# iWD Deployment Guide

intelligent Workload Distribution 9.0.0

9/1/2024

# Table of Contents

# intelligent Workload Distribution 9.0 Deployment Guide

Welcome to the *intelligent Workload Distribution 9.0 Deployment Guide*. This document describes how to install and configure intelligent Workload Distribution (iWD).This document is valid for 9.0.x releases of this product.

This document is intended for:

- IT staff responsible for iWD installation and configuration.
- Business analysts responsible for Department and Process configuration.

It assumes that you understand:

- The workflow concepts implemented in the various enterprise source systems (for example, business process management (BPM) systems, host systems, CRM systems, and so on) from which the iWD solution captures tasks.
- Network design and operation.
- Your own network configurations.

Interaction Server Configuration

Installing iWD GAX Plug-in

Uploading IPs Using GAX

Creating the Tenant in GAX

Working with Integrated Capture Points

Manual Setup of the IWDBPs

## Installation 3

Logging

Preparing iWD for use with Social Engagement

Installing/Removing Language Packs

## Configuration

iWD Manager

iWD History Node

Roles and Privileges in GAX

iWD Business Solution

Task Attributes and Interaction Properties

iWD Services

## iWD Web Installation and Configuration

Installing iWD Web for Windows Manually

Installing iWD Web on Linux

Create/Configure iWD Web in GAX

Configure Web Service Capture Point

Configuring iWD Web Services

## Further Configuration

Customer and Partner Translation

High Availability

## Troubleshooting

Troubleshooting

Support for RHEL 64-bit Platforms

## Best Practice Overview

Design Phase

Task Capture

Blocked Lookup Tables

Business Rules Configuration

Configuration

Task Distribution and Routing

Reporting

# New Features in Release 9.0

> **Important**
> In release 9, you must upgrade all iWD components to the same release version (9.0.xxx must be the same for all components) to guarantee correct functioning.

## 9.0.019

- Starting with 9.0.019.06, Genesys recommends using Kafka instead of JMS in the iWD History Node setup because of a more straightforward configuration process and a more reliable delivery of events in the correct sequence when using multiple iWD History Nodes.
- Various third-party libraries are updated to the latest versions to improve security.

### iWD Runtime Node

- The calculation logic for the **Accept Time** metric and all related database columns has been updated. Now if a task is accepted multiple times, this metric corresponds to the first assigned time instead of the last assigned time.
  The following iWD DataMart database fields now correspond to the first assigned time instead of the last assigned time:
  - TASK_FACT Tables
    - ASSIGNED_DATE_KEY
    - ASSIGNED_TIME_KEY
    - ASSIGNED_INTERVAL
    - ASSIGN_TIME_FROM_CREATED_SEC
  - TASK_CAPT_FACT Aggregate
    - CMPL_TASK_ASSIGN_TIME
    - CMPL_TASK_AVG_ASSIGN_TIME
  - TASK_CLASSIF_FACT Aggregate
    - CMPL_TASK_ASSIGN_TIME
    - CMPL_TASK_AVG_ASSIGN_TIME
  - TASK_QUEUE_FACT Aggregate
    - CMPL_TASK_ASSIGN_TIME
    - CMPL_TASK_AVG_ASSIGN_TIME

This change also impacts the following GCXI iWD Metrics and related reports:

- Accept Time
- Avg Accept Time

## 9.0.018

- Various third-party libraries are updated to the latest versions to improve security.

### iWD Runtime Node

- Performance of the standard Aggregate Historical plugins is improved. Most of the aggregation is now performed on the database server side. As a result, the amount of data transferred from database to the application and the amount of memory consumed by the application are reduced by up to 80 percent.

## 9.0.017

- Various third-party libraries are updated to the latest versions to improve security.

### iWD Runtime Node

- A new job, **Load GIM**, is introduced. The job uses the Genesys Info Mart (GIM) database to load agents' information into iWD Data Mart database. Once agents' information is loaded, the **Agent Name** column is shown (instead of **Employee ID**) in the following reports:
  - Resource Performance Report
  - Task Detail Report
  - Task Work Detail Report
  - Resource Performance Dashboard

### Platform Changes

- Support for RedHat OpenJDK 11.

## 9.0.016

- **oAuth changes**
  - The parameter **[oauth/]redirect-uri** has been removed from iWD Manager Server application options.

- **SAML-related changes**
  - The SAML-related properties **logged_with_saml** and **slo_enabled** have been removed from the **/user_settings** endpoint. The **/api/security/saml_enabled** endpoint has also been removed. DETAILS

## 9.0.015

### iWD Data Mart

- Support for new dimensions, such as multimedia type and interaction type, has been added.
  - The core fact tables (TASK_WORK_FACT, TASK_FACT) have been expanded. They now allow you to describe facts with media type and interaction type.
  - Aggregate tables have been extended with new attributes:
    - MEDIA_TYPE_KEY for Age, Agent, Capt, Classif aggregate plugins.
    - INTERACTION_TYPE_KEY for Agent aggregate plugins.

    Starting with this release, Genesys strongly recommends using native plugins provided with iWD. Kettle aggregation plugins do NOT support dimensions added in this version. Use of custom Kettle plugins is fully supported.

### iWD Manager

- iWD Manager now supports OAuth user authentication with GWS Authentication Service. MORE INFORMATION.

## 9.0.014

### iWD Data Mart

- The iWD Data Mart database schema has been changed. Please follow the detailed migration information in the migration section of the Deployment Guide.
  - Aggregate Intraday tables (**I_TASK_<SUBJ>_FACT_15MIN**) have been removed from the iWD Data Mart schema. From now on the Aggregate Intraday job directly populates **H_TASK_<SUBJ>_FACT_15MIN** tables. There is no need to move data from Aggregate Intraday to Aggregate Historical tables any more.
  - Aggregate Historical tables (**H_TASK_<SUBJ>_FACT_15MIN**) now contain the data for every aggregated 15-minute interval, which expedites data fetching for reporting purposes.
  - Population of **H_TASK_<SUBJ>_FACT_DAY** table is now triggered by the fact that the corresponding **H_TASK_<SUBJ>_FACT_15MIN** table contains a full set of 15-minute intervals for that particular day.
- Performance of CAPT, CLASSIF and QUEUE aggregation plugins has been improved by removing

unnecessary table joins that were previously used in the calculation of the "number of pending overdue tasks" column (TOTAL_OVERDUE_TASK_COUNT).

- CAPT and QUEUE aggregation plugin logic has been enhanced to make segmentation/aggregation using just one fifth of the BUSINESS_VALUE and PRIORITY dimensions' FK values without affecting current reports. This reduces both iWD Data Mart storage usage and ETL calculation time.

### iWD History Node

- The behavior of configuration options stop-processing-delay and stop-processing-interval has been corrected, and documentation has been updated.

### iWD REST API

- **REST API changes**—Logout via the GET method is no longer supported. From release 9.0.014, only the POST method is now supported. See the *iWD REST API Reference Guide* for details.

## 9.0.013

### iWD Manager

- **iWD Query Language improvements**—You can now can save and edit iWD QL queries and make them public to share with other users. More information
- **Filter and View changes**:
  - iWD Manager now remembers the latest GTL view settings and restores them on returning from another page.
  - iWD Manager now loads lists of possible values for Disposition Codes and Agent Groups to be used in Filters, iWD Query Language (iWD QL) queries and the Custom Filter Query Wizard.
- **UI improvements**—Several UI improvements and fixes have been made, including:
  - The iWD QL autosuggest panel can now be hidden by pressing the Escape button on the keyboard.
  - When the mouse pointer is over the iWD QL status icon, a detailed status message is shown.
- **REST API changes**—Changes to the iWD Manager REST API are described in the following topics:
  - In the Fetch paginated list of tasks query the request format was changed from GET to POST.
- HTTP Status Code 302 Found is replaced with 307 Temporary Redirect.
- The Request Body format for the following queries has been changed:
  - Bulk operations.
  - Get common task attributes for modification.
  - Export selected tasks.

- **Standard Rules Template changes**—iWD Standard Rules Templates are now supplied in both the Eclipse project format and XML format for compatibility with GRAT 9.0.

### iWD Runtime Node

- **Migration changes**—Improved and extended Data Migration from iWD 8.5.1 to iWD 9.0 by adding the ability to migrate data from the EventLog database to iWD Data Mart. More information

### iWD History Node

- **Performance**—iWD History Node processing performance has been improved by up to more than two times in synthetic data scenarios. Event Logger DAP templates with the recommended settings for this improvement are now provided with the iWD History Node IP.
- **Migration**—Improved and extended Data Migration from iWD 8.5.1 to iWD 9.0:
  - Added ability to migrate data from the EventLog database to iWD Data Mart.
  - Re-worked the Migrating GTL History Events procedure.
  - Added a new option taskEventLimit to protect against tasks with a large amount of events.
  - The fetchSize option is deprecated.
  - More information
- **Event Logging**—Configuration option **event-logger-mode** is deprecated. Starting from this version:
  - History Node determines the Event Logger type automatically based on the DAP connection provided.
  - There must be one and only one Event Logger DAP connected to History Node.

## 9.0.012

- **Kafka Support**
  - iWD now fully supports the Interaction Server Kafka Capture Point.
  - iWD now fully supports the Kafka Binary Event Logger provided by Interaction Server. Please see the following documents for more details:
    - How to configure the Kafka Binary Event Logger
    - How to configure History Node

    Starting with this release, you no longer have to have JMS as transport between Interaction Server and History node—you can have Kafka-only deployments of iWD.

    You can still use JMS Capture Point and/or JMS as transport between Interaction Server and History Node.
- **iWD Manager**
  - iWD Manager now loads lists of possible values for Queue Type and Media Type to be used in Filters, iWD Query Language (iWD QL) queries and the Custom Filter Query Wizard.

    - Significant improvements for iWD QL have been introduced:
        - iWD QL now suggests possible values of attribute names, attribute values and operators.
        - iWD QL now supports the IN <LIST> operator.

          Please see the documentation for details.
    - iWD Query Language is now fully localized.
- **iWD Runtime Node**
    - In views that are based on the TASK_AGENT_FACT table, measures are now summed up from both intraday and historical facts.
- **Support for Genesys Rules System 9.0**
    - iWD now fully supports GRS 9.0.

### Platform Changes

- Support for the following databases is implemented:
    - Oracle 19c RAC
    - Oracle 18c RAC
- Support for Red Hat Enterprise Linux 6 is discontinued.

## 9.0.011

- **iWD GAX Plug-In**
    - **Compatibility with GAX**
      iWD Plug-in for GAX is now fully compatible with Genesys Administrator Extension (GAX) core version 9.0.100.52.
- **iWD Manager**
    - **iWD Query Language**
      Support for query capabilities in iWD Manager enabling users to perform SQL-like queries on the Global Task List.
    - **ORS Business Process improvements**
      Error handling: fixed several cases where error messages were not provided.
- **iWD Data Mart**
    - **Performance and audit improvements**
        - Significantly increased performance of the Load Intraday job: up to 4 times in certain synthetic data cases (real-life performance improvement may be lower).
        - The ETL Audit table is expanded with the following new columns — `ETL_AUDIT_START_TIME`, `ETL_AUDIT_FINISH_TIME`, `DURATION`, `STATUS`. For more information, see the iWD Data Mart Reference Guide.

## Platform Changes

- Support for CentOS Linux 7 is implemented.
- Support for Red Hat Enterprise Linux 8 is implemented.
- Support for Windows Server 2019 is implemented.
- Support for MSSQL Server 2019 is implemented.
- Support for PostgreSQL 10 is implemented.

# 9.0.010

- **iWD Manager**
  - **Support for custom attribute types**
    You can now configure custom attributes types in Genesys configuration layer. iWD Manager uses these types for validation in filter criteria and for proper display of task details. Configuration details are here. Please perform the migration as described here.
  - **ORS Business Process improvements**
    - The maximum number of retries to connect GRE/UCS is now restricted for the InvokeGRE and InvokeUCS strategies in order to prevent an infinite loop if GRE/UCS is down.
    - The iWD Business Process now properly handles ORS timeouts and GRE/UCS unavailability.
  - You can now create a bookmark for any page in the Global Task List (for example, for a custom filter). If you navigate to this bookmark after being logged out, you are routed to login and then redirected to the desired URL after authentication. In previous versions, bookmarked pages could be used only after logging in explicitly.
- **iWD Data Mart reporting enhancements**
  - **Default Values**
    iWD Data Mart now calculates and aggregates tasks which are not yet classified or prioritized at the moment of aggregation. The default value `Unclassified` is used to populate undefined values in the following dimensions:
    - DEPARTMENT
    - PROCESS
    - PRIORITY
    - BUSINESS VALUE

    Undefined values in other dimensions are now set as `Unknown`.
  - **New configuration option**
    A new configuration option—**aggregation-delay-interval**—is introduced. You can now specify a number of minutes (in 15-minute increments—15, 30, 45, and so on) to be waited before aggregation. Setting this parameter helps to avoid getting default values (`Unclassified`, `Unknown`) in reports. Set the value to a sufficient amount of time to ensure pre-routing activity is already completed and all tasks' properties are populated.
  - **Pending Metrics**

The calculation logic for some rare cases has been fixed. The following measures are affected:

- TOTAL_PENDING_TASK_COUNT
- TOTAL_OVERDUE_TASK_COUNT

These metrics belong to the aggregates:

- TASK_CAPT_FACT
- TASK_CLASSIF_FACT
- TASK_QUEUE_FACT

- **Real-time reporting**
  Real-time reporting metrics are now calculated over facts, not over aggregates. This means that real-time reporting will receive the latest processed data, even if an aggregation delay is set to more than `0` (zero) or if the aggregation process is delayed for some reason.

- Support for the following databases. See the IWD section in the *Supported Operating Environment Reference Guide* for more detailed information and a list of all supported databases.
  - Oracle 12c R2
  - Oracle 12c R2 RAC
  - Microsoft SQL Server 2017

## 9.0.009

- Support for OpenJDK 11.
- iWD Manager:
  - Provision of a default filter to be applied when users open the Global Task List.
  - New **Create Copy** button for copying filters.
  - A Queue selector for filter criteria now shows all iWD queues for the selected tenant.
  - A mandatory `tenantId` parameter has been added to the FetchFilterCriteriaTemplates method.
  - Improvements to display of non-English characters.
- iWD History Node:
  - Additional JMS parameter support.
  - Support for the IBM MQ JMS provider.
- iWD Data Mart:
  - Support for native plugins to improve performance.
  - Improvements to the KETTLE version of the AGE plugin.
- Improvements to the iWD Business Process for Composer/ORS.

## 9.0.008

- Support for Genesys CX Insights (GCXI) has been introduced.
- The Statistics Adapter job can now automatically create Virtual Queues for real-time reporting. MORE DETAILS
  New Pulse widgets are provided within the iWD Runtime Node IP. (IWD-8105) MORE DETAILS
- iWD Manager filters now support "is like" and "value in list" criteria. MORE DETAILS
- Logging configuration been unified for all iWD components. LMS files now contain full log messages, so Message Server alarms can be set up in the same way as other Genesys products. MORE DETAILS
- Support for OpenJDK 8 is introduced.

## 9.0.007

- **9.0.007.07**:
  - iWD Manager, iWD Plug-in for GAX, and iWD Web are now localized into the following languages: Chinese (Simplified), French (France), German (Germany), Portuguese (Brazilian), Russian, and Japanese.
  - The iWD Data Mart database schema has been changed. See migration information for more details.
- **9.0.007.05**:
  - iWD Manager and iWD Web are now localized into Spanish (International). iWD Plug-in for GAX is now localized into Spanish (Latin America) and Spanish (Mexico).
  - The Statistics Adapter job can now write settings into multiple Stat Server primary/backup pairs. Previously, it could only write to one such pair.
  - JDBC Driver Class, JDBC Driver JAR File and JDBC URL can be explicitly configured for iWD Stat Extensions, using iWD GAX Plug-in. These configuration options are optional. By default iWD Stat Server Extensions will use values from iWD Data Mart and default values. MORE DETAILS.

## 9.0.006

- iWD Manager and iWD Web now fully support Local Control Agent (LCA).

## 9.0.005

- **Changes to iWD History Node**
  - iWD History Node now fully supports Local Control Agent (LCA). iWD History Node's Application type has been changed from type `Third Party Server` to type `Genesys Generic Server` to support this change.

    - A new configuration option—**centralized-logging**—now enables configurable centralized logging to Message Server.
- **iWD Data Mart**—IWD Data Mart now supports centralized logging. A checkbox that enables logging has been added to the Data Mart logging settings in the GAX plugin.
- **Changes to Application types**—The Application types of iWD Manager Server, iWD History Node and iWD Web have been changed from type `Third Party Server` to type `Genesys Generic Server`, in order to support the implementation of Local Control Agent (LCA) functionality for iWD History Node.
- **Extension of TLS Support**—Connections from iWD Data Mart and iWD History Node to Message Server now support TLS. Please read this topic.

## 9.0.004

- Support for Transport Layer Security (TLS) 1.2 is implemented. Please refer to the new Configuring TLS for iWD guide.
- iWD Manager configuration—iWD Manager configuration has been moved from the **iwd.properties** file to the iWD Manager Application object in Configuration Server. Please refer to iWD manager configuration options.
- A **Pause** block with a configurable delay has been added to the InvokeGRE workflow of the IWD Business Process for Composer/ORS, in order to guarantee that interaction updates will be received. For more information, see IWDBP Strategies & Subroutines. (IWD-7465)

## 9.0.003

### Platforms/Tools Changes

- Support for Microsoft SQL Server 2016 is implemented.
- Support for MS SQL 2016 Cluster & AlwaysOn is implemented.
- Support for Windows Server 2016 is implemented.
- Support for Jetty 9.4 is implemented.

> **Important**
> Please see the release note Known Issues topic.

# 9.0.002

## iWD History Node

iWD History Node is an application that filters the flow of data from the Interaction Server, so that only iWD-relevant data is made available to iWD, thereby boosting performance.

### Background

In iWD deployments with a large number of interactions, very high volumes of events can be stored in the Event Log database. The size of the Event Log database can affect performance for both iWD and Interaction Server. Besides using database tables used by iWD, Interaction Server can provide Event Log data via the Java Messaging Service (JMS) queue. The new iWD History Node application utilizes this mechanism to filter uninteresting records before they enter the database. It also splits the source data stream into separate tables, one for Data Mart and one for the Global Task List. Thus each table can be maintained separately, reducing the complexity of the Prune job.

History Nodes can also be configured in clusters.

### New Process Summary

1. Interaction Server sends reporting events to the JMS queue instead of storing them in the Event Log Database.
2. History Node reads events from the JMS queue, filters them and extracts important information. The JMS queue remains almost empty the whole time.
3. Only events needed by iWD are processed and saved in History Node's database.
4. Only relevant information from the events is stored in History Node's database.
5. Events are stored in two tables in History Node's database, one for use by Data Mart, one for use by GTL.
6. Events can be queried by iWD Manager via the REST API.
7. GTL events are removed from History Node's database when the interaction is completed.
8. Events can be queried by iWD Data Mart via the REST API.
9. Data Mart events are removed from History Node's database immediately after they are processed by Data Mart.

### More Information

- iWD Architecture diagram
- Installing History Node
- Configuring History Node
- History Node Configuration Options
- History Node Migration
- History Node Limitations

- iWD Reporting architecture

## REST API for iWD Manager

- The REST API for iWD Manager is now available to customers who want to develop custom solutions.

## iWD Manager Updates

- The user interface of iWD Manager has been updated to bring it into line with other Genesys applications and components. As part of this update, the Custom Interaction Properties functionality has been moved into the iWD Plug-in for GAX. iWD Manager Help and iWD Plug-in for GAX Help have also been updated accordingly.
- iWD Manager now runs as a standalone Java application without needing any external servers.

### Removal of IWD Manager Configuration Database

- iWD Manager no longer has a separate database for configuration objects. Configuration of iWD Manager filters, icons and user settings has been moved to Configuration Server.

## Support for Interaction Server Cluster via Proxy

Before release 9.0, iWD supported multi-Interaction-Server deployments with tasks segmented into logical entities called Solutions. In this architecture, each Solution is handled by a single Interaction Server (possibly in primary-backup configuration) and each Interaction Server can handle multiple Solutions. iWD 9.0 expands this capability with an architecture where a single Solution can be handled by multiple Interaction Servers via an Interaction Server Proxy.

### More Information

- Installing Interaction Server
- Business Structure (GAX Plug-in Help)

## Support for Automated Import of XLS/CSV Files in iWD Web

In release 9.0, you can configure automatic import of .XLS and .CSV files with column headings, based on a time interval of the customer's choosing. Source systems export tasks/work items to a safe location that iWD Web can access. iWD Web checks for the presence of a new file in this location, checks the file format, then imports the individual tasks to iWD / Interaction Server for insertion into the Global Task List. Automatically uploaded jobs can be viewed (alongside any manual uploads) in the **Jobs List** tab of iWD Web.

To use this feature, you must create and configure a special iWD Web capture point, similar to a File capture point.

> **Important**

> Please note that handling of New tasks only is supported in the initial release.

### More Information

- Automated Upload Overview
- Creating an iWD Web Capture Point for Automatic Job Upload
- Defining Customized Mapping for iWD Web Capture Point

## Platforms/Tools Changes

- Support for Java 8 (mandatory for iWD 9.0 components) is implemented.
- Support for the Safari 10 browser is implemented.
- Support for embedded Tomcat is implemented.
- Support for Redhat Linux 7 is implemented.

## Discontinued Support

- Support for Java 7 and Java 6.
- Support for IBM WebSphere.
- Support for Microsoft Internet Explorer versions 8, 9 and 10.
- Support for Windows Server 2008.
- Support for MS SQL Server 2005.
- Support for the ACME samples tools and demonstration.
- The iWD Data Mart Reference Guide has been updated to reflect the fact that distribution points have been removed from iWD Data Mart.

# Systems, Platforms and Compatibilities Information

Please refer to these documents for further information.

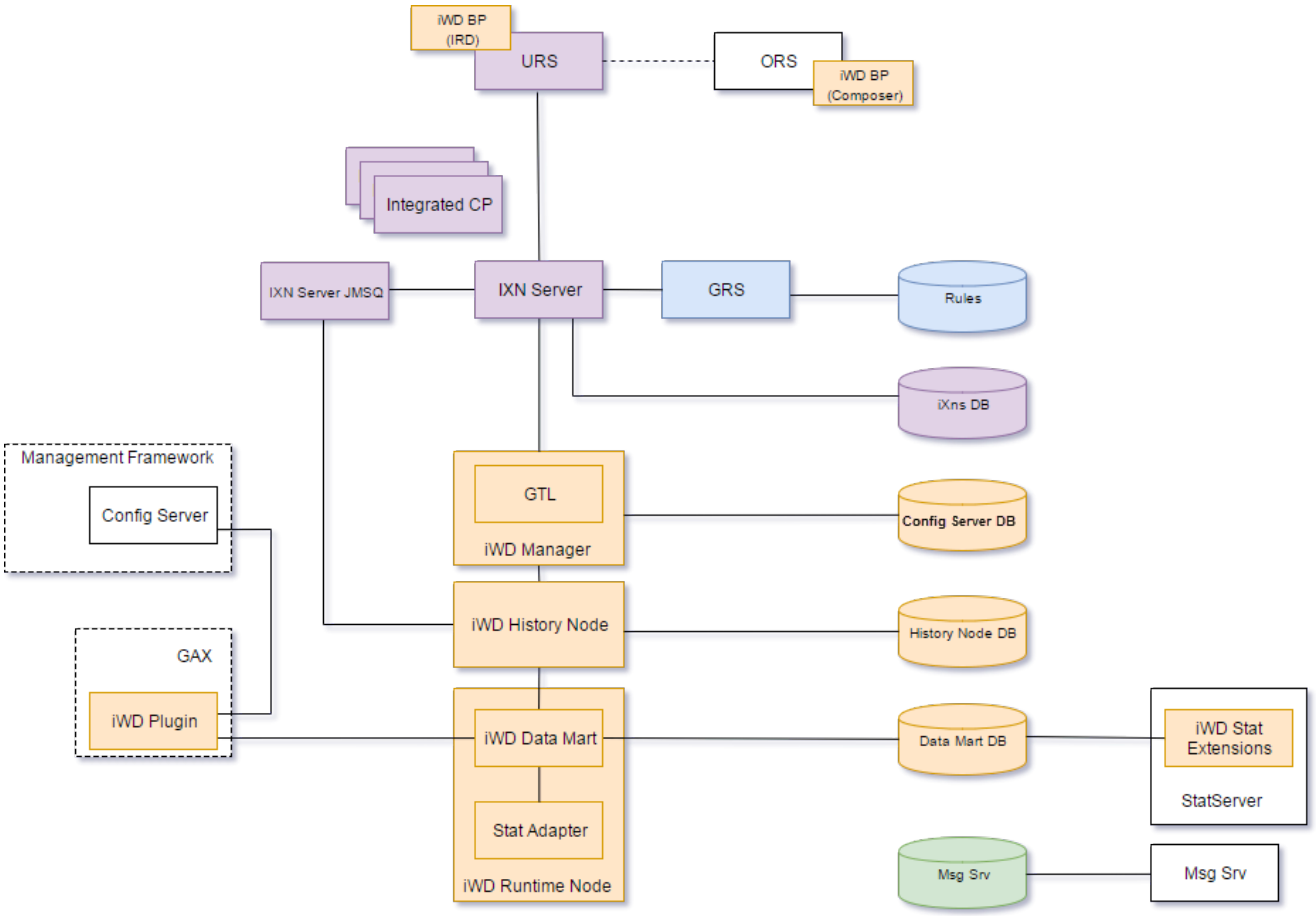
- Genesys Supported Operating Environments Reference Guide for IWD—Provides operating environments information required to run Genesys iWD.
- Genesys Interoperability Guide—Use this document for product availability and interoperability when you plan to add or upgrade Genesys products.
- Genesys Supported Media Interfaces Reference Manual—Use this document for information on media interfaces.
- Genesys Security Deployment Guide—Use this document for information on how to configure security levels for your environment.

# iWD Architecture Diagrams for 9.0.x

## 9.0.2 Component Architecture

New items in release 9.0.2 are:

- Interaction Server JMS Queue
- iWD History Node
- History Node database
- Removal of iWD Configuration Database and replacement by Configuration Server database



# Migration from 8.5.1x to 9.0.x

## Prerequisites

- iWD 9.0 requires Java 1.8. Support for Java 1.6 and 1.7 is withdrawn in release 9.0.

