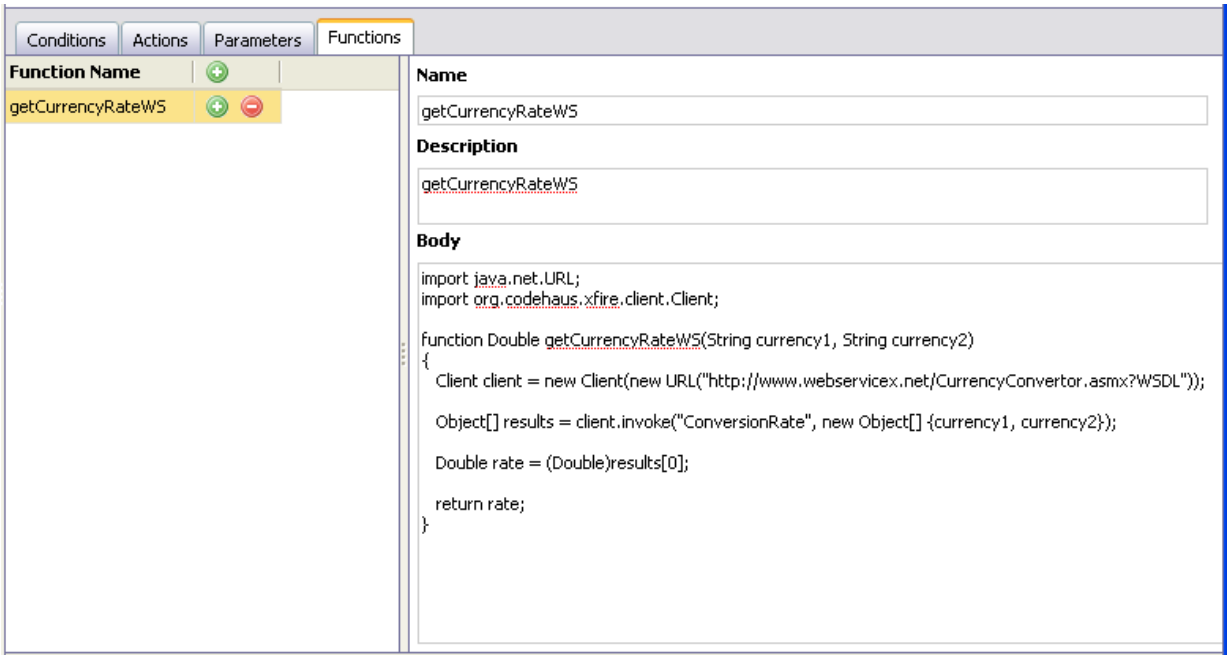
Property	Description
	The pattern for SelectDynamic is the following: <package name>.<class name> This field is mandatory for SelectConstant and SelectDynamic type parameters.

Functions

During definition of Rule templates, elements that are more complex than conditions and actions that allow only one input line are required. The Functions tab enables you to write specific Java functions for different purposes in Rule templates. The specified functions are used in the Rule Language Mappings.

Important: During use of a rule function in a condition or action, make sure that each function name and parameter is prefixed with a space, such as the following; otherwise, the function might not be recognized by the rules parser:

```
eval( someFunction( param1, param2) )
```



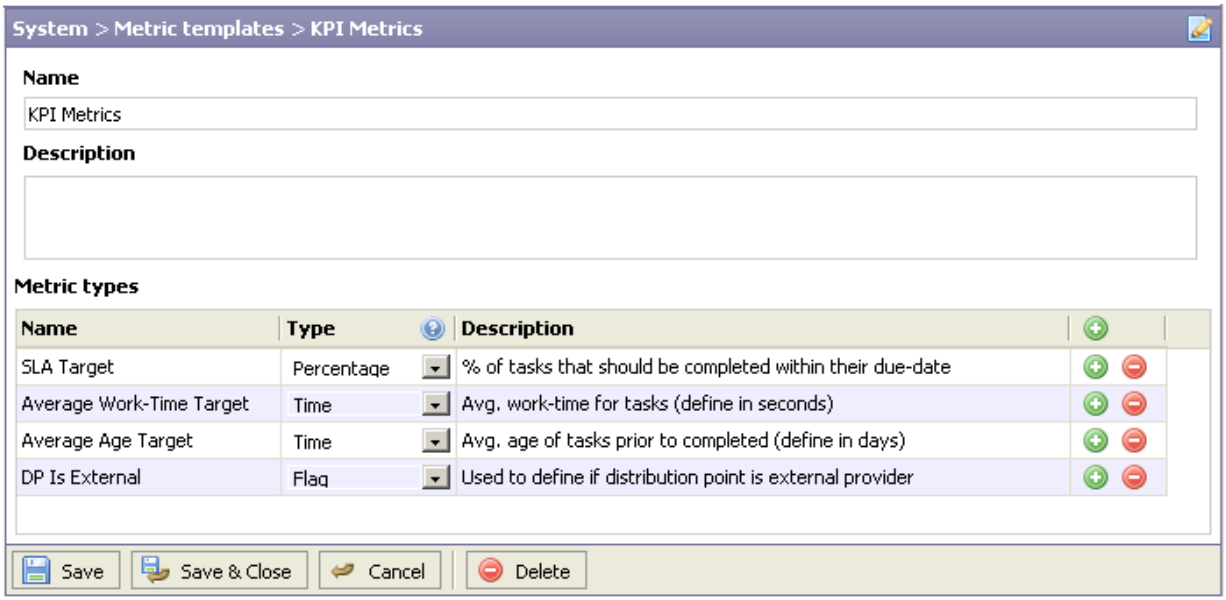
The following properties are available in the Functions tab:

Property	Description
Function Name	The name of the function. This property is mandatory.
Description	A short description of the function. This property is optional.

Property	Description
Body	The function code, with the following syntax: function <return type> <function name>(<arguments>) { ... } This field is mandatory.

Metrics Templates

Metrics templates can be used to define various types of metrics that will be available at the business-user level to specify actual values for contracts/processes and distribution points. Metrics templates are just like any other module components and must be assigned to a module to be available under contracts and processes.



The following shows value types for the Metric template:

Value type	Description
Time	Time value in days, hours, minutes, or hours
Flag	Check box for true/false values
Date	Simple date control
Number	Number value, basic number validation
Percentage	Percentage from 0% to 100%

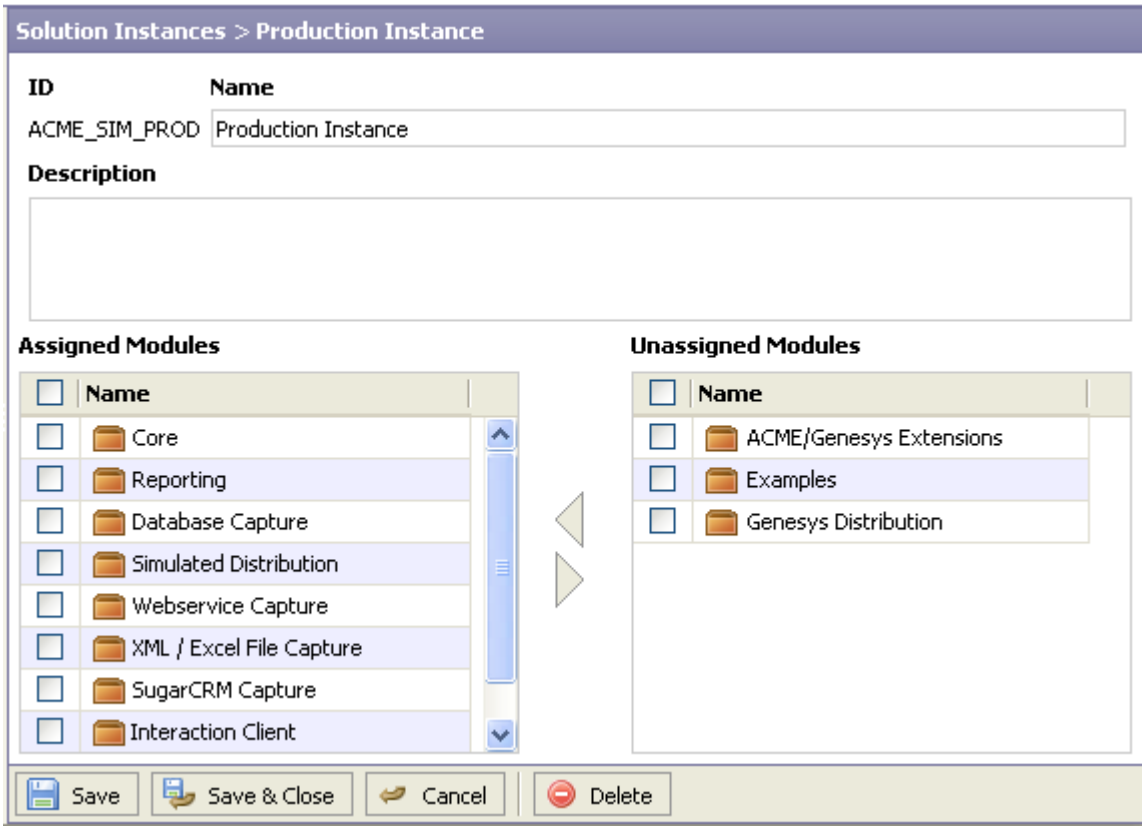
Solutions & Services

The Solutions & Services configuration section allows the setup of iWD runtime instances for a specific infrastructure and environment:

- Use Solution Details view to create new solutions, as well as to view and modify general solution attributes
- Use Runtime Nodes view to define runtime nodes in which services will run.
- Use Services view to monitor and manage service status.
- Use Service Detailsview to create new services, as well as to view and modify configuration of existing services.
- Use Deployment view to deploy solution configuration to runtime nodes, as well as to activate services.
- Use Change History view to check the history of each deployed configuration version.

Solution Details

Solution Details view displays, and lets you modify, general solution attributes.



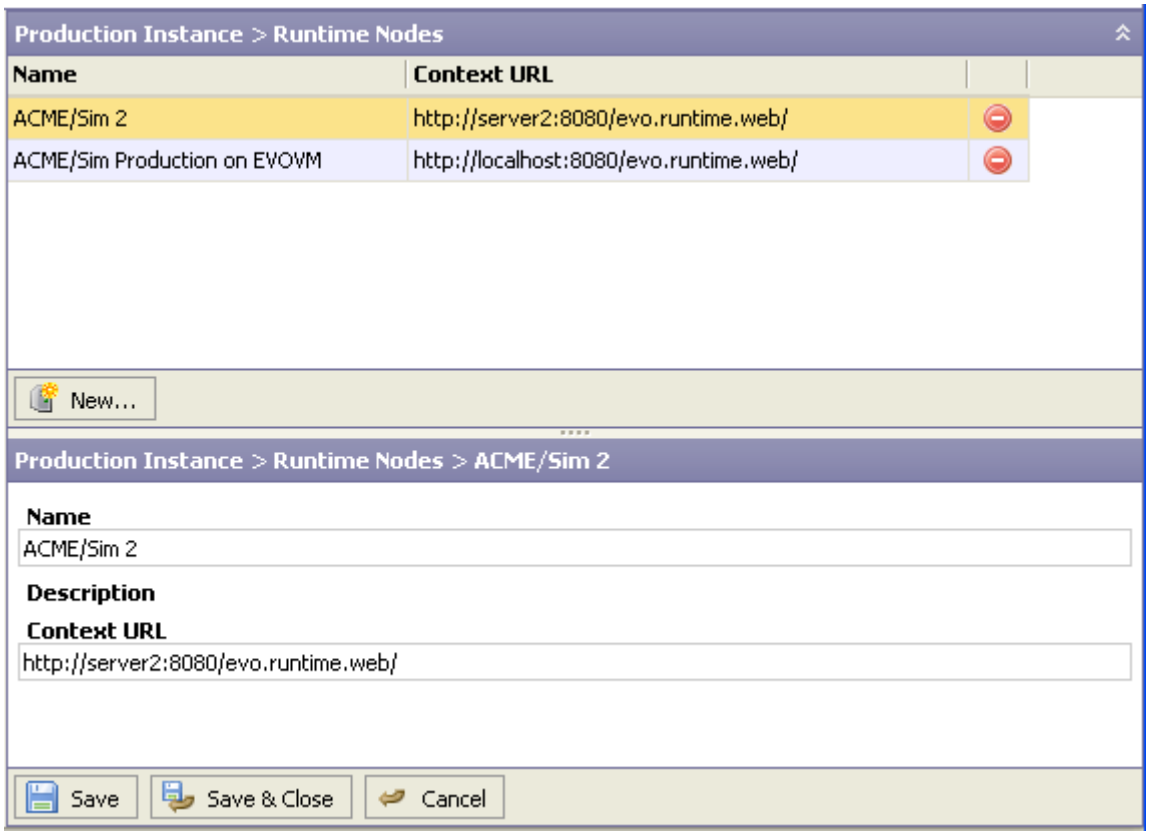
The following attributes and actions are available in Solution Details view:

Attribute / Action	Description

Attribute / Action	Description
ID	The ID of the solution. This field becomes read-only after the solution has been created.
Name	The name of the solution.
Description	A description of the solution.
Assigned Modules	Modules (that is, functionality) that have been assigned to the solution. Components such as Services and Rule templates from these modules are available for use in the solution.
Unassigned Modules	Modules that have not been assigned to the solution. Components from these modules are not available for use in the solution.
Save/Save & Close/Cancel/Delete	Standard iWD Manager actions as described in iWD Manager Overview

Runtime Nodes

Runtime Nodes view lets you view and configure runtime nodes in which the solution's services are running.



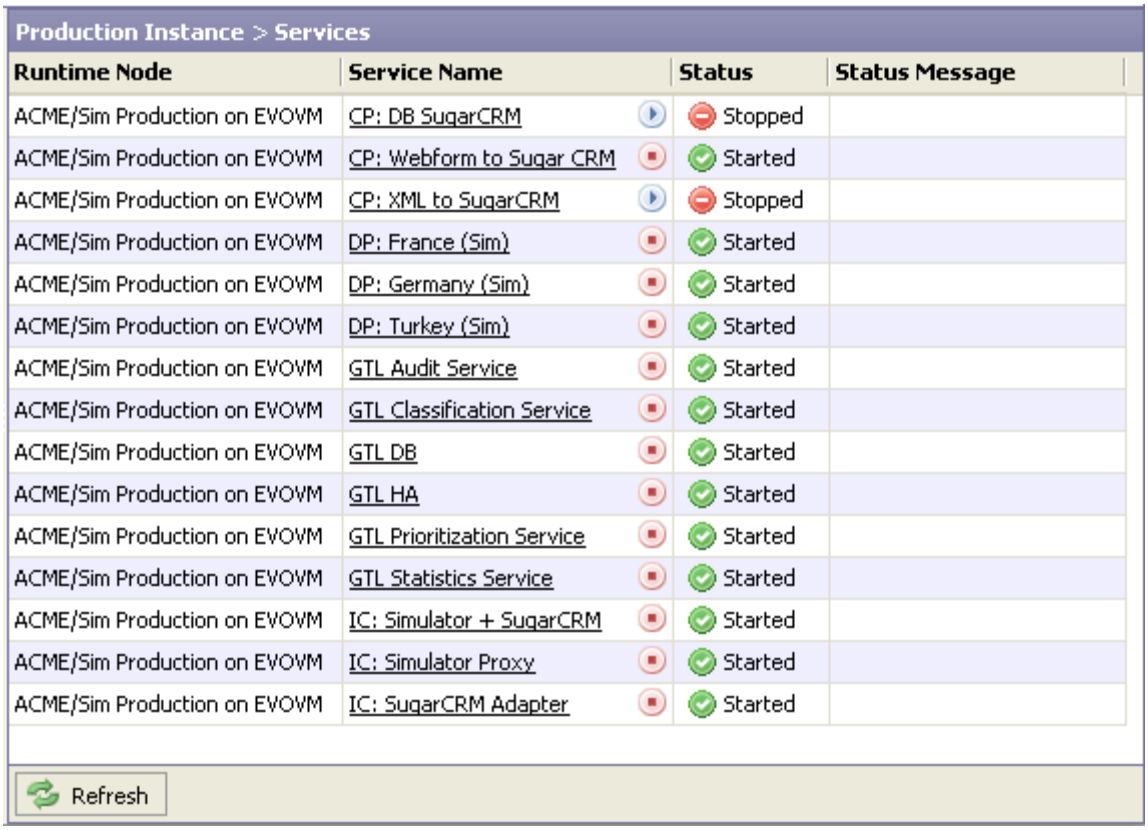
The following attributes and actions are available in Runtime Nodes view:

Attribute/Action	Description

Attribute/Action	Description
Name	The name of the runtime node.
Context URL	The URL of the runtime node (depends on how the runtime node was installed).
New...	Allows definition of a new runtime node.
Save / Save & Close / Cancel /	Standard iWD Manager actions as described in iWD Manager Overview

Services

Services view displays status of services that have been configured in the solution, and allows you to manage service status manually.