## General Migration Procedure

1. Backup all databases.
2. Upgrade iWD dependencies to supported versions (see Preparing for Installation).
3. Stop all iWD 8.5 components.
    1. iWD Manager's Tomcat.
    2. iWD Runtime Node.
    3. iWD Web.
    4. Interaction Server.
4. Install all components.
    1. Upgrade GAX plug-in to a new version (at least 9.0.002.07) by uninstalling the previous GAX plug-in and installing the new one (details here).
    2. History Node
        1. Install History Node using this procedure. Set up JMS and JDBC Database Access Points (DAPs).
    3. Data Mart
        1. History Node should be installed already.
        2. Migration
            1. Create a new database for Data Mart.
            2. Complete a fresh installation of Data Mart as a separate application.
            3. Copy all necessary settings from the iWD Runtime Node 8.5 application.
            4. Copy all customized ETL scripts to the iWD Runtime Node 9.0 ETL directory (**../iWD Runtime Node 9.0/etl**). Note: If your Kettle plugins fail to work on an Oracle database, please set the following database parameters:
                - `JDBC URL: jdbc:evo:oracle:@//<databasename>:1521/<SID>`
                - `JDBC Driver Class: evo.database.oracle.EvoDriver`

                For more information about setting database parameters please refer to iWD GAX Plug-in Datamart help.

5. Create a Configuration Server database JDBC DAP with configuration option **[iWD]/role = cfgserver** and add it to the Runtime Node connections (details here).
6. Add a **jdbc** section with options **url** and **driver-class** to all created DAPs listed in the connections (details here).
7. Change or set the Data Mart application to a new one for all necessary solutions using **GAX / iWD / Datamart / <Solution>**.
8. Using GAX, change the **ETL Scripts Directory** to a new one for each solution.
9. Check that your Data Mart Business Structure Solution has a timezone set.
10. Check that in the Data Mart application the **schedules/historical** and **schedules/intraday** are set as valid Quartz cron expressions.
11. [Optional] Disable the EventLog as it was replaced by History Node and JMS event logger.

4. Perform Data Migration from iWD 8.5.1x to iWD 9.0 if necessary.
5. Stat Server Extensions
    1. Install the new 9.0 version. No other changes are required.
6. iWD Manager
    1. iWD GAX Plugin must be updated already.
    2. Make sure iWD Manager 8.5 is stopped. Disable its autostart.
    3. Import new the iWD Manager 9.0 application templates (App and Server) as described in Installing iWD Manager and make applications from them.
    4. Install the iWD Manager 9.0 application itself to a separate folder.
    5. Copy the required settings from the old iWD Manager application to the new iWD Manager server application by adding a connection from iWD Manager Server to Interaction Server.
    6. Copy the required settings from the old iWD Manager application to the new iWD Manager Application by adding the necessary permissions for the iWD access group (to make users able to login, to read filter list, and so on).
    7. Migrate options from the old **iwd.properties** file to iWD Manager Server Application Options. The detailed description of the properties can be found here.
        - Note: Parameter **iwd.cfgConnectionTimeout** is replaced by four independent parameters for each connection:
            - **cfgConnectionTimeout** for the Configuration Server connection.
            - **ixnConnectionTimeout** for the Interaction Server connection.
            - **ucsConnectionTimeout** for the Universal Contact Server connection.
            - **msgConnectionTimeout** for the Message Server connection.
    8. Migrate logging settings.
        1. If you use Centralized Logging, make sure that you've created a connection from iWD Manager Server to Message Server.
        2. Configure logging in accordance with the details in Logging.
    9. Make sure that host and port of the primary Configuration Server, and the application name of the iWD Manager Server, are presented in the **General -> Command Line Arguments** of the

iWD Manager 9.0 Server.

10. Run the separate iWD Manager Configuration Database to Configuration Server migration procedure.
11. Open GAX and make sure that the **Scripts -> iWD Manager** folder is created and permissions to that folder are granted for the **EVERYONE** group.

7. iWD Web
    1. Install a new iWD Web application as described in this series of topics.

5. For each configured Solution you must complete following steps:
    1. Add the corresponding History Node using **GAX > iWD > Business Structure > '<Solution>' > History Node Settings**. For more details please refer to the IWD Business Solution Configuration.
    2. Make sure that the JDBC URL of every Interaction Server used by the Solution is configured in DAP. For more details please refer to the IWD Business Solution Configuration.
6. Start:
    1. All iWD 9.0 components.
    2. Interaction Server.

## iWD Manager Configuration Database to Configuration Server Migration Information

iWD Configuration Database migration is performed manually by a Python script similar to that used in 8.5.0 to 8.5.1 migration. To run the script, you need to provide the iWD Manager Configuration Database connection parameters (URL, login and password), and the GAX URL, username and password. The Python script takes the data from the IWD Configuration Database and sends it into the Configuration Server via the GAX API.

### Procedure

1. Install Python 2.7.
2. Add environment variables:
    - Path=C:\Python27;C:\Python27\Scripts
3. For Linux users some additional libraries may be required. To resolve this please deploy the following RPMs with the version appropriate for your OS version:
    - unixODBC-*.rpm (for example, for 64-bit RHEL6: unixODBC-2.2.14-14.el6.x86_64.rpm)
    - freetds-*.rpm (for example, for 64-bit RHEL6: freetds-0.91-2.el6.x86_64.rpm)
4. Download the required Instant Client packages for your platform. All installations require either the Basic or Basic Light package.
5. Unzip the packages into a single directory such as '"instantclient_12_2'".
6. Set your environment's library loading path (for example **LD_LIBRARY_PATH** on Linux, or **PATH** on Windows) to the directory created in Step 2.
7. Start your application.

8. Refer to any detailed instructions on the download page of your platform for more information. Note: Python and Oracle client versions must both be either 32- or 64-bit.
9. If there are any data with Unicode charaters to be migrated, ensure that the GAX application is configured to support UTF-8 encoding. You can do the following:
   - For Windows, add `-Dfile.encoding=utf-8` to JavaServerStarter.ini under the **[JavaArgs]** section.
   - For Linux, add `export JAVA_OPTS="$JAVA_OPTS -Dfile.encoding=utf-8` to setenv.sh.
10. Restart GAX.
11. Unzip the migration.zip file. This should be located under the **config/ folder** in iWD Manager's installation directory.
12. Install dependencies.
13. From the folder where migration.zip was unpacked, run the following command:
    ```
    pip install -—upgrade -r requirements.txt
    ```
14. Run the migration script for each tenant, include the Environment tenant:
    ```
    python main.py -g <gaxurl> -u <gax username> -p <gax password> -t <tenant> -y <db type> -d <db host> -r <db port> -n <db name> -s <db user> -w <db user password> --verbose
    ```

    The script parameters are as follows:

    - `-g, --gaxurl`—GAX URL address
    - `-u, --username`—GAX user name
    - `-p, --password`—GAX user password
    - `-t, --tenant`—Tenant Name
    - `-y, --dbtype`—iWD Configuration Database type (either Oracle or MSSQL)
    - `-d, --dbhost`—iWD Configuration Database host
    - `-r, --dbport`—iWD Configuration Database port
    - `-n, --dbname`—iWD Configuration Database name
    - `-s, --dbuser`—iWD Configuration Database user
    - `-w, --dbpass`—iWD Configuration Database password
    - `--verbose`—verbosity level

    **Oracle example:**

    ```
    python main.py -g http://192.168.123.132:8080/gax -u default -p password -t selenium -y oracle -d 192.168.123.123 -r 1521 -n xe -s iwd_user -w iwd_password --verbose
    ```

    **MSSQL example:**

    ```
    python main.py -g http://192.168.123.123:8080/gax -u default -p password -t selenium -y mssql -d 192.168.123.123 -r 1433 -n iwd_config -s iwd_user -w iwd_password --verbose
    ```

### History Node Specific Migration Information

- iWD History Node Migration

## TLS Configuration

To configure TLS, please refer to Configuring TLS for iWD.

# Migration to Interaction Server 9.0.x

## Prerequisites

Follow the recommendations provided in Recommendation Guide for Upgrading Interaction Server 8.5.x to 9.0.x for the Migration of the Interaction Server. Once the upgrade is done, a new Application Object for Interaction Server will be available in the Configuration Server.

## General Migration Procedure

Perform the below steps to complete the migration process for the new Interaction Server Application Object:

1. GRS: Creating the Connection to Interaction Server
2. iWD Manager: Application object configuration
3. For each Integrated Capture Point, change its connection to the new Interaction Server
4. For each solution in your Business Structure, update the Integrated Capture Point

# Data Migration from iWD 8.5.1x to iWD 9.0

There are two different options for data migration:

- **Migrating GTL History Events**—transfer GTL history data to iWD Manager 9.0.
- **Migrating Data Mart History Events**—transfer iWD task data to iWD Data Mart 9.0.

> **Important**
>
> 1. Data migration is not a mandatory step. Depending on your needs you can choose one option, both or none.
> 2. The migration process might be intensive and time consuming. It is recommended to avoid migration unless absolutely necessary.
> 3. If you cannot avoid the GTL History migration, please consider using the **taskEventLimit** option to reduce the amount of data to be migrated.

Please also note that depending on the selected iWD History Node configuration, IF:

- The option **[iWD]/process-gtl** = `false`, there is no point in migrating GTL History Events.
- The option **[iWD]/process-dm** = `false`, there is no point in migrating Data Mart History Events.

## Migrating GTL History Events

The purpose of this migration is to transfer task history data to iWD Manager 9.0.

You can find Task History by clicking the **History** tab in the GTL view. It displays all the task attributes' values and the task history events.

### Limitations

Task history migration is supported for tasks that are still in the Interaction Server database.

### Prerequisites

1. Installing iWD History Node is done.
2. iWD History Node Configuration is done
3. Stop all iWD 8.5 components.

1. iWD Manager's Tomcat.
2. iWD Runtime Node.
3. iWD Web.
4. Interaction Server.

### Warning

If Interaction Server is not stopped, no iWD task activity during the migration process will be recorded.

## Procedure

You can migrate GTL history events from the Event Log database to the History Node database by using History Node's built-in mechanism, which can be run with the following command:

### On Linux

```
./iwd_history.sh -host CFG_SRV_HOST -port CFG_SRV_PORT -app HN_APP events-migrate <yaml-config-filename>
```

### On Windows

```
JavaServerStarter.exe -host CFG_SRV_HOST -port CFG_SRV_PORT -app HN_APP events-migrate <yaml-config-filename>
```

where *<yaml-config-filename>* is the path to the YAML configuration file providing connection settings to the old environment.

### Warning

When events migration is in process, make sure that one of the following conditions applies:

- No History Node applications are running.
- If any History Node applications are running, they must have history events removal disabled (that is, the **stop-processing** option must be set to keep)
- The GTL History Events Migration mechanism only migrates events into the iWD Manager database and doesn't support migration to the Data Mart database.

## Prepare YAML Configuration File

1. Please find the configuration template file according to your database vendor:
   **Template path**

   ```
   ${INSTALLATION DIRECTORY}/config/template_iwd_events_migrate_<DB-VENDOR>.yaml.template
   ```
2. Copy and rename.
3. Edit the template and set the following parameters:
   - **solutionId**: *<solution-id>*
   - **taskEventLimit**: *<number>*
   - **interaction-server**:
     - **url**: *<url>*
     - **user**: *<user>*
     - **password**: *<password>*
   - **eventlog**:
     - **url**: *<url>*
     - **user**: *<user>*
     - **password**: *<password>*

> **Important**
> For security reasons, delete the passwords once migration/creation is complete.

## Configuration Options

- **logging**—Defines History Node logging configuration.
- **solutionId**—The runtime id of the solution for which events will be migrated. Solution with this id must be configured in Configuration Server.
- **taskEventLimit**—Allows to specify the maximum number of events to be migrated for one iWD task.
  - If a task consists of more events than specified, only the last of them will be migrated. It might be helpful for stability and performance reasons.
  - All the tasks with skipped events will be displayed in the migration log.
    - Valid values:
      `0`—No limit (default)

      1 to 2147483647—The maximum number of events

> **Important**
>
> iWD Manager cannot show more than 5000 events per one interaction. If you don't have any other specific needs then 5000 is the recommended value.

- **interaction-server**—Points to the Interaction Server database.
- **eventlog**—Points to the Event Log database.

## Working Command Line with Output Example

### Output Example

```
INFO [2020-06-22 17:03:02,104] com.genesyslab.iwd.eventlog.EventLogEventsMigrator: #50011|Command description: Migrate events from Event
Log db to the History Node
INFO [2020-06-22 17:03:10,229] com.genesyslab.iwd.eventlog.EventLogEventsMigrator: #50012|Connecting to the Interaction Server database
INFO [2020-06-22 17:03:11,979] com.genesyslab.iwd.eventlog.EventLogEventsMigrator: #50013|Events for 74480 interaction(s) will be migrated
INFO [2020-06-22 17:03:11,979] com.genesyslab.iwd.eventlog.EventLogEventsMigrator: #50014|Connecting to the Event Log database
WARN [2020-06-22 17:03:12,925] com.genesyslab.iwd.eventlog.EventLogEventsMigrator: #50071|Events limit exceeded, interaction:
```

```
01T6XTJXMYTMNEFR
INFO [2020-06-22 17:03:54,544] com.genesyslab.iwd.eventlog.EventLogEventsMigrator: #50016|Migration progress: 7448/74480
...
INFO [2020-06-22 17:08:14,220] com.genesyslab.iwd.eventlog.EventLogEventsMigrator: #50016|Migration progress: 67032/74480
WARN [2020-06-22 17:08:44,247] com.genesyslab.iwd.eventlog.EventLogEventsMigrator: #50071|Events limit exceeded, interaction:
02RHK42XSWDA4009
INFO [2020-06-22 17:08:45,247] com.genesyslab.iwd.eventlog.EventLogEventsMigrator: #50016|Migration progress: 74480/74480
INFO [2020-06-22 17:08:45,247] com.genesyslab.iwd.eventlog.EventLogEventsMigrator: #50015|Migration finished
INFO [2020-06-22 17:08:45,263] com.genesyslab.iwd.history.HistoryApplication: #50047|Shutdown hook triggered
INFO [2020-06-22 17:08:45,263] com.genesyslab.iwd.history.HistoryApplication: #50048|Shutdown hook completed
info
```

> **Important**
>
> Message #50071 shows interactions with skipped events due to specified **taskEventLimit** option.

## Migrating iWD Data Mart History Events

The purpose of this migration is to transfer the current tasks data to the iWD Data Mart 9.0. This way you will be able to see that data in iWD 9.0 reporting with some limitations (see below).

The procedure consists of two steps:

1. Migration from the EventLog database to iWD History Node.
2. Loading the data from iWD History Node to iWD Data Mart.

### Limitations

- Only unfinished tasks are migrated (backlog).
  - The Fact Tables and the Aggregate Tables are not populated with already finished tasks.
- The whole task event history is not migrated.
  - Included:
    - Agent's activity events: ASSIGN, FINISH, FINISH_RETURN, TRANSFER
    - A task "snapshot" or the latest task state - migrated as NEW
  - Excluded:
    - All other event types not listed as included (the list of event types: TASK_EVENT_TYPE)
    - The following metrics are not calculated:
      - Held Activity
        - TASK_FACT.TOTAL_HELD_TIME_SEC
        - TASK_WORK_FACT.HELD_TIME_SEC
      - Queue Metrics
        - TASK_QUEUE_FACT.ENTERED_TASK_COUNT
        - TASK_QUEUE_FACT.EXITED_TASK_COUNT

## Step 1: Migration from EventLog Database to iWD History Node

### Prerequisites

1. Installing iWD History Node is done.

2. iWD History Node Configuration is done.
3. Stop all iWD 8.5 components.
    1. iWD Manager's Tomcat.
    2. iWD Runtime Node.
    3. iWD Web.
    4. Interaction Server.

> **Warning**
> If the Interaction Server is not stopped, all iWD tasks activity during the migration process will not be recorded.

## Procedure

You can migrate Data Mart history events from the Event Log database to the History Node database by using History Node's built-in mechanism, which can be run with the following command: **On Linux**

```
./iwd_history.sh -host CFG_SRV_HOST -port CFG_SRV_PORT -app HN_APP datamart-migrate <yaml-config-filename>
```

**On Windows**

```
JavaServerStarter.exe -host CFG_SRV_HOST -port CFG_SRV_PORT -app HN_APP datamart-migrate <yaml-config-filename>
```

Where *<yaml-config-filename>* is the path to the YAML configuration file providing connection settings to the old environment.

## Prepare YAML Configuration File

1. Please find the configuration template file according to your database vendor:
    **Template path**

    ```
    ${INSTALLATION DIRECTORY}/config/template_iwd_datamart_migrate_<DB-VENDOR>.yaml.template
    ```

2. Copy and rename.
3. Please modify the following options:
    - **solutionID**: *<solution-id>*
    - **fetchSize**: `100`
    - **customAttributes**:
      - *<attribute1>*
      - ...
      - *<attribute10>*
    - **interaction-server**:
      *<url>*
      *<user>*
      *<password>*
    - **eventlog**:
      *<url>*
      *<user>*
      *<password>*

> **Warning**
> Take care to set the **customAttributes** section properly, otherwise they won't be migrated.

## Configuration Options

- **logging**—Defines History Node logging configuration.

- **solutionId**—The runtime id of the solution for which events will be migrated. Solution with this id must be configured in Configuration Server.
- **fetchSize**—Determines how many interactions are processed per batch.
  This option in combination with **processor-threads** could be useful for performance and stability tuning.
  - Valid values:
    - 1 to 2147483647 (100 by default)
- **customAttributes**—Defines a list of the custom attributes.

> **Important**
>
> You should provide the full list of Task Attributes from iWD 8.5. Attributes not listed in this field will not be migrated.

- **interaction-server**—Points to the Interaction Server database.
- **eventlog**—Points to the Event Log database.

## Working Command Line with Output Example

```
INFO  [2020-06-10 16:40:22,668] com.genesyslab.iwd.eventlog.EventLogData MartMigrator: #50011|Command description: Migrate events from
Event Log db to the Data Mart
INFO  [2020-06-10 16:40:32,356] com.genesyslab.iwd.eventlog.EventLogData MartMigrator: #50012|Connecting to the Interaction Server
database
INFO  [2020-06-10 16:40:33,496] com.genesyslab.iwd.eventlog.EventLogData MartMigrator: #50013|Events for 71320 interaction(s) will be
migrated
INFO  [2020-06-10 16:40:33,496] com.genesyslab.iwd.eventlog.EventLogData MartMigrator: #50014|Connecting to the Event Log database
INFO  [2020-06-10 16:40:39,584] com.genesyslab.iwd.eventlog.EventLogData MartMigrator: #50016|Migration progress: 10/71320
...
INFO  [2020-06-10 17:42:29,923] com.genesyslab.iwd.eventlog.EventLogData MartMigrator: #50016|Migration progress: 71320/71320
INFO  [2020-06-10 17:42:29,923] com.genesyslab.iwd.eventlog.EventLogData MartMigrator: #50015|Migration finished
```

> **Important**
> Message '#50016 Migration progress' shows the current migration progress.

Any errors are logged. For more detailed information you can use DEBUG or TRACE log level.

## Step 2: Loading the data from iWD History Node to iWD Data Mart

### Prerequisites

1. Step 1 Migration from EventLog DB to iWD History Node has been completed.
2. The Installing Runtime Node is done.
3. The iWD Runtime Node Options are configured.
4. The iWD Aggregate Plugins are configured.

### Procedure

1. Configure the specific migration options
    1. Open the iWD Data Mart Application Properties: **Configuration → iWD Datamart Properties → options**
    2. Add a new option **[settings]\etl-start-date**.
        - **Description**: the earliest date for which iWD Data Mart initialize time data in a new deployment.
          **Valid values**: date in format `yyyy-MM-dd HH:mm`, like `2015-10-26 00:00`

          **Default value**: the beginning of the current year.

          **Note:** Perform the following query to the Interaction Server Database in order to know the proper value.

```
SELECT min(submitted_at) as ETL_START_DATE
FROM interactions
```

3. Add a new option settings\aggregate-start-date
   - **Description**: the earliest date for which iWD Data Mart aggregates data in a new deployment.
     **Valid values**: date in format `yyyy-MM-dd HH:mm`, like `2015-10-26 00:00`
     Should be greater than **etl-start-date**.

     **Default value**: today - 30 days

> **Warning**
> If the **etl-start-date** or **aggregate-start-date** options are not set (or set inappropriately), it may cause "orphaned" rows in facts and aggregates."Orphaned" rows means that such records don't have a link to the **DATE_TIME** dimension and cannot be selected or filtered by date.

4. Add a new option **[schedules]\manual-mode** = `true`.
   - **Description**: It switches off a scheduled mode and allows to start ETL jobs manually

2. Start the Runtime Node 9.0.
   1. Open Data Mart Dashboard—A real-time view of the status of iWD Services.
   2. Run ETL jobs manually in the following pattern:
      1. Run Initialize job
      2. Run Load Config
      3. Run Load Intraday Job
      4. Run Aggregate Intraday Job

> **Important**

> Aggregate Intraday Job can process only 1 DAY of data per run. In order to catch up faster you should repeat this job as many times as needed until all the data is aggregated.

5. Run Load Historical Job
6. Run Aggregate Historical Job

3. Make sure that all the data was loaded
   1. The iWD History Node datamart table is empty.

```
SELECT count(*) cnt_rows
FROM <HISTORY_NODE_DB>.datamart
```

   2. The iWD Data Mart loaded data without fails.

```
SELECT *
FROM <DATAMART_DB>.etl_audit
WHERE STATUS in ('IN PROGRESS', 'FAILED')
```

3. Rollback the specific migration options:
   1. Remove **[settings]\etl-start-date**.
   2. Remove **[settings]\aggregate-start-date**.
   3. Remove **[schedules]\manual-mode**.

# Migration of the 9.0.x release to newer versions

> **Important**
> In release 9, you must upgrade all iWD components to the same release version (9.0.xxx must be the same for all components) to guarantee correct functioning.

## How to use this guide

1. Check your iWD version.
2. Install the latest version.
3. Perform all the steps one by one between your version and version you installed in step 2.
   For example, if you have 9.0.009.07 and migrate to the 9.0.011.xx, then you need to perform the following steps sequentially:
   1. Migration to 9.0.009.23+
   2. Migration to 9.0.010.xx

## Migration to 9.0.015.xx

Starting with this release, Genesys strongly recommends using native plugins provided with iWD. Kettle aggregation plugins do NOT support the new dimensions added in this release. Use of custom Kettle plugins is fully supported.

To switch to using the native version of the aggregation plugins, adjust the **plugins.properties** file.

> **Important**
> If you have already installed 9.0.015.03, please follow the migration procedure here.

### Migration procedure

1. Stop iWD Data Mart runtime node.
2. If you use aggregation plugins, it is recommended to switch aggregation to the native version of

plugins. Using an ASCII editor, change **plugins.properties** file, for each plug-in that you have enabled, as shown below:

```
${KETTLE_REPOS_DIR}\plugins\classif,task_classif_fact,native

${KETTLE_REPOS_DIR}\plugins\age,task_age_fact,native

${KETTLE_REPOS_DIR}\plugins\agent,task_agent_fact,native

${KETTLE_REPOS_DIR}\plugins\capt,task_capt_fact,native

${KETTLE_REPOS_DIR}\plugins\queue,task_queue_fact,native
```

3. Install the new iWD Runtime Node application.
4. Start the iWD Runtime Node application.

# Migration to 9.0.014.xx

## Changes to aggregate tables

This version contains the following changes related to aggregate tables.

- Intraday aggregates tables **I_TASK_[Subj]_FACT_15MIN** have been removed in this release. The Aggregate Intraday job now directly populates **H_TASK_[Subj]_FACT_15MIN** tables.
- For backward compatibility, the **I_TASK_[Subj]_FACT_15MIN** view has been introduced to reflect intraday aggregates tables behavior.
- Population of **H_TASK_[Subj]_FACT_DAY** tables is triggered by the fact that corresponding **H_TASK_[Subj]_FACT_15MIN** table contains a full set of 15-minute intervals for that particular day.

## Migration procedure

1. Stop the iWD Runtime Node application.
2. Back up the Data Mart database.
3. Install the new iWD Runtime Node application.
4. If you use the **H_TASK_[Subj]_FACT_15MIN** table directly in your reports or plugins, you need to take into account that it may now contain intervals for the current day. If you expect the prior behavior, you need to add a condition such as the one below to filter the incomplete aggregated day.
   For example:

   ```
   SELECT * FROM H_TASK_<subj>_FACT_15MIN WHERE interval_date_key <= (SELECT
   COALESCE(MAX(interval_date_key), 0) FROM H_TASK_<subj>_FACT_DAY)
   ```

   The <subj> can take values from the list of aggregate names: AGE, AGENT, CAPT, CLASSIF, QUEUE.
5. Start the iWD Runtime Node application.

## Migration to 9.0.013.xx

The following default attributes types were changed from **string** to **list**:

- iwd_ext_resultCode
- iwd_ext_requestedAgentGroup

To update the default attributes please follow this procedure.

## Migration to 9.0.010.xx

Custom Attributes can now have types configured. To update the default attributes please follow this procedure.

## Migration to 9.0.009.23+

> **Important**
> Interaction Server 8.5.3+ is required for this release of iWD and higher. DB Server connection is not supported: an ODBC connection must be used.

All standard iWD columns created in the Interaction Server database have been converted to type **nvarchar**. The length of all standard columns is also aligned with the Data Mart database.

Standard indexes are recreated during the migration procedure. If you have custom indexes using standard iWD columns, you will need to drop them before the migration run and recreate them manually after the migration.

The following procedure must be repeated for all iWD Solutions with different Interaction Servers.

### Migration procedure

1. Stop the Interaction Server application.
2. Back up the Interaction Server database.
3. If you have any non-standard indexes that use standard iWD columns, drop them manually to speed up the migration procedure.
4. Log into iWD Plug-in for GAX.
5. Open **Business Structure**.
6. Navigate to your iWD tenant.

7. Navigate to your Solution from the navigation tree and select the **Migration** tab. The **Interaction custom properties and migration issues** table on the right side notifies you of the updates that must be made.
8. Press the **Configure** button.
9. Recreate the non-standard indexes dropped in step 3.
10. Start the Interaction Server application.

## Migration to 9.0.008.xx

If you have not changed the default value of iWD Manager's **[iWD]/filterElementValuePattern** option, then either:

- Update this option to the new default value from Manager Configuration Options; or;
- Remove this option from the iWD Manager configuration object. This will force the application to use the default option value.

If there is a customized value stored in **filterElementValuePattern** then make sure that your regex allows percent (%), underscore (_) and backslash (\ ) for use with SQL wildcards within iWD Manager's filters.

### Migrate iWD Manager Logging options

Previously iWD Manager logging was configured via the **log4j.properties** file. This file has been removed and configuration properties have moved to Configuration Server. Move existing options according to the following mapping:

| log4j Property | <section.property> | Comment |
|---|---|---|
| log4j.rootLogger=DEBUG... | log.level | add rootLogger level to log.level option |
| log4j.appender.Console=org.apache.log4j.ConsoleAppender | log.log-to-console | If Console logging is enabled, set log.log-to-console to true |
| log4j.appender.Console.Threshold=TRACE | log.console-log-level | |
| log4j.appender.centralized_manager=com.genesys.logging.log.CentralizedAppender | log.centralized-logging | If Centralized logging is enabled set log.log-to-file to true |
| log4j.appender.centralized_manager.Threshold=DEBUG | log.console-log-level | DEBUG, TRACE, WARN, INFO => STANDARD, ERROR => ALARM |
| log4j.appender.manager=org.apache.log4j.RollingFileAppender | log.log-to-file<br>log.archive | • If File logging is enabled set log.log-to-file to true<br>• If Logger class is org.apache.log4j.FileAppender set archive to false,<br>• If Logger class is org.apache.log4j.RollingFileAppender |

| log4j Property | <section.property> | Comment |
|---|---|---|
| | | set archive to true |
| log4j.appender.manager.Threshold=DEBUG | log.log-level | |
| log4j.appender.manager.File=/GCTI/iWD/iwd_manager.log | log.log-filename | |
| log4j.appender.manager.MaxBackupIndex=9 | log.max-history | |
| log4j.appender.manager.MaxFileSize=1024MB | log.max-file-size | |

### Migrate Data Mart and History Node logging options

Data Mart and History Node logging options have been updated. Migration to the new version of logging configuration may be performed via iWD Plugin for GAX. Note that iWD Plugin for GAX should be updated to version 9.0.008.0x

1. Login to GAX.
2. Navigate to **GAX -> Configuration -> Environment > Tenants**.
3. Go to the **iWD Attributes** tab
4. Check that the fields "'Current Configuration Version"' and "'Actual Configuration Version"' have the same value—`9.0.0.2`.
5. If the "'Current Configuration Version'" is lower than the actual one, click the "'Update Configuration"' button.
6. If the '"Current Configuration Version"' is still not updated, refer to the GAX logs.
7. If the "'Current Configuration Version"' and "'Actual Configuration Version'" are equal to 9.0.0.2, you can check and modify options as described on this Logging page.

## Migration to 9.0.007.07

If you have already installed a 9.0.x iWD Runtime Node prior to the 9.0.007.07 release, you must manually upgrade the database schema before starting the new iWD Runtime Node versions. The upgrade procedure alters the `H_TASK_FACT` table, adding the NOT NULL constraint to the `last_task_event_id` column and changing the primary key.

### Migration Procedure

1. Stop the iWD Runtime Node application.
2. Back up the Data Mart database.
3. Install a new iWD Runtime Node application.
4. In the <iWD Runtime Node>/etl/migration/manual directory, find a migration script for the appropriate database type and run it on the Data Mart database.

5. Start the iWD Runtime Node application.

# Troubleshooting iWD Runtime Node 9.0.015.03

## Important

This article troubleshoots the following issues:

- Existing historical data was not updated with the default media type and interaction type value after installing 9.0.015.03 iWD Runtime Node.
- Follow this procedure if you have installed 9.0.15.03 IWD and your Data Mart runs on MSSQL or POSTGRES databases.
- Do NOT execute this procedure if you have not installed 9.0.015.03.

## Procedure

1. Obtain the latest IWD release (9.0.15.05 Release Note).
2. Stop the iWD Runtime Node application.
3. Back up the Data Mart database.
4. Depending on your database vendor:
   - For MSSQL go to the folder **..\iWD Runtime Node\etl\migration\manual\mssql**.
   - For POSTGRES go to the folder **..\iWD Runtime Node\etl\migration\manual\postgres**.
5. Run the **manual_migration_fix_90011.sql** script against your Data Mart database.

# Deployment Overview

> **Important**
> Genesys strongly recommends that you perform all installations manually and subsequently configure the applications using the Genesys Administrator Extension (GAX) and the iWD Plug-in for GAX.

Deployment of IWD consists of three main phases:

- Installation phase
- System configuration phase
- Business logic configuration phase

# Installation Phase Overview

## Definitions

Installation results in a fully functional iWD Manager application and prepared iWD Runtime Node, iWD History Node and iWD Stat Server Java Extensions (iWD SSJE). Almost all the configuration is carried out using the Genesys Administrator Extension (GAX) component and the iWD Plug-in for GAX component (from now on referred to as iWD GAX Plugin).

The iWD Runtime Node is a container in which iWD Data Mart is packaged.

The iWD History Node utilizes Interaction Server's ability to provide EventLog data via either Kafka or Java Messaging Service (JMS) queue to filter uninteresting records before they enter the database. It also splits the source datastream into separate tables, one for Data Mart and one for the Global Task List.

The iWD SSJE provides real-time statistics aggregated in the iWD Data Mart.

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

iWD installation consists of three steps:

- Configuration of either Kafka or JMS
- Preparing databases
- Installing applications

## Configuration of either Kafka or JMS

### Kafka

Genesys recommends using Kafka for Event Logging purposes. No additional libraries are needed for iWD History Node to operate. The Kafka broker itself however is not supplied in iWD software distribution, so it is your responsibility to obtain and install Kafka.

### JMS

Neither JMS broker nor JMS driver is supplied in the iWD software distribution. Make sure that you have installed a suitable driver for the JMS queue. A list of supported JMS brokers is in the Genesys Supported Operating Environment Guide.

### Important

The JMS queue driver must be placed in the History Node installation directory:

```
{INSTALLATION DIR}/lib
```

### Important

The JMS URL configured in Interaction Server must have following option:

```
jms.redeliveryPolicy.maximumRedeliveries=-1
```

Example URL:

```
tcp://jmshost:61616?jms.redeliveryPolicy.maximumRedeliveries=-1
```

## Preparing Databases

A basic iWD deployment uses three operational databases:

- Genesys Rules System (GRS) rule repository—GRS stores business rule templates and business rules. For more information, see the Genesys Rules System 8.5 Deployment Guide.
- Interaction Server databases—Interaction Server database stores iWD tasks. Make sure that you have installed and configured the Interaction Server and its associated databases, as described in the *eServices Deployment Guide*.
- History Node database—The History Node database stores a filtered subset of iWD events. Only sample JDBC drivers are supplied in the iWD software distribution. Make sure that you have installed and configured a suitable database and JDBC driver for it.

JDBC drivers do not require any configuration. However their **.jar** files must be put into the **{HISTORY NODE INSTALLATION DIR}/lib** directory.

- Data Mart database—Uses Fact tables, aggregates, dimension and measures to provide sophisticated reporting options. As with History Node, make sure that you have installed and configured a suitable database and JDBC driver for it. In the iWD software distribution only sample JDBC drivers are provided. JDBC **.jar** files must be put into the **{IWD DATAMART INSTALLATION DIR}/lib** directory.

For a list of the database engines that are supported by the iWD application, Genesys Rules System, and Interaction Server, refer to the Genesys Supported Operating Environment Guide. Please note that the iWD application might not support all of the databases that are supported by Interaction Server or the Genesys Rules System.

## Application Installation

After the iWD operational databases have been prepared, the iWD GAX Plugin component, the iWD Manager application, the iWD History Node, the iWD Runtime Nodes and iWD SSJE should be installed manually.

> **Important**
> From release 9.0, all iWD components operate as stand-alone Java applications.

All five can be configured in release 9.0 by using Genesys Administrator Extension (GAX).

A basic iWD deployment, such as in a lab environment, utilizes a single instance of iWD Manager, as well as a single instance of an iWD Runtime Node and an iWD History Node. In more complex scenarios, such as multi-tenant, high-volume or high-availability deployments, multiple iWD Runtime Node instances can be installed. These deployments provide more controlled resource partitioning and allow load distribution across multiple physical servers.

In addition to the iWD GAX Plugin, the iWD Manager and iWD Runtime Node installation and iWD History Node, Genesys Rules System must be installed. This installation includes:

- Genesys Rules Development Tool, which is a Genesys component for business rule template development
- Genesys Rules Authoring Tool, which is used for authoring and deploying business rules
- Genesys Rules Engine, which is an engine that receives and processes requests for business rule evaluation from clients, such as the routing strategies in the iWD Business Process (IWDBP).

# System Configuration Phase Overview

## Tenants

iWD configuration supports multi-tenancy. iWD automatically creates a root tenant that is named `System` by default. The `System` tenant is automatically mapped to the `Environment` tenant in Genesys Configuration Server. Although it is possible to configure iWD solutions, services, and business logic directly at the `System` tenant level, it is recommended that you create a subtenant for that configuration. In iWD terminology, such a subtenant is called a managed tenant. This managed tenant is associated directly with a tenant defined in Genesys Configuration Server. In a single-tenant environment the configured managed tenant maps to the `Resources` tenant in Genesys Configuration Server.

A user who is configuring the system in Genesys Administrator Extension (GAX) can have access to one or more managed tenants. Access is defined by the role-based access controls that are configured in GAX, per tenant. The policy allows definition of an arbitrary numbers of user roles, where each role can be assigned to any Genesys Configuration Server users or access groups. Each role has a number of associated iWD permissions. Note, that iWD Manager and iWD GAX Plugin have separate lists of permissions.

> **Important**
>
> The tenancy model that is supported in the iWD application currently has a one-to-one relationship with the tenancy model in Genesys Configuration Server.

### Multi-tenant Configurations

If you are using a Multi-Tenant Configuration Server, please be aware that the iWD tenant can only access resources (such as Skills and Agent Groups) that are specifically configured under the corresponding Genesys Configuration Server Tenant in GAX. Therefore, any Skills, Agent Groups, or other resources that are configured at the Environment Level will not be accessible in rules at a child-tenant level in iWD. To make these resources accessible, they must be configured as resources under the child tenant in Configuration Server.

## Solutions

Solutions are used for partitioning logical and physical resources for purposes of user access control and load partitioning (performance). Normally there will be one Solution per Tenant.

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

- Runtime nodes—iWD runtime application instances that are stand-alone Java applications. Runtime

nodes require simple preparation during installation.

- StatServer—Provides statistics to CCPulse+ via iWD Stat Extensions
- Business logic configuration—Primarily the configuration of iWD departments and processes.
- Multiple solution instances can be configured per tenant, if necessary (for example, “Production” and “Test”).

## Runtime Nodes

Runtime Nodes are instances of the iWD runtime application and are run as a stand-alone Java application. Services that run within a runtime node are configured in (and managed through) the iWD GAX Plugin application.

## Services

iWD Services implement actual iWD functionality, such as loading data into the Data Mart. Refer to iWD Services for more information about iWD Services, including the recommended order of configuration.

# Business Logic Configuration Phase Overview

The iWD business logic configuration phase is where iWD business context is introduced. This includes definition of departments and processes. This phase also includes the definition of business rules for use in task processing. 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 will be handled by iWD. This includes business processes, service-level agreements (SLAs), and other factors that influence task-handling logic.

## Departments and Processes

Departments represent organizational entities for which iWD will perform task prioritization and routing. Processes represent the business processes that are within those enterprise departments. In iWD, processes are always grouped within (associated with) a department. Departments and Processes allow for the definition of task-handling business rules that are specific to a department or process context.

Departments and Processes are created in iWD GAX Plugin and stored on Genesys Configuration Server. They are used by iWD Manager for task filtering and Genesys Rules System for managing rules.

Each department and process allows for the definition of a maximum of 5 custom attributes to a department/process in order to provide additional enterprise-specific context for reporting purposes.

> ### Important
> You can create new business processes (under the same Tenant) that can support clear logical distinctions between Processes and Departments For example, interactions from different media types (email, chat, SMS and so on) can be handled by separate business processes with their own customized queue names, and this in turn can provide clear logical distinctions in reporting, because the queue name is the basis for handling reporting requirements. See either **Working with the iWDBP for IRD** or **Working with the iWDBP for Composer/ORS**'

## Rules

Part of configuring the business logic for departments and processes in iWD is configuring and associating rules. All business rule authoring for iWD is done through the Genesys Rules Authoring Tool, which is a component of the Genesys Rules System. These rules define the task-handling business logic that is applicable to the departments, processes, or the entire system. Generally, a

rule is represented by zero, one or more conditions and one or more actions. If there are no conditions for the rule, its actions will always be executed. If all of the conditions are true, all of the actions are executed. If any condition is not true, none of the actions are executed.

Rules are expressed in an easy-to-understand human language, such as:

```
If the 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
- As a decision table.

Linear rules are intended for complex rules that have many conditions and/or actions. Each condition or action is represented by a single line in the rule.

Decision tables represent a more compact form of rule representation; however, they might not be as well suited to complex rules. In a decision table, multiple rules are grouped together, so that each condition or action is represented by a column in a table, and each row represents a rule. The number and type of conditions and actions (columns) is constant across all of the rules in the list.

## Task Classification

Task classification does the following:

- Associates a task with a configured process.
- (Optionally) assign values to other 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:

- Package-level rules (also known as Global Rules)
- Department
- Process

If rules are defined for more than one context, they are evaluated in sequence, as previously listed. After a process has been assigned to a task, additional classification rules are evaluated that have been defined, first at the rule package or global level, then at the department level, and finally at the process level. The figure below shows an example of Task Classification.

A process must be assigned during the classification phase. It can be assigned in two places:

- At the global level
- At the department level.

(A department can be assigned at the global level and then, the rule evaluation can continue at the department level to actually select a process.)

> **Important**
> Capture Point rules are configured at the rule package (global) level, by using the rule condition `Capture Point is`.

.

### Task Classification Using the “Capture Point is” Rule Condition

For more information about task classification, refer to **Working with the iWD Business Process for IRD** or **Working with the iWD Business Process for Composer/ORS**.

## Task Prioritization

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

Prioritization rules are initially applied immediately after classification rules and then reapplied after a specified reprioritization period. The reprioritization period is expressed in the same way as any other rule action.

### Task Prioritization

If a reprioritization period is not set for a task during the prioritization phase in business rules, the `IWD_reprioritizeDateTime` attribute is set to `Dec 31, 2030`. Therefore, for all intents and purposes. the task will not undergo further reprioritization unless it is restarted.

The Standard Rules Template includes two rule conditions, `Is first prioritization` and `Is reprioritization` that should be used in prioritization rules to ensure that the reprioritization interval is set correctly, while avoiding any unnecessary immediate reprioritization of a task (that is, the first time prioritization rules are evaluated).

For example, suppose you have a task that, during the classification phase, gets an initial priority of 100. You wish to increase the priority by 15 every 2 hours, if the task is due in less than 24 hours. You want to do the first check 1 hour after the task is classified. You would set this up by using two different prioritization rules, configured in the order shown below. The example `Rule 1–Is First Prioritization` shows the first rule, which includes the `Is first prioritization` condition. The second rule, shown in `Rule 2–Is Reprioritization`, includes the `Is reprioritization` condition.

### Rule 1—Is First Prioritization Example


### Rule 2—Is Reprioritization Example


For more information about task prioritization, refer to the Working with IWD Business Process documentation.

## Business Calendars

A business calendar is a set of configuration parameters 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 a working week and working hours that apply to all working days.

A definition of a working week from Monday to Friday—in which 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 to 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. Business calendars can either be assigned to a rule itself, or can be assigned in a rule action. In either case, the business calendar must be assigned before other rules that use the business calendar can be evaluated.

> **Important**
>
> Business Calendars are created in the Genesys Rules Authoring Tool. In Genesys Rules System release 8.5.1, business calendar functionality has been enhanced. See **Business Calendar Enhancements** (GRS Best Practice/User Guide).

## Business Calendar Example

Standard ACME calendar

| Name | Entry Type | Calendar Placement | Definition | | | | Start Time | End Time | |
|---|---|---|---|---|---|---|---|---|---|
| New Year's Day | Holiday | Fixed | Jan 1, 2012 | | | | | | |
| Thanksgiving | Holiday | Relative | Fourth | Thursday | of | November | | | |
| Day after Thanksgiving | Holiday | Relative | Fourth | Friday | of | November | | | |
| Day before Thanksgiving | Time Change | Relative | Fourth | Wednesday | of | November | 8:00 AM | 12:00 PM | |

# Task Attributes and Interaction Property Analysis Overview

Before you configure a working iWD environment with Capture Points (or if you are planning to bypass Capture Points and upload tasks directly by using iWD Web) you need to carry out an analysis of what task attributes and interaction properties (including custom properties) you need to capture from the source system and how they will be mapped in iWD and processed by the Genesys Rules engine and reported on subsequently.

This topic is covered in detail in **Working with Task Attributes and Interaction Properties**.

There are also best practice recommendations for **Task Capture here**.

# Installation

## Task Summary: Installing iWD 9.0

The following table outlines the task flow for installing iWD 9.0. The procedures referenced in this table provide instructions about installing iWD components on Windows hosts. See **Preparing for Installation** for software prerequisites.

**Installing iWD 9.0**

| Objective | Related procedures and actions |
|---|---|
| 1. Prepare for installation and review prerequisites. | Ensure that your environment meets the prerequisites. Ensure that the required CDs are available. See **Preparing for Installation**. |
| 2. Install Interaction Server. | Genesys recommends that you use the latest version of Interaction Server that is included with iWD. If your environment does not have an instance of Interaction Server installed already (such as for an eServices solution), you must install one. See **Install Interaction Server**. |
| 3. Install Genesys Administrator Extension (GAX). | If it is not already installed, you must install GAX. |
| 4. (Recommended) Upload the iWD Installation Packages in GAX. | Genesys recommends that you upload all the iWD installation packages (IPs) in GAX. See **Uploading IPs Using GAX**. Alternatively, you can do this on a component-by-component basis, in which case the templates must be uploaded separately. |
| 5. Install the iWD GAX Plugin. | In Genesys Administrator Extension, install the iWD GAX Plugin component, which is required to complete iWD configuration. See **Install iWD GAX Plugin**. |
| 6. Install the Genesys Rules System. | Complete steps 1 to 6 (only) in the **Task Summary: Installing Genesys Rules System** section of the *Genesys Rules System 8.5 Deployment Guide*. **Note:** Step 7 in the Task Summary is about defining the business structure, whichis performed in the iWD GAX Plugin. |
| 7. Install iWD History Node. | See **Install IWD History Node**. |
| 8. Install iWD Runtime Node. | Install Runtime Node and associate a Person account to the iWD Runtime Node application. **Install Runtime Node**. |
| 9. Install iWD Manager. | See **Install IWD Manager**. |

| Objective | Related procedures and actions |
|---|---|
| 10. Create an iWD Tenant in GAX. | See **Creating a Tenant in GAX**. |
| 11. Create an iWD Business Solution in the iWD GAX Plug-in. | See **Create an iWD Business Solution in the iWD GAX Plug-in**. |
| 12. Configure the necessary Integrated Capture Points for use with iWD. | For background on Capture Points, see **eServices Integrated Capture Points Guide**. For iWD-specific details, see **Capture Point Details**. |
| 13. Configure the various mandatory configuration objects **manually**. | |
| 14. Install iWD Stat Server Extensions. | See **Installing the iWD Stat Extensions**. |
| 15. Configure logging for iWD Manager and iWD Runtime Node. | See **Logging**. |
| 16. Update the Interaction Server database. | See **Working with Task Attributes**. |
| 17. Configure remaining iWD services and objects. | See the **iWD GAX Plug-in Help topics**. |
| 18. Optional: Configure iWD Reporting. | See **iWD Reporting** for details on configuring CCPulse+ for iWD and creating CCPulse+ templates. |
| 19. Optional: Configure Multiple Business Processes if required. | See **Configuring Multiple Business Processes** for details on how to clone the iWD Business Process to create new business processes with custom queue names. |
| 20. Optional: Configure the system to start/stop application server service through Genesys Solution Control Interface (SCI) or Genesys Administrator. | |
| 21. Optional: Install and configure iWD Web. | iWD Web allows you to upload tasks to INX/iWD directly from spreadsheets. See **Installing iWD Web** for details. |

# Preparing for Installation

### Important

- In release 9.0, all iWD components operate as standalone components.
- JDBC drivers provided along with IPs (iWD History Node, iWD Datamart and iWD SSJE) are only for sample purposes. Customers must provide appropriate JDBC drivers for iWD History Node, iWD Datamart, iWD SSJE and GAX (for iWD Plugin for GAX).
- iWD does not support UCS 9.x.

## Prerequisites

### Software Environment

- From release 9.0.2, iWD requires:
  - Eclipse in order to use the Composer version of the iWD BP.
  - ActiveMQ installation with appropriate driver (see Installation Phase Overview).
- For the new iWD Manager user interface, these are the minimum browser requirements (Genesys strongly recommends latest versions at all times):
  - Google Chrome (40+)
  - Mozilla Firefox (52+)
  - Microsoft Internet Explorer 11+
  - Apple Safari 9+
- For production deployments, install the Genesys Rules System web applications on a separate instance of the application server.
- A supported database server must be installed, with the database created and accessible for both the iWD Data Mart and the iWD History Node database. Since connection to these databases will be made via JDBC make sure to have in possession suitable JDBC drivers for your database of choice.

### Important

From release 9.0, Java 6 and 7 are no longer supported. Refer to the iWD page in the Genesys Supported Operating Environment Reference Manual for supported platforms and database servers.

## Genesys Components

### Warning

iWD components are provided with UTF-8 encoding enabled by the default. If your environment is not configured for Unicode support and no non-Latin characters are expected in it, then remove encoding arguments from the startup scripts of iWD components after their installation.

The following procedure applies for iWD Manager, iWD History Node and iWD Runtime Node:

1. Under your installation folder, locate the **JavaServerStarter.ini** file (for Windows) or the **iwd_<component_name>.sh** file (for Linux, **<component_name>** is either `manager`, `history` or `runtime` for the corresponding component).
2. Remove the **-Dfile.encoding=UTF-8** part from the **[JavaArgs]** section (for Windows) or from **JAVA_OPTS** (for Linux).

**For iWD Manager only**, also add the following Java options under the same location as described above to avoid possible localization issues:

- **-Duser.language=en**
- **-Duser.region=EN**

However if non-Latin characters are expected, then refer to the following links for more details on how to apply Unicode support:

- Management Framework
- Interaction Server

For instance, Transport Protocol Parameter **string-attributes-encoding** should be passed in the connection to Configuration Server for non-UTF-8 data processing by iWD components. You can find more details in the links above.

## Components List (minimum)

- Management Framework—8.5.1
- Interaction Server—
  - 8.5.110 for releases of iWD up to 9.0.008.xx.
  - 8.5.3+ for releases of iWD from 9.0.009.23 onwards. DB Server connection is not supported: an ODBC connection must be used.
- Universal Routing Server—8.1.400
- Orchestration Server—8.1.400.48
- Universal Contact Server—8.5.300.09 (Note: iWD does not support UCS 9.x.)
- Genesys Rules Engine—8.5.302

- Genesys Rules Authoring Tool—8.5.302
- GAX:
  - 8.5.250.11 for releases of iWD GAX Plug-in up to and including 9.0.002.20.
  - 8.5.270.06 for iWD GAX Plug-in 9.0.003.07.
  - 8.5.280.06 for releases of iWD GAX Plug-in from 9.0.004 up to and including 9.0.017.01.
  - 9.0.103.05 for releases of iWD GAX Plug-in from 9.0.017.10 onwards.

> **Important**
> iWD GAX Plug-in 9.0.016 does not support versions of GAX higher than 9.0.101.29.

- Stat Server 8.5.107
- Pulse 9.0.0

## Support for RHEL 64-bit Platforms

Installers support installation on Red Hat Linux 64-bit platforms but require additional packages to be installed:

- `libnsl.so.1`

Prior to iWD version 9.0.020.13, installers required standard compatibility packs for 32-bit platforms. These packages have generic names like:

- `compat-glibc*`
- `compat-libstdc++*`
- `libstdc++*`

## Java configuration

Depending on what version of Java will be used to run iWD components, you need to set different environment variables:

- For Oracle JDK 8 and OpenJDK 8, set **JRE_HOME**.
- For OpenJDK 11, set **JAVA_HOME**.

## iWD DVD

This DVD contains the following components:

- iWD Runtime Node—Runs scheduled iWD Data Mart services.
- iWD Manager—A graphical user interface (GUI). iWD Manager is used for real-time management of tasks.
- iWD History Node—Reporting data source for iWD Data Mart and iWD Manager .
- iWD GAX Plugin—Business configuration of the iWD solution.
- iWD Stat Extensions—Stat Server Java Extensions that provide Stat Server clients, such as CCPulse+, the ability to request and display current-day statistics from iWD's Data Mart.
- iWD Web—A web-based, easy-to-deploy workload management solution.

> **Important**
>
> For production deployments, install the iWD web applications and the Genesys Rules System web applications as separate application server instances, for improved scalability and memory management.

## Installing as Standalone Application

### On Windows

To deploy on Windows, find the following files:

- `JavaServerStarter.exe`—This is the executable binary that is used in Windows service definition and is used to launch the application defined in the `JavaServerStarter.ini` file.
- `JavaServerStarter.ini`—This is a configuration file that allows you to define the application and its starting options. Because iWD depends on Java you can define Java options here. For example;

```
[Service]
AppTitle="Genesys iWD Data Mart"
AppVersion=9.0.000.xx
JVMPath=JRE_PATH\bin\server\jvm.dll
MainClass=evo/main/NodeLauncher
EventHandlerMethod=stopNode

[JavaArgs]
-Xms256m
-Xmx1536m
-XX:MaxPermSize=128m

[Djava.class.path]
iwd_node.jar
```

## On UNIX

On UNIX there are no `JavaServerStarter` files. The application uses a regular script, for example; `iwd_runtime.sh`, where Java with its optional arguments is explicitly called. Then you can simply use the following example:

```
JAVA_JRE -XX:MaxPermSize=128m -Xms256M -Xmx1536M -host Configuration Server Host -port Configuration Server Port -app Application_Name
```

> **Important**
>
> On 64-bit platforms, you can define more than 1.5GB for the `Xmx` parameter. For example, Genesys recommends **12GB** for the Runtime Node application.

# Interaction Server Configuration

iWD 9.0 supports single Interaction Servers or clusters via a cluster proxy.

## Single Interaction Server

### Preparation steps

#### Existing Deployments

If you are an existing eServices customer, and Interaction Server and its databases are already installed and configured for your environment, please check the items in the *iWD-Specific Configuration* section, then you can proceed with installation of other iWD components.

#### New Deployments

Please install Interaction Server by using the procedures in the eServices Deployment Guide.

A Multimedia Switching Office and Multimedia Switch must be created in Genesys Configuration Database, to support Stat Server and URS operations. Refer to the eServices Deployment Guide for more details on these topics.

### iWD-specific configuration

#### Interaction Server database configuration

If you use PostgreSQL DBMS for Interaction Server, make sure parameter **STANDARD_CONFORMING_STRINGS** is turned `ON` for the Interaction Server database. To change this parameter without affecting the rest of your database server you can set it directly for the user by which Interaction Server accesses its database. Here is an example SQL command:

```
ALTER USER <ixn_db_user> SET STANDARD_CONFORMING_STRINGS TO ON;
```

where *<ixn_db_user>* is the user that is configured in Interaction Server's DAP configuration object.

#### Configuring the JDBC connection link

For automatic update procedures to be carried out by iWD Plugin for GAX, you must provide the JDBC connection URL to the Interaction Server database by following these steps:

1. Using GAX, open Interaction Server's DAP configuration object.
2. Open the **Application Options** tab.
3. Create a new option: **jdbc / url / <IXN_RDBMS-specific-JDBC-URL>**

### completed-queues

There is a specific Interaction Server configuration option named completed-queues that specifies a list of queues for completed interactions. When an interaction is placed into one of these queues, the **CompletedAt** timestamp is set for that interaction. This is also the timestamp that will be used to calculate the **Age** of the interaction that is displayed on the Global Task List. This option, if it is not already present, will be added for you automatically by using the **Configure Ixn Custom Properties** feature (formerly in iWD Manager but moved to the iWD Plug-in for GAX from release 9.0). However, this will only add the **iWD_Completed** queue to the option. You might want to add other queues to this option, based on how you want this **Age** to be calculated. For example:

- When using the iWDBP for IRD you may wish to set it to: `iWD_Completed, iWD_Canceled, iWD_Rejected`
- When using the iWDBP for Composer you may wish to set it to: `iwd_bp_comp.Main.iWD_Completed, iwd_bp_comp.Main.iWD_Canceled, iwd_bp_comp.Main.iWD_Rejected`
- Section: **[settings]**
- Option name: **completed-queues**
- Valid Values: Comma-separated list of queue names

### enable-revoke-from-agent

From Interaction Server 8.1.2, a configuration option, enable-revoke-from-agent has been provided to support enhancements to task management in the Global Task List. Specifically, this option allows an Assigned interaction to be revoked from an employee desktop when the interaction is put on hold, canceled, or completed, from the Global Task List or through an integrated capture point.

## Interaction Server Clusters

iWD 9.0 supports connection to one of the following:

- Interaction Server (as described in previous section Single Interaction Server)
- Interaction Server Proxy
- Interaction Server Proxy cluster

> **Important**
> Direct connection to an Interaction Server cluster is not supported. Use Interaction Server Proxy to add a connection to the Interaction Server cluster.

This support requires minimum versions of 8.5.106.x of Interaction Server and 8.5.107.x of Interaction Server Proxy.

Deployment of Interaction Server Proxies, Interaction Server clusters and Interaction Server Proxy

clusters is described in the eServices Deployment Guide.

When a Tenant for an iWD Solution is created (see Creating the Tenant) add a connection to this new Tenant from any newly created Interaction Servers, Interaction Server Proxies and Interaction Server Proxy clusters.

# Uploading IPs Using GAX

Once you have installed the GAX, you can (optional at this stage—you can do this on a component-by-component basis) upload the relevant IPs that are required for the configuration process.

Click **here** (new document) for the GAX IP Upload Procedure.

# Installing the iWD GAX Plug-in

> **Important**
>
> Before any install of the GAX Plug-in, if you have installed it previously you must perform the removal procedure below to ensure a clean installation.

## Installation steps

1. Copy the iWD GAX Plug-in IP to the host machine.
2. Stop GAX (if it is running).
3. Run the **setup.exe** (Windows) or **install.sh** (Linux) installation file.
4. Follow the prompts in the installer to install the iWD GAX Plug-in. Install the plug-in in the GAX working directory under the **plug-ins** sub-folder: for example; **/GCTI/GAX/plug-ins**.
5. Restart GAX.

## Notes on Localization

- The base GAX application and its plugins (including iWD GAX plug-in) are not fully separated with their localization.
- iWD GAX Plug-in is delivered with all available localizations at the time of its release.

## Notes on Installing/Removing the iWD GAX Plug-in

> **Important**
>
> If you perform the **GAX Plug-in removal procedure** for the iWD GAX Plug-in, you need to make sure that following files have been deleted (if you are working on Windows, some of these files might be already removed by the generic Windows uninstaller):
>
> - **<GAX_HOME>/webapp/WEB-INF/lib/gax-iwd.jar** file

- **<GAX_HOME>/plug-ins/gax-iwd.jar** file
- **<GAX_HOME>/webapp/plugins/gax-iwd** directory

### Important

Before you perform the **installation of the new version of iWD GAX Plug-in** or after you perform the **GAX Plug-in removal procedure** , you need to make sure that following files have been deleted:

- **<GAX_HOME>/webapp/WEB-INF/lib/gax-iwd-lp-<*lang-build*>.jar** (for example, **gax-iwd-lp-de-de-9.0.007.01.jar** file)
- **<GAX_HOME>/plug-ins/gax-iwd-lp-<*lang-build*>.jar** (for example, **gax-iwd-lp-de-de-9.0.007.01.jar** file)
- **<GAX_HOME>/webapp/plugins/gax-iwd-lp-<*lang*>** (for example, **gax-iwd-lp-de-de** directory)

## Upgrading Plug-ins

To upgrade any plug-in, you need to follow the instructions above to remove the old plug-in, then install the new one.

# Installing iWD History Node

## Prerequisites

- The environment meets the requirements that are described in *Installation Prerequisites*
- The computer on which the iWD History Node is going to be installed has network access to the computer that is hosting Genesys Configuration Server.
- You have access rights to execute **install.sh** or **setup.exe**, depending on the operating system.

## Creating the Database Schema and DAP

This is a general procedure. Work with your enterprise's database administrator to follow the specific procedure that is required by your database management system and your enterprise policies.

1. Ensure that the database server is running.
2. Log on to the database server's administrative interface (such as Oracle Enterprise Manager).
3. Create a new database user account (for example, `iwdhistorynodeuser`).
4. Create a new database (for example, `iwdhistorynodedb`).
5. Ensure that there is a user who has access to the iWD History Node database, who has the following permissions:
   - CREATE TABLE
   - CREATE INDEX
   - CREATE VIEW
   - CREATE TRIGGER (Oracle)
   - CREATE SEQUENCE (Oracle)
6. Create a Database Access Point (DAP), filling in the usual mandatory settings on the **General** and **DB Info** tabs. This DAP must have a **[jdbc]** section in the configuration application options, with the following parameters:

   Microsoft SQL

   - `driver-class = "com.microsoft.sqlserver.jdbc.SQLServerDriver"`
   - `url = "jdbc:sqlserver://<hostname>:<port>;databaseName=<history_node_DB>;"`
   - `validation-query = "select 1"`

   Oracle

   - `driver-class = "oracle.jdbc.OracleDriver"`

- `url = "jdbc:oracle:thin:@//<hostname>:<port>/<history_node_DB>"`
- `validation-query = "select 1 from dual"`

  Postgres

- `driver-class = "org.postgresql.Driver"`
- `url = "jdbc:postgresql://<hostname>:<port>/<history_node_DB >"`
- `validation-query = "select 1"`

# On Windows

## Prerequisites

- Installation Packages have been installed.

## Purpose

To install the iWD History Node application on the Windows platform. The current procedure assumes that the application already exists in Configuration Server—the required steps are described in iWD History Node Application Definition.

## Procedure

1. Locate and double-click **setup.exe** in the iWD History Node directory of the iWD DVD.
2. The iWD History Node Installation Wizard opens. Click **Next** in the **Welcome** screen.
3. In the **Connection Parameters** to the Configuration Server screen, enter the login details to connect to Genesys Configuration Server and then click **Next**:
   - **Host name**—The host of Genesys Configuration Server.
   - **Port**—The port that is used by Genesys Configuration Server.
   - **User name**—The user name of the Person (or User) as defined in Genesys Configuration Manager or Genesys Administrator.
   - **Password**—The password that is associated with the Person (or User).
4. From the list of available choices, choose the iWD History Node application that you want to install and click **Next**.
5. Choose the destination location for iWD History Node.
6. Click **Next**.
7. In the **Ready to Install** screen, click **Install** to begin the installation of iWD History Node.
8. When installation has been completed, click **Finish**.

**End of procedure**

# On UNIX

## Prerequisites

- Installation Packages have been installed.

## Purpose

To install the iWD History Node application on the UNIX platform. The current procedure assumes that the application already exists in Configuration Server—the required steps are described in iWD History Node Application Definition.

## Procedure

1. Locate the install directory and enter **./install.sh**.
2. When the following output is displayed, enter the required information, as indicated at each prompt.

```
************************************************
* Welcome to the Genesys 9.0 Installation Script *
************************************************

Installing iWD History Node, version 9.0.000.xx

Please enter the hostname or press enter for "xx-yy-zzzzzz" =>

Please enter the following information about your Configuration Server:

Configuration Server Hostname =>xx-yy-zzzzzz
Network port =>8888
User name =>default
Password =>

Please choose which application to install:
1 : iwd_history_1
2 : iwd_mgr_1_server
=>1

Press ENTER to confirm /genesys/iwd_history_1 as
the destination directory or enter a new one =>

Extracting tarfile: data.tar.gz to directory: /genesys/iwd_history_1
config/
config/iwd_history.yaml
iwd_history.jar
iwd_history.sh
iwd_history_service.sh
lib/

Installation of iWD History Node, version 9.0.000.xx has completed successfully.
```

## iWD History Node Application Definition

1. Log into Genesys Administrator or GAX and import the iWD History Node Application template from the iWD DVD. Double-check to see whether metadata were correctly imported. This is important for definition of roles in Genesys Administrator. For GAX, importing the IP automatically also imports privileges. In that case, metadata are options since you would manage roles using GAX in those circumstances.
2. Create a new **Application** object based on the iWD History Node Application template.
    1. To begin the create procedure, navigate to **Configuration > Environment > Applications** and click **New**.
    2. On the **General** tab:
        1. Enter a name for the iWD History Node.
        2. Select the application **Template**—This must of type:
            - `Third Party Server` for releases prior to 9.0.005.
            - `Genesys Generic Server` for release 9.0.005+.
        3. **Version** and **Is Application Server** boxes are pre-selected according to the template type.
        4. Enter the name of the **Tenant** for which History Node will be working.
        5. **State Enabled**—If selected, indicates that the object is in regular operating condition and can be used without any restrictions.
    3. On the **Connections** tab:
        1. Add a connection to the Configuration Server. If you have both primary and backup Configuration Servers, add a connection only to the primary.
        2. Add a connection to the History Node database's Database Access Point (DAP).
        3. Add a connection to an Event Logger DAP in one of the following ways:
            Either:
            - Add a connection to an existing Interaction Server Event Logger DAP. You can use either JMS or Kafka Event Logger;

            Or;
            - Create a new Event Logger DAP:

            1. To create a JMS Event Logger DAP:
                1. Import the **iWD_IXN_Logger_JMS_DAP** template from the **{HISTORY NODE INSTALLATION DIR}/event_logger_templates** folder.
                2. Create a new application based on the **iWD_IXN_Logger_JMS_DAP** template.
                3. Redefine required parameters according to specific JMS vendor documentation:
                    - ActiveMQ
                    - IBMMQ
                    - Tibco
            2. To create a Kafka Event Logger DAP:

1. Import the **iWD_IXN_Logger_Kafka_DAP** template from the **{HISTORY NODE INSTALLATION DIR}/event_logger_templates** folder.
2. Create a new application based on the **iWD_IXN_Logger_Kafka_DAP** template.
3. Redefine the following required parameters from the **[kafka-settings]** section in the **Options** tab:
   - **servers** = <comma-separated list of the Kafka broker(s) to connect to initially>
4. Refer to Using Kafka Event Logger with History Node for detailed information.

   This Event Logger must use the same JMS/Kafka instance as Interaction Server Event Logger uses.

3. Add the DAP to the **Connections** tab of the Interaction Server application.

3. Add a Redelivery Policy.

> **Important**
> The order of events must not be broken in case of redelivery or database failure.

- Example for ActiveMQ:
  - Update the JMS DAP configuration object's **jms-provider-url** option under the **logger-settings** section by adding the following parameters (highlighted in **bold**) to the URI:

    ```
    jms-provider-url =
    tcp://<hostname>:<port>?jms.redeliveryPolicy.maximumRedeliveries=-1&jms.redeliveryPolicy.redeliveryDelay=0&jms.redeliveryPolicy.initialRedeliveryDelay=0
    ```

- IBMMQ and Tibco do not require additional redelivery settings. Redelivery Policy should work by default.
- Refer to the vendor specific documentation for other JMS.

4. Add other connection details:

   1. If you need to, specify the port ID and its number that iWD History Node will connect to.
      - Optionally, specify the **Connection Protocol**—`simple` or `addp`
      - Optionally, specify the **Local Timeout** and the **Remote Timeout**—Leave empty. These values are required only if you specified addp in **Connection Protocol**. This value specifies the heartbeat polling interval, measured in seconds, on the client side. This indicates how often the client application sends polling signals to the server application. To enable this functionality, specify any integer as the value.
   2. Specify a Trace Mode—The connection trace mode used between a server and its client.
      - **Trace Is Turned Off**—Select if you do not want either the client or the server application to print ADDP-related messages in its log.
      - **Trace On Client Side**—Select if you want the client application to print ADDP-related messages in its log.
      - **Trace On Server Side**—Select if you want the server application to print ADDP-related messages in its log.

- **Trace On Both Sides**—Select if you want both the client and server applications to print ADDP-related messages in their log.

3. Specify **Transport Protocol Parameters**—Any text, usually key/value pairs, separated by a semicolon (;). This property is application-specific.
4. Specify **Application Parameters**—Any text, usually key/value pairs, separated by a semicolon (;). This property is application-specific.

> **Important**
>
> These connection parameters are effective for Configuration Server and ignored for DAPs.

> **Important**
>
> JDBC and JMS drivers do not require any configuration, but they are not delivered in the iWD software distribution artifacts. However their .jar files must be put into the **{HISTORY NODE INSTALLATION DIR}/lib** directory.

5. The **Ports** tab lists communication ports used by the clients of an application to connect to a server. To support specific high-availability configurations, more than one server can be registered on the same port within the same host. Otherwise, do not assign the port number to any other server on the same host. Click **Add** to add a connection. You need to add two ports:
   - `default`—Provides History Node's main functionality.
   - `admin`—Provides administrative tasks.
6. In the **Application Options** tab in the **[IWD]** section, set the value of these two options:
   - **process-dm**—`true`
   - **process-gtl**—`true`

> **Important**
>
> See also the '**Configuring History Node to Process Single Event Type**' section of the iWD History Node Configuration topic.

7. Ignore the **Options** tab.
8. Click **Save** to save the Application object.

## Creating the Database Schema

To create the History Node database schema, perform the migration steps described in iWD History Node Database Schema Migration to ensure that the database schema is created correctly.

## Installing an iWD History Node Cluster

1. Create an application for every History Node application in the cluster as described in *iWD History Node Application Definition*.
2. Create a new **Application** object based on the Application Cluster template.
   a. To begin the create procedure, navigate to **Configuration > Environment > Applications** and click **New**.
   b. On the **General** tab:
      i. Enter a name for the iWD History Node Cluster.
      ii. Select the application **Template**—This must of type `Application Cluster`.
   c. On the **Connections** tab, add connections to all History Node applications working in this cluster.

## Configuring TLS

> **Important**
> To configure TLS, please refer to the Configuring TLS for iWD guide.

# Installing Runtime Node

This section describes the procedures that are used to install iWD Runtime Node.

## Prerequisites

The iWD Runtime Node must be installed as a standalone Java application server before it can run iWD services. When deploying services for multiple solutions, each solution should have its own dedicated Runtime Node and the installation procedure must be repeated for each of them.

- The environment meets the requirements that are described in Installation Prerequisites.
- The computer on which the iWD Runtime Node is going to be installed has network access to the computer that is hosting Genesys Configuration Server, Interaction Server and Message Server databases.
- You have access rights to execute **install.sh** or **setup.exe**, depending on the operating system.
- For upgrades from 8.5.1 to 9.0.0:
  - iWD Runtime Node is stopped.
  - The previous version of iWD Runtime Node is uninstalled.

> **Important**
> Genesys recommends 12GB memory for the Runtime Node application.

## On Windows

### Purpose

To install the iWD Runtime Node application on the Windows platform.

### Summary

Installation of iWD Runtime Node saves the required database scripts and Kettle files in the working directory. For more information about the database go to the Create the Data Mart Database section. The current procedure assumes that the application already exists in Configuration Server. If not, see the required steps in the iWD Runtime Node Application Definition section.

> **Important**
>
> If you are upgrading, ensure that you have uninstalled the previous version and proceed with the current installation procedure. Also make sure not to skip the Application Definition step, because new versions might deliver new options.

### Procedure

1. Locate and double-click **setup.exe** in the iWD Runtime Node directory of the iWD DVD.
2. The iWD Runtime Node Installation Wizard opens. Click **Next** in the **Welcome** screen.
3. In the **Connection Parameters** to the Configuration Server screen, enter the login details to connect to Genesys Configuration Server and then click **Next**:
   - `Host name`—The host of Genesys Configuration Server
   - `Port`—The port that is used by Genesys Configuration Server
   - `User name`—The user name of the Person (or User) as defined in Genesys Configuration Manager or Genesys Administrator.
   - `Password`—The password that is associated with the Person (or User).
4. Select application from proposed list. Those selection will provide details for rest of dialog.
5. Click **Next**.
6. The installer suggests an installation path. Change it if needed, then accept by clicking **Next'**.
7. In the **Ready to Install** screen, click **Install** to begin the installation of iWD Runtime Node.
8. When installation has been completed, click **Finish**.
9. The installed `JavaServerStarter.ini` file should not require any updates but you can review it.
10. The appropriate Windows service should be created and should appear on the list of Windows services. This allows you to start/stop it.
11. Turn on any of the plug-ins by editing the `plugins.properties` file, if needed.
12. Start the iWD Runtime Node Windows service.

**End of procedure**

## On UNIX

### Purpose

To install the iWD Runtime Node application on the UNIX platform.

## Summary

Installation of iWD Runtime Node saves the required database scripts and Kettle files in the working directory. For more information about the database go to the Create the Data Mart Database section. The current procedure assumes that the application already exists in Configuration Server. If not, see the required steps in the iWD Runtime Node Application Definition section.

> **Important**
>
> If you are upgrading, ensure that you have uninstalled the previous version and proceed with the current installation procedure. Also make sure not to skip the Application Definition step, because new versions might deliver new options.

## Procedure

1. As the iWD Runtime Node user, browse to the `install` directory and enter `./install.sh`.
2. When the following output is displayed, enter the required information, as indicated at each prompt.

```
************************************************
* Welcome to the Genesys 9.0 Installation Script *
************************************************

Installing iWD Runtime Node, version 9.0.XXX.XX

Please enter the hostname or press enter for "HOSTNAME" =>

Please enter the following information about your Configuration Server:

Configuration Server Hostname =>CONFIGURATION_SERVER_HOSTNAME
Network port =>CONFIGURATION_SERVER_PORT
User name =>CONFIGURATION_SERVER_USERNAME
Password =>CONFIGURATION_SERVER_PASSWORD

Please choose which application to install:
1 : iWD_Runtime_Node
=>1

Please enter full path of the destination directory for installation =>/YOUR_PATH/iwd/
YOUR_APP_NAME/

Extracting tarfile: data.tar.gz to directory: /YOUR_PATH/iwd/YOUR_APP_NAME/
...
etl/
etl/plugins/
etl/plugins/all_plugins_example.properties
etl/plugins/plugins.properties
...
iWD_Runtime_Service.sh
iwd_dm.jar
iwd_runtime.sh
...
sql_scripts/
sql_scripts/mssql/
sql_scripts/mssql/iwd_dm_mssql.sql
```

```
sql_scripts/oracle/
sql_scripts/oracle/iwd_dm_oracle.sql

Installation of iWD Runtime Node, version 9.0.XXX.XX has completed successfully.
```

3. Turn on any of the plug-ins by editing the **plugins.properties** file, if needed.
4. Review and make changes, if needed, to the **iwd_runtime.sh** or **iWD_Runtime_Service.sh** file, for example, providing flags to Java. White characters (such as spaces) are not supported in the script variables—for example, `DM_APPL=iWD Runtime Node` is illegal.

> **Important**
>
> Please note that the path must be provided in the application configuration.

5. Start the iWD Runtime Node process.

## iWD Runtime Node Application Definition

1. Log into Genesys Administrator or GAX and import the iWD Runtime Node Application template from the iWD DVD.
2. Create a new **Application** object based on the template. For upgrades from 8.5.1 to 9.0.x you can either create a new application in place of the previous one or simply update the application by using the new template. To begin the Create procedure, navigate to **Configuration > Environment > Applications** and click **New**.
3. On the **General** tab:
   a. Enter your chosen name, for example, `iWDRuntimeSolutionName`. Using spaces is not recommended.
   b. Select the application **Template**—Use type `iWD Runtime Node`.
   c. **Version**, **Tenant** and **Is Application Server** boxes are pre-selected according to the template type.
   d. Fill in fields **Working Directory**, **Command Line** and **Command Line Arguments** per the example below:

| Attribute name | Windows | Unix |
|---|---|---|
| `Working Directory` | C:\Program Files (x86)\GCTI\ iWD Runtime Node | /IWD_APP_PATH/ iWDRuntimeNode/ |
| `Command Line` | JavaServerStarter.exe | /INIT_PATH/ iWD_Runtime_Service.sh |
| `Command Line Arguments` | -host YOUR_CFG_SRV_HOST -port YOUR_CFG_SRV_PORT -app YOUR_APP_NAME -service YOUR_SERVICE_NAME | start |

e. Choose the host where the application will run.

f. **Optional fields**

- **Certificate**—The security certificate value. In Windows, select the certificate value from the list of installed certificates. In UNIX, enter the full path to the **<serial_#>_<host_name>_cert.pem** file.
- **Certificate Description**—An optional description of the Certificate.
- **Certificate Key**—The full path to the **<serial_#>_<host_name>_priv_key.pem** file of the security certificate key. This field is used only if Genesys Security is deployed on UNIX; otherwise this field is empty.
- **State Enabled**— If selected, indicates that the object is in regular operating condition and can be used without any restrictions.
- **Login As Account**— Select a user account that has full access to the relevant Tenant, as well as to Applications, Applications Templates and Switches. These fields are displayed after the application is created.

> **Important**
>
> The Load Config service will need to read certain settings from your Tenant and parent tenant (if the latter exists). For example, it needs to have read permission to all applications involved in the Solution, including Capture Points and their application templates. The Statistics Adapter service will need to update the options in the Stat Server application, as well as create Virtual Queue objects under Multimedia Switch, and it does this through the iWD Runtime Node application that has been configured in Configuration Server. For this reason iWD Runtime Node should be configured so that it can make the required changes to both the Stat Server application and Multimedia Switch on behalf of a user with appropriate security permissions.

7. On the **Connections** tab, add:
   - A Data Mart DAP to point at the database that will be used. It should contain the following parameters:
     - `jdbc / url = "jdbc:sqlserver://<hostname>:<port>;databaseName=<Datamart_DB>"` (example for Microsoft SQL)
   - History Node—The iWD History Node application defined in Installing iWD History Node.
   - Configuration Server—If you have both primary and backup Configuration Servers, add a connection only to the primary.
   - A Configuration Server DAP to point to Configuration Server's database. It should contains the following parameters:
     - `iWD / role = "cfgserver"`.
     - `jdbc / url = "jdbc:sqlserver://<hostname>:<port>;databaseName=<config_server_DB>"` (example for Microsoft SQL).
8. The **Ports** tab lists communication ports used by the clients of an application to connect to a server. To support specific high-availability configurations, more than one server can be registered on the same port within the same host. Otherwise, do not assign the port number to any other server on the same host. Click **Add** to add ports. You need to add two ports:
   - `webservice`—With `http` connection type. This will be used to communicate between the GAX Data Mart Dashboard and iWD Runtime Node.

- `admin`—Provides administrative tasks.

9. On the **Tenants** tab choose only one Tenant where your solution is located.
10. Ignore the **Options** tab.
11. Ignore the **Application Options** tab.
12. Click **Save** to save the Application object.

If you have re-created the iWD Runtime Node application you should go to **GAX > Configuration > Data Mart** section to confirm that the selected application for your solution is correct and has a suitable configuration.

> **Important**
> Any changes made to application settings require iWD Runtime Node restart in order to be applied.

## Create the Data Mart Database

This is a general procedure. Work with your enterprise's database administrator to follow the specific procedure that is required by your database management system and your enterprise policies.

### Prerequisites

- iWD Runtime Node must be installed.
- A Database Access Point (DAP) application with connection type `JDBC` needs to created.

> **Important**
> Remember to set up the port in that DAP to the value of the database server port—for example, for MSSQL the default port is 1433. This number needs to be set up as the port in the DAP configuration.

### Procedure

1. Ensure that the database server is running.
2. Log on to the database server's administrative interface (such as Oracle Enterprise Manager).
3. Create a new database user account (for example, `iwddatamartuser`).
4. Create a new database (for example, `iwddatamartdb`).

5. Ensure that there is a user, who has access to the Data Mart database, who has the following permissions:
   - `CREATE TABLE`
   - `CREATE INDEX`
   - `CREATE VIEW`
   - `CREATE TRIGGER` (Oracle)
   - `CREATE SEQUENCE` (Oracle)
6. The iWD Data Mart database will be initialized automatically the first time the Database Service and Kettle ETL Service are started. If the Database Service's **Auto Sync** option is selected, this initialization is automatic, and the Database Service will also check for updates to the iWD Data Mart database whenever a new version of iWD Data Mart is installed. The **Auto Sync** option will also initialize ETL plug-ins.

> **Important**
>
> You can find the **Auto Sync** option by following this path: **GAX** (with iWD plugin installed) **-> Configuration -> Datamart -> *YOUR_SOLUTION*-> Database**. Any changes to these options require iWD Runtime Node to be restarted take effect.

### Using Data Mart Database with Genesys Historical Reporting

The previous reporting product, Genesys Interactive Insights (GI2), is not supported by iWD 9.0. GI2 has entered End of Life (EOL) phase for all versions. Please refer to the Interactive Insights product page for detailed information.

For information about built-in iWD historical reporting product, please refer to the CX Insights for iWD reports guide.

If you will be using the CX Insights for iWD product for historical reporting based on the iWD Data Mart, you must enable several aggregates that are not enabled by default. Please read Before using the iWD reports for detailed information.

## High Availability for iWD Runtime Node

### Overview

iWD Runtime Node supports High Availability in cold standby mode. That means for every iWD Runtime Node you should configure a redundant application which is manually brought up in case of outage of a given iWD Runtime Node.

### Configuration of a redundant iWD Runtime Node

Create a new Application as described in iWD Runtime Node Application Definition or clone an

existing one. Make sure that, for the new redundant application and the primary one:

- They have the same Connections, Permissions and linked Tenant.
- They have the same Options and Application Options, except for logging options and **[settings] / etl-scripts-directory**.
- The ETL scripts pointed to in option **[settings] / etl-scripts-directory** are the same.

> **Important**
> During deployment on different servers, the locations of ETL scripts and written logs might differ. However, it is important to have the same scripts distributed across these servers.

### Manual switch-over procedure

1. In GAX, navigate to **Configuration > intelligent Workload Distribution > Datamart > [tenant_name] > [solution_name]**.
2. On the **General** tab, change the Application to the corresponding redundant iWD Runtime Node application.
3. Save your changes.
4. Make sure that iWD History Node is listed within the connections of the selected iWD Runtime Node.
5. Start iWD Runtime Node.

## Configuring TLS

> **Important**
> To configure TLS, please refer to the Configuring TLS for iWD guide.

# Installing IWD Manager

## Prerequisites

- The environment meets the requirements that are described in *Installation Prerequisites*
- The computer on which the iWD Manager is going to be installed has network access to the computer that is hosting Genesys Configuration Server. Users of iWD Manager will be authenticated through Genesys Configuration Server.
- You have access rights to execute **install.sh** or **setup.exe**, depending on the operating system.
- For upgrades from 8.5.1 to 9.0.x:
  - The previous version of iWD Manager is uninstalled—the **iwd_manager** directory from the web server is removed.

## On Windows

### Prerequisites

- Installation Packages have been installed.

### Purpose

To install the iWD Manager application on the Windows platform.

### Summary

> **Important**
>
> For Interaction Server synchronization, from release 9.0 of iWD, the functionality previously in the **Configure Ixn Custom Properties** in iWD Manager is relocated to the **Migration** tab in the Solution definition in iWD GAX Plugin (new document). For this reason, adding a connection to IXN Server is no longer mandatory for iWD Manager during its installation phase. It will be added automatically later by the iWD Plugin for GAX.

The current procedure assumes that the application already exists in Configuration Server—the required steps are described in *iWD Manager Application Definition*.

> **Important**
>
> If you are upgrading, ensure that you have uninstalled the previous version and proceed with the current installation procedure. Also make sure not to skip the Application Definition step, because new versions might deliver new options.

### Procedure

1. Locate and double-click **setup.exe** in the iWD Manager directory of, for example, the iWD DVD.
2. The iWD Manager Installation Wizard opens. Click **Next** in the **Welcome** screen.
3. In the **Connection Parameters** to the Configuration Server screen, enter the login details to connect to Genesys Configuration Server and then click **Next**:
   - **Host name**—The host of Genesys Configuration Server
   - **Port**—The port that is used by Genesys Configuration Server
   - **User name**—The user name of the Person (or User) as defined in Genesys Configuration Manager or Genesys Administrator.
   - **Password**—The password that is associated with the Person (or User).
4. From the list of available choices, choose the iWD Manager Server application that you want to install and click **Next**.
5. Choose the destination location for iWD Manager.
6. Click **Next**.
7. Specify the address of the network interface where iWD Manager listens to connections, and the iWD Manager application name. Click **Next**.
8. In the **Ready to Install** screen, click **Install** to begin the installation of iWD Manager.
9. When installation has been completed, click **Finish**.
10. Perform any optional steps or install localization if needed.

**End of procedure**

## On UNIX

### Prerequisites

- Installation Packages have been installed.

## Purpose

To install the iWD Manager application on the UNIX platform.

## Summary

> **Important**
>
> For Interaction Server synchronization, from release 9.0 of iWD, the functionality previously in the **Configure Ixn Custom Properties** in iWD Manager is relocated to the **Migration** tab in the Solution definition in iWD GAX Plugin (new document). For this reason, adding a connection to IXN Server is no longer mandatory for iWD Manager during its installation phase. It will be added automatically later by the iWD Plugin for GAX.

The current procedure assumes that the application already exists in Configuration Server—the required steps are described in *iWD Manager Application Definition*.

> **Important**
>
> If you are upgrading, ensure that you have uninstalled the previous version and proceed with the current installation procedure. Also make sure not to skip the Application Definition step, because new versions might deliver new options.

## Procedure

1. Locate the install directory and enter `./install.sh`.
2. When the following output is displayed, enter the required information, as indicated at each prompt.