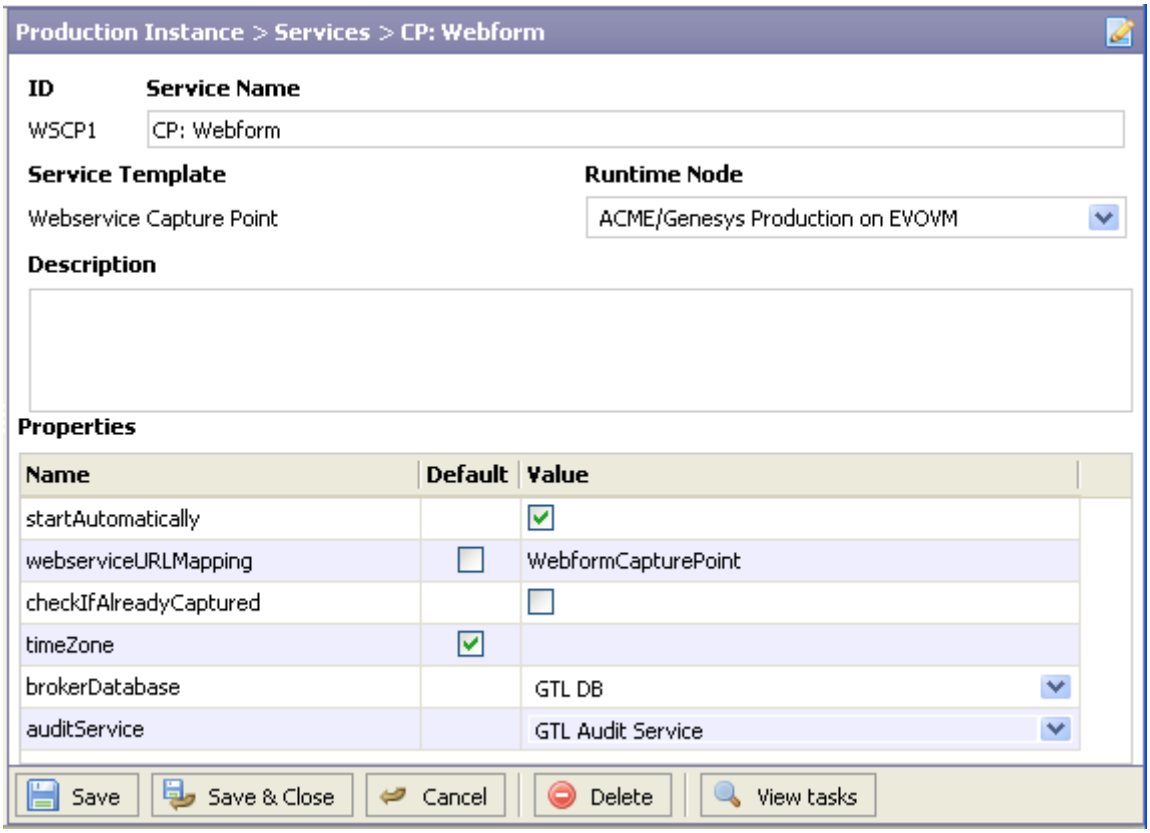
The following attributes and actions are available in Services view:

Attribute/Action	Description
Runtime Node	Runtime node on which the service is running. Most of the services run on a single runtime node; however, some lightweight services (such as iWD Database), run on all available runtime nodes.
Service Name	The name of the service; click here to open the Service Details view.
	Starts a stopped service.

Attribute/Action	Description
	Stops a started service.
Status	Service status: • Started: The service is active (running). • Scheduled: The service is scheduled to run in the future. • Queue: The service is scheduled to run after another job that is in the same execution queue has finished. • Stopped: The service is stopped. • Error: The service has encountered an error during its last operation; error details are provided in the Status Message column. • Not Deployed: The service is configured in the solution, but the configuration is not yet deployed to the runtime node. • Not Accessible: The runtime node is not accessible. This can be caused by an invalid runtime node configuration, or because the application server on which the node is installed is not running.
Status Message	Displays additional service-status details, when available, such as an error message; click it to view details in a larger window (shortcut menu).
Log	Opens Log Viewer dialog box for the specified service

Service Details

Service Details view displays, and lets you modify, service attributes. See also iWD Services, for a detailed reference on specific service types.



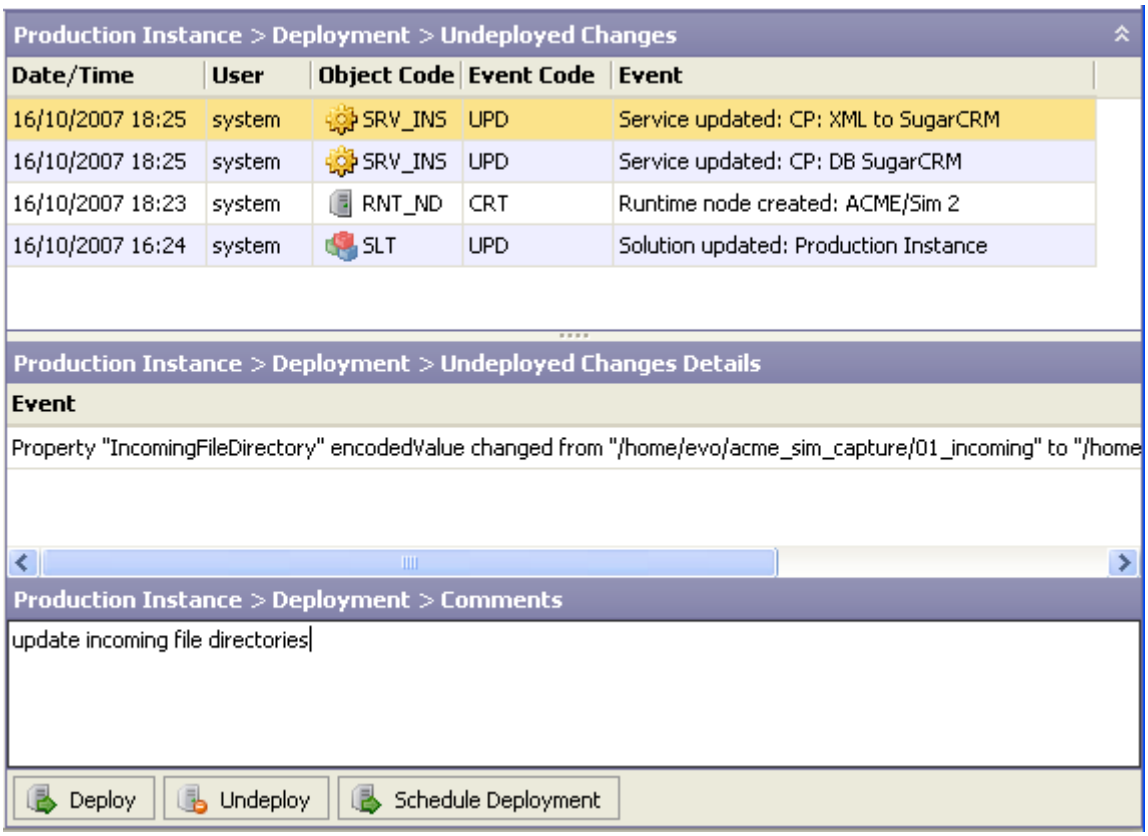
The following properties and actions are available in Service Details view:

Property	Description
ID	The ID of the service. This field becomes read-only after the service has been created.
Service Name	The name of the service.
Service Template	The template on which the service is based. This field becomes readonly when the service is created; see also Modules and Components for more information on service templates.
Runtime Node	The runtime node on which the service runs. This field is absent for any lightweight service that is running on all available runtime nodes, such as iWD Database.
Description	A description of the service.
Properties	The properties of the service that configure its functionality: • Name: The name of the property. • Default: Whether or not to use the property's default value. • Value: The property's value. Depending on property type, this field can be a text box, check box, or dropdown box. When "Default" box has been checked, this field is read-only.

Property	Description
	Different service types have different properties. For a detailed reference on specific service types, see iWD Services.
Save/Save & Close/Cancel/Delete	Standard iWD Manager functions as described in iWD Manager Overview
View Tasks	Opens a corresponding task view in iWD Manager. This action is available only for capture points and distribution points.

Deployment

Any change to the configuration in iWD Manager is not automatically activated. Deployment view performs this task by distributing a solution's configuration across the defined runtime nodes.



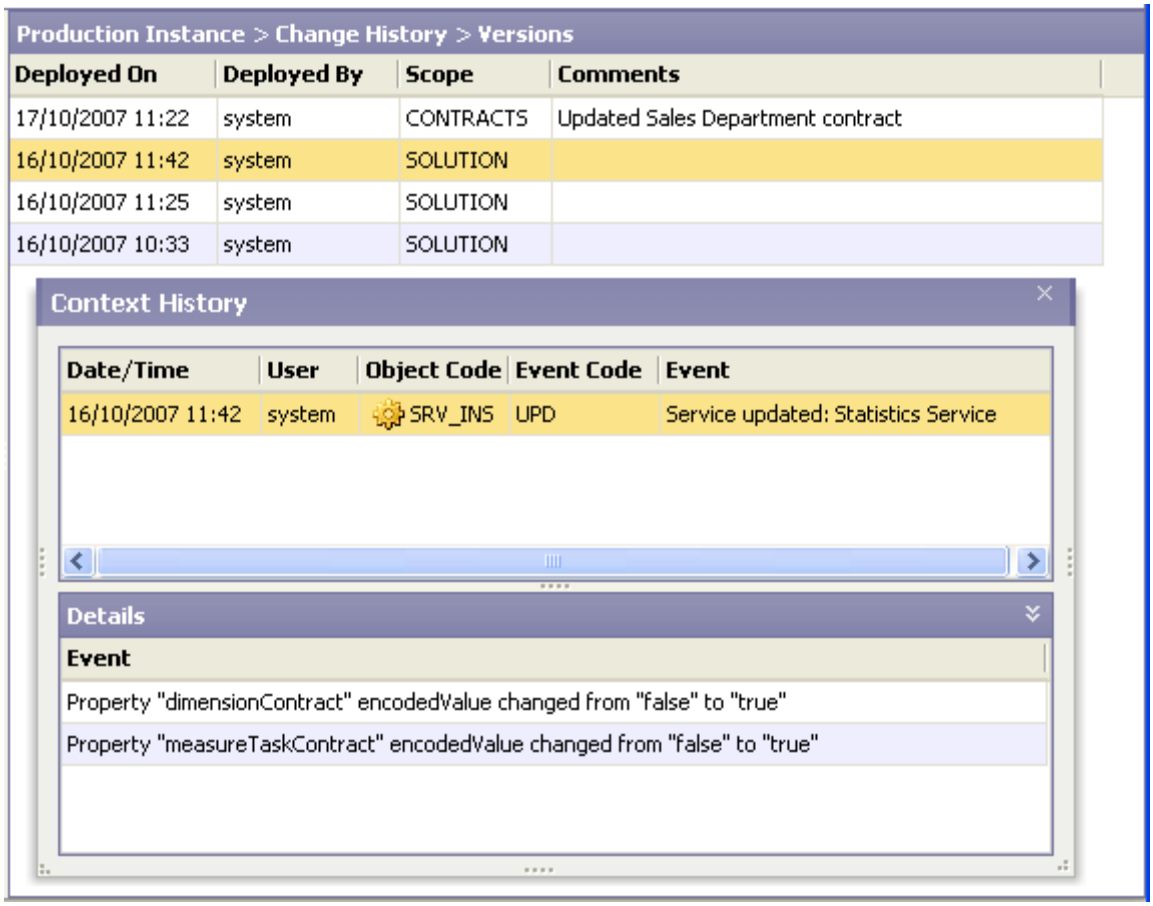
The following attributes and actions are available in Deployment view:

Attribute/Action	Description
Undeployed Changes/Undeployed Changes Details	Detailed information about activities performed in iWD Manager since the last deployment; for more details, see History.
Comments	Deployment comments; will be displayed as version comments in Change History.

Attribute/Action	Description
Deploy	Deploys the configuration. Depending on the configuration's complexity and runtime environment characteristics, this action can take several seconds.
Undeploy	Stops all of the services of a solution, and undeploys the configuration from all available runtime nodes. If any runtime node that is defined in the solution is not available, iWD Manager will display a warning message.
Schedule Deployment	Allows deployment to be performed on a specific date/hour. This requires the evo.host parameter to be set in databaseconfiguration XML (must be unique for every host that is running iWD Manager with the same configuration database).

Change History

Change History maintains the history of deployed configuration versions. Whenever a solution configuration is deployed, a new configuration version record is created that contains all of the changes that have been made in the configuration since the previous version (deployment).

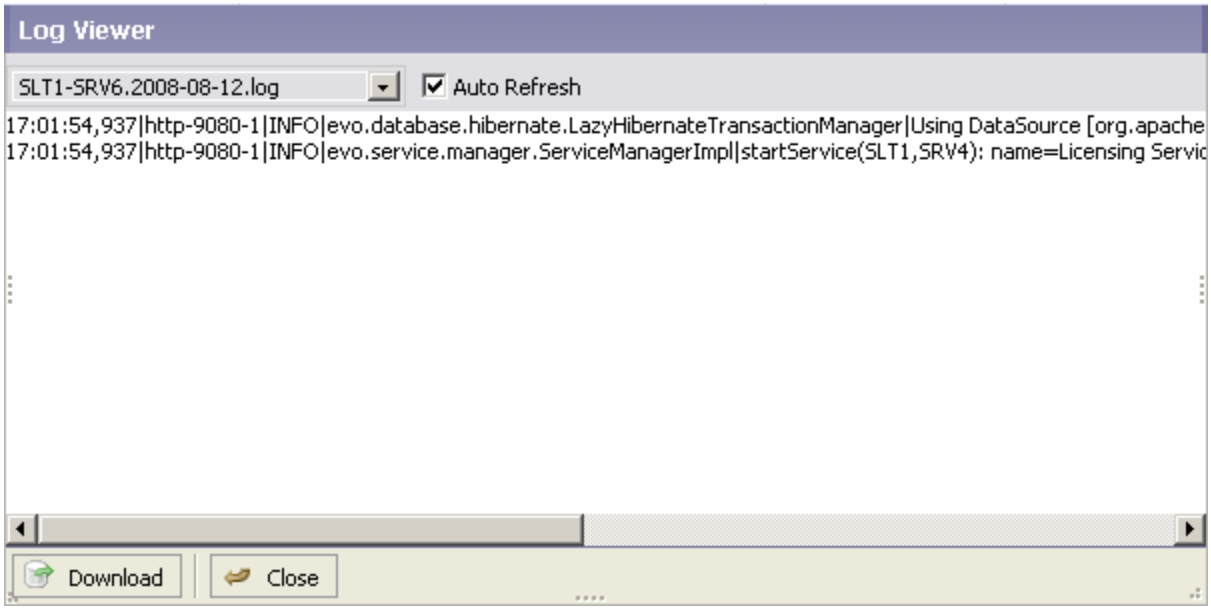


The following attributes and actions are available in Change History view:

Attribute/Action	Description
Deployed On	When the configuration was deployed.
Deployed By	Who deployed the configuration.
Scope	Whether the whole solution configuration was deployed (SOLUTION) from Services deployment view or only iWD contracts, processes, and rules (CONTRACTS) from Contract & Process deployment view.
Comments	Deployment comments.
Context History	Opens in a new window by clicking a version in the Change History table. Contains detailed information about activities performed in iWD Manager for the selected version; for more details, see History.