```
************************************************
* Welcome to the Genesys 9.0 Installation Script *
************************************************

Installing iWD Manager, version 9.X.XXX.XX


Please enter the hostname or press enter for "<hostname>" =>

Please enter the following information about your Configuration Server:

Configuration Server Hostname =>X.X.X.X
Network port =>2020
User name =><username>
Password =><password>

Please choose which application to install:
```

```
1 : iWD_Manager_Server
 =>1

Press ENTER to confirm "0.0.0.0" as
the Tomcat network interface or enter a new one =>X.X.X.X

Please enter the iWD Manager Application Name =>iWD_Manager

Press ENTER to confirm /home/genesys/GCTI/iWDManager as
the destination directory or enter a new one =>/home/genesys/GCTI/iWDManager

Extracting tarfile: data.tar.gz to directory: /home/genesys/GCTI/iWDManager
...
webapp/
...
webapp/WEB-INF/
webapp/WEB-INF/application.properties
webapp/WEB-INF/web.xml
webapp/WEB-INF/faces-config.xml
webapp/WEB-INF/lib/
webapp/WEB-INF/lib/hibernate.jar
webapp/WEB-INF/lib/packagedstatisticsdeprecated.jar
webapp/WEB-INF/lib/commons-lang.jar
webapp/WEB-INF/lib/commons-logging.jar
...
webapp/META-INF/MANIFEST.MF

Installation of iWD Manager, version 9.X.XXX.XX has completed successfully.
```

3. Perform any optional steps or install localization if needed.

## iWD Manager Application Definition

1. Log into Genesys Administrator or GAX and import the iWD Manager Server and iWD Manager Application templates from the iWD DVD. Double-check to see whether metadata were correctly imported.
2. Import the iWD Manager IP into GAX to import privileges. This is important for the definition of roles in Genesys Administrator.
3. Create a new **Application** object based on the iWD Manager Application template.
    1. To begin create procedure navigate to **Configuration > Environment > Applications** and click **New**.
    2. On the **General** tab:
        1. Enter a name for the iWD Manager.
        2. Select the application **Template**—This must of type `iWD Manager`.
        3. **Version**, **Tenant** and **Is Application Server** boxes are pre-selected according to the template type.
        4. **State Enabled**—If selected, indicates that the object is in regular operating condition and can be used without any restrictions.
    3. On the **Connections** tab, add the connections to the Configuration Server. If you need to:

1. Add the **Port ID** on the Configuration Server that iWD Manager will connect to.
2. Specify the **Connection Protocol**: `simple` or `addp`.
3. Specify the **Local Timeout** and the **Remote Timeout**—These values are required only if you specified addp in **Connection Protocol**. This value specifies the heartbeat polling interval, measured in seconds, on a client side. This indicates how often the client application sends polling signals to the server application. To enable this functionality, specify any integer as the value.
4. Specify a Trace Mode—The connection trace mode used between a server and its client.
   - `Trace Is Turned Off`—Select if you do not want either the client or the server application to print ADDP-related messages in its log.
   - `Trace On Client Side`—Select if you want the client application to print ADDP-related messages in its log.
   - `Trace On Server Side`—Select if you want the server application to print ADDP-related messages in its log.
   - `Trace On Both Sides`—Select if you want both the client and server applications to print ADDP-related messages in their log.
5. Specify **Transport Protocol Parameters**—Any text, usually key/value pairs, separated by a semicolon (;). This property is application-specific.
6. Specify **Application Parameters**—Any text, usually key=value pairs, separated by a semicolon (;). This property is application-specific.

4. Ignore the **Application Options** tab.
5. Click **Save** to save the Application object.

4. Create a new **Application** object based on the iWD Manager Server template.
   1. To begin the create procedure navigate to **Configuration > Environment > Applications** and click **New**.
   2. On the **General** tab:
      1. Enter a name for the iWD Manager Server.
      2. Select the application **Template**—This must be of type:
         - `Third Party Server` for releases prior to 9.0.005.
         - `Genesys Generic Server` for release 9.0.005+.
      3. **Version**, **Tenant** and **Is Application Server** boxes are pre-selected according to the template type.
      4. **State Enabled**—If selected, indicates that the object is in regular operating condition and can be used without any restrictions.
   3. On the **Connections** tab:
      1. Add the connection to the Configuration Server. If you have both primary and backup Configuration Servers, add a connection only to the primary.
      2. Add the connection to the UCS.
      3. (Optional) Add the connection to the Message Server if you plan to use the Centralized Logging feature.

> **Important**
>
> Connections to Interaction Server and iWD History Node should be added to the iWD Manager Server application object by the iWD Plugin for GAX automatically after the Business Structure is saved. See Interaction Server Settings and History Node Settings for more details.

4. On the **Ports** tab add the port where iWD Manager REST server will listen for connections. The **Ports** tab lists communication ports used by the clients of an application to connect to a server. To support specific high-availability configurations, more than one server can be registered on the same port within the same host. Otherwise, do not assign the port number to any other server on the same host.
5. On the **Application Options** tab set **[clientApplication] \ name** to the name of iWD Manager application created in the previous step.
6. On the **Application Options** tab configure any additional options if needed. A detailed description of these options can be found here.
7. Click **Save** to save the Application object.

# Configuring High Availability

## Purpose

iWD Manager is a web application and its multiple instances can be deployed in a load-balanced environment to support High Availability.

## Procedure

1. Create an application for every web server of iWD Manager as described in iWD Manager Application Definition. Please refer to the second part starting from step 4 ("Create a new Application object based on the iWD Manager Server template") and use the same value for **[clientApplication] \ name** option for all the Server applications.
2. Deploy an IP for each of the Server applications created in step 1.
3. Set up load-balancing software against the configured web servers of iWD Manager. iWD Manager supports any third-party load balancer that supports sticky sessions. You should configure session affinity (sticky sessions) based on JSESSIONID. The following is a sample configuration for Apache httpd deployed against two iWD Manager servers running on the same host:

```
<Proxy balancer://iwd_mgr_balancer>
BalancerMember http://localhost:9190/iwd_manager lbset=0
BalancerMember http://localhost:9290/iwd_manager lbset=1
ProxySet lbmethod=bybusyness
</Proxy>
RewriteEngine On
RewriteRule ^/iwd_manager/(.*) balancer://iwd_mgr_balancer/$1 [P] stickysession=JSESSIONID
ProxyPassReverse "/iwd_manager" "balancer://iwd_mgr_balancer"
```

# Configuring TLS

> **Important**
> To configure TLS, please refer to the Configuring TLS for iWD guide.

# Integration with Genesys Rules System and Universal Contact Server

## Installing Genesys Rules System for Use with iWD

For production deployments, install the Genesys Rules System web applications (Genesys Rules Engine and the Genesys Rules Authoring Tool) on a dedicated application server instance that is separate from the application server instance that is dedicated to the iWD web applications.

> **Important**
>
> To install Genesys Rules System for use with iWD, complete steps 1 to 6 of the **GRS Installation Task Summary**. Step 7 of the Task Summary is not required. For iWD installations, you will create the business structure in the iWD GAX Plugin.

## Creating the Connection to Interaction Server

Create a connection by using the **Connections** tab in the Interaction Server Application to the Genesys Rules Engine application. If GRE is configured as a cluster, add the **GRE_Cluster** application on the **Connections** tab instead of the Genesys Rules Engine application.

## Configuring Universal Contact Server (UCS)

Configuring UCS is optional. However, if you provide any UCS name during the setup process then a dedicated mapping is created inside the `Iwd_Esp_List` transaction object content—and if iWD Manager does not have that UCS application in its connections, an alarm will be generated.

## GRE Cluster Additional Steps

If you want to use GRE clusters you must correct the transaction object content of `Iwd_Esp_List`. Instead of using a particular GRE application name you need to add the GRE cluster application name attached to your solution. Then this name will be used by the iWD Business Process—for example, by the sub-strategy `invoke_GRE`.

# Enabling/Disabling ADDP Connections

1. After installation, navigate in GAX to **Configuration > Environment > Applications**.
2. Locate the iWD Manager or Runtime Node in the list and open it.
3. Select the **Connections** tab and then select the server connection you want to change.
4. Open it and select addp from the **Connection Protocol** drop-down options.

# Creating the Tenant in GAX

## Procedure

1. Navigate to **GAX -> Configuration -> Environment > Tenants**.
2. Click **New** to open a configuration page. This has three tabs: **General, Options** and **iWD Attributes**. Two other tabs, **Permissions**, **Dependencies** appear when the Tenant is saved.
3. Enter the following information. For some fields, you can either enter the name of a value or click **Browse** to select a value from a list:

   **General tab**

   - **Name**—The name of the Tenant. You must specify a value for this property, and that value must be unique within the Configuration Database.
   - **Password**—A password that must be used to access this Tenant.
   - **Confirm Password**—A confirmation of the password.
   - **Parent Tenant**—The parent Tenant of this Tenant. By default, the parent Tenant is the Tenant in which you are creating the new Tenant. If you change this field, the new Tenant will be created as a new child Tenant under the specified parent Tenant. To subsequently change the parent Tenant, refer to the Structure tab, above.
   - **Chargeable Number**—The account number to which activities for this Tenant are charged, for cost-tracking purposes.
   - **Default Contract**—The default cost contract applied to resources of this Tenant. For more information, refer to the Routing Solutions chapter of the Universal Routing 8.0 Routing Application Configuration Guide.
   - **State Enabled**—If selected, indicates that the object is in regular operating condition and can be used without any restrictions.

4. Ignore the **Options tab**.
5. Configure user permissions for this Tenant.
6. Configure any Dependencies for this Tenant.
7. Configure the iWD Attributes for this Tenant.
   - **ID**—The Tenant's runtime ID, generated automatically.
   - Description of the tenant.
   - **Social Messaging Enabled**—Check to enable social engagement integration for this Tenant.
   - **Rule Authoring Tool URL**—The URL of the Genesys Rules Authoring Tool for this iWD Tenant.
   - **Current Configuration Version**—Contains the version of iWD configuration (Filters, Media Icons, Account Settings) which is applied for the current tenant.
   - **Actual Configuration Version**—Contains the latest available configuration version.

- Click the "Update Configuration" button to update the Tenant configuration to the actual version. Usually this button is disabled, and updates are automatically applied during the login into GAX if the user has sufficient permissions. But during the process of creating a new tenant, users must invoke the update explicitly.
- Click **Inventory Report** to print to screen a complete view of the Tenant hierarchy, including Solutions, Departments and Processes.
- Add any Custom Tenant attributes by clicking **Add** and filling in the **Name Type and Value** table.

8. Save the Tenant.
9. Open the Tenant again, go to the **iWD Attributes** tab and click the **Update Configuration** button to create the default set of iWD Manager filters.
10. Add the created Tenant to the Genesys applications used. To do this:
    a. Open the configuration object of the application that you have deployed and configured to which connections will be made.
    b. Open the **Tenants** tab.
    c. Add the created tenant.
    d. Save changes to the configuration object.

> **Important**
>
> If you are not logged in as the default User, or are not a member of the **SuperAdministrators** Access Group, you must have special permissions and role privileges to create a Tenant. Refer to the **Genesys 8.5 Security Deployment Guide for details about the security requirements for creating a Tenant.**

## Next Steps

Configure your IWD Solution manually by following the procedure **here**. Create as many iWD solutions as are required for your business needs. Each iWD solution requires its own dedicated Stat Server. For each iWD Solution that you you create, configure iWD manually and install the Stat Server Java Extension.

> **Important**
>
> It is recommended that you do not create any Solutions and Services under the System Tenant. You should do so under a managed Tenant.

# Working with Integrated Capture Points

Capture point (CP) functionality allows iWD to create new tasks, based on data coming from an enterprise application or *source system*. CPs also enable existing tasks to be canceled, completed, held/resumed, restarted, or modified. CP functionality is integrated into Interaction Server.

Genesys recommends that you use the Integrated CPs (ICPs). Because iWD no longer supports legacy capture points, you will need to migrate to using ICPs. See the Moving from Legacy to ICPs section on this page.

> **Important**
> See the **iWD Web topics** for information regarding capture points for iWD Web.

## Installing iWD ICPs

Installation procedures for ICPs are described in the eServices Integrated Capture Points 8.1 Guide.

> **Important**
> Remember that legacy capture points are no longer supported in iWD.

The prerequisites and other information will differ in the eServices 8.1 Deployment Guide, depending on the ICP being used. However, for all ICPs there are two common installation steps:

- Creating the Capture Point application.
- Configuring Capture Point iWD attributes.

## Creating a Generic Capture Point

The procedures in this section are applicable to all types of Capture Point. Differences in configuration between pre-8.5 ICPs are mentioned specifically in the procedures are documented in the e-Service Capture Point Guide.

### Creating the Capture Point application

**Purpose**

- The Capture Point functionality is built within Interaction Server 8.1, which means that there is no separate installation package for any type of Capture Points. An `Application` object for the Capture Point must be configured in GAX. One `Application` must be configured for each instance of the Capture Point. Interaction Server supports multiple capture points.

### Prerequisites

- Interaction Server must be installed.
- iWD 9.0 must be installed.

### Start

1. Login to GAX.
2. Navigate to Environment > Applications.
3. Create a new `Application` object based on the chosen Capture Point template. The `CapturePointId` will be automatically set to the name of the Capture Point application as configured in GAX. In iWD compatibility mode, it will also be saved as the `IWD_capturePointId` property in user data. When the Capture Point is configured, the Capture Point ID must be the same as the application name in order to ensure accurate events history reporting and accurate filtering. (The Capture Point *Name* can be anything).

   > **Important**
   >
   > The name of the Capture Point **Application** must start with a letter, contain only alpha-numeric characters and underscores, and cannot be longer than 16 characters and cannot contain spaces.

4. Because the Capture Point is integrated with Interaction Server, the Host and Port information is taken from Interaction Server (which must be listed as a connection on the **Connections** tab). Host and Port information comes from Interaction Server.
5. Add a connection to Interaction Server. Multiple Capture Point **Application** objects can connect to the same Interaction Server.
6. On the **Ports** tab, add any additional ports required for connection to Interaction Server.
7. On the **Tenants** tab, add the relevant Tenant.
8. Ignore the **Options** tab.
9. Ignore the **Application Options** tab.
10. On the **iWD Attributes** tab, select the **Solution** from the drop-down list and add a **Description**. The runtime **ID** of the capture point is generated by GAX to a default initial value and it can be changed. Interaction queues configured in the Solution are copied to the appropriate Capture Point's options. If the Solution has defined incorrect or disallowed queue names, they should be fixed in the Solution configuration first. Otherwise the Capture Point object cannot be saved.
11. Save the `Application` object.
12. When configuration is complete, click Save.

### End

# Moving from Legacy to ICPs

## Mapping Native iWD Task Actions to Native IXS Message Operations

This section is intended to help you understand how the native Interaction Server (IXS) XML message operations compare to the native iWD XML message operations. The information provided here is mainly derived from the *eServices 8.1 User's Guide.*

This page shows how the operations you would specify in the iWD message format correspond, or translate, to the operations you would specify in the IXS message format.

| Task action | iWD XML message operation | Interaction Server message operation |
|---|---|---|
| Create a new task/interaction in iWD | `CreateTask` | `<interaction operation="submit">` |
| Get information about a task/interaction in iWD | `GetTaskInfo` | `<interaction operation="getinfo">` |
| Update a task/interaction in iWD | `UpdateTask` | `<interaction operation="update">` |
| Complete a task/interaction in iWD | `CompleteTask` | `<interaction operation="update">` |
| Hold a task/interaction in iWD | `HoldTask` | `<interaction operation="hold">` |
| Resume a held task/interaction in iWD | `ResumeTask` | `<interaction operation="resume">` |
| Restart a task/interaction in iWD | `RestartTask` | `<interaction operation="update">` |
| Cancel a task/interaction in iWD | `CancelTask` | `<interaction operation="update">` |

# Creating a New Task or Interaction

When you are creating a new task/interaction by using the `<interaction operation="submit">` operation to specify the `all initial` properties of the interaction, use the `properties` element, which is a direct child of the `interaction` element. Alternatively, you can provide default values for any interaction properties, as part of the Integrated Capture Point `Application` configuration. See the descriptions of the default-values sections (which are repeated for each capture type) in the *eServices 8.1 Reference Manual.*

Configuration of some properties is mandatory to ensure that iWD behaves as expected. Also, in Task Attribute Mapping in the columnInteraction Attached Data Key, the names of the properties are described as you would refer to them in this message. You can cross-reference this with the information in the tables titled Translation Table for Known Attributes-Inbound and Translation Table for Known Attributes-Outbound in the *eServices 8.1 User's Guide.*

## Updating, Completing, Canceling, or Restarting a Task/Interaction

Use the <interaction operation="update"> operation when you are updating, completing, canceling, or restarting a task/interaction. Use the properties element, which is a direct child of the interaction element, to specify which properties to update.

In the special cases, when you are completing, canceling, or restarting a task/interaction, specify the Queue property, to move the task/interaction to the iWD_Completed/iwd_bp_comp.Main.iWD_Completedqueues, iWD_Canceled/iwd_bp_comp.Main.iWD_Canceled queues, or the iWD_New/iwd_bp_comp.Main.iWD_New queues, respectively.

> **Important**
>
> When you use the iWD_Completed/iwd_bp_comp.Main.iWD_Completed, iWD_Canceled/iwd_bp_comp.Main.iWD_Canceled and iWD_New/iwd_bp_comp.Main.iWD_New queue names, it is assumed that you are using the out-of-the-box business process that is provided with iWD (IWDBP). If you have modified this business process to add/change queue names, you will need to consider this when you are specifying the Queue property for the update action.

### Differences in Task Restart Process

Restarting a held task differs for legacy CPs and integrated CPs. With legacy CPs, if a held task is restarted, its state changes to New and it is processed again as a new task. However, for integrated CPs, after a task is restarted, it must first be resumed before processing continues. If the task is not resumed, then it stays in the iWD_New queue as held.

## Mandatory Interaction Properties

When you submit a new interaction to Interaction Server (<interaction operation="submit">), the following properties are mandatory. Therefore, they are also mandatory for the Integrated Capture Points:

- InteractionType
- InteractionSubtype
- MediaType
- Queue

- TenantID

> **Important**
>
> In this case, TenantID is the Genesys tenant ID, not the ID of the managed tenant that is created in the iWD configuration.

You do not need to specify these properties in the XML message, because several of them are normally not relevant to iWD. There is an configuration option for the Capture Point Application template, called default-values (for a description, see the *eServices 8.1 Reference Guide*), which enables you to enter default values for any of these properties (as well as any other interaction properties). Therefore, if the default values for these five attributes/properties are specified in the options, then the message itself need not contain any other properties for Interaction Server to accept and process it.

The Queue property is an exception, because you can configure the default queue as part of the business process configuration by using the endpoints section.

In addition to these mandatory properties, three other properties, specific to iWD interactions, are used to ensure that interactions are accounted for correctly by iWD Data Mart and are displayed properly through the Global Task List. The properties are:

- iWD_TenantId
- iWD_SolutionId
- iWD_CapturePointId

The iWD_CapturePointId property will be equivalent to the name of your Capture Point Application in Configuration Server. See the procedure, “Creating the capture point application” in the *eServices 8.1 Deployment Guide*.

When you deploy the Integrated Capture Point, you must also set the ID and Solution on the iWD Attributes tab.

The iWD_SolutionId and the iWD_TenantId properties are the ID of the Solution and the Tenant, respectively, under which the capture point application has been created in GAX. Similar to the mandatory interaction properties, these three additional properties are set automatically by the iWD GAX Plugin when the iWD Attributes tab is completed.

## More Information About CPs

For a description of some optional ways to format the information inside the <properties> element, see the section, “Properties Element” in the *eServices 8.1 User's Guide.*

For additional information about other elements, such as changed, deleted, reason, actor, and party, and about how responses to capture point requests are processed, see Chapter 9 in the *eServices 8.1 User's Guide.*

By default, when you specify any interaction property that is not considered a core property of the Interaction Server interaction data model, or that is not a core or extended attribute in terms of the iWD data model, the value of that property will be stored in a BLOB in the interactions table of the Interaction Server database. If you need to use this custom property in any Queue Views, or to use filtering in the Global Task List, see the **Working with Task Attributes and Interaction Properties** topic.

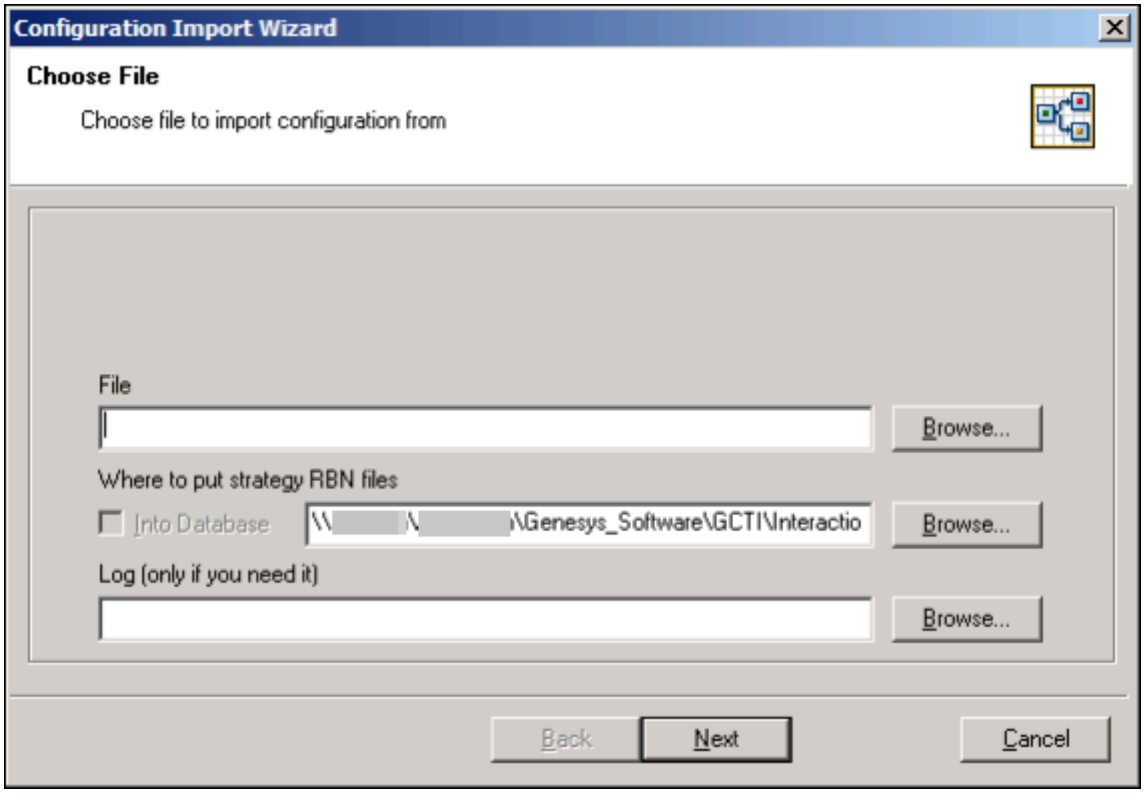
# Manual Installation of IWDBPs

## Manual Installation of iWDBP for IRD

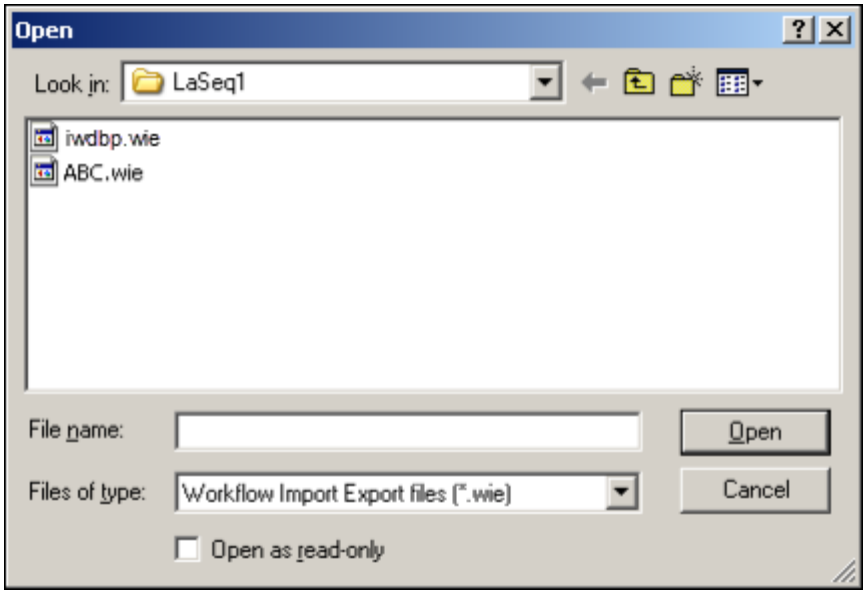
The business process for iWD is provided as a **.wie** file that can be imported manually through the Genesys Interaction Routing Designer. The iWD business process **.wie** files are saved to your file system when the iWD Manager Installation Package is installed.

### Procedure

1. In the IRD main window, click **Interaction Design**.
2. Click **Business Processes**. The existing Business Processes are listed for selection.
3. Double-click a Business Process to open the **Interaction Design** window.
4. From the **File** menu, select **Import**. The **Configuration Import Wizard Choose File** page appears.



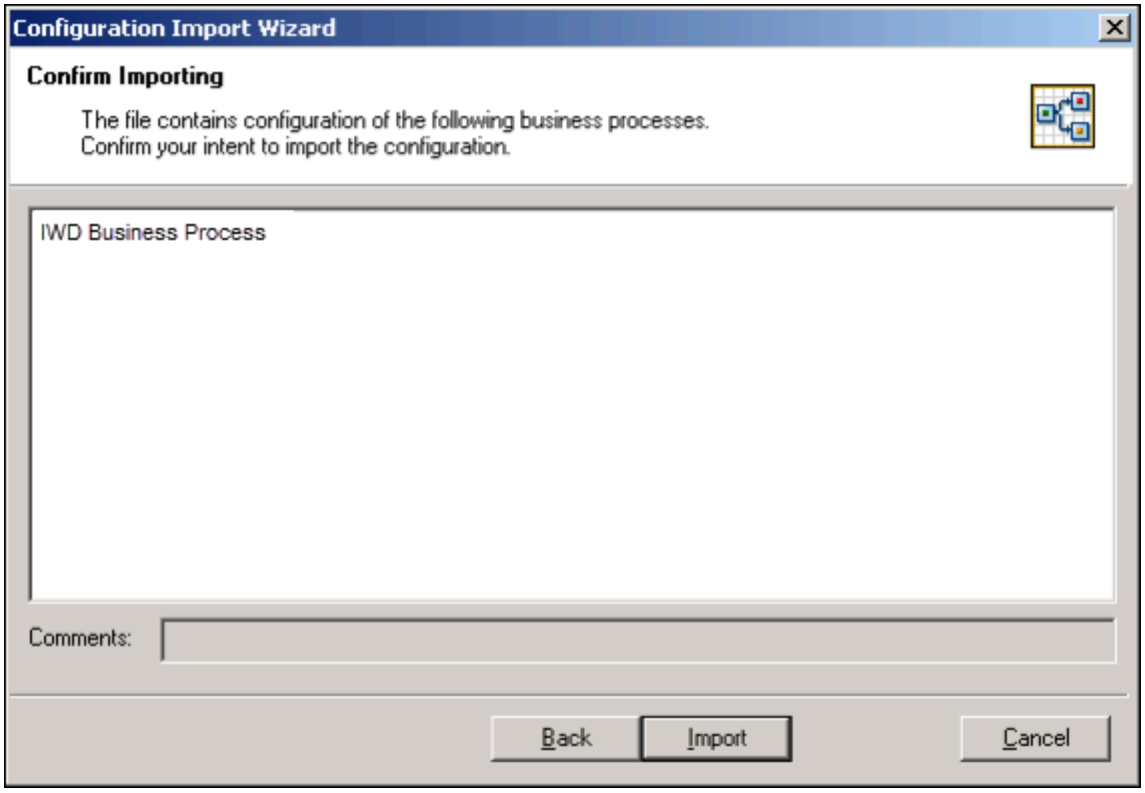

5. Next to the **File** text box, click **Browse**. The **Open** dialog box appears.




6. Locate the **iwdbp.wie** file, and double-click it. The **Open** dialog box closes. IRD inserts the selected directory path and file name into the **File** text box in the Configuration Import Wizard.
7. Under **Where to put strategy RBN files**, do one of the following:
    - Click **Browse**, and then locate the folder that will hold the graphical portion of strategies in the Business Process (strategy ***.rbn** files, which consume more space than the script portion of strategies).
    - Select the **Into Database** checkbox (if it is enabled), to save ***.rbn** files in a database.

    **Notes**:

    - In order for the **Into Database** checkbox to be enabled, a Database Access Point must appear in the **Connections** tab of the IRD Application in Configuration Manager.
    - The database used is the Configuration Server database. The table name is **ird_strategies**.
    - If security is a consideration, you may want to store the **.rbn** files in a database. For example, if you are a Service Provider, you might not want your subscribers to have access to your corporate drives. In this case, saving to the database in the recommended method.
    - If you want to save the details of the import operation to a log file, click **Browse** next to the **Log** text box (unless the log file is already selected), and then locate or create the log file.
8. Under **Log** (only if you need it), you have the option of saving business process import details in a log file. If you wish to do this, click the **Browse** button and locate/name the log file.
9. Click **Next**. The Configuration Wizard Confirm Importing page appears.




10. If you are satisfied with your entries, click **Import**; otherwise, click **Back** to return to the **Configuration Import Wizard Choose File** page, and repeat Steps 5-9.
    After you click **Import**, the **Configuration Import** dialog box appears. A progress bar shows the progress of the import.
11. Do one of the following:
    - To update your configuration settings immediately after the update, click **Refresh**.
    - Click **Close**; then at any time, you can update your configuration settings by selecting **Refresh** from the **File** menu.
12. Respond to any messages that appear.

## Manual Installation of iWDBP for Composer

Import the Composer IWDBP project template (**iwd_bp_comp**) which is provided with the iWD software distribution artifacts here:

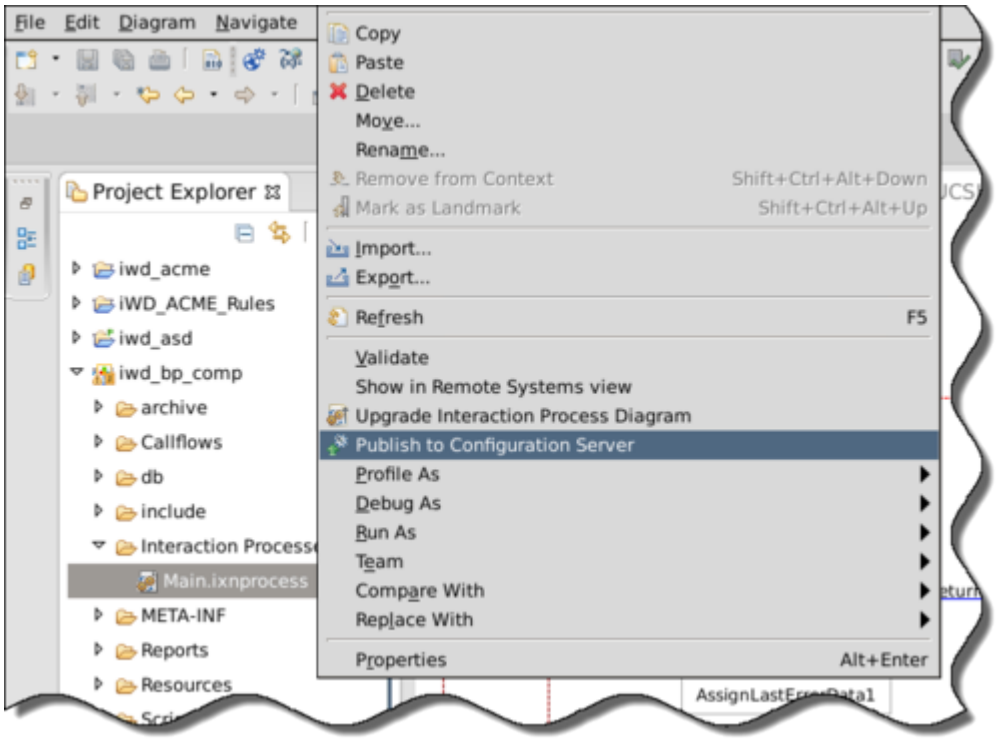
- ***<iWD Manager Installation Directory>*/config/iwd_bp_comp**.

## Prerequisites

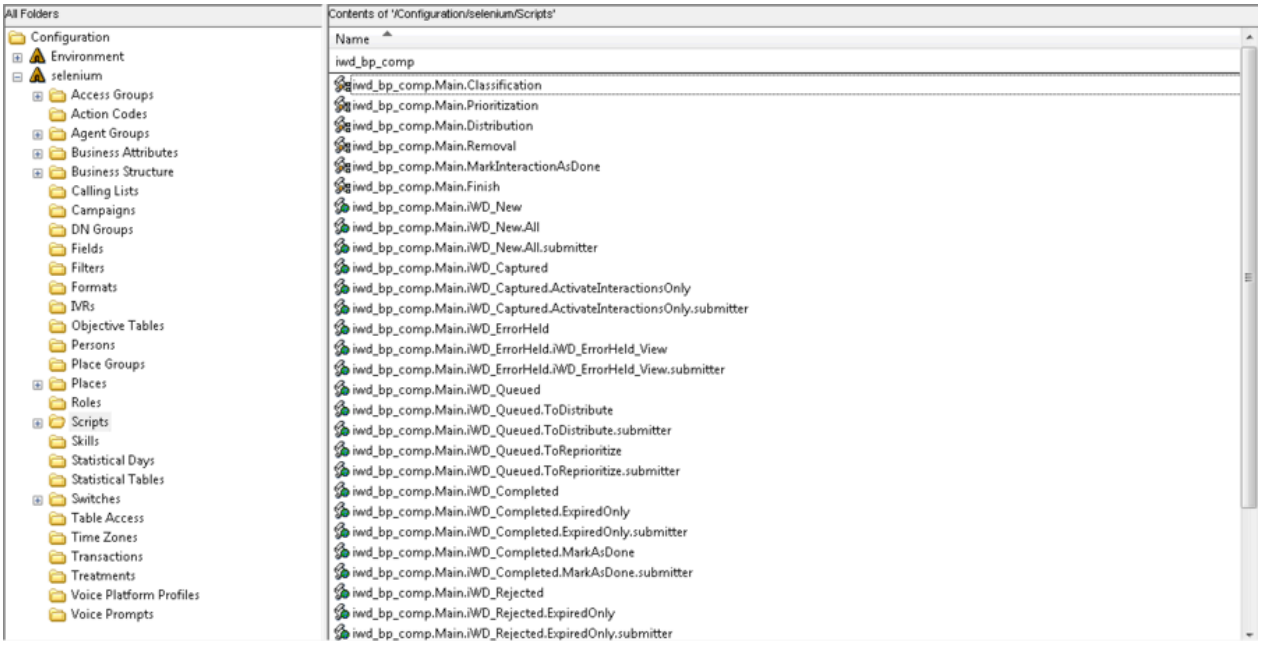
Configure your Web Application Server by following the application server requirements guidelines **here** (new document).

## Procedure

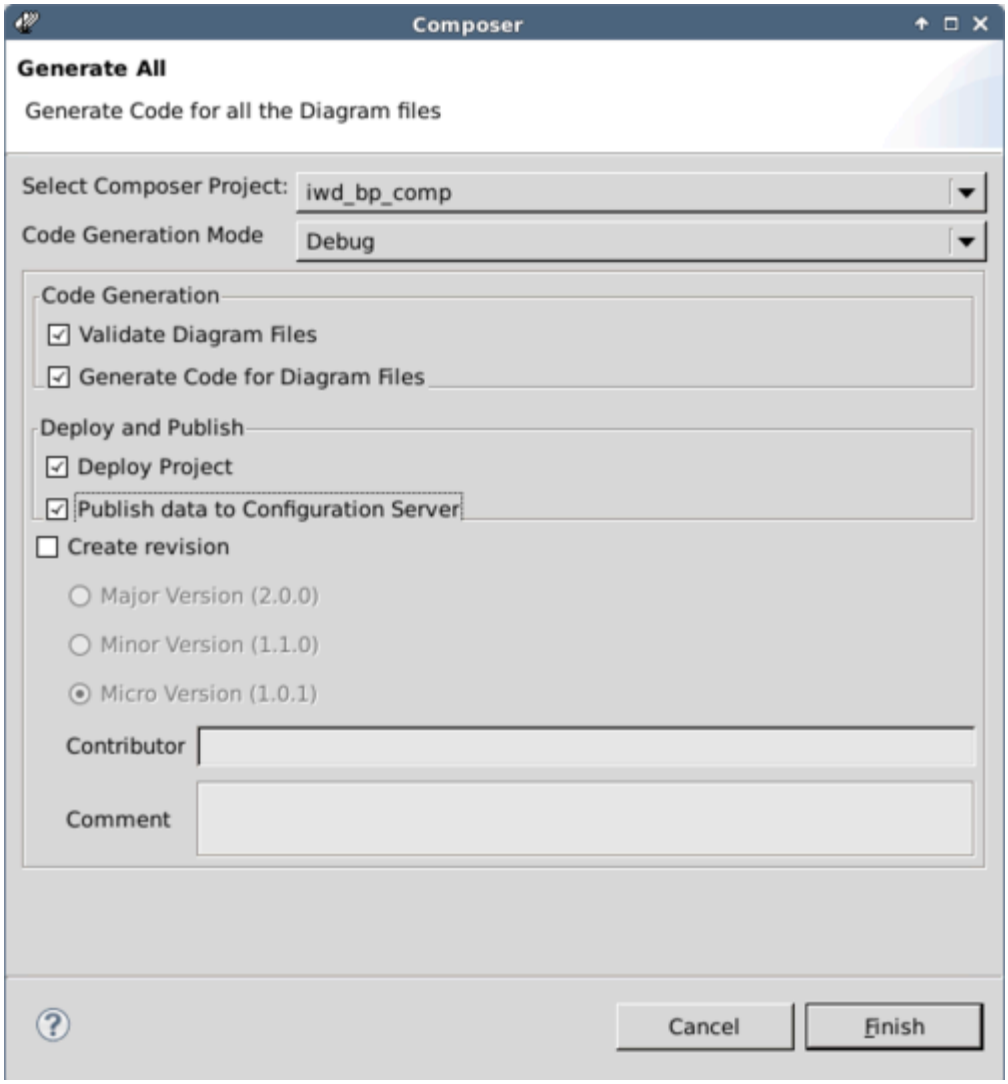
1. Run Composer (or Eclipse with the Composer plug-in).
2. Switch to **Composer Design** or **Composer** perspective.
3. Select **File > Import > General -> Existing Project into Workspace**.
4. Select the **iwd_bp_comp** template and click Next.
5. Select the Project locale and click **Finish**.
6. Connect to Configuration Server.
7. Select the Tenant that will be used for the business process.
8. Publish the Composer configuration objects to Configuration Server. Do this by right-clicking on the **Interaction Processes/Main.ixnprocess** script, then clicking **Publish to Configuration Server**.



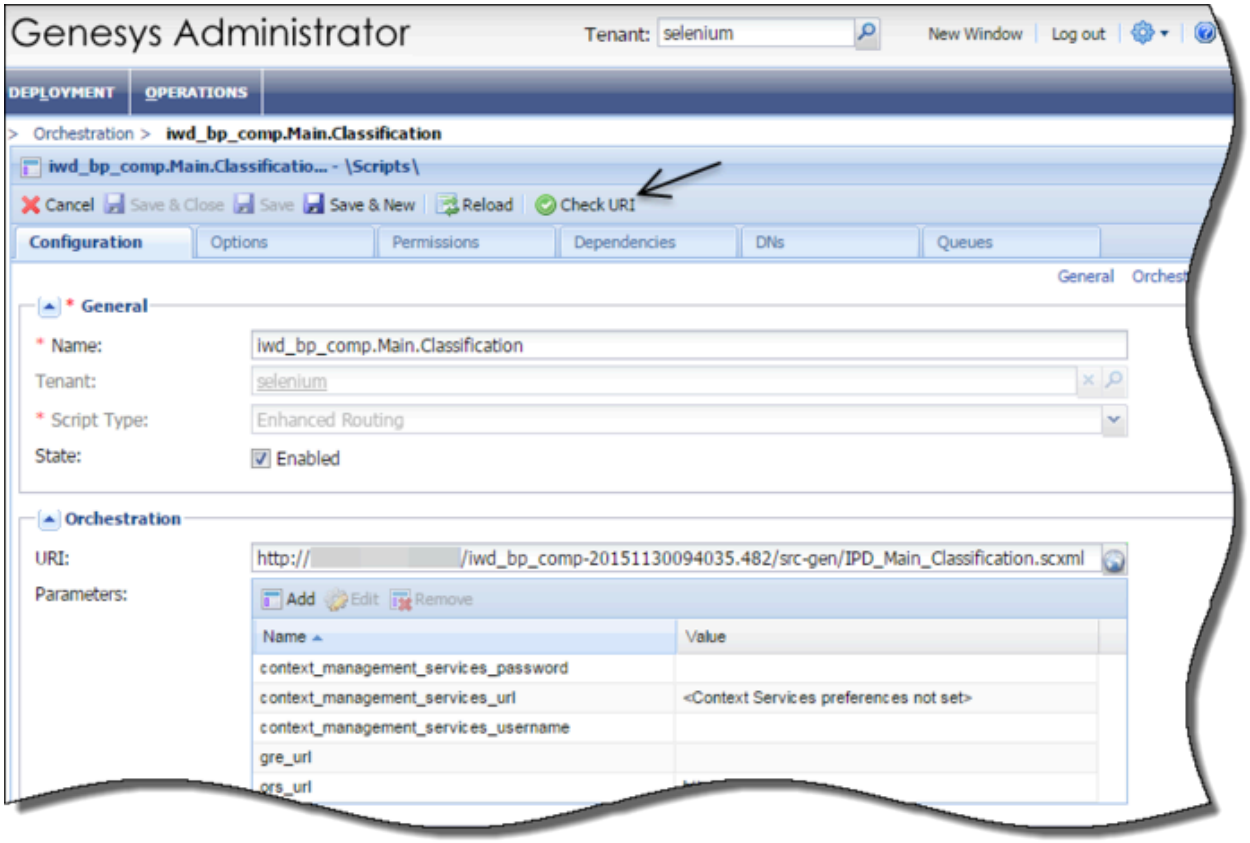

9. When the project is successfully published to Configuration Server the following objects will be created under the ***<selected tenant>*** **Scripts** folder on Configuration Server:




10. Composer can show warnings in Queue Interaction blocks. In these blocks select **Properties->Destination->Interaction Queue Name** even if it is already set and save changes.
11. Right-click the **iwd_bp_comp** project and click **Generate All** to display the **Generate All** dialog.




12. Check the **Deploy Project** and **Publish data to Configuration Server** checkboxes and click **Finish**. This deploys the **iwd_bp_comp** application to Tomcat and generates ***scxml** files that can be accessed via http.
13. To verify that the URL to the generated files is valid, log in to Genesys Administrator.
14. Navigate to ***<Used tenant>*** **-> Provisioning -> Routing/eServices -> Orchestration ->** `iwd_bp_comp.Main.Classification.Classification`.




15. Click **Check URI** to verify whether the Composer-generated URL is valid.

# Configuring iWD Manually

> **Important**
> iWD Setup Utility is not supported in iWD 9.0.

## Procedure

1. Log in to GAX.
2. Go to **Administration >Installation Packages** and click **New**.
3. Check **Installation Package Upload (includes templates)** and click **Next**.
4. Click **Upload**, select the iWD Manager installation package and click **Finish**. Upload can take a few minutes. When it is completed you will see the uploaded iWD Manager installation package.
5. Go to **Configuration > Accounts > Capacity Rule** and create `iwd_capacity_rule` (that is, one interaction of any type) in the tenant's **Script** folder.
6. Go to **Configuration > Switching > Places** and create `iwd_place` for the `iwd_admin` user.
7. Go to **Configuration > Accounts > Persons** and create the `iwd_admin` user . In the `iwd_admin` user, set the **Default Place** to `iwd_place` and the **Capacity Rule** to `iwd_capacity_rule`.
8. Go to **Configuration > Accounts > Access Groups** and create the `iwd_access_group`. Add the `iwd_admin` user in the **Members** tab.
9. Go to **Configuration > Accounts > Agent Groups** and create the `IWD` agent group. Add the `iwd_admin` user to the **Agents** tab.
10. Go to **Configuration > Environment > Tenants** and in the **Permissions** tab assign `Create, Read, Update,` and `Execute` permissions to the `iwd_access_group`.
11. Go to **Configuration > Accounts > Roles** and create the `iwd_role`. In the **Assigned Privileges** tab, check all the required privileges from the `iWD` and `CfgiWDManager` groups. Then add role members (such as Persons or Access Groups) to the `iwd_role`.
12. Go to **Configuration > Environment > Scripts**. Check that "'iWD Manager'" folder exists (create if not) and all permissions for the **EVERYONE** access group are granted for this folder.
13. Go to **Configuration** > **Environment** > **Scripts**. Ensure that **Environment** tenant is selected. Using breadcrumbs, navigate to one level above, that is from **Home** > **Scripts** > **Scripts** to **Home** > **Scripts**. Create a new item with name **iWD Audit** by clicking **More** and then **New Configuration Unit**. Once saved, select it, and click **Edit**. Go to the **Permissions** tab, add **SYSTEM** user (the one that is chosen to **Login As** by the GAX application object) and grant the `Create`, `Read`, `Execute`, `Delete`, and `Propagate` permissions.
14. Go to **Configuration > Routing/Digital > Transactions** and create the `Iwd_Package_List` object. Set the **Name** and **Alias** to `Iwd_Package_List`. In the **Options** tab create:
    a. A section called `RulePackageList`.

b. A key called <Solution runtime id>, with the value <GRAT package name>. For multiple rule packages, specify each Package Name separated by a comma—for example; <GRAT package1>, <GRAT package2>, <GRAT package3>

c. A **Transaction Type** with the value list.

4. Go to **Configuration > Routing/Digital > Transactions** and create Iwd_Esp_List object. Set the **Name** and **Alias** to Iwd_Esp_List. In the **Options** tab create:

a. A section called GREServerList.

b. A key called <Solution runtime id> with the value <GRE application name>.

c. A **Transaction Type** with the value list.

4. (Optional) If UCS will be used, create:

a. A section called ContactServerList.

b. A key called <Solution runtime id> with the value <UCS application name>

3. Import the iWD Business Process using the procedures described **here**.

4. Activate the **Classification**, **Prioritization**, **Distribution**, **Mark Interaction as Done** and **Removal** strategies.

5. Go to **Configuration > Switching > DNs > Switches > <*multimedia switch*> > DNs** and create a virtual queue called iWD. Set its **Number** and **Alias** to iWD and its **Routing Type** to Default.

6. Configure the iWD Stat Extensions:

a. Extract the iWD Stat Extensions IP to a temporary folder (such as iwdstatext).

b. Copy ./iwdstatext/java/ext/* to ../statserver/java/ext.

c. Copy ./iwdstatext/java/lib/* to ../statserver/java/lib.

d. The following table provides possible values of the Stat Server configuration options. All information regarding Stat Server options is described here.

> **Important**
>
> When the Stat Adapter job is run, it rewrites the Stat Server options.

| Section | Key | Value |
|---|---|---|
| statserver | enable-java | true |
| java-config | java-libraries-dir | ./java/lib |
| | java-extensions-dir | ./java/ext |
| | jvm-path | <jdk install dir>/jre/lib/amd64/server/libjvm.so for Linux/UNIX<br><jdk install dir>\jre\lib\amd64\server\libjvm.dll for Windows |
| java-extensions | BPR_iWD_Extension.jar | true |

| Section | Key | Value |
|---|---|---|
| java-extensions-bpr-iwd | dimension-mapping-1 | Virtual Queue |
| | java-extension | BPR_iWD_Extension.jar |
| | java-extension-jar | BPR_iWD_Extension.jar |
| | jdbc-driver-jar | See JDBC driver JAR information. |
| | jdbc-driver | See JDBC driver information. |
| | jdbc-url | See JDBC URL information. |
| | password | <iWD datamart password> |
| | refresh-interval | 15 |
| | service-id-1 | STAT_1 |
| | service-tenant-1 | Resources |
| | user | <iWD datamart database user> |
| | verbose | debug |
| | virtual-queue-name-1 | iWD_ |

5. Restart the Stat Servers.

# Stat Server Extensions

After you have installed iWD, you can install the iWD Stat Extensions, which provide access to the aggregated data in the Data Mart.

> **Important**
> iWD Setup Utility is not supported in release 9.0.

## Installing Stat Server Extensions

> **Important**
> If the Data Mart is not used, Stat Extensions are not required and the solution will work fine.

### Purpose

To provide access to the aggregated data in the Data Mart.

### Prerequisites

- An instance of Stat Server is installed, dedicated for use with iWD. Refer to the *Stat Server 8.5 Deployment Guide* for more information.
- ORS 8.1.400.48 is installed if you wish to use the Stat Server Extensions features implemented in iWD release 8.5.106.03.

### Procedure

1. From the server that is running Stat Server, navigate to the **iWD Stat Extensions** folder of the iWD CD. Locate and double-click **Setup.exe**.
2. Click **Next** on the **Welcome** screen.
3. Select the appropriate Stat Server instance from the list that is displayed and click **Next**.
4. Click **Install** to install iWD Stat Extensions. Click **Finish** when the installation has been completed.

# Stat Server Configuration Options

## [java-extensions] section

During installation, a new option—**BPR_iWD_Extension.jar**—is added to the [java-extensions] section, with a default value of `true`.

## [java-extensions-bpr-iwd] section

The **[java-extensions-bpr-iwd]** section contains options which specify the JDBC connection driver and parameters for access to the iWD Data Mart database. Most options are managed by the Stat Adapter job of iWD Data Mart and are rewritten each time the Stat Adapter is run. For reference, those options are listed below:

- **jdbc-driver-jar**—The **.jar** file with the JDBC driver. The path is relative to the directory specified as **java-libraries-dir** in the Stat Server configuration (which defaults to **./java/lib** in the Stat Server installation directory). Valid values include:
  - `mssql-jdbc-6.1.0.jre8.jar` (for MS SQL)
  - `ojdbc8.jar` (for Oracle)
  - `postgresql-9.4.1212.jar` (for PostgreSQL)
- **jdbc-driver**—The class name for the corresponding JDBC driver. Valid values include:
  - `com.microsoft.sqlserver.jdbc.SQLServerDriver` (for MS SQL)
  - `oracle.jdbc.OracleDriver` (for Oracle)
  - `org.postgresql.Driver` (for PostgreSQL)
- **jdbc-url**—The JDBC URL, which describes RDBMS-specific access parameters. Below are some sample values:
  - `jdbc:sqlserver://hostname:1433;databaseName=databasename` (for MS SQL)
  - `jdbc:oracle:thin:@//hostname:1521/databasename` (for Oracle)
  - `jdbc:postgresql://hostname:5432/databasename` (for PostgreSQL)
- **user**—The user name for database access.
- **password**—The password for database access.
- **verbose**—The level to control debug information, provided in the Stat Server log file. Possible values are `debug`, `trace`, or `standard`.
- **refresh-interval**—The interval (in minutes) for data updates from the database.
- **service-id-1**—The runtime ID of the Statistics Adapter service in the iWD configuration.
- **service-tenant-1**—The name of the configuration tenant—this is linked to the iWD managed tenant where the Statistics Adapter service is configured.

- **virtual-queue-name-1**—
  - If Dimension Mapping is set to `Filter`, it is the name of the single Genesys virtual queue to which statistics are distributed.
  - If Dimension Mapping is set to `Virtual Queue`, it is the prefix to be added to the names of Genesys virtual queues to which statistics are distributed.
- **dimension-mapping-1**—The type of the dimension mapping between iWD Data Mart and Stat Server. Valid values are `Filter` and `Virtual Queue`.

You can set a small subset of options manually:

- **java-extension-jar**—Name of the BPR iWD extension file. Defaults to `BPR_iWD_Extension.jar`. This must match the option in **java-extensions** section in the Stat Server configuration.
- **jdbc-properties-file**—Path to JDBC driver properties file relative to directory specified as `java-libraries-dir` in Stat Server configuration (defaults to `./java/lib` in Stat Server installation directory). Can be used to specify optional driver-specific JDBC options. Please consult driver documentation for further information.
- **reconnection-timeout**—Delay in milliseconds between database reconnection attempts performed by the BPR iWD extension. Defaults to `10000`.
- **tenant-ids**—A list of Configuration Server/Genesys Administrator tenant names mapped to iWD tenant IDs, separated by a comma. For a system with one tenant with name `TenantA` and ID `T2`, this option should be set to value:`Environment=1,TenantA=2`.

## Creating iWD Virtual Queues automatically

The Statistics Adapter job can automatically create Virtual Queues for statistics to be reported. This feature is disabled by default.

For every unique value stored for the tenant in the **gtl_stat** table in the **dimensionId** column, the Statistics Adapter job does the following:

1. Checks whether the Virtual Queue with the defined name exists.
2. If the Virtual Queue does not exist, the Statistics Adapter job creates the Virtual Queue.
3. If the Virtual Queue does exist, the Statistics Adapter skips Virtual Queue creation.

These steps are continuously performed on each run of the Statistic Adapter job. No automatic update or delete is implemented.

### Configuring Virtual Queues creation

To enable this feature, the following options in iWD Runtime Node application must be set:

- **stat-server \ dimension-mapping** = `Virtual Queue`
- **stat-server \ virtual-queue-name** = *<VQ_name_prefix>* (`iWD_` by default)
- **virtual-queues \ create-vqs** = `true`

- **virtual-queues \ switch** = *<switch_name>* (name of the switch where VQs will be created)

You can configure these options in a more convenient way by using Data Mart settings in the iWD Plug-in for GAX, as described in this article.

### Virtual Queues naming convention

Virtual Queues are created according to the following naming convention:

- <VQ prefix> + <dimensionId value>

where:

- <VQ prefix> is taken from iWD Runtime Node options > **stat-server\virtual-queue-name** option.
- <dimensionId value> is taken from **dimensionId** column of **gtl_stat** table.

## Report Stats for each Dimension on its own Virtual Queue

The iWD Stat Server Java Extension can be configured to report statistics in two different ways:

- All statistics for all dimensions can be reported on one Virtual Queue, or;
- Each dimension can have its statistics reported on its own Virtual Queue.

Manual setup of IWD configures the Stat Server to use the iWD Stat Server Java Extension to report each statistic on its own Virtual Queue. If you want to change this, you must make the change manually.

### Procedure

To report each dimension on its own Virtual Queue:

1. Set the the option **dimension-mapping-1** to the value `Virtual Queue`.
2. Instead of using the Virtual Queue name as a value for the option **virtual-queue-name-1**, indicate the prefix that will be used for Virtual Queue names. For example, if **virtual-queue-name-1**=`dim-`, then Virtual Queues with the names `dim-CNT_T2_C106, dim-CNT_T2_C107` and so on, have to be created.

To report all statistics on one Virtual Queue:

1. Set the option **dimension-mapping-1** to the value `Filter`.
2. Set the Virtual Queue name as the value for the option **virtual-queue-name-1**.
3. Create the Virtual Queue using the name value from step 2.

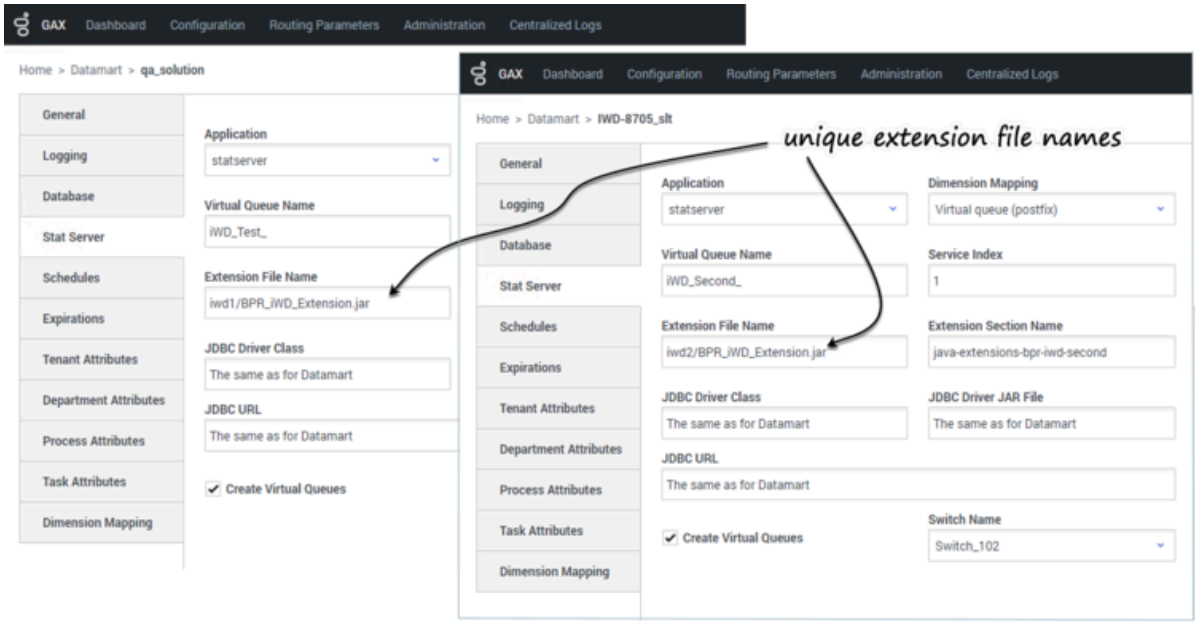
## Configure Stat Extensions for separate iWD Solutions using one

## Stat Server

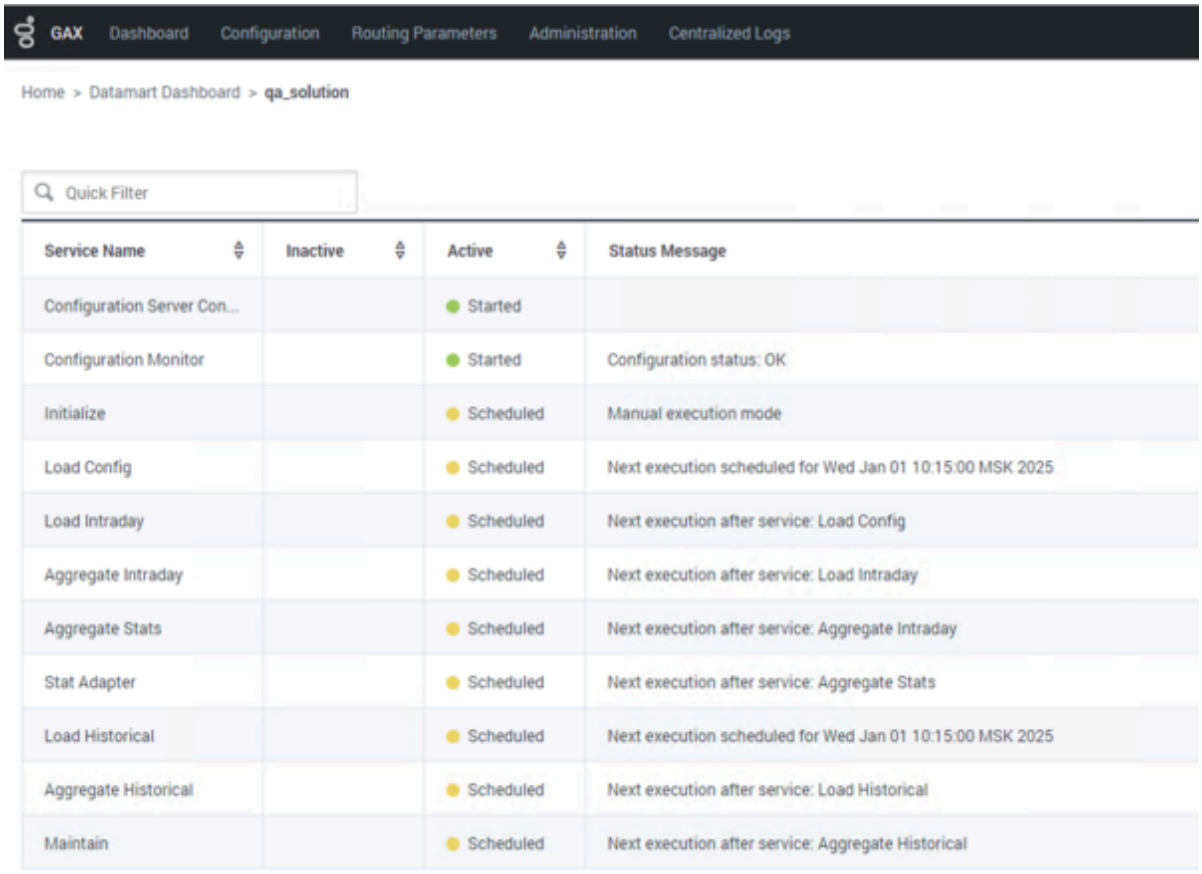
1. Copy the iWD SSJE binary on the back-end for each solution to provide statistics. In this example they are put into separate directories:

```
$ mkdir statserver/java/ext/iwd1 statserver/java/ext/iwd2

$ mv statserver/java/ext/BPR_iWD_Extension.jar statserver/java/ext/iwd1

$ cp statserver/java/ext/iwd1/BPR_iWD_Extension.jar statserver/java/ext/iwd2/


$ ls statserver/java/ext/*

statserver/java/ext/iwd1:

BPR_iWD_Extension.jar


statserver/java/ext/iwd2:

BPR_iWD_Extension.jar
```

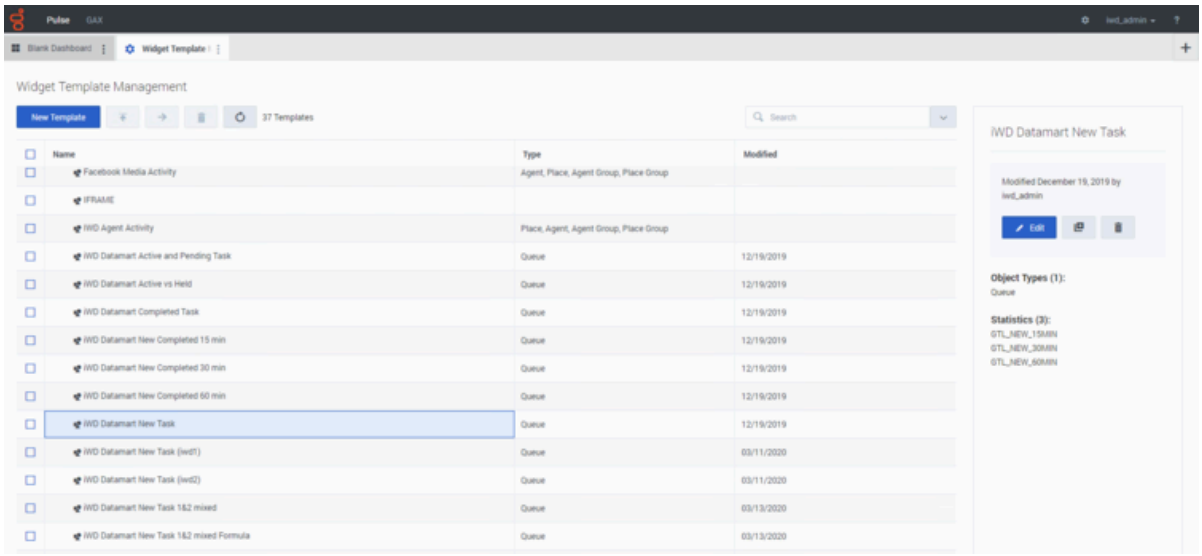
2. Reflect these changes in each iWD Solution by using iWD Plug-in for GAX (**Configuration > iWD > Datamart > [tenant] > [solution] > Stat Server**). Ensure that the **Extension File Name** is unique for each solution:



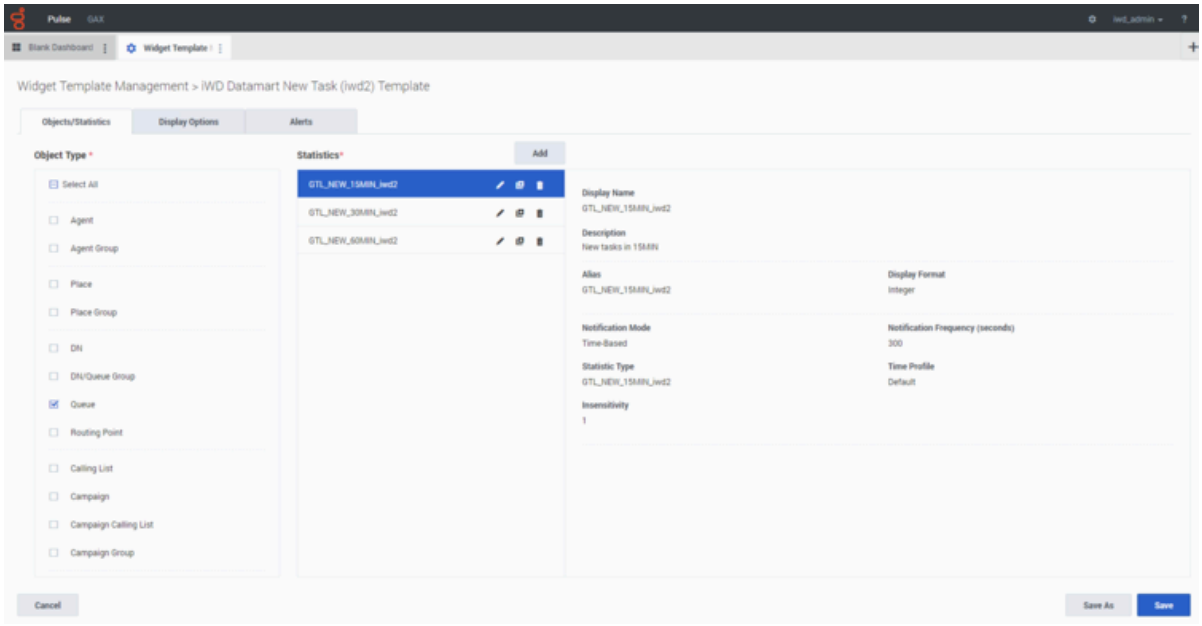
3. Run the **Stat Adapter** job for each Data Mart you have configured (**Configuration > iWD > Datamart Dashboard > [tenant] > [solution] > Stat Adapter > Start**):



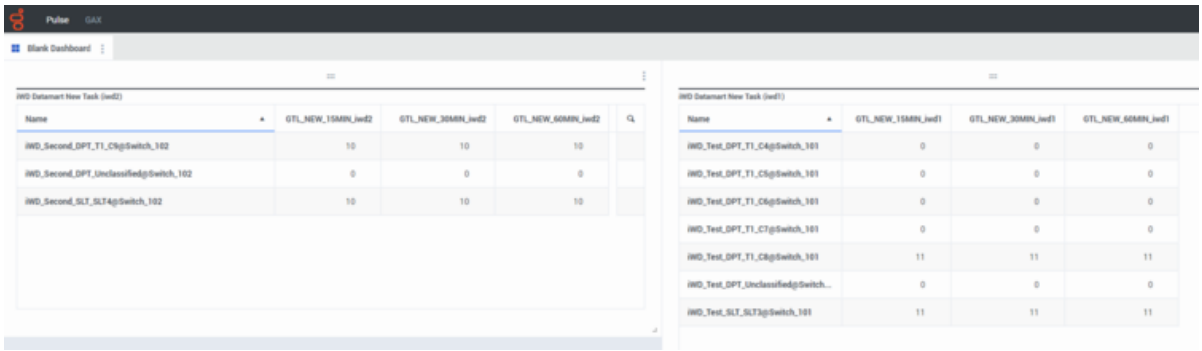

4. Open your Stat Server application object, open **Application Options** and export them to either a cfg or csv file.
5. Within these options there are Stat Types (options sections) with names starting with `GTL_`; such as `GTL_ACTIVE`, `GTL_NEW_15MIN`, and more. Clone these sections while providing some unique affix to them, like `_iwd1` and `_iwd2`, for each solution. You should end up with sections `GTL_ACTIVE_iwd1`, `GTL_ACTIVE_iwd2`, and so on.
6. Within each such section amend option **JavaSubCategory** with the name of the Java Extension name. Please refer to the sample configuration file for two IWD solutions below.
7. Import the amended configurations to the Stat Server application object.
8. Restart Stat Server.
9. Open Pulse and clone the default iWD widget templates for each solution:




10. In the templates, change the **Statistics Type** for each Statistic to match with the section names given in Stat Server:




11. With new templates created, you can now add widgets and assign proper Virtual Queues to them, depending on the solution. Here's the result of a sample configuration:




## Sample configuration file for two IWD solutions

```
[java-config]
java-extensions-dir=./java/ext
jvm-path=/usr/java/jre-1.8/lib/amd64/server/libjvm.so

[java-extensions-bpr-iwd-test]
service-id-1=STAT_1
service-tenant-1=selenium
virtual-queue-name-1=iWD_Test_
dimension-mapping-1=Virtual Queue
java-extension=iwd1/BPR_iWD_Extension.jar
java-extension-jar=iwd1/BPR_iWD_Extension.jar
jdbc-driver-jar=mssql-jdbc-6.1.0.jre8.jar
jdbc-driver=com.microsoft.sqlserver.jdbc.SQLServerDriver
jdbc-url=jdbc:sqlserver://<db_host>:1433;databaseName=<iwd_dm_1_db_name>
user=<iwd_dm_1_db_user>
refresh-interval=15
verbose=debug

[java-extensions-bpr-iwd-second]
service-id-1=STAT_1
service-tenant-1=selenium
virtual-queue-name-1=iWD_Second_
dimension-mapping-1=Virtual Queue
java-extension=iwd2/BPR_iWD_Extension.jar
java-extension-jar=iwd2/BPR_iWD_Extension.jar
jdbc-driver-jar=mssql-jdbc-6.1.0.jre8.jar
jdbc-driver=com.microsoft.sqlserver.jdbc.SQLServerDriver
jdbc-url=jdbc:sqlserver://<db_host>:1433;databaseName=<iwd_dm_2_db_name>
user=<iwd_dm_2_db_user>
refresh-interval=15
verbose=debug

[GTL_NEW_15MIN_iwd1]
Objects=Queue
Category=JavaCategory
JavaSubCategory=iwd1/BPR_iWD_Extension.jar:BPR iWD Statistics
Description=BPR iWD Statistics
measure-id=NEW_15MIN
```

```
[GTL_COMPLETED_15MIN_iwd1]
Objects=Queue
Category=JavaCategory
JavaSubCategory=iwd1/BPR_iWD_Extension.jar:BPR iWD Statistics
Description=BPR iWD Statistics
measure-id=COMPLETED_15MIN

[GTL_NEW_30MIN_iwd1]
Objects=Queue
Category=JavaCategory
JavaSubCategory=iwd1/BPR_iWD_Extension.jar:BPR iWD Statistics
Description=BPR iWD Statistics
measure-id=NEW_30MIN

[GTL_COMPLETED_30MIN_iwd1]
Objects=Queue
Category=JavaCategory
JavaSubCategory=iwd1/BPR_iWD_Extension.jar:BPR iWD Statistics
Description=BPR iWD Statistics
measure-id=COMPLETED_30MIN

[GTL_NEW_60MIN_iwd1]
Objects=Queue
Category=JavaCategory
JavaSubCategory=iwd1/BPR_iWD_Extension.jar:BPR iWD Statistics
Description=BPR iWD Statistics
measure-id=NEW_60MIN

[GTL_COMPLETED_60MIN_iwd1]
Objects=Queue
Category=JavaCategory
JavaSubCategory=iwd1/BPR_iWD_Extension.jar:BPR iWD Statistics
Description=BPR iWD Statistics
measure-id=COMPLETED_60MIN

[GTL_OVERDUE_15MIN_iwd1]
Objects=Queue
Category=JavaCategory
JavaSubCategory=iwd1/BPR_iWD_Extension.jar:BPR iWD Statistics
Description=BPR iWD Statistics
measure-id=OVERDUE_15MIN

[GTL_PENDING_15MIN_iwd1]
Objects=Queue
Category=JavaCategory
JavaSubCategory=iwd1/BPR_iWD_Extension.jar:BPR iWD Statistics
Description=BPR iWD Statistics
measure-id=PENDING_15MIN

[GTL_ACTIVE_iwd1]
Objects=Queue
Category=JavaCategory
JavaSubCategory=iwd1/BPR_iWD_Extension.jar:BPR iWD Statistics
Description=BPR iWD Statistics
measure-id=ACTIVE

[GTL_HELD_iwd1]
Objects=Queue
Category=JavaCategory
JavaSubCategory=iwd1/BPR_iWD_Extension.jar:BPR iWD Statistics
Description=BPR iWD Statistics
measure-id=HELD
```

```
[GTL_NEW_15MIN_iwd2]
Objects=Queue
Category=JavaCategory
JavaSubCategory=iwd2/BPR_iWD_Extension.jar:BPR iWD Statistics
Description=BPR iWD Statistics
measure-id=NEW_15MIN

[GTL_COMPLETED_15MIN_iwd2]
Objects=Queue
Category=JavaCategory
JavaSubCategory=iwd2/BPR_iWD_Extension.jar:BPR iWD Statistics
Description=BPR iWD Statistics
measure-id=COMPLETED_15MIN

[GTL_NEW_30MIN_iwd2]
Objects=Queue
Category=JavaCategory
JavaSubCategory=iwd2/BPR_iWD_Extension.jar:BPR iWD Statistics
Description=BPR iWD Statistics
measure-id=NEW_30MIN

[GTL_COMPLETED_30MIN_iwd2]
Objects=Queue
Category=JavaCategory
JavaSubCategory=iwd2/BPR_iWD_Extension.jar:BPR iWD Statistics
Description=BPR iWD Statistics
measure-id=COMPLETED_30MIN

[GTL_NEW_60MIN_iwd2]
Objects=Queue
Category=JavaCategory
JavaSubCategory=iwd2/BPR_iWD_Extension.jar:BPR iWD Statistics
Description=BPR iWD Statistics
measure-id=NEW_60MIN

[GTL_COMPLETED_60MIN_iwd2]
Objects=Queue
Category=JavaCategory
JavaSubCategory=iwd2/BPR_iWD_Extension.jar:BPR iWD Statistics
Description=BPR iWD Statistics
measure-id=COMPLETED_60MIN

[GTL_OVERDUE_15MIN_iwd2]
Objects=Queue
Category=JavaCategory
JavaSubCategory=iwd2/BPR_iWD_Extension.jar:BPR iWD Statistics
Description=BPR iWD Statistics
measure-id=OVERDUE_15MIN

[GTL_PENDING_15MIN_iwd2]
Objects=Queue
Category=JavaCategory
JavaSubCategory=iwd2/BPR_iWD_Extension.jar:BPR iWD Statistics
Description=BPR iWD Statistics
measure-id=PENDING_15MIN

[GTL_ACTIVE_iwd2]
Objects=Queue
Category=JavaCategory
JavaSubCategory=iwd2/BPR_iWD_Extension.jar:BPR iWD Statistics
Description=BPR iWD Statistics
measure-id=ACTIVE
```

```
[GTL_HELD_iwd2]
Objects=Queue
Category=JavaCategory
JavaSubCategory=iwd2/BPR_iWD_Extension.jar:BPR iWD Statistics
Description=BPR iWD Statistics
measure-id=HELD

[java-extensions]
iwd1/BPR_iWD_Extension.jar=true
iwd2/BPR_iWD_Extension.jar=true
```

# Logging

iWD Manager and iWD Runtime Node support creation of their own log files for troubleshooting purposes. In addition, iWD Manager supports centralized logging through Genesys Message Server (optional). The following procedures explains how to configure the required parameters.

> **Important**
>
> Write the pathnames using either a single forward ( **/** ) slash or a double backwards slash ( **** ) as a separator.

## Configuring Logging for iWD Manager

> **Important**
>
> For centralized logging, add the connection from iWD Manager Server to Message Server. Refer to the 8.5.1 Management Layer User's Guide for more information about Message Server.

### Process

The process of startup logging contains two stages:

1. Initial Phase.
    1. When iWD Manager starts, initial logging configuration is read from the logback.xml configuration file **<IWD_MANAGER_FOLDER>/config/logback.xml**. One log file will be created: **<IWD_MANAGER_FOLDER>/iwd_manager_startup.log**.
2. iWD Manager retrieves the configuration from Configuration Server.
    1. iWD Manager will reconfigure itself to log according to configuration stored in Configuration Server.
    2. If logging configuration is missing in Configuration Server, iWD Manager will use default logging options values.
    3. If logging configuration is incorrect and iWD Manager cannot create any logger it will exit with an error.

For information about configuring IWD Manager Logging, refer to the iWD Manager configuration options.

### Log Files

- **iwd_manager.log**—Main log.
- **iwd_manager_startup.log**—Initial phase log.
- **iwd_manager_access.log**—iWD Manager API log.

## Configuring Logging for iWD Runtime Node

> **Important**
> For centralized logging, add the connection from iWD Runtime Node to MessageServer. Refer to the Framework 8.5.1 Management Layer User's Guide for more information about Message Server.

### Process

The process of startup logging contains two stages:

1. Initial Phase.
   a. When iWD Runtime Node starts, initial logging configuration is read from the yaml configuration file (**<RUNTIMENODE_FOLDER>/config/iwd_dm.yaml**).
   b. One log file will be created—**<RUNTIMENODE_FOLDER>/datamart-startup.log**.
3. iWD Runtime Node retrieves the configuration from Configuration Server.
   a. iWD Runtime Node will reconfigure itself to log according to configuration stored in Configuration Server.
   b. If logging configuration is missing in Configuration Server, iWD Runtime Node will use default logging options values.
   c. If logging configuration is incorrect and iWD Runtime Node cannot create a logger, it will exit with an error.
   d. All log files are created (see Log Files below).

For information about configuring IWD Runtime Node Logging, refer to the iWD Genesys Administrator Extension Plug-in Help.

### Log Files

- **datamart.log**—Main log.
- **datamart-startup.log**—Initial phase log.

- **datamart-access.log**—Data Mart API log.

Log files for corresponding Data Mart services:

- **aggregate-historical.log**
- **aggregate-intraday.log**
- **aggregate-stats.log**
- **configuration-monitor.log**
- **conf-server-connector.log**
- **initialize.log**
- **load-config.log**
- **load-historical.log**
- **load-intraday.log**
- **maintain.log**
- **stat-adapter.log**

## Configuring Logging for iWD History Node

> **Important**
>
> For centralized logging, add the connection from iWD History Node to MessageServer. 8.5.1 Management Layer User's Guide for more information about Message Server.

### Process

The process of startup logging contains two stages:

1. Initial Phase.
    1. When iWD History Node starts, initial logging configuration is read from the **yaml** configuration file **<IWD_HISTORY_FOLDER>/config/iwd_history.yaml**. One log file will be created: **<IWD_HISTORY_FOLDER>/iwd_history_startup.log**.
2. iWD History Node retrieves the configuration from Configuration Server.
3. iWD History Node will reconfigure itself to log according to configuration stored in Configuration Server.
4. If logging configuration is missing in Configuration Server, iWD History Node will use default logging options values.
5. If logging configuration is incorrect and iWD History Node cannot create any logger, it will exit with an error.

For information about configuring IWD History Node Logging, refer to the iWD History Node

Configuration.

### Log Files

- **iwd_history.log**—Main log.
- **iwd_history_startup.log**—Initial phase log.
- **iwd_history-access.log**—iWD History API log.

## Hiding or Tagging Sensitive Data in Logs

### Hiding All Sensitive Data

To hide all sensitive data in logs, you must create the following in the application options:

- Section—**[log-filter]**
- Option name—**default-filter-type** with the value set to `hide`.

### Hiding Specific Data

To hide specific data (for example, password) in logs, you must create the following in the application options:

- Section—**[log-filter-data]**
- Option name—**password** with the value set to `hide`.

### Common Settings

The settings described at the links below are common for most Genesys applications.

- *PSDK Developer's Guide* (a detailed description of "Hide or Tag Sensitive Data in Logs" feature)
- *Security Deployment Guide*

# Preparing iWD For Use With Genesys Social Engagement

By default, the installation and configuration procedures described above will prepare iWD to support Genesys Social Engagement. That is, the required database fields will be present in the Interaction Server and Interaction Server Event Log databases, the required Interaction Custom Properties (Business Attributes) will be created in Configuration Server, and the required Interaction Server and Event Log Database Access Point application options will be configured.

Enabling this feature will have the following effect:

- A Global Task List custom filter that uses specific social media-related columns will be visible in the **Filters** drop-down list.
- Social media-related attributes will be available in custom filters in the custom filters drop-down list of the Global Task List.
- Social media-related attributes will be visible on the **Attributes** panel of the Global Task List.

## Updating the Tenant to enable display of Social Media Attributes and Filters

1. Log in to the Genesys Administrator Extension.
2. Navigate to **Configuration Manager > Environment > Tenant**.
3. Select the Tenant to work with.
4. In the **iWD Attributes** tab,check the **Social Messaging Enabled** checkbox and save the change.

# Installing/Removing Language Packs

> **Important**
> From version 9.0, iWD products are delivered with all available localizations at the time of their releases.

## Notes on Localizations

- For localization related to the iWD Plug-in for GAX, please go **here**.
- To make custom translations for your iWD products, please go **here**.

# Configuration

The following topics describe the configuration required for iWD:

# IWD Manager Configuration

## Logging in

### User Login

Anyone who needs access to iWD Manager must be configured as a `Person` or in Genesys Administrator (GA) or Genesys Administrator Extension (GAX) and also must have the required privileges (usually defined as iWD roles) assigned either in GA or GAX.

The login screen prompts for the following information:

- **Username**—The username for the Person or User as configured in GA/GAX.
- **Password**—The password for the Person or User as configured in GA/GAX.

## Programmatic Login

You can log into iWD Manager programmatically by providing the URL for iWD Manager login page along with a valid username and password. This can facilitate a single sign-on process.

The URL format to use is:

```
http://<appserverhost>:<appserverport>/<iwdmanagerapplication>/ui/login.jsf?username=<username>
&password=<password>&passwordEncoded
```

Where:

- <appserverhost> and <appserverport> are the host and port for the application server where iWD Manager is deployed.
- <iwdmanagerapplication> is the iWD Manager application running on the application server (for example, iwd_manager).
- <username> and <password> are a valid username and password combination for the user logging into iWD Manager.

> **Important**
>
> If you want to include the &passwordEncoded at the end of the URL, then the value for <password> should be encoded using the BASE64 algorithm. Otherwise, you can pass a plain-text password in the <**password**> parameter and omit &passwordEncoded.

An example URL is:

```
<http://myTomcatHost:8080/iwd_manager/ui/login.jsf?username=jsmith&password=myPassword&ConfigApplication=iWDManager
```

## General Conditions for Configuring an iWD Manager User

**To access the Global Task List,** the Person must be associated with a Place in Genesys Configuration. If the Person is not an Agent, the Place can be configured as an option on the **Annex** tab of the Person/User object. Create a section called **[iWD]** (if it does not already exist) and within it, create an option **iWDManagerPlace** with the value of a valid Place name.

**If you want a user to have access to a specific managed tenant in iWD,** that user must have at least Read access to the Configuration Server tenant that is linked to that iWD managed tenant—either directly or by being a member of an Access Group that does. The user must have at least Read and Execute permissions to the iWD Manager application in Configuration Server, and Read permissions to the Configuration Server Host object where the Interaction Server is running, either directly or by being a member of an Access Group that does.

**If the user is going to have access to the Global Task List,** then the user must have at least Read and Execute permissions to these application objects in Configuration Server:

- Interaction Server
- The Database Access Point for the Interaction Server database
- The Database Access Point for the History Node database.

These permissions may be applied directly to the user, or by the user being a member of an Access Group that has such permissions.

The Place that must be associated with the user must be a Place configured under the Configuration Server tenant that maps to the iWD managed tenant. (This presumes that the Interaction Server application also has an association with this Configuration Server tenant.) The Place is only used to give the user access to the Global Task List.

**If you want to restrict what the user can do in iWD Manager,** use roles and privileges as described in Role-Based Access Control.

**If the user is created under a Configuration Server child tenant,** then the following additional conditions must be met:

- If you want the user to have access to both the iWD SYSTEM tenant as well as the iWD managed tenant to which the Configuration Server tenant maps, then this user must be either be a member of at least two Access Groups with the relevant permissions, or must have permissions granted directly to them.

Such Access Groups must have at least Read permission to the tenants.

**If the user has no role privilege in a Tenant,** then the Tenant is hidden from that user.

**If the user is created under the Environment tenant** (for example, such as the default user that is in the Configuration Server database when it is initially deployed) then in order for this user to have full control of the Global Task List (not just read-only access), two conditions must be met:

- The user must have a valid Place configured, where the Place is created under a child Configuration Server tenant to which the Interaction Server is associated.

- The user must have an Employee ID that also belongs to a user who is created under a child Configuration Server tenant to which the Interaction Server is associated.
- The user must have all role privileges from all the "iWD*" groups.

# Importing Media Icons

## Prerequisites

- iWD Manager, iWD Runtime Node are installed.
- Application servers are started (if running on WebSphere).

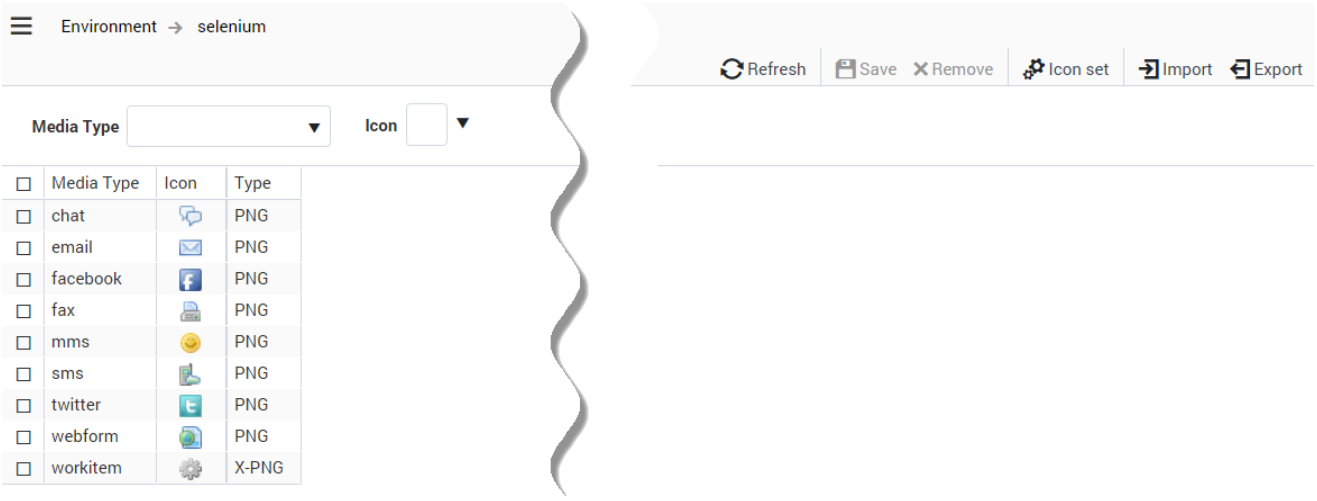
## Procedure

1. Log in to iWD Manager.
2. Select "Media Icons" from the main menu.
3. Select Import from the top right toolbar.
4. Import the relevant media icons file: for example;
    - ***<iWD Manager installation directory>*\config\iwd_mediaicons.xml**.

> **Important**
> The XML file is saved to the installation directory where the iWD Manager supporting files were installed.

5. After each import a message appears at the bottom of the screen to indicate whether the import was successful.

## Media Icons View in iWD Manager




The **Media Icons** view lets you map graphical icons to media types for display in the GTL. iWD comes with some pre-loaded icons, but you can import additional icons and map them to existing media types, or new media types. You can also export the icon set to an XML file.

### Basic data

- **Media Type**— The media type to which the icon will be mapped. This list of media types is retrieved dynamically from Genesys Configuration Server. Media types are a type of **Business Attribute** in Configuration Server. Genesys provides out-of-the-box media types but new custom media types can be added.
- **Icon**—The graphical 16x16 icon that will be displayed in the GTL when a task is of the media type to which the icon is mapped.
- **Type**—The icon's file type. Icons must be 16x16, but can be in any of the following file formats:
  - GIF
  - JPG
  - BMP
  - PNG
  - X-PNG

## Actions

Use the toolbar commands to do the following:

- **Remove**—Remove a selected icon.
- **Save**—Save changes.
- **Refresh**—Refresh the display.
- **Icon set**—Add new icons from a local file system.

- **Import**—Allows you to import the iWD Media Icons configuration from an XML file. Click Import, select a file on your local drive, select it and and click Open.
- **Export**—Export the iWD Media Icons configuration to a local XML file.

## Filters

Configuration of Filters is performed in iWD Manager. Filters let you refine the list of tasks displayed in the GTL. Each filter is defined by a set of filter criteria (optional) and a list of table columns (Filter columns) that will be displayed in the GTL.

Some filters are preconfigured for iWD Manager, and you can create custom filters. Full details are in the **Filters** (new document) topic of iWD Manager Help.

# OAuth User Authentication for iWD Manager using GWS Auth Service

> **Important**
>
> This functionality requires GWS Authentication Service 9.0.

You can set up iWD Manager to use the OAuth 2.0 protocol for user authorization. OAuth, short for "open authorization," is an open standard protocol that allows secure API authorization without requiring the user to provide their credentials to a third party. You can read more about OAuth here.

When OAuth is enabled, users can log in to iWD manager with accounts from Genesys Web Services (GWS).

To enable the OAuth 2.0 authentication mechanism in iWD manager:

1. Set the enabled option to `true` in the **[oauth]** section.
2. Configure the OAuth authentication settings such as an Authorization Service Base URI, Client ID, and Client Secret, Redirect URI in the oauth section.
3. Enable the token-based authentication as described in Secure Communication with Configuration Server.

# iWD Manager configuration options

iWD Manager establishes a connection to Configuration Server at the moment of user login. The configuration needed to start the iWD Manager web application is read from property files. There are two levels of connection:

- System level connection—Connects to Configuration Server on behalf of the SYSTEM user at the moment of iWD Manager application startup before Tomcat starts, and reads the initial configuration required for web application.
- User level connection—Connects to Configuration Server on behalf of an actual user at the moment of login, and reads user-specific configuration.

iWD Manager configuration options are described here.

# iWD Runtime Node Configuration

## Configuration options

iWD Runtime Node configuration options are described here.

Additional configuration can be carried out using iWD GAX Plugin as described in iWD Business Solution Configuration.

## .yaml File

The **.yaml** configuration file:

- Provides initial log configuration
- Provides configuration of a list with HTTP methods to reject on open sockets

On startup, Runtime Node attempts to read it from a default location:

```
${INSTALLATION DIRECTORY}/config/iwd_dm.yaml
```

### Sample File

```
 logging:
  level: INFO
  appenders:
    - type: file
      currentLogFilename: datamart-startup.log
      threshold: ALL
      archive: false
      timeZone: UTC

httpDisabledMethods: null
adminHttpDisabledMethods: null
```

where:

- **logging**—Defines Runtime Node logging configuration.
    - **httpDisabledMethods**— One or more (comma delimited) HTTP methods to reject on the webservice port. Null indicates allowing all methods. After an update, you must restart the component. Example: OPTIONS,TRACE.
    - **adminHttpDisabledMethods**— One or more (comma delimited) HTTP methods to reject on the admin port. Null indicates allowing all methods. After an update, you must restart the component. Example: OPTIONS,TRACE.

# Enabling or disabling Load GIM job

This section describes how to enable or disable the Load GIM job.

Log into Genesys Administrator or GAX, open the iWD Runtime Node Application, and configure the following:

- On the **Connections** tab, add a Genesys Info Mart (GIM) DAP to point at the Genesys Info Mart source that will be used.
  The GIM DAP should contain the role `INFO_MART` in the options. For more information, see DAP Options. **Note**: Only one GIM DAP can be specified.
- On the **Options** tab, add the **gim-load-enabled** option into the **settings** section. To enable the Load GIM job, set the option value as `true`.

| Section | Option | Default | Valid values | Changes Take Effect | Description |
|---|---|---|---|---|---|
| settings | gim-load-enabled | false | true, false | After restart | The option that enables Load GIM ETL job |

# iWD History Node Configuration

## .yaml File

The **.yaml** configuration file provides:

- Initial log configuration
- Configuration for the events migration mechanism from the Event Log database to the History Node database
- Configuration of a list with HTTP methods to reject on open sockets

On startup, History Node attempts to read it from a default location:

```
${INSTALLATION DIRECTORY}/config/iwd_history.yaml
```

However, different **.yaml** configuration file can be provided by using the **-config** command-line option.

### Sample File

```
 logging:
  level: INFO
  appenders:
    - type: console
    - type: file
      currentLogFilename: /var/log/historynode.log
      threshold: ALL
      archive: true
      archivedLogFilenamePattern: /var/log/historynode-%i.log
      archivedFileCount: 5
      timeZone: UTC
      maxFileSize: 100MB

event-log-migration:
  solutionId: SLT1
  fetchSize: 10
  interaction-server:
    driverClass: com.microsoft.sqlserver.jdbc.SQLServerDriver
    url: jdbc:sqlserver://<db_host>:<db_port>;databaseName=inx_db
    user: <db_username>
    password: <db_password>
  eventlog:
    driverClass: com.microsoft.sqlserver.jdbc.SQLServerDriver
    url: jdbc:sqlserver://<db_host>:<db_port>;databaseName=eventlog_db
    user: <db_username>
    password: <db_password>

httpDisabledMethods: null
adminHttpDisabledMethods: null
```

where:

- **logging**—Defines History Node logging configuration.
- **event-log-migration**—Contains configuration of the Event Log database events to History Node events migration process. This section is optional.
  - **solutionId**—The runtime id of the solution for which events will be migrated. Solution with this id must be configured in Configuration Server
  - **fetchSize**—Determines how many interactions' events will be queried at once from the Event Log database. Please note it is number of interactions, not number of events. Default value: `10`.
  - **interaction-server**—Points to the Interaction Server database.
  - **eventlog**—Points to the Event Log database.
  - **httpDisabledMethods**— One or more (comma delimited) HTTP methods to reject on the default port. Null indicates allowing all methods. After an update, you must restart the component. Example: OPTIONS,TRACE.
  - **adminHttpDisabledMethods**— One or more (comma delimited) HTTP methods to reject on the admin port. Null indicates allowing all methods. After an update, you must restart the component. Example: OPTIONS,TRACE.

## Configuring History Node to Process a Single Event Type

History Node can provide data for iWD Manager or iWD Data Mart or both. By default both event types are processed; however you can change this with following options:

- **process-gtl**—setting this option to `false` will disable processing of the events for the iWD Manager.
- **process-dm**—setting this option to `false` will disable processing of the events for the iWD Data Mart.

### Important

1. iWD Manager will not connect to a History Node application unless the **process-gtl** option is explicitly set to `true`.

2. iWD Data Mart will not connect to a History Node application unless the **process-dm** option is explicitly set to `true`

### Warning

History Node applications connecting to the same JMS queue must have exactly the same configuration for both **process-gtl** and **process-dm** options. Configuring them differently would result in inconsistent data provided by History Node applications.

## Configuring History Node to Process UTF-8

If you plan to use UTF-8 encoded characters in the task updates (for example, in task attributes values), you must enable the UTF-8 encoding support.

**To do this for Windows**, add `-Dfile.encoding=utf-8` to **JavaServerStarter.ini** under the **[JavaArgs]** section.

**To do this for UNIX**, add `-Dfile.encoding=utf-8` to **JAVA_OPTS** in **iwd_history.sh**.

Then restart History Node.

# iWD History Node Configuration Options

iWD History Node configuration options are described here.

# iWD History Node Database Schema Migration

There are two scenarios to note when you should perform database schema migration:

- Creating a database schema from scratch in the initial startup process.
- Updating the database schema when upgrading iWD History Node to a new version.

In both cases it works the same way. Complete either one of the following procedures:

- Enable automatic database schema migration by setting the value of Configuration Server option **[iWD]/migrate-schema** to `true`. This means that the application will check the current data schema at startup and update (or create) it if necessary. Note that default value of **[iWD]/migrate-schema** is `false`. Or;
- Migrate the schema manually using the command line as shown below:
  - **Linux**—`./iwd_history.sh -host CFG_SRV_HOST -port CFG_SRV_PORT -app HN_APP db migrate`
  - **Windows**—`JavaServerStarter.exe -host CFG_SRV_HOST -port CFG_SRV_PORT -app HN_APP db migrate`

> **Important**
>
> All commands like "./iwd_history.sh" and "JavaServerStarter.exe" should be executed from the directory where History Node has been installed.

## Troubleshooting

Schema migration can sometimes get stuck in a loop that prevents History Node starting. When this happens, the following entries are written to the log file:

```
INFO  [2017-05-26 10:10:57,497] com.genesyslab.iwd.history.HistoryApplication: Running database schema migration
INFO  [2017-05-26 10:11:03,313] liquibase: Waiting for changelog lock....
INFO  [2017-05-26 10:11:13,317] liquibase: Waiting for changelog lock....
...
```

To correct this, after stopping History Node application, do the following:

- On Linux

1. List any database locks.

```
./iwd_history.sh -host CFG_SRV_HOST -port CFG_SRV_PORT -app HN_APP db locks --list
```

2. Release the database locks.

```
./iwd_history.sh -host CFG_SRV_HOST -port CFG_SRV_PORT -app HN_APP db locks --force-release
```

- On Windows:

1. List any database locks.

```
JavaServerStarter.exe -host CFG_SRV_HOST -port CFG_SRV_PORT -app HN_APP db locks --list
```

2. Release the database locks.

```
JavaServerStarter.exe -host CFG_SRV_HOST -port CFG_SRV_PORT -app HN_APP db locks --force-release
```

# Using Kafka Event Logger with History Node

You can use Apache Kafka as a transport for Interaction Server reporting events. It is especially useful if you plan to use Interaction Server Kafka Capture Point, allowing you not to install a separate JMS broker. The same Kafka installation can be used as a transport both for Capture Point and Event Logger needs.

## Prerequisites

You can choose either:

1. Import template from the **iwd_history/event_logger_templates** folder as described in the iWD History Node Application Definition topic.
2. Configure Interaction Server Kafka Event Logger from scratch as described in the Using the Groovy Event Logger with Apache Kafka topic in the *eServices Administration Guide*.

## Kafka Settings

For iWD History Node to be able to consume data written by Interaction Server, set the following option in the Kafka Event Logger Application Object that is used by these iWD History Node and Interaction Server respectively:

**[kafka-settings]\binary-output = true**

## Consumer Options Description

- **[consumer-options]\poll-delay**—The maximum time in milliseconds to block the consumer when it polls for records.
- **[consumer-options]\max.poll.records**—The maximum number of records returned in a single call to poll().

### Important

1. **max.poll.records** and **poll-delay** values can be tuned for better performance depending on your environment and the size of interaction user data. Please change

these values only if you understand what you are doing and why.

2. Option **[iWD]\batch-size** is not taken into account when Kafka Event Logger is used. Use option **[consumer-options]\max.poll.records** instead.

All other options are standard Kafka consumer options and can be found in the official Kafka documentation.

## Additional Considerations

iWD History Node configured to use Kafka Event Logger must use the same Kafka deployment and its configurations once data (events) are being produced and written by Interaction Server to Kafka and the same are consumed by iWD History Node.

### Important

If at any point, offsets in Kafka are reset or a different deployment of Kafka is being used then use a different topic for Event Logger. In this scenario, you must also update the Interaction Server configuration. iWD History Node will skip all the events up to the latest offset number that it has consumed prior to reset to safeguard itself against possible duplicated data consumption.

# iWD History Node Limitations

- History Node and History Node clusters can only support one Tenant. For multi-tenant configurations, a separate History Node instance is needed for each Tenant.
- Migration of data from iWD Data Mart 8.5 to iWD Data Mart 9.0 is supported with the following limitations:
  - GTL history events
  - iWD Data Mart history events
- iWD Data Mart does not support horizontal scalability. Possible mitigations include:
  - Scaling the application vertically.
  - Splitting the business structure into solutions.
- Authentication and encryption is only supported for the Configuration Server connection.

# Roles and Privileges

iWD 9.0 uses Genesys Role-Based Access Control (RBAC). For details of how to use this feature please see:

- Genesys Administrator documentation set
- Genesys Security Guide
- Roles in the main GAX Help (new document).

> **Important**
> Role privileges cannot be seen in GA, only in GAX. GAX iWD