Log Viewer

Log Viewer allows the user to view iWD service log files remotely. Various viewing methods can be used, including near-real-time viewing with Auto Refresh functionality, as well as viewing of historical log files for the particular service.



iWD Services

The following services are listed in their recommended configuration order:

Name	Mandatory	Category	Dependencies	Notes
Logging Service	x	Core		
Database	x	Core		Typically, two databases need to be defined: one for iWD runtime

Name	Mandatory	Category	Dependencies	Notes
				services, other for iWD Data Mart.
Audit Service	x	Core	Database	
Message Queue Service	x	Core	Database	
Scripting Service		Core		Required only when iWD transformation is being used in Websphere MQ or XML Capture Points.
Broker Service	x	Core	Database Audit Service Message Queue Service	
Rules Service	x	Core		
Classification Service	x	Core	Broker Service Message Queue Service Rules Service	
Prioritization Service	x	Core	iWD Broker Service Message Queue Service Rules Service	
Archiving Service		Core	Broker Service Rules Service*	* Rules service is not required if Archiving Service is in delete mode
Webservice Capture Point		Capture Point	Broker Service	
XML File Capture Point		Capture Point	Broker Service Message Queue Service Scripting Service*	* Scripting Service is only required when iWD transformation is being used.
Database Capture Point		Capture	Broker Service	

Name	Mandatory	Category	Dependencies	Notes
		Point	Message Queue Service	
Websphere MQ Capture Point		Capture Point	Broker Service Message Queue Service Scripting Service*	* Scripting Service is only required when iWD transformation is being used.
Genesys Configuration Server Connector		Distribution Point		
Genesys Interaction Server Connector		Distribution Point	Genesys Configuration Sever Connector	
Genesys Synchronization Service		Distribution Point	Genesys Interaction Server Connector	
Task Info Service		Distribution Point	Broker Service	
Genesys Distribution Point		Distribution Point	Broker Service Message Queue Service Genesys Interaction Server Connector Task Info Service*	* iWD Task Info Service is optional
Simulator Distribution Point		Distribution Point	Broker Service Message Queue Service	
Simulator Distribution Point Assigner		Distribution Point	Broker Service Simulator Distribution Point	
Simulator Distribution Point Completer		Distribution Point	Broker Service Simulator Distribution Point	
Genesys TServer Connector		Reporting	Genesys Configuration Server Connector	
Genesys Statistics Adapter		Reporting	Genesys TServer	Must be configured on the

Name	Mandatory	Category	Dependencies	Notes
			Connector Genesys Configuration Server Connector	same runtime node as iWD Statistics Service.
Extended Statistics Service		Reporting	Genesys Statistics Adapter	
Kettle ETL Service		Reporting	Database	ETL Service has dependencies on two databases - one for runtime (source) and other for datamart (target).
Scheduled ETL Job		Reporting	Kettle ETL Service	A separate scheduled ETL job must be created for each job type.
_Ref1536074933Interaction Client Simulator Proxy		Client	Simulator Distribution Point	

Archiving Service

The Archiving Service removes all Completed, Canceled, or Rejected tasks whose expiration date/time is in the past. The Archiving Service can be scheduled by a CRON expression, launched manually, or triggered by some other scheduled service. There are three types of archiving that the service can perform:

- Remove tasks from database: Deletes expired tasks from a iWD runtime database
- Archive to XML files: Applies archiving rules on tasks (if defined), exports tasks as XML files, and deletes original tasks from a iWD runtime database
- Archive to another solution: Applies archiving rules on tasks (if defined), exports tasks to the runtime database of another iWD solution, and deletes original tasks from a iWD runtime database

The following properties are configurable for the Archiving Service:

Property	Description
startAutomatically	Whether the service should be started automatically after the configuration deployment.
logLevel	Service log level. Should be set to Default, unless otherwise

Property	Description
	specified by Genesys Support.
logAgeInDays	Sets the number of days that log files should be kept in the system. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logMaxFileSizeMegaBytes	Sets a limit on the size of a single log file, in megabytes. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logFilesToKeep	Sets a limit on the number of log files that are kept for this service, excluding the current log file. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
triggerMode	How the Archiving Service will be triggered (started): • Trigger using CRON expression: Starts a job automatically, based on a CRON scheduling expression. • Trigger after another scheduled service: Starts a job automatically after another scheduled job has finished. • Trigger manually: Job can be started manually via iWD Manager Services Status screen.
cronExpression	Standard CRON scheduling expression when triggerMode is set to "Trigger using CRON expression". For example, the following expression will cause the job to be executed every 15 minutes: "0 0,15,30,45 * * * ?". See http://en.wikipedia.org/wiki/Cron for more information.
executionQueueName	Scheduled services that have the same executionQueueName will never be run in parallel within the same solution. If a scheduled service is triggered while another scheduled service that has the same executionQueueName is in progress, it will be queued until the other service has completed processing.
mode	• *Remove tasks from DB*: Deletes expired tasks from an iWD runtime database; • *Archive to XML files*: Applies archiving rules on tasks (if defined), exports tasks as XML files and deletes original tasks from an iWD runtime database; • *Archive to another solution*: Applies archiving rules on tasks (if defined), exports tasks to runtime database of another iWD solution and deletes original tasks from an iWD runtime database.
archiveSolution	The iWD solution in which to place exported tasks. Applicable

Property	Description
	only when mode is set to *Archive to another solution*
archiveDirectory	The directory in which to place XML files that contain exported tasks. Applicable only when mode is set to *Archive to XML files*.
BrokerService	*Mandatory dependency:* Broker Service.
RulesService	*Mandatory dependency:* Rules Service. Applicable only when mode is set to *Archive to another solution or Archive to XML files.*
threads	*Performance tuning:* Number of parallel execution threads.
batchSize	*Performance tuning:* The size of the task batch that is processed in one cycle.

Database

The Database service provides a connection to a database server. A typical iWD solution has two defined database connections: one for runtime data (task, audit, and statistics), and another for historical reporting data(iWD Data Mart).

The configured database and user must exist in the database server. The creation and upgrade of database structures are handled automatically by iWD; so, besides the read/writer permissions, the database user also must be able to create and alter tables and views (plus sequences and triggers for Oracle).

The following properties are configurable for the Database service:

Property	Description
startAutomatically	Whether the service should be started automatically after the configuration deployment.
logLevel	Service log level. Should be set to Default, unless otherwise specified by Genesys Support.
logAgeInDays	Sets the number of days that log files should be kept in the system. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logMaxFileSizeMegaBytes	Sets a limit on the size of a single log file, in megabytes. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logFilesToKeep	Sets a limit on the number of log files that are kept for this service, excluding the current log file. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
DatabaseType	The type of the database server. intelligent Workload Distribution currently supports MySQL 5.0 and Oracle 10.2.
OracleCustomURL	If this is checked, the connection is configured by specifying a custom Oracle JDBC URL. This is available only when DatabaseType is Oracle.

Property	Description
OracleJDBCURL	The JDBC connection URL for Oracle. This is available when OracleCustomURL is checked.
ServerName	The host name of the database server.
MysqlServerPort	The TCP port of the MySQL database server. This is available only when DatabaseType is MySQL.
OracleServerPort	The TCP port of the Oracle database server. This is available only when DatabaseType is Oracle.
DatabaseName	The name of the database. This is available only when DatabaseType is MySQL.
Username	The user name of the database.
Password	The password of the database.
ConnectionPoolSize	*Performance tuning:* How many connections to the database are established.
MysqlConnectionTimeoutMs	*Exception handling:* Timeout, in milliseconds, for the MySQL connection attempt. This is available only when DatabaseType is MySQL.
MysqlSocketTimeoutMs	*Exception handling:* Timeout, in milliseconds, for an established MySQL connection. This is available only when DatabaseType is MySQL.

Audit Service

The Audit Service is a mandatory iWD service that keep's track of all changes that occur to tasks within iWD.

The following properties are configurable for iWD Audit Service:

Property	Description
startAutomatically	Whether the service should be started automatically after the configuration deployment.
logLevel	Service log level. Should be set to Default, unless otherwise specified by Genesys Support.
logAgeInDays	Sets the number of days that log files should be kept in the system. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logMaxFileSizeMegaBytes	Sets a limit on the size of a single log file, in megabytes. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logFilesToKeep	Sets a limit on the number of log files that are kept for this service, excluding the current log file. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.

Property	Description
Database	*Mandatory dependency:* Database in which to store the audit information.

Logging Service

The Logging Service is a mandatory iWD service that provides internal logging capabilities within a iWD solution.

The following properties are configurable for the Logging Service:

Property	Description
startAutomatically	Whether the service should be started automatically after the configuration deployment.
logLevel	Service log level; should be set to Info, unless otherwise specified by Genesys Support. This property's value will be propagated to any other service that has logLevel set to Default.
logAgeInDays	Sets the number of days that log files should be kept in the system. A value of zero disables this limit. This property's value will be propagated to any other service that has logLevel set to Default.
logMaxFileSizeMegaBytes	Sets a limit on the size of a single log file, in megabytes. A value of zero disables this limit. This property's value will be propagated to any other service that has logLevel set to Default.
logFilesToKeep	Sets a limit on the number of log files that are kept for this service, excluding the current log file. This property's value will be propagated to any other service that has logLevel set to Default.
logDirectory	The directory in which the log files will be stored, for all services. If it starts with "/" (on Linux) or a drive letter (on Windows), an absolute path will be used; otherwise, the path is relative from the web applications folder on the application server.

Message Queue Service

The Message Queue Service is a mandatory iWD service that provides internal message-queue capabilities within a iWD solution.

The following properties are configurable for the Message Queue Service:

Property	Description
startAutomatically	Whether the service should be started automatically after the configuration deployment.
logLevel	Service log level. Should be set to Default, unless otherwise specified by Genesys Support.

Property	Description
logAgeInDays	Sets the number of days that log files should be kept in the system. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logMaxFileSizeMegaBytes	Sets a limit on the size of a single log file, in megabytes. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logFilesToKeep	Sets a limit on the number of log files that are kept for this service, excluding the current log file. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
Database	*Mandatory dependency:* Database where the messages will be persisted.

Scripting Service

The Scripting Service is an optional iWD service that provides scripting capabilities for message-transformation purposes (for XML File Capture Point and Websphere MQ Capture Point).

The following properties are configurable for the Scripting Service:

Property	Description
startAutomatically	Whether the service should be started automatically after the configuration deployment.
logLevel	Service log level. Should be set to Default, unless otherwise specified by Genesys Support.
logAgeInDays	Sets the number of days that log files should be kept in the system. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logMaxFileSizeMegaBytes	Sets a limit on the size of a single log file, in megabytes. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logFilesToKeep	Sets a limit on the number of log files that are kept for this service, excluding the current log file. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.

Broker Service

The Broker Service is a mandatory iWD service that provides common functionality, such as task persistence to other iWD services.

The following properties are configurable for the Broker Service:

Property	Description

Property	Description
startAutomatically	Whether the service should be started automatically after the configuration deployment.
Database	*Mandatory dependency:* Database where the task data will be stored.
logLevel	Service log level. Should be set to Default, unless otherwise specified by Genesys Support.
logAgeInDays	Sets the number of days that log files should be kept in the system. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logMaxFileSizeMegaBytes	Sets a limit on the size of a single log file, in megabytes. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logFilesToKeep	Sets a limit on the number of log files that are kept for this service, excluding the current log file. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
AuditService	*Mandatory dependency:* Audit Service.
MQService	*Mandatory dependency:* Message Queue Service.
notificationBatchSize	*Performance tuning:* The size of the batch in which the iWD internal notifications are handled.
maxThreads:	*Performance tuning:* Size of notification-processing thread pool.
idleSleepTimeSeconds	*Performance tuning:* The service-idle period when there are no more tasks to process.
stopProcessTimeoutSeconds	*Exception handling:* How long to wait for notification-handling threads to exit before a forced shutdown.

Rules Service

The Rules Service is a mandatory iWD service that allows iWD services to execute various rules.

The following properties are configurable for the Rules Service:

Property	Description
startAutomatically	Whether the service should be started automatically after the configuration deployment.
logLevel	Service log level. Should be set to Default, unless otherwise specified by Genesys Support.
logAgeInDays	Sets the number of days that log files should be kept in the system. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logMaxFileSizeMegaBytes	Sets a limit on the size of a single log file, in megabytes. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.

Property	Description
logFilesToKeep	Sets a limit on the number of log files that are kept for this service, excluding the current log file. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.

Classification Service

The Classification Service is a mandatory iWD service that performs task classification, based on business rules that are configured for capture points, contracts, and processes. The classification is performed, on a task-by-task basis, in four steps:

- Apply capture-point level rules, based on capture point via which the task is captured. After this step, the task must have an assigned process; otherwise, its status is set to ErrorHeld, and classification of the task is halted.
- Validate activation start/end dates of assigned contract and process, and reject task (change status to Rejected) if it falls outside of the configured date/time ranges.
- Apply contract-level rules
- Apply process-level rules

The following properties are configurable for the Classification Service:

Property	Description
startAutomatically	Whether the service should be started automatically after the configuration deployment.
logLevel	Service log level. Should be set to Default, unless otherwise specified by Genesys Support.
logAgeInDays	Sets the number of days that log files should be kept in the system. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logMaxFileSizeMegaBytes	Sets a limit on the size of a single log file, in megabytes. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logFilesToKeep	Sets a limit on the number of log files that are kept for this service, excluding the current log file. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
BrokerService	*Mandatory dependency:* Broker Service.
MQService	*Mandatory dependency:* Message Queue Service.
RulesService	*Mandatory dependency:* Rules Service.
backupFor	*High availability:* Primary classification service in a high-availability setup scenario. This service will perform a backup role in such a setup, and will take over processing if primary service becomes unavailable.

Property	Description
threads	*Performance tuning:* Number of parallel execution threads.
batchSize	*Performance tuning:* The size of the task batch that is processed in one cycle.
idleSleepTimeSeconds	*Performance tuning:* The service-idle period when there are no more tasks to process.

Prioritization Service

The Prioritization Service is a mandatory iWD service that performs task prioritization and repriotization, based on business rules configured for contract and/or process associated to a task.

Initially, each task is prioritized once, right after classification. After that, the task is reprioritized, based on the assigned reprioritization date/time of the task.

During each prioritization cycle, priority must be updated and a next reprioritization date/time must be set. If a reprioritization date/time is not set, the task will not be reprioritized any more. The same effect is achieved when no configured rule condition matches the task state during (re)prioritization (that is, no rule is applied and, hence, a reprioritization date/time is not set).

After the initial prioritization cycle, a task must have an assigned distribution point. If that's not the case, the task status will be set to ErrorHeld, and no more processing will take place until manual intervention.

The following properties are configurable for the Prioritization Service:

Property	Description
startAutomatically	Whether the service should be started automatically after the configuration deployment.
logLevel	Service log level. Should be set to Default, unless otherwise specified by Genesys Support.
logAgeInDays	Sets the number of days that log files should be kept in the system. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logMaxFileSizeMegaBytes	Sets a limit on the size of a single log file, in megabytes. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logFilesToKeep	Sets a limit on the number of log files that are kept for this service, excluding the current log file. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
BrokerService	*Mandatory dependency:* Broker Service.
MQService	*Mandatory dependency:* Message Queue Service.
RulesService	*Mandatory dependency:* Rules Service.
backupFor	*High availability:* Primary classification service in a high-availability setup scenario. This service will perform a backup

Property	Description
	role in such a setup, and will take over processing if primary service becomes unavailable.
maxPriority	The priority will be automatically limited to this value, if it exceeds it.
threads	*Performance tuning:* Number of parallel execution threads.
batchSize	*Performance tuning:* The size of the task batch that is processed in one cycle.
idleSleepTimeSeconds	*Performance tuning:* The service-idle period when there are no more tasks to process.

Webservice Capture Point

The Webservice Capture Point is a iWD service that allows third-party systems to submit and manipulate tasks in iWD via a SOAP Webservice interface. For more information on the Webservice Capture Point, see the iWD Webservice Capture API documentation.

The following properties are configurable for the Webservice Capture Point:

Property	Description
startAutomatically	Whether the service should be started automatically after the configuration deployment.
logLevel	Service log level. Should be set to Default, unless otherwise specified by Genesys Support.
logAgeInDays	Sets the number of days that log files should be kept in the system. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logMaxFileSizeMegaBytes	Sets a limit on the size of a single log file, in megabytes. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logFilesToKeep	Sets a limit on the number of log files that are kept for this service, excluding the current log file. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
webserviceURLMapping	The Webservice URL mapping. The URL is composed as follows: <runtime node context URL>/services/<webserviceURLMapping>, for example http://server:8080/evo.runtime.web/services/WebserviceCapturePoint. To retrieve a WSDL file for the Webservice, attach ?WSDL to the URL.

Property	Description
checkIfAlreadyCaptured	If true, the iWD will verify whether another task that has a given captureId already has been captured.
timezone	The time zone of the Webservice Capture Point. Date/time values will be converted from the specified time zone to UTC, before those values are stored in the iWD. If this parameter is not specified, it defaults to tenant time zone.
BrokerService	*Mandatory dependency:* Broker Service.

XML File Capture Point

The XML File Capture Point is a iWD service for capturing tasks from XML files. The XML files can be formatted according to the standard iWD XML message schema (described in detail in the iWD Websphere MQ Capture Point API document) or in a custom XML format.

To handle custom XML formats, two transformation scripts must be created: one for input transformation, and one for output. The iWD currently supports the Groovy scripting language for these purposes. For more information on transformation scripts, see the standard iWD input and output transformation scripts that are included in the iWD core package.

All tasks that are captured from a single XML file are grouped in a task batch.

The following properties are configurable for the XML File Capture Point:

Property	Description
startAutomatically	Whether the service should be started automatically after the configuration deployment.
logLevel	Service log level. Should be set to Default, unless otherwise specified by Genesys Support.
logAgeInDays	Sets the number of days that log files should be kept in the system. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logMaxFileSizeMegaBytes	Sets a limit on the size of a single log file, in megabytes. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logFilesToKeep	Sets a limit on the number of log files that are kept for this service, excluding the current log file. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
IncomingFileDirectory	The directory in which XML File Capture Point is looking for new files to capture.
CapturedFileDirectory	The directory in which captured files are put.
CompletedFileDirectory	The directory in which completed files are put. A file is considered completed when all of the tasks that are captured from the file are either completed, canceled, or rejected.

Property	Description
RejectedFileDirectory	The directory in which files for rejected tasks are put.
ErrorFileDirectory	The directory in which files that contain errors are put.
checkIfAlreadyCaptured	If true, the iWD will verify whether another task that has a given captureId already has been captured.
captureBatchSize	The maximum number of rows that are to be returned by the query that is specified in the captureQuerySql parameter. A value of zero sets the JDBC driver default value.
idleSleepTimeSeconds	Service-idle period when there are no more tasks to process.
timezone	The time zone of the XML File Capture Point. Date/time values will be converted from the specified time zone to UTC, before those values are stored in the iWD. Also, any date/time values that are included in response XML files will be converted to the specified time zone. If this parameter is not specified, it defaults to tenant time zone.
BrokerService	*Mandatory dependency:* Broker Service.
MQService	*Mandatory dependency:* Message Queue Service.
TransformScriptingService	*Optional dependency:* Scripting Service. If specified, XML input/output will be transformed using input-/output-transformation scripts.
TransformInputScript	The input-transformation script; must be specified when TransformScriptingService is selected.
TransformOutputScript	The output-transformation script; must be specified when TransformScriptingService is selected.
TransformOutputRootNode	The name of the XML root node in output files.
backupFor	*High Availability:* The primary XML File Capture Point in a high-availability setup scenario. This service will perform a backup role in such a setup, and will take over processing if the primary service becomes unavailable.

Websphere MQ Capture Point

The Websphere MQ Capture Point is a iWD service for capturing and manipulating tasks via the Websphere Message Queue interface. The messages must be formatted in XML.

iWD provides a standard iWD XML message schema (described in detail in the iWD Websphere MQ Capture Point API document). Alternatively, it is possible to use custom XML formats via message-transformation scripts.

To handle custom XML formats, two transformation scripts must be created: one for input transformation, and one for output. The iWD currently supports the Groovy scripting language for these purposes. For more information on transformation scripts, see the standard iWD input and output transformation scripts that are included in the iWD core package.

The following properties are configurable for the Websphere MQ Capture Point:

Property	Description
startAutomatically	Whether the service should be started automatically after the configuration deployment.
logLevel	Service log level. Should be set to Default, unless otherwise specified by Genesys Support.
logAgeInDays	Sets the number of days that log files should be kept in the system. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logMaxFileSizeMegaBytes	Sets a limit on the size of a single log file, in megabytes. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logFilesToKeep	Sets a limit on the number of log files that are kept for this service, excluding the current log file. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
mqHost	The Websphere MQ server address
mqPort	The Websphere MQ server port
userName	The Websphere MQ server username
password	The Websphere MQ server password
queueManagerName	The Websphere MQ queue manager name
managerChannel	The Websphere MQ queue manager channel
managerCCSID	The Websphere MQ queue manager CCSID
messageEncoding	The Websphere MQ message encoding
queueInbound	The inbound queue from which the iWD messages will be retrieved.
queueOutbound	The outbound queue to which the iWD notifications will be submitted.
queueWorkerThreads	*Perfomance tuning:*The size of the thread pool.
reconnectTimeoutSeconds	*Exception handling:* The connection timeout.
checkIfAlreadyCaptured	If true, the iWD will verify whether another task that has a given captureId already has been captured.
timezone	The time zone of the Websphere MQ Capture Point. Date/time values will be converted from the specified time zone to UTC, before those values are stored in the iWD. Also, any date/time values that are included in response messages will be converted to the specified time zone. If this parameter is not specified, it defaults to tenant time zone.
BrokerService	*Mandatory dependency:* Broker Service.
MQService	*Mandatory dependency:* Message Queue Service.
backupFor	*High availability:* The primary Websphere MQ Capture

Property	Description
	Point in a high-availability setup scenario. The service will perform a backup role in such a setup, and will take over processing if the primary service becomes unavailable.
TransformScriptingService	*Optional dependency:* Scripting Service. If specified, XML input/output will be transformed using input-/output-transformation scripts.
TransformInputScript	The input-transformation script; must be specified when TransformScriptingService is selected.
TransformOutputScript	The output-transformation script; must be specified when TransformScriptingService is selected.
TransformOutputRootNode	The name of the XML root node in output files.
NotifyError	Whether to send error notifications to an outbound queue; available only when transformation is not enabled.
NotifyTaskAssigned	Whether to submit "task assigned" notifications to outbound queue; available only when transformation is not enabled.
NotifyTaskCanceled	Whether to submit "task cancelled" notifications to outbound queue; available only when transformation is not enabled.
NotifyTaskCreated	Whether to submit "task created" notifications to outbound queue; available only when transformation is not enabled.
NotifyTaskRejected	Whether to submit "task rejected" notifications to outbound queue; available only when transformation is not enabled.
NotifyTaskCompleted	Whether to submit "task completed" notifications to outbound queue; available only when transformation is not enabled.
NotifyTaskDistributed	Whether to submit "task distributed" notifications to outbound queue; available only when transformation is not enabled.
NotifyTaskDistributedQueue	Whether to submit "task moved into distribution queue" notifications to outbound queue; available only when transformation is not enabled.
NotifyTaskFinished	Whether to submit "task finished" notifications to outbound queue; available only when transformation is not enabled.
NotifyTaskHeld	Whether to submit "task held" notifications to outbound queue; available only when transformation is not enabled.
NotifyTaskRestarted	Whether to submit "task restarted" notifications to outbound queue; available only when transformation is not enabled.
NotifyTaskReturned	Whether to submit "task returned" notifications to outbound queue; available only when transformation is not enabled.
NotifyTaskResumed	Whether to submit "task resumed" notifications to outbound queue; available only when transformation is not enabled.

Property	Description
NotifyTaskUpdated	Whether to submit "task updated" notifications to outbound queue; available only when transformation is not enabled.

Database Capture Point

The Database Capture Point is an asynchronous capture-point service that creates tasks in the iWD, based on records in a database. As a task in the iWD goes through its life-cycle stages, the Database Capture Point allows you to make updates to the database, to mimic the task life cycle in the originating system. The Database Capture Point allows execution of a certain database query or update statement, depending on the task state:

- Capture: The Database Capture Point scans the result set that is returned from the originating database by a special "capture" query and creates tasks in the iWD that are based on the information that is provided by each result-set record.
- Captured: After the task is created in the iWD, the Database Capture Point may execute a "captured" update query to update the respective database record and notify the originating system that the task for the respective row in the database already has been created.
- Distributed: The iWD has distributed the task to an external system for further processing. The Database Capture Point may execute a "distributed" update query to update the respective database record and notify the originating system that the iWD task for the respective row in the database has been distributed.
- Assigned: The task has been assigned to a user within an external system. The Database Capture Point may execute an "assigned" update query to update the respective database record and notify the originating system that the iWD task for the respective row in the database has been assigned to the user.
- Completed: The task has been completed by a user within an external system. The Database Capture Point may execute a "completed" update query to update the respective database record and notify the originating system that the iWD task for the respective row in the database has been completed.
- Returned: The task has been returned to the iWD for distribution at a later time. The Database Capture Point may execute a "returned" update query to update the respective database record and notify the originating system that the iWD task for the respective row in the database has been returned to the iWD.
- Restarted: The task has been restarted (that is, returned to the iWD for reprocessing and distribution at a later time). The Database Capture Point may execute a "restarted" update query to update the respective database record and notify the originating system that the iWD task for the respective row in the database has been restarted.

The following properties are configurable for the Database Capture Point:

Property	Description
startAutomatically	Whether the service should be started automatically after the configuration deployment.

Property	Description
logLevel	Service log level. Should be set to Default, unless otherwise specified by Genesys Support.
logAgeInDays	Sets the number of days that log files should be kept in the system. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logMaxFileSizeMegaBytes	Sets a limit on the size of a single log file, in megabytes. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logFilesToKeep	Sets a limit on the number of log files that are kept for this service, excluding the current log file. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
jdbcDriver	The name of the JDBC driver class to access the database.
jdbcURL	The URL of your JDBC connection. For information on how to construct a valid URL, see your JDBC driver documentation.
username	The name of the database user.
password	The password of the database user.
idField	The name of the column that uniquely identifies the record in in the result set that is returned by the database query that is specified in the captureQuerySql parameter. The value of the field is mapped to the captureId parameter in the consecutive update statements.
processIdField	The name of the column that contains the process ID. If the value of this field matches a process ID in the iWD, a process and contract will be assigned to the task.
fieldsToAttach	A list of fields that will be attached to task data. Multiple fields should be separated by a comma. This property also supports * and ? wildcards. For example: `field1,field2,special*` This expression means that the fields that have the names "field1" and "field2" will be attached, as well as any field that has a name that starts with "special".
captureQuerySql	The database query that returns the result set in which each row will be captured as a task in the iWD. The result set that is returned by this query must contain also the column that is specified in the idField parameter.

Property	Description
	For example: `select * from TABLE where STATUS="new"`
capturedUpdateSql	The database update statement that updates the database to reflect that certain data already has been captured as a task in the iWD. The captureId parameter can be used to reference the particular row. For example: `update TABLE set STATUS='captured' where ROW_ID=:captureId`
errorUpdateSql	The database update statement that updates the database to reflect that the associated task in the iWD has been held by an error. The captureId parameter can be used to reference the particular row. For example: `update TABLE set STATUS='error' where ROW_ID=:captureId`
distributedUpdateSql	The database update statement that updates the database to reflect that the associated task in the iWD has been distributed. The captureId parameter can be used to reference the particular row. For example: `update TABLE set STATUS='distributed' where ROW_ID=:captureId`
assignedUpdateSql	The database update statement that updates the database to reflect that the associated task in the iWD has been assigned. The captureId parameter can be used to reference the particular row. The userId parameter can be used to set the name of the user who is working on the task. For example: `update TABLE set STATUS='assigned', USER=:userId where ROW_ID=:captureId`
completedUpdateSql	The database update statement that updates the database to reflect that the associated task in the iWD has been completed. The captureId parameter can be used to reference the particular row. For example:

Property	Description
	`update TABLE set STATUS='restarted' where ROW_ID=:captureId`
rejectedUpdateSql	The database update statement that updates the database to reflect that the associated task has been rejected in the iWD. The captureId parameter can be used to reference the particular row. For example: `update TABLE set STATUS='rejected' where ROW_ID=:captureId`
restartedUpdateSql	The database update statement that updates the database to reflect that the associated task in the iWD has been restarted in the iWD. The captureId parameter can be used to reference the particular row. For example: `update TABLE set STATUS='restarted' where ROW_ID=:captureId`
returnedUpdateSql	The database update statement that updates the database to reflect that the associated task in the iWD has been returned to the iWD for later distribution. The captureId parameter can be used to reference the particular row. For example: `update TABLE set STATUS='returned' where ROW_ID=:captureId`
timezone	The time zone of the target database. Date/time values will be converted from the specified time zone to UTC, before those values are stored in the iWD. If this parameter is not specified, it defaults to tenant time zone.
checkIfAlreadyCaptured	If true, the iWD will verify whether another task that has the given captureId already has been captured.
BrokerService	*Mandatory dependency:* Broker Service.
MQService	*Mandatory dependency:* Message Queue Service.
backupFor	*High Availability:* The primary Database Capture Point in a high-availability setup scenario. This service will perform a backup role in such a setup, and will take over processing if the primary service becomes unavailable.
captureBatchSize	*Perfomance tuning:* The maximum number of rows that are to be returned by the query specified in the captureQuerySql parameter. A value of 0 sets the JDBC driver default value.
threads	*Perfomance tuning:* The size of the thread pool.

Property	Description
idleSleepTimeSeconds	*Perfomance tuning:* The service-idle period when there are no more tasks to process.

Genesys Distribution Point

The Genesys Distribution Point consists of the following services:

- Genesys Distribution Point service
- Genesys Configuration Server Connector service
- Genesys Interaction Server Connector service
- Genesys T-Server Connector service
- Genesys Synchronization Service
- Genesys Statistics Adapter

Genesys connector services (Genesys Configuration Server Connector, Genesys Interaction Server Connector, and Genesys T-Server Connector) provide physical connection to respective Genesys servers and can be shared by several Genesys Distribution Point services, if necessary.

Genesys Distribution Point service

The Genesys Distribution Point service is a daemon service that scans its task stack in a timely fashion and distributes iWD tasks to the Genesys system via the Genesys Interaction Server Connector service. iWD tasks are distributed as Genesys interactions to the specified Genesys "inbound" interaction queue; here, the interactions can be picked up by a Genesys routing strategy. The Genesys Distribution Point service receives notifications about the processing progress of each task in the Genesys system and propagates any state changes back to the iWD and the capture-point service that initially captured the task. A task is considered to be completed when the Genesys interaction is put in the specified "outbound" Genesys interaction queue.

The following properties are configurable for the Genesys Distribution Point service:

Property	Description
startAutomatically	Whether the service should be started automatically after the configuration deployment.
logLevel	Service log level. Should be set to Default, unless otherwise specified by Genesys Support.
logAgeInDays	Sets the number of days that log files should be kept in the system. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logMaxFileSizeMegaBytes	Sets a limit on the size of a single log file, in megabytes. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logFilesToKeep	Sets a limit on the number of log files that are kept for

Property	Description
	this service, excluding the current log file. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
AttachIncludeFilter	A comma-separated list of wildcards that specify what attached data name-value pairs should be included in a name-value exchange with Genesys. The subset that is defined by this filter can be negated by the filter that is specified in the AttachExcludeFilter property. Note: A name-value pair exchange includes the transfer of iWD-task data to a Genesys interaction, as well as the propagation of added/changed key-value pairs from a Genesys interaction to the iWD task. The wildcards that are allowed are the following: * (asterisk) any number of any characters ? (question mark) any character For example: *Agent*: Any pair whose name contains "Agent" ?code: Any pair whose name starts with any character and ends with "code"
AttachExcludeFilter	A comma-separated list of wildcards that specify what attached data name-value pairs should be excluded from a name-value exchange with Genesys. This filter applies only to the subset of names that are allowed by the AttachIncludeFilter property. The wildcards that are allowed are the following: * (asterisk) any number of any characters ? (question mark) any character For example: *Agent*: Any pair whose name contains "Agent" ?code: Any pair whose name starts with any character and ends with "code"
inboundQueueName	The name of the Genesys "outbound" interaction queue. All iWD tasks are submitted to this interaction queue as Genesys interactions.
outboundQueueName	The name of the Genesys "outbound" interaction queue. The iWD marks all tasks as complete, if their associated Genesys interactions are put into this interaction queue.
restartQueueName	The name of the Genesys "restart" interaction queue. The iWD resets the task state to New for each task, if its associated Genesys interactions are put into this

Property	Description
	interaction queue. This action subjects the iWD task to a series of procedures, as if the task had just been captured; that is, a full set of classification and prioritization rules is run against the task.
returnQueueName	The name of the Genesys "return" interaction queue. The iWD resets the task state to Queued for each task, if its associated Genesys interactions are put into this interaction queue. A iWD prioritization rule set is run against such an interaction, until it is redistributed to Genesys.
idleSleepTimeSeconds	*Performance tuning:* The time, in seconds, that the service is idle between individual task-distribution iterations.
distributionThreshold	The maximum number of distributed tasks for the distribution point. When this number is reached, the distribution-point service submits no new tasks to Genesys, until some tasks are completed or returned to iWD.
notInWorkbinThreshold	The maximum number of distributed tasks for the distribution point, excluding tasks in a workbin. When this number is reached, the distribution-point service submits no new tasks to Genesys, until some tasks are completed, returned to the iWD, or placed in a workbin.
enableTaskReplace	When distributionTreshhold is reached, the distribution point will check for tasks that have a higher priority than tasks that already have been distributed. If such tasks are detected, the distribution point will automatically revoke some tasks that have a lower priority; thus, freeing up space for tasks that have a higher priority.
threads	*Performance tuning:* The number of concurrent working threads. The Distribution Point service maintains a pool of threads, and each thread can perform distribution of a single task. A Multi threaded approach helps to increase performance of the Distribution Point service.
distributionBatchSize	*Performance tuning:* The maximum number of tasks that the distribution point attempts to distribute during one iteration.
updateGenesysPriority	If checked, the Genesys interaction priority will be set

Property	Description
	directly from the iWD task priority; otherwise, the iWD priority will be attached as a regular CAD field that has the key GTL_priority.
pingGenesysWhenIdling	If checked, Genesys Distribution Point will periodically ping Genesys Interaction Server to ensure that the connection is alive.
stopInteractionOnComplete	When this option is checked, and when the task has been completed in the iWD by a third-party system via a capture point, the associated interaction will be revoked from the Genesys Interaction Server.
emergencySleepThreshold	*Exception handling:* After how many failed distribution attempts to hold processing.
emergencySleepSeconds	*Exception handling:* For how long to hold processing after emergencySleepThreshold failed distribution attempts.
timeZone	The time zone of the Genesys system. All date/time values that are attached to a Genesys interaction will be converted to the specified time zone. If this parameter is not specified, it defaults to tenant time zone.
genesysInteractionServerConnector	*Mandatory dependency:* The Genesys Interaction Server Connector service that should be used for connectivity to to the Genesys Interaction Server.
BrokerService	*Mandatory dependency:* Broker Service
MQService	*Mandatory dependency:* Message Queue Service
taskInfoService	An iWD task-information service; provides detailed iWD task information to external systems.
backpFor	*High Availability:* Specifies that the service is a backup service for another Genesys Distribution Point service.

Genesys Configuration Server Connector service

The Genesys Configuration Server Connector service maintains a physical connection to the Genesys configuration server and allows other iWD Genesys Distribution Point services to access the Genesys configuration-management environment.

The following properties are configurable for the Genesys Configuration Server Connector service:

Property	Description

Property	Description
startAutomatically	Whether the service should be started automatically after the configuration deployment.
logLevel	Service log level. Should be set to Default, unless otherwise specified by Genesys Support.
logAgeInDays	Sets the number of days that log files should be kept in the system. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logMaxFileSizeMegaBytes	Sets a limit on the size of a single log file, in megabytes. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logFilesToKeep	Sets a limit on the number of log files that are kept for this service, excluding the current log file. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
configServerPrimaryHost	The host name of the primary Genesys configuration server.
configServerPrimaryPort	The port of the primary Genesys configuration server.
configServerBackupHost	The host name of the backup Genesys configuration server.
configServerBackupPort	The port of the backup Genesys configuration server.
genesysUserName	User name for Genesys configuration-management environment.
genesysApplicationName	The Genesys Configuration Server Connector application in the Genesys configuration-management environment.
genesysUserPassword	Password of the Genesys configuration-management environment.
reconnectionAttempts	*Exception handling:* The maximum number of attempts to connect to the Genesys configuration server, before connecting to the backup server.
reconnectionPeriod	*Exception handling:* The time, in seconds, between individual reconnection attempts.
protocolTimeout	*Exception handling:* The timeout of the Genesys configurationserver protocol; specifies the number of milliseconds that the connector will wait for Genesys to respond to the request.
eventBufferSize	*Performance tuning:* The number of Genesys events that can be queued up in memory, until they are processed by the service.

Genesys Interaction Server Connector service

The Genesys Interaction Server Connector service maintains a physical connection to the Genesys interaction server and allows the iWD Genesys Distribution Point to interface with the Genesys interaction server. This service uses the Genesys Configuration Server Connector service to retrieve additional configuration data, such as the names of the host and the ports of the target Genesys interaction server.

The following properties are configurable for the Genesys Interaction Server Connector service:

Property	Description
startAutomatically	Whether the service should be started automatically after the configuration deployment.
logLevel	Service log level. Should be set to Default, unless otherwise specified by Genesys Support.
logAgeInDays	Sets the number of days that log files should be kept in the system. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logMaxFileSizeMegaBytes	Sets a limit on the size of a single log file, in megabytes. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logFilesToKeep	Sets a limit on the number of log files that are kept for this service, excluding the current log file. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
genesysApplicationName	The name of the Genesys Interaction Server Connector application in the Genesys configuration-management environment.
genesysTenantName	The name of Genesys tenant, as defined in Genesys configuration-management environment.
genesysConfigurationServerConnector	*Mandatory dependency:* The Genesys Configuration Server Connector service; provides access to the Genesys configuration-management environment.
protocolTimeout	*Exception handling:* The timeout of the Genesys interaction-server protocol; specifies the number of milliseconds that the connector will wait for Genesys to respond to the request.
eventBufferSize	*Performance tuning:* The number of Genesys events that can be queued up in memory, until they are processed by the service.

Genesys T-Server Connector service

The Genesys T-Server Connector service maintains a physical connection to Genesys T-Server and allows the iWD Genesys Statistics Adapter service to interface with Genesys T-Server. The service uses the Genesys Configuration Server Connector service to retrieve additional configuration data, such as the names of the host and the ports of the target Genesys T-Server.

The following properties are configurable for the Genesys T-Server Connector service:

Property	Description
startAutomatically	Whether the service should be started automatically after the configuration deployment.
logLevel	Service log level. Should be set to Default, unless otherwise specified by Genesys Support.
logAgeInDays	Sets the number of days that log files should be kept in the system. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logMaxFileSizeMegaBytes	Sets a limit on the size of a single log file, in megabytes. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logFilesToKeep	Sets a limit on the number of log files that are kept for this service, excluding the current log file. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
genesysApplicationName	The name of the Genesys T-Server Connector application in the Genesys configuration-management environment.
genesysTenantName	The name of the Genesys tenant, as defined in the Genesys configuration-management environment.
genesysConfigurationServerConnector	*Mandatory dependency:* The Genesys Configuration Server Connector service; provides access to the Genesys configuration-management environment.
protocolTimeout	*Exception handling:* The timeout of the Genesys T-Server protocol; specifies the time, in milliseconds, that the connector will wait for Genesys to respond to the request.
eventBufferSize	*Performance tuning:* The number of Genesys events that can be queued up in memory, until they are processed by the service.

Genesys Synchronization Service

The Genesys Synchronization service synchronizes the task state in iWD with interactions in the Genesys interaction server. This service will complete a task that has been distributed, but for which the associated interaction in the Genesys interaction server is missing. The Genesys Synchronization service can be scheduled to run, based on the specified CRON expression. Additionally, this service automatically can schedule itself, if the Genesys Interaction Server Connector service sends a notification about a lost connection to the Genesys interaction server.

The following properties are configurable for the Genesys Synchronization service:

Property	Description
startAutomatically	Whether the service should be started automatically after the configuration deployment.
logLevel	Service log level. Should be set to Default, unless otherwise specified by Genesys Support.
logAgeInDays	Sets the number of days that log files should be kept in the system. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logMaxFileSizeMegaBytes	Sets a limit on the size of a single log file, in megabytes. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logFilesToKeep	Sets a limit on the number of log files that are kept for this service, excluding the current log file. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
synchronizationPlace	*Exception handling:* The Genesys Place ID that is used by the Genesys Distribution Point to log in, for task-synchronization purposes (performed each time that Genesys Distribution Point is started or a connection to the Genesys interaction server is resumed after a disconnect).
synchronizationPlaceBackup	*Exception handling:* The backup Genesys Place ID that is used if the primary place ID (synchronizationPlace property) is busy.
threads	*Performance tuning:* The number of concurrent working threads. The Genesys Synchronization Service maintains a pool of threads, and each thread can perform the synchronization of a single task. A multithreaded approach helps to increase performance of the Genesys Synchronization Service.
synchronizationDelayMinutes	How many minutes to wait before synchronization, in case an associated Genesys Interaction Server Connector

Property	Description
	service reports an error with connectivity to the Genesys interaction server. This value is used to schedule the Genesys Synchronization service automatically.
genesysInteractionServerConnector	*Mandatory dependency:* The Genesys Interaction Server Connector service that should be used for connectivity to to the Genesys interaction server.

Genesys Statistics Adapter service

The Genesys Statistics Adapter service delivers real-time statistics from the iWD Statistics service to the Genesys Stat Server. Statistics are delivered via Genesys T-Server as custom UserEvents and are associated with a virtual queue object. The service uses the Genesys Configuration Server Connector service to update Genesys Stat Server application options with all of the necessary statistical-object definitions and filters, which allows viewing of these statistics in the Genesys CCPulse+ application. The exact set of statistics that are submitted depends on the configuration of the iWD Statistics service.

The following properties are configurable for the Genesys Statistics Adapter service:

Property	Description
startAutomatically	Whether the service should be started automatically after the configuration deployment.
logLevel	Service log level. Should be set to Default, unless otherwise specified by Genesys Support.
logAgeInDays	Sets the number of days that log files should be kept in the system. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logMaxFileSizeMegaBytes	Sets a limit on the size of a single log file, in megabytes. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logFilesToKeep	Sets a limit on the number of log files that are kept for this service, excluding the current log file. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
dimensionMapping	Defines how statistical dimensions are mapped: Filter: Dimensions are mapped to CCPulse+ filters; filter definitions can be pushed automatically to configuration of the Genesys Stat Server (see genesysStatServerName property for details). Virtual Queue: Dimensions are mapped to Genesys

Property	Description
	virtual queues.
virtualQueueName	The name of the Genesys virtual queue with which UserEvents should be associated.
genesysStatServerName	The name of the Genesys Stat Server that will work with submitted statistics. The property is required only if there is a need to automatically update Genesys Stat Server application options with all of the necessary statistical-object definitions and filters. The Genesys Statistics Adapter service will use this name to access the Genesys Stat Server configuration in the Genesys configuration-management environment.
genesysTenantName	The name of the Genesys tenant, as defined in the Genesys configuration-management environment.
genesysConfigurationServerConnector	*Mandatory dependency:* The Genesys Configuration Server Connector.
genesysTServerConnector	*Mandatory dependency: The* Genesys T-Server Connector.

Configuration of iWD Real-Time Statistics in Genesys CCPulse+

In Genesys, iWD real-time-statistic measure types are presented as statistic objects, and iWD statistic dimensions are defined as filters. Therefore, it is necessary to combine statistic objects with filters in order to get a measure type for a dimension.

- Start CCPulse+, and connect to the Genesys Stat Server (File > New).
- In the Object Types dialog box, select the Virtual Queue object type for the corresponding Genesys tenant and switch, and click Next.
- In the Object Instances dialog box, select the virtual queue that is used by the iWD to submit statistics, and click Finish.

To be able to view iWD real-time statistics in Genesys CCPulse+, it is necessary to create a CCPulse+ template:

- Open the CCPulse+ template wizard (Tools > Template Wizard...).
- In the Template Definition dialog box, select the Virtual Queue object type in the Available Object Types list, then select the Create New Template option, and click Next.
- In the Pre-defined Statistics dialog box, enter the template name, then click the New Group button, and enter a name for the newly created statistic group.

- Expand the TotalCustomValue stat type in Available Statistics tree view, and select the iWD statistic that you want to add to the template (all iWD statistic names have a "GTL" prefix).
- Click the >> button to add the selected statistic to the newly created statistic group.
- In Requested Statistics tree view, select the newly added statistic, and then click the Properties button.
- In the Statistic Properties dialog box, enter Alias (which is how the statistic is displayed in CCPulse+). Set Insensitivity Value to 1; set Interval Type to Selection and GTLAggregated; set Notification Mode to Changes Based; and set Filter to the filter type that represents the required iWD dimension; and then click OK.
- Add more statistics and statistic groups, if necessary, and then click the Next button.
- In the Graph dialog box, adjust graph parameters, if necessary, and then click Finish.

In the CCPulse+ main window, select the virtual-queue instance that is used by the iWD to submit statistics. Create a real-time view for this virtual queue:

- In the Real-Time Data Template dialog box, select the previously created template from the list, and click OK.

iWD Extended Statistics Service

The iWD Extended Statistics service is a scheduled ETL service that executes a set of preconfigured ETL plug-ins and submits results to the Genesys Stat Server via the Genesys Statistics Adapter service. Extended statistics are created as Kettle plug-ins that are located in the `$KETTLE_REPOS_DIR/aggregate_stats/stats` directory. In order to add a new plug-in, you need to add a path to its transformation file to the `$KETTLE_REPOS_DIR/aggregate_stats/stats.properties` file.

The following properties are configurable for the Extended Statistics service:

Property	Description
startAutomatically	Whether the service should be started automatically after the configuration deployment.
logLevel	Service log level. Should be set to Default, unless otherwise specified by Genesys Support.
logAgeInDays	Sets the number of days that log files should be kept in the system. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logMaxFileSizeMegaBytes	Sets a limit on the size of a single log file, in megabytes. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logFilesToKeep	Sets a limit on the number of log files that are kept for this service, excluding the current log file. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.

Property	Description
triggerMode	How the ETL job will be triggered (started): • Trigger using CRON expression: Starts a job automatically, based on a CRON scheduling expression. • Trigger after another scheduled service: Starts a job automatically after another scheduled job has finished. • Trigger manually: Job can be started manually via iWD Manager Services Status screen.
cronExpression	Standard CRON scheduling expression when triggerMode is set to "Trigger using CRON expression". For example, the following expression will cause the job to be executed every 15 minutes: "0 0,15,30,45 * * * ?". See http://en.wikipedia.org/wiki/Cron for more information.
executionQueueName	Scheduled services that have the same executionQueueName will never be run in parallel within the same solution. If a scheduled service is triggered while another scheduled service that has the same executionQueueName is in progress, it will be queued until the other service has completed processing.
StatAdapter	*Mandatory dependency:* The Genesys Statistics Adapter service that will submit statistics to the Genesys Stat Server

Simulator Distribution Point

Simulator Distribution Point is a iWD service that allows to simulate task distribution without an actual routing system in place. It also supports simulation of agents via two additional helper services - Simulator Assigner and Simulator Completer described below.

The following properties are configurable for this service:

Property	Description
startAutomatically	Whether the service should be started automatically after the configuration deployment.
logLevel	Service log level. Should be set to Default, unless otherwise specified by Genesys Support.
logAgeInDays	Sets the number of days that log files should be kept in the system. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logMaxFileSizeMegaBytes	Sets a limit on the size of a single log file, in megabytes. A value of zero disables this limit. This property is not available if

Property	Description
	logLevel is set to Off or Default.
logFilesToKeep	Sets a limit on the number of log files that are kept for this service, excluding the current log file. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
AttachIncludeFilter	A comma-separated list of wildcards that specify what attached data name-value pairs should be included in a name-value exchange with Genesys. The subset that is defined by this filter can be negated by the filter that is specified in the AttachExcludeFilter property. Note: A name-value pair exchange includes the transfer of iWD-task data to a Genesys interaction, as well as the propagation of added/changed key-value pairs from a Genesys interaction to the iWD task. The wildcards that are allowed are the following: * (asterisk) any number of any characters ? (question mark) any character For example: *Agent*: Any pair whose name contains "Agent" ?code: Any pair whose name starts with any character and ends with "code"
AttachExcludeFilter	A comma-separated list of wildcards that specify what attached data name-value pairs should be excluded from a name-value exchange with Genesys. This filter applies only to the subset of names that are allowed by the AttachIncludeFilter property. The wildcards that are allowed are the following: * (asterisk) any number of any characters ? (question mark) any character For example: *Agent*: Any pair whose name contains "Agent" ?code: Any pair whose name starts with any character and ends with "code"
idleSleepTimeSeconds	*Performance tuning:* The time, in seconds, that the service is idle between individual task-distribution iterations.
distributionThreshold	The maximum number of distributed tasks for the distribution point. When this number is reached, the distribution-point service submits no new tasks to Genesys, until some tasks are completed or returned to iWD.
threads	*Perfomance tuning:The number of concurrent working threads. The Distribution Point service maintains a pool of threads, and each thread can perform distribution of a single task. A Multi*

Property	Description
	threaded approach helps to increase performance of the Distribution Point service.
distributionBatchSize	*Perfomance tuning:* The maximum number of tasks that the distribution point attempts to distribute during one iteration.
timeZone	The time zone of the Simulator Distribution Point. All date/time values that are attached to the simulated interaction will be converted to the specified time zone. If this parameter is not specified, it defaults to tenant time zone.
BrokerService	*Mandatory dependency:* Broker Service.
MQService	*Mandatory dependency:* Message Queue Service.
backpFor	*High Availability*: Specifies that the service is a backup service for another Simulator Distribution Point service.

Simulator Assigner

The Simulator Assigner service assigns tasks that are distributed via the Simulator Distribution Point to virtual (simulated) agents.

Property	Description
startAutomatically	Whether the service should be started automatically after the configuration deployment.
logLevel	Service log level. Should be set to Default, unless otherwise specified by Genesys Support.
logAgeInDays	Sets the number of days that log files should be kept in the system. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logMaxFileSizeMegaBytes	Sets a limit on the size of a single log file, in megabytes. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logFilesToKeep	Sets a limit on the number of log files that are kept for this service, excluding the current log file. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
batchSize	*Performance tuning:* The number of tasks that are processed in each processing cycle.
numberOfVirtualAgents	The number of virtual agents.
agentIdleTimeSeconds	Time, in seconds, that an agent is idle before accepting a new task.
threads	*Performance tuning:* The number of concurrent working threads.

Property	Description
idleSleepTimeSeconds	*Performance tuning:* The time, in seconds, that the service is idle after an empty processing cycle.
BrokerService	*Mandatory dependency:* Broker Service.
simulatorDistributionPoint	*Mandatory dependency:* The Simulator Distribution Point.

Simulator Completer

The Simulator Completer service completes tasks that are assigned to virtual agents by the Simulator Assigner.

Property	Description
startAutomatically	Whether the service should be started automatically after the configuration deployment.
logLevel	Service log level. Should be set to Default, unless otherwise specified by Genesys Support.
logAgeInDays	Sets the number of days that log files should be kept in the system. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logMaxFileSizeMegaBytes	Sets a limit on the size of a single log file, in megabytes. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logFilesToKeep	Sets a limit on the number of log files that are kept for this service, excluding the current log file. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
batchSize	*Performance tuning:* The number of tasks that are processed in each processing cycle.
agentWorkTimeSeconds	The time, in seconds, that an agent works on a task.
threads	*Perfomance tuning:* The number of concurrent working threads.
idleSleepTimeSeconds	*Perfomance tuning:* The time, in seconds, that the service is idle between processing cycles.
BrokerService	*Mandatory dependency:* Broker Service.
simulatorDistributionPoint	*Mandatory dependency:* The Simulator Distribution Point.

Datamart ETL Services

The Data Mart ETL services enable batch loading and aggregation of data into the iWD Data Mart. The Kettle ETL service defines general ETL configuration, such as the database in which the aggregated data will be stored. The Scheduled ETL Job service defines execution characteristics (such as scheduling) for each specific ETL job.

Kettle ETL Service

The Kettle ETL service represents an embedded Kettle ETL Engine that runs ETL scripts for loading and aggregating task and configuration data into the iWD Data Mart.

Property	Description
startAutomatically	Whether the service should be started automatically after the configuration deployment.
logLevel	Service log level. Should be set to Default, unless otherwise specified by Genesys Support.
logAgeInDays	Sets the number of days that log files should be kept in the system. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logMaxFileSizeMegaBytes	Sets a limit on the size of a single log file, in megabytes. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logFilesToKeep	Sets a limit on the number of log files that are kept for this service, excluding the current log file. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
repositoryType	The Kettle repository type; must be set to "Directory".
repositoryDirectory	The directory on the server in which iWD Data Mart ETL scripts are stored. If the iWD Manager application and the iWD runtime node are located on different computers, this directory should be accessible on both via the same path.
customTaskAttributeMapping	Up to 10 comma-separated names of a task's custom attribute that will be loaded into task_fact custom-attribute fields (CUSTOM_ATTRIBUTE1-10).
customTaskDimensionMapping	Up to 5 comma-separated names of a task's custom attribute that will be loaded into the CUSTOM_DIM dimension and associated to the task via the CUSTOM_DIM_KEY field.
customTenantAttributeMapping	Up to 5 comma-separated names of a tenant's custom attribute attribute that will be loaded into the CUSTOM_DIM dimension and associated to the tenant via the CUSTOM_DIM_KEY field.
customContractAttributeMapping	Up to 5 comma-separated names of a contract's custom attribute that will be loaded into the CUSTOM_DIM dimension and associated to the contract via the CUSTOM_DIM_KEY field.
customProcessAttributeMapping	Up to 5 comma-separated names of a process's custom attribute that will be loaded into the CUSTOM_DIM dimension and associated to the process via the CUSTOM_DIM_KEY field.

Property	Description
detailsExpirationDays	The number of days after which the detailed task (task_fact, task_event_fact, and task_work_fact) data will be removed from the database.
aggregation15minExpirationDays	The number of days after which the data will be removed from 15-min aggregation tables.
brokerDatabase	*Mandatory dependency:* Database from which to load the iWD task and audit data.
datamartDatabase	*Mandatory dependency:* Database in which to load and aggregate reportable data (Data Mart).
extendedStatisticsService	*Optional depandency:* Extended statistics service to use for the delivery of statistics.

Scheduled ETL Job

The following properties are configurable for the Schedued ETL Job service:

Property	Description
startAutomatically	Whether the service should be started automatically after the configuration deployment.
logLevel	Service log level. Should be set to Default, unless otherwise specified by Genesys Support.
logAgeInDays	Sets the number of days that log files should be kept in the system. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logMaxFileSizeMegaBytes	Sets a limit on the size of a single log file, in megabytes. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logFilesToKeep	Sets a limit on the number of log files that are kept for this service, excluding the current log file. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
triggerMode	How the ETL job will be triggered (started): • Trigger using CRON expression: Starts a job automatically, based on a CRON scheduling expression. • Trigger after another scheduled service: Starts a job automatically after another scheduled job has finished. • Trigger manually: Job can be started manually via iWD Manager Services Status screen.
cronExpression	Standard CRON scheduling expression when triggerMode is set

Property	Description
	to "Trigger using CRON expression". For example, the following expression will cause the job to be executed every 15 minutes: "0 0,15,30,45 * * * ?". See http://en.wikipedia.org/wiki/Cron for more information.
executionQueueName	Scheduled services that have the same executionQueueName will never be run in parallel within the same solution. If a scheduled service is triggered while another scheduled service that has the same executionQueueName is in progress, it will be queued until the other service has completed processing.
jobName	The name (type) of the ETL job. Please see the iWD Data Mart Reference Guide for detailed descriptions of the various types of iWD Data Mart ETL jobs.
etlService	*Mandatory dependency:* The kettle ETL Service

iWD Task Info Service

The iWD Task Info service is an optional iWD service that allows displaying of iWD task information within third-party applications, such as Genesys Agent Desktop.

The following properties are configurable for the iWD Task Info service:

Property	Description
startAutomatically	Whether the service should be started automatically after the configuration deployment.
logLevel	Service log level. Should be set to Default, unless otherwise specified by Genesys Support.
logAgeInDays	Sets the number of days that log files should be kept in the system. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logMaxFileSizeMegaBytes	Sets a limit on the size of a single log file, in megabytes. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
logFilesToKeep	Sets a limit on the number of log files that are kept for this service, excluding the current log file. A value of zero disables this limit. This property is not available if logLevel is set to Off or Default.
BrokerService	*Mandatory dependency:* Broker Service.

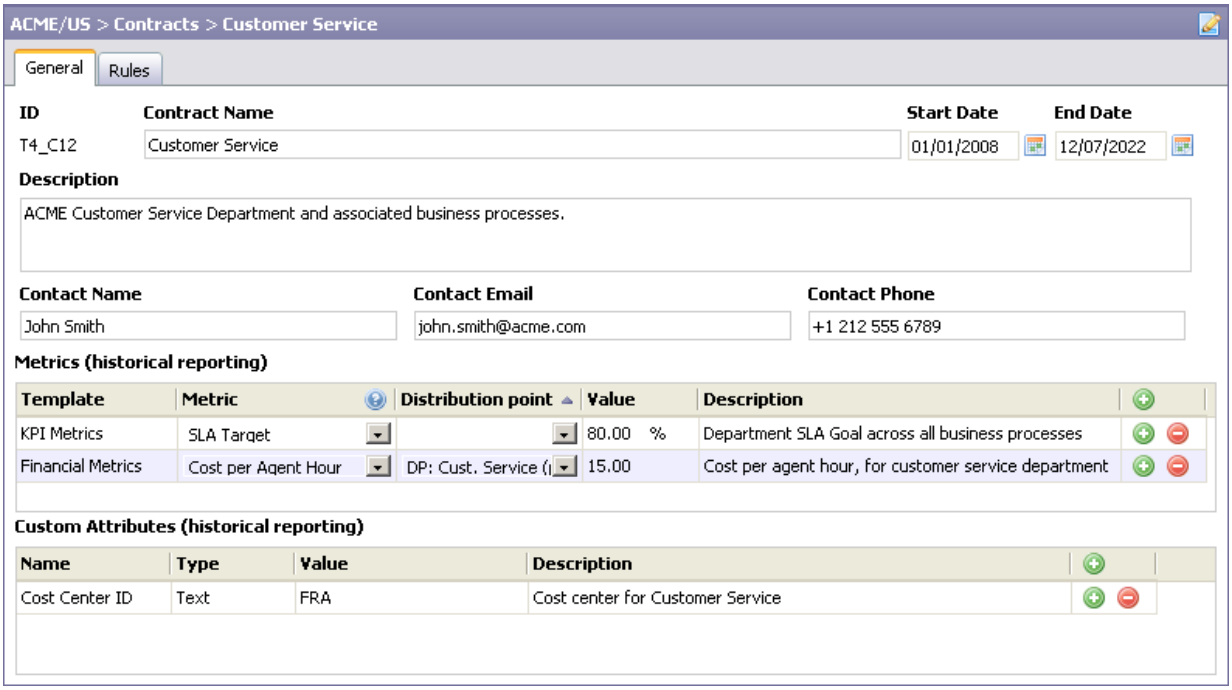
Contracts & Processes

The Contracts & Process configuration section allows configuration of iWD business logic.

- Use Contract Detailsview to create new contracts, and to view and modify general contract attributes and contract-level Metrics
- Use Process Details view to create new processes, and to view and modify general process attributes and process-level Metrics
- Use Rules view to create, view, and modify iWD business rules.
- Use Business Calendars view to create, view, and modify iWD business calendars.
- Use Deployment view to deploy solution configurations.

Contract Details

Contract Details view displays, and lets you modify, general contract attributes.



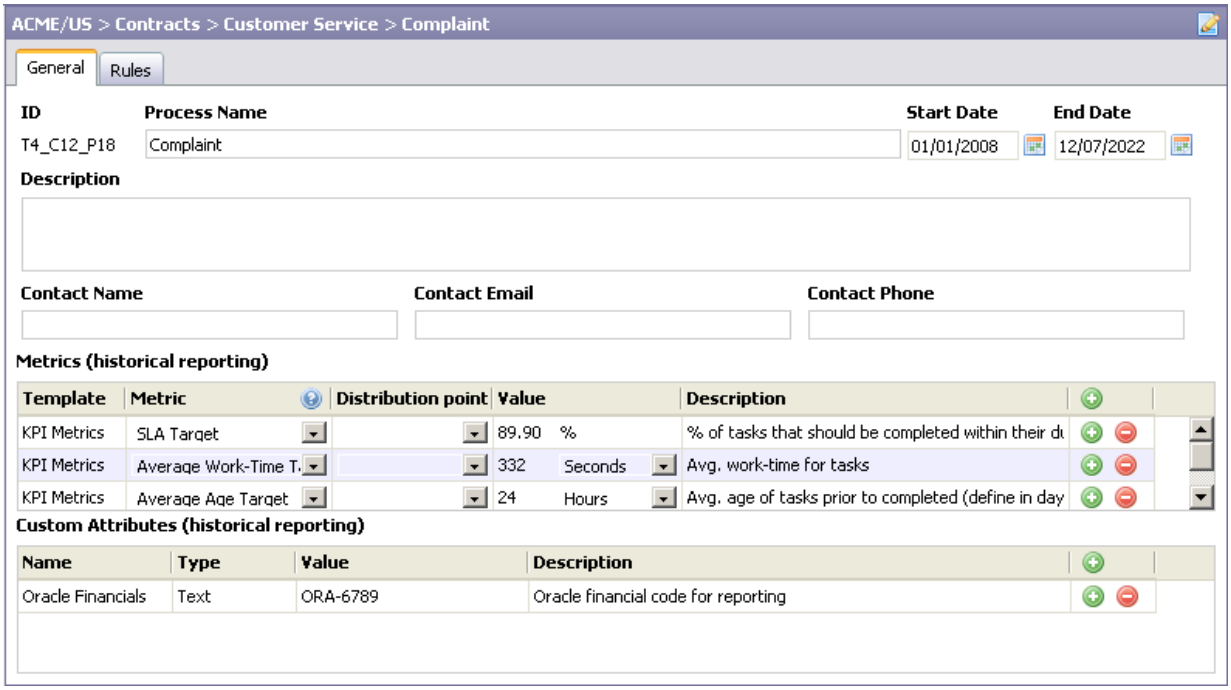
The following attributes and actions are available in Contract Details view:

Attribute/Action	Description
ID	The ID of the contract.
Contract Name	The name of the contract. This field is mandatory.
Start Date	The date on which the contract becomes active.

Attribute/Action	Description
End Date	The last day that the contract is active. If left empty, the period end date is unconstrained (that is, the contract will be active infinitely).
Description	A free-form description of the contract.
Contact Name, Phone and E-Mail	The contact information for the contract, for informational purposes.
Metrics	A set of user-defined metrics, for reporting purposes (described in Metrics).
Custom Attributes	A set of user-defined attributes, for informational and reporting purposes. Each attribute must have a name, type (either Text, Number, Currency, or Date), value, and optional description.
Save/Save & Close/Cancel/Delete	Standard iWD Manager functions as described in iWD Manager Overview
View Tasks	Opens a corresponding task view in iWD Manager.

Process Details

Process Details view displays, and lets you modify, general process attributes.



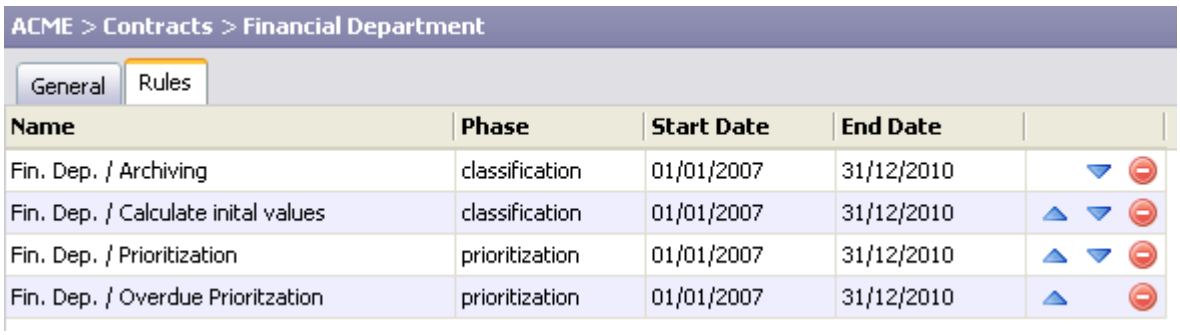
The following attributes and actions are available in Process Details view:

Attribute/Action	Description

Attribute/Action	Description
ID	The ID of the process.
Process Name	The name of the process. This field is mandatory.
Start Date	The date on which the process becomes active.
End Date	The last day that the process is active. If left empty, the period end date is unconstrained (that is, the process will be active infinitely).
Description	A free-form description of the process.
Contact Name, Phone and E-Mail	The contact information for the process, for informational purposes..
Metrics	A set of user-defined metrics, for reporting purposes. Described in Metrics
Custom Attributes	A set of user-defined attributes for informational and reporting purposes. Each attribute must have a name, a type (either Text, Number, Currency or Date), a value and an optional description.
Save/Save & Close/Cancel/Delete	Standard iWD Manager functions as described in iWD Manager Overview
View Tasks	Open a corresponding task view in iWD Manager.

Rules

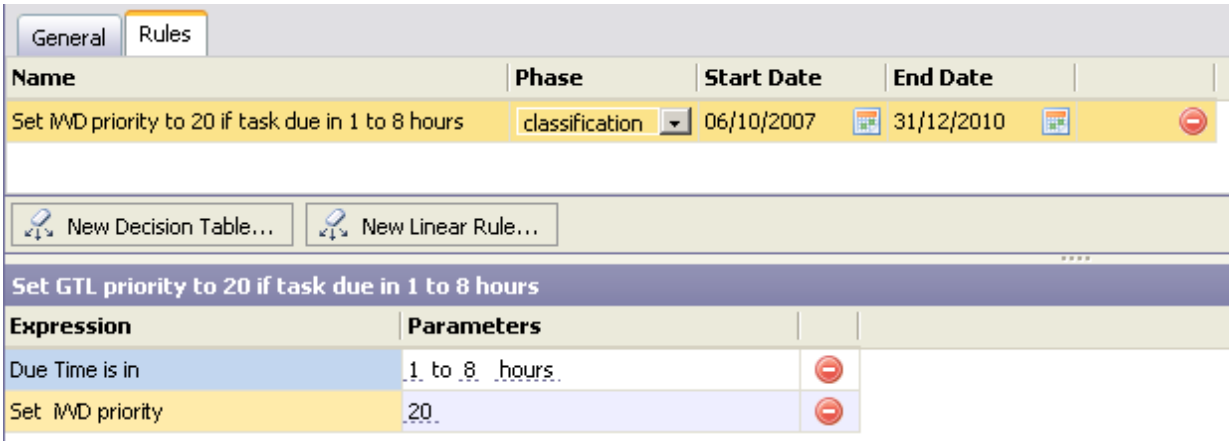
In iWD Manager, when a contract, process, or capture point is selected in the navigation tree, the rules for that particular context are shown on the right side of iWD Manager, under the Rules tab. Rules are presented in a list, and in exactly same order as they are applied to tasks. You can change the order of rules by clicking the Up and Down buttons. The rules are applied to tasks in the order specified in the table.



The logic of a particular rule can be expressed as either a linear rule or a decision table.

Linear Rule

A linear rule is a business rule that has a set of conditions (when) and actions (then), and is used for a simple (linear) business case. For example, when a task is due in 1 to 8 hours, set the task's priority to 20.



To specify a linear rule:

- In iWD Manager, in the Contracts & Processes navigation tree, select a contract, process, or capture point.
- On the right side of iWD Manager, in the Rules tab, click New Linear Rule.
- In the rule Name, enter a name that identifies the rule.
- Select the Phase in which to apply the rule. Depending on the current context (capture point, contract, or process) this can be either classification, prioritization, or archiving.
- If needed, set the Start Date and End Date.
  Leaving these empty means that the rule activation period is unconstrained.
- Select one or several conditions for the rule from the Add condition combo box.
  Conditions are logically concatenated by using the logical AND function.

Note: Available actions and conditions are defined in Rules templates. If a new action or condition type is required, add it to either an existing rules template or a new template.

- Select one or several actions for the rule from the Add action combo box.
- To save the specified rule, click either Save or Save & Close.

Decision Table

Decision tables have a set of the same conditions (when) and actions (then), but have different parameters and are used for a complex (structured) business case.

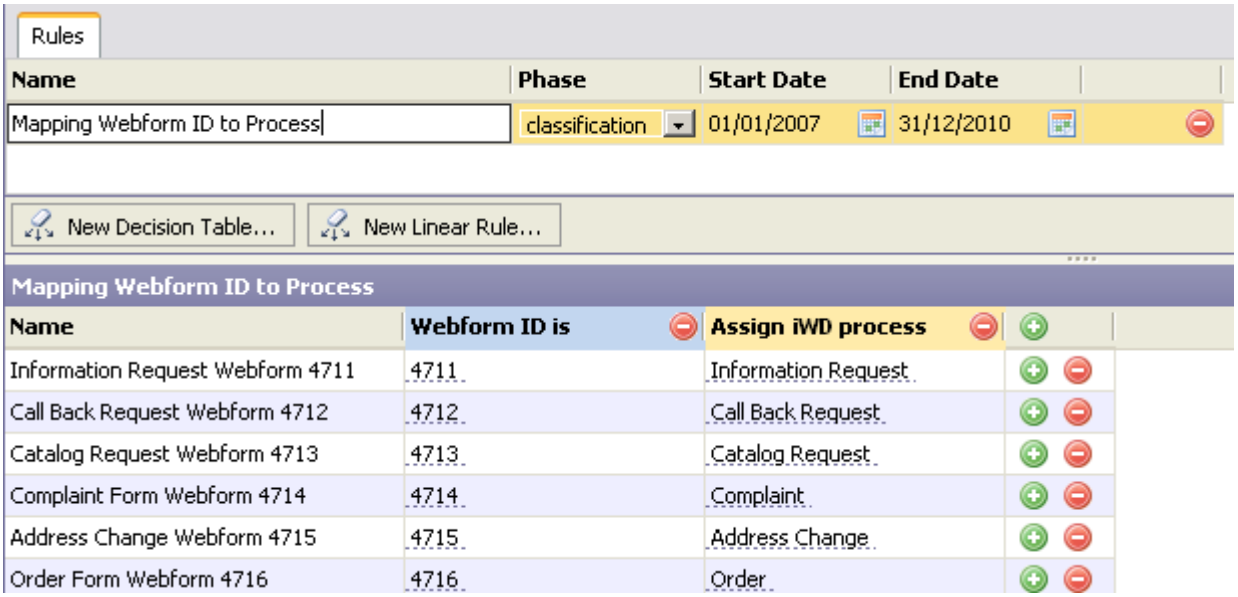
Use decision tables to avoid dozens of linear rules in the system.

Defining a decision table is similar to defining a linear rule.

To specify a decision table:

- In the Rules tab, click New Decision Table.

- Specify the rule Name, Phase, Start Date, and End Date, as required (in the same way as for a linear rule).
- Click Add Rule , and then enter the name of the rule.
- Select one or several conditions for the rule from the Add condition combo box. For each selected condition, set the parameter value.
- Select one or several actions for the rule from the Add action combo box. For each selected action, set the parameter value.
- To delete condition, action, or rule, click Delete , which is located on the right side of the column (for conditions and actions), or on the right side of the row (for rules).
- Repeat steps 3-5, until you have set all of the required cases. The result is a sort of table in which the columns represent rule conditions and actions and the rows contain real conditions and action parameter values, as shown in the following example:



The preceding example shows how to assign a iWD process to a task, based on the ID of the webform that was used to submit the task.

- To save the specified rule, click either Save or Save & Close.

Standard Rules Template

The standard Rules template defines the most commonly used rule conditions, actions, parameters, and functions.

Conditions

The following table describes conditions for the standard Rules template:

Condition	Explanation
Business value is *"from"* to *"to"*	When business value of the task is between *from* and *to*, then...

Condition	Explanation
Due Time is in *"from"* to *"to"* *"units"*	When the task due date/time is between *from* and *to* specified time *units*, then...
iWD Priority is below *"priority"*	When iWD priority of the task is below *value*, then...
No process selected	If process is not assigned to the task, then..
Process is *"process"*	If the task is assigned to *process*, then...
Task is overdue	When the task is overdue, then...

Actions

The following table describes standard rules template actions:

Action	Explanation
Activate task in *"amount"* *"units"*	Activate the task in the specified *amount* of time *units*
Activate task in *"amount"* working *"units"*	Activate the task in the specified *amount* of time *units* according to the task's business calendar (working days, working hours and so on). **Note:** Business calendar must be assigned to the task before this action can be used.
Assign business calendar *"business calendar"*	Assign the specified *business calendar* to the task.
Assign distribution point *"distribution point"*	Assign the specified *distribution point* to the task.
Assign iWD process *"process"*	Assign the task to the specified *process.*
Assign skill *"skill"*	Specify what *skill* is required to process this task.
Increase iWD priority *"amount"*	Increase the iWD priority of the task by the specified *amount.*
Multiply business value *"amount"*	Multiply the business value of the task by the specified *amount.*
Reprioritize after "*period*" "*periodType*"	Reapply prioritization rules to the task after the specified *amount* of time *units.*
Reprioritize after "*period*" working "*periodType*"	Reapply prioritization rules to the task after the specified amount of time units, according to the task's business calendar (working days, working hours, and so on). **Note:** Business calendar must be assigned to the task before this action can be used.

Action	Explanation
Set activation date from "*customAttribute*"	Set the activation date/time of the task from the specified *custom attribute* of the task.
Set activation time "*time*"	Set the *time* when the task will be activated.
Set business value "*businessValue*"	Set business value of the task to the specified *value*.
Set due date from "*customAttribute*"	Set the due date/time of the task from the specified *custom attribute* of the task.
Set due time "*time*"	Set the *time* when the task is due.
Set iWD priority "*priority*"	Set the iWD priority of the task to the specified *value*.
Task Due in "*period*" "*periodType*"	The task is due in, after the specified *amount* of time *units*.
Task Due in "*period*" working "*periodType*"	The task is due in, after the specified *amount* of time *units*, according to the task's business calendar (working days, working hours, and so on). **Note:** Business calendar must be assigned to the task, before this action can be used.

Parameters

Parameters in a Rules template describe the nature of the parameters that are used in rule conditions and actions. This includes how parameters are presented to the user (who is creating the rule), how input is validated, and so on.

The following table describes parameters for the standard Rules template:

Parameter	Description
distributionPoint	Presents a user with a list of distribution points that are defined in the solution. This list is dynamic; it changes as distribution points get added or removed.
periodType	Presents a user with predefined types of time periods, such as *days*, *hours* or *minutes*.
businessValue	Allows a user to input the numeric value that represents business value.
period	Allows a user to input the numeric value. Combined with the period type, it gives the actual value of the time period.
priority	Allows a user to input the numeric value that represents iWD priority.
process	Presents a user with a list of processes that are defined in the solution. This list is dynamic; it changes as processes get added or removed.
periodFrom	Allows a user to input the numeric value that represents the start point of a period, in time units, according to period type.
periodTo	Allows a user to input the numeric value that represents the end point of a

Parameter	Description
	period, in time units, according to period type.
customAttribute	Allows a user to input text that represents the name of a task custom attribute.
skill	Presents a user with a list of skills that are read from the Genesys configurationenvironment. This list is dynamic; it changes as skills get added or removed.
businessCalendar	Presents a user with a list of business calendars that are defined in the solution. This list is dynamic; it changes as business calendars get added or removed.

Functions

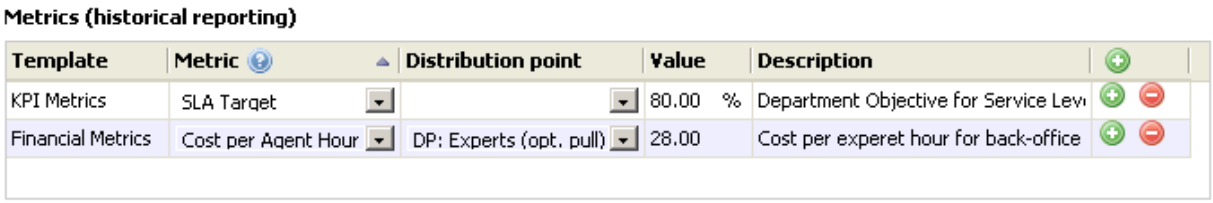
The following table describes functions for the standard Rules template:

Function	Description
getCurrentDT	Returns the current date and time, in UTC.
getPeriodDTFrom	Adds the specified amount of time units (minutes, hours, days) to the specified date/time value.
setContractAndProcess	Sets the contract and process of the given task from a given string, in*contract*|*process* format.
isProcess	Is true if a given task is assigned to a given *process*, false otherwise.
setTime	Sets the time portion of a given date/time value from a string value, in *hh:mm* format.
getDueDT	Returns the task due date or throws an exception if a due date is not set.

Metrics

A key component of dashboards and reports is the comparison of actual metrics against target goals. Understanding the effectiveness or efficiency of organizations is measuring performance against important goals that have been set by the organization.

Targets can be associated with a number of objects in iWD Manager, such as distribution point, process, contract, or event tenant. For example, a cost-per-task target might differ between a distribution point (center) in Eastern Europe and one in Western Europe. A work-time goal for a task will differ, based on its process; for example, orders will take longer than address changes.



Metrics can be configured at the business-user level for contracts and processes using metric types that are defined in available Metrics templates. When a metrics value is set, it will be stored as a named attribute in the runtime and Data Mart. When a metrics value is set, it will be stored as a named attribute in Data Mart. If the value is changed, the updates are pushed through

to Data Mart with a valid_from and valid_to date/time stamp. This is important for historical reporting; example, if you update the target on November 1 from 2.5 to 3.5, all tasks up to November 1 will use 2.5, and all new tasks will use 3.5.

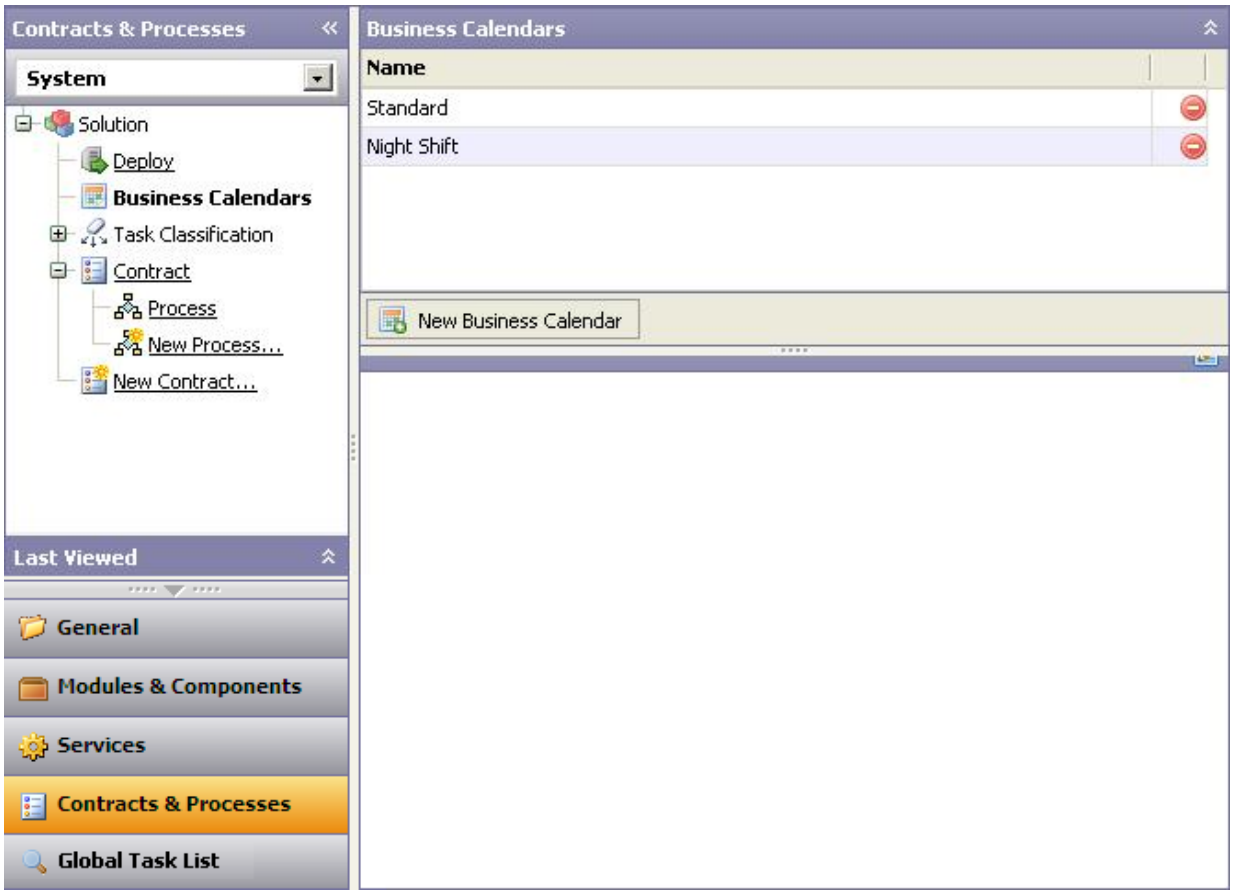
If the value is set at a contract level, it applies to all processes, unless there is a specific value for that process. For example, Contract 1 has four processes: A, B, C, and D. Cost/Task @ Contract 1 = 2.50, which applies to Processes B, C, and D. Cost/Task @ Process A = 1.50, which applies only to Process A.

Business Calendars

A business calendar is a set of rules that define working days and hours, as well as holidays,that are applicable for the business. Business calendars can be used in iWD rules to perform date and time calculations, taking into account the working schedule of the business. Business calendars are defined on a solution level and can be used in rules only for that particular solution.

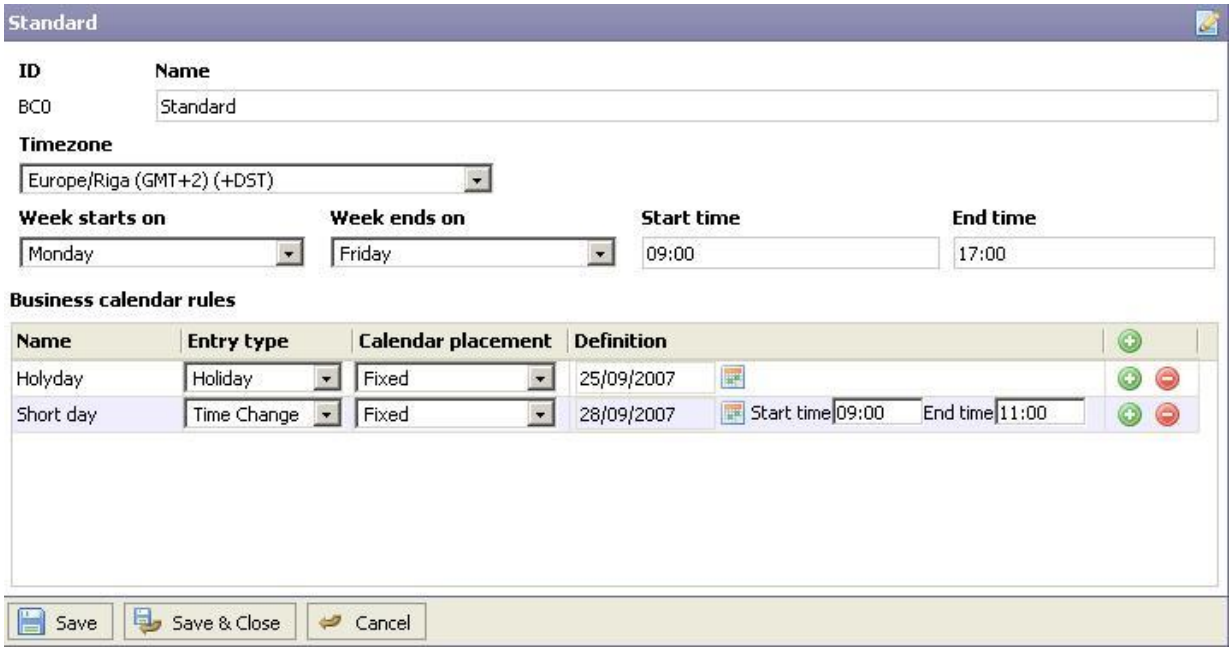
Managing business calendars in iWD Manager

To create new (or to manage existing) business calendars, open Contracts & Processes view in iWD Manager, and select the Business Calendars entry in the tree on the left-hand side of iWD Manager.



List of solution's business calendars

The right-hand side of iWD Manager is split horizontally into two panes. The upper displays a list of a solution's business calendars. The New Business Calendar button that is underneath this list is used to create a new business calendar. You can delete business calendars by clicking the button that is on the right-hand side of the business calendar in the business-calendar list. When a calendar in the business- calendar list is selected, the lower pane displays the attributes of the selected business calendar.



Attributes of a business calendar

Business calendars consist of a set of standard mandatory attributes, as well as optional businesscalendar rules. The following table describes the standard attributes of a business calendar:

Attribute name	Description
ID	The business-calendar identifier. This attribute is generated automatically by iWD Manager when the calendar is created.
Name	The name of the business calendar as it will appear throughout iWD Manager.
Timezone	The time zone of the business calendar.
Week starts on	The first day of a working week.
Week ends on	The last day of a working week.
Start time	The start time of a working day. This attribute is a representation of time in 24-hour format (*hh:mm*, where *hh* are hours and *mm* are minutes).
End time	The end time of a working day. This attribute is a representation of time in 24-hour format (*hh:mm*, where *hh* are hours and *mm* are minutes).

Business-calendar rules describe exceptions to the regular working schedule that is defined by business-calendar standard attributes. It is possible to add a new business-calendar rule by clicking on button that is found in the list of business-calendar rules. It is possible to delete a business-calendar rule by clicking the button that is located on the right side of the business-calendar rule in the business- calendar rule list. Depending on calendar placement and entry type, definition of a business-calendar rule can have different attributes. The following table describes the attributes of a business-calendar rule:

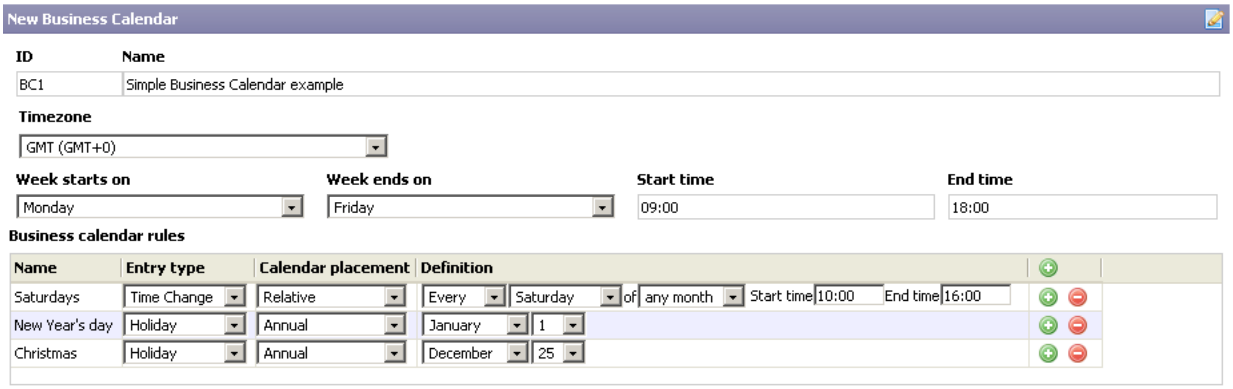
Attribute name	Description
Name	The name of the business-calendar rule.
Entry type	The type of business-calendar rule. This can be one of the following: **Holiday:** The business-calendar rule describes a holiday. **Time Change:** The business-calendar rule describes working hours that differ from the working hours of the business calendar. *Note:* A business calendar's holidays precede time changes; therefore, if any rule sets a day as a holiday, and there is a time change for that specific day, the time change will have no effect. This applies only to holidays that are specified by the rules of the business calendar.
Calendar placement	The calendar placement of the business-calendar rule. This can be one of the following: **Fixed:** The business-calendar rule is tied to a specific date. **Relative**: The business-calendar rule is described in a relative way and is not tied to a specific date; for example, the first Monday of September would be a relative description. **Annual**: The business-calendar rule that happens every year on a specific date.
Definition	The contents of this attribute depend on the selected calendar placement and entry type: **Date input control** This parameter applies only to business-calendar rules that have a fixed calendar placement. The year portion of this date is not taken into account, when performing calculations; thus, making this date recurrent on yearly basis. **Relative calendar placement controls** This set of controls applies only to business-calendar rules that have a relative calendar placement. There are three dropdown-list controls that allow specification of Place in month, Day of week, and Month of year. Place in month is defined by a value of Every, First, Second, Third, or Fourth, and is used in conjunction with Day of week (Monday, Tuesday, and so on) to describe relatively the day of month. Month of year defines to what month a business-calendar applies; besides month names, it allows also specification of "any month," if the business-calendar rule must be recurrent on a monthly basis. The following are examples of definitions of a relative calendar placement: [Second] [Tuesday] of [September]

Attribute name	Description
	[Every][Saturday] of [May] [First][Monday] of [any month] **Annual calendar placement controls** This set of controls applies only to business-calendar rules that have an anual calendar placement. There are two dropdown-list controls that allow specification of month and date. **Working hours definition controls** This set of controls applies only to business-calendar rules that have a time-change type. Two input controls allow the entering of a start time and end time of the working day. Each value is a representation of time in 24-hour format (*hh:mm*, where *hh* are hours and *mm* are minutes).

Simple example

To configure a business calendar in which Mon-Fri are standard working days 09:00 to 18:00, Sat 10:00 to 16:00, the user would proceed with the steps that are shown in the following image:

- Create a new calendar, select Timezone, and set Week starts on Monday and Week ends on Friday. Then, set Start time and End time.
- Add a time-change rule for Saturdays (it will mark it as working day).
- Also, some holidays can be specified with rules, such as Christmas or New Year's Day.



Using business calendars in iWD rules

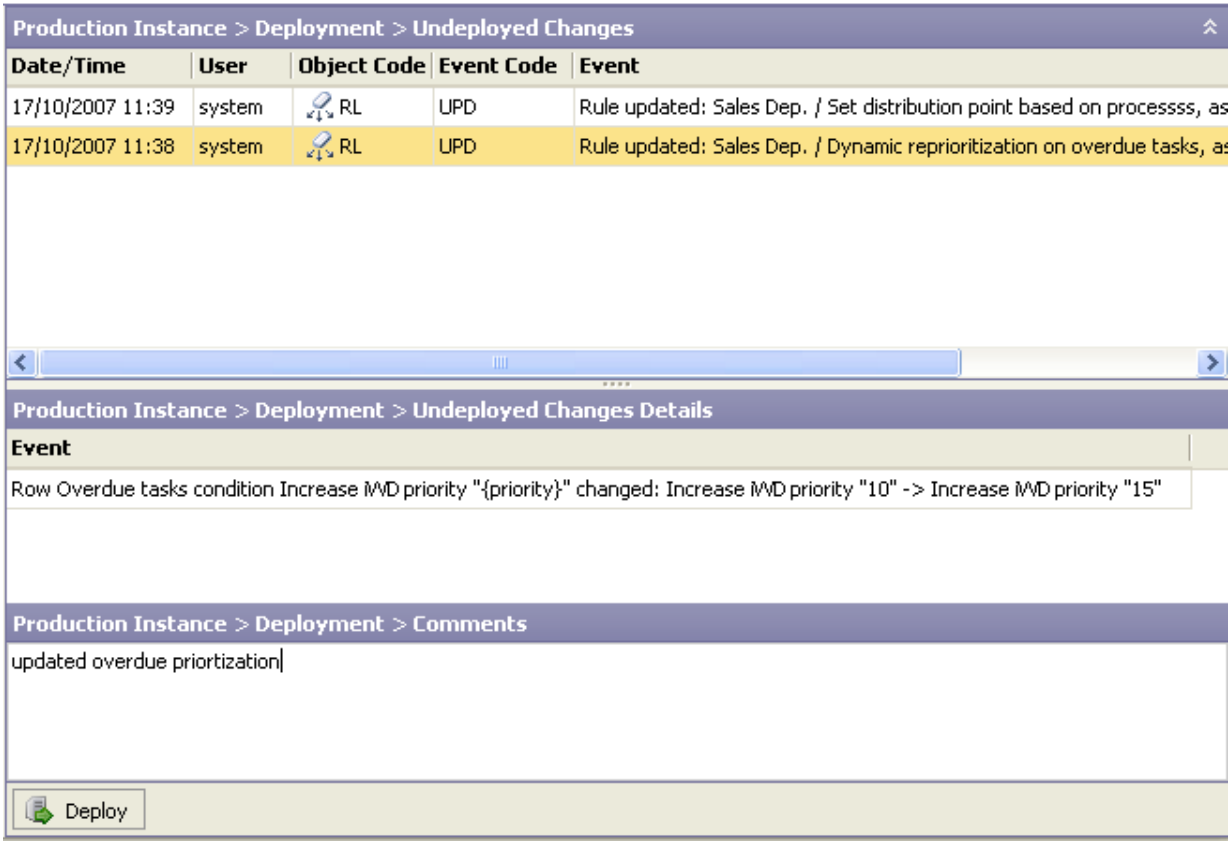
After business calendars are defined, it is possible to use them in GTL rules. Business calendars should be assigned to a task before any business-calendar-related calculations can be performed on task values. The following business-calendar-related actions are available in GTL rules:

Action name	Parameters	Description
Select business calendar	{businessCalendar}	Assigns a business calendar to a task. A business calendar must be assigned to a task, before any business-calendar-related

Action name	Parameters	Description
		calculations can be performed on task values. A dropdown list displays a list of business calendars that are defined for the solution.
Reprioritize after	{period} in working {periodType}	Sets a task's re-prioritization date/time to value that is calculated, based on current date/time, the task's business calendar, and specified parameters. {period} is a numeric value, and {periodType} specifies "working" minutes, hours, or days.
Task Due in	{period} in working {periodType}	Sets a task's due date/time to a value that is calculated, based on the task's creation date/time, the task's business calendar, and specified parameters. {period} is a numeric value, and {periodType} specifies "working" minutes, hours, or days.
Activate task in	{period} in working {periodType}	Sets a task's re-prioritization date/time to a value that is calculated, based on task's creation date/time, the task's business calendar, and specified parameters. {period} is a numeric value, and {periodType} specifies "working" minutes, hours, or days.

Deployment

Any changes to a configuration in iWD Manager are not activated automatically. Deployment view performs this task by distributing contract, process, and rule configuration to iWD services. This functionality is similar to Services deployment, with one difference: Only the business-logic configuration (contract, process, rules, and business calendars) is deployed.



The following attributes and actions are available in Deployment view:

Attribute/Action	Description
Undeployed Changes/ Undeployed Changes Details	Detailed information about activities, performed in iWD Manager, since the last deployment (for more details, see History).
Comments	Deployment comments; will be displayed as version comments in Change History.
Deploy	Deploys the configuration. Depending on the configuration's complexity and runtime-environment characteristics, this action can take several seconds.

Solution User Access

Contracts and processes support instance-level security. The User Access page under Solution allows the assignation of configured User roles to specific contracts or processes. Any roles that have been specified on a contract level are automatically inherited by all child processes. Process roles are appended to inherited ones.

User access can be copied to another solution, if there are contracts or processes that have matching IDs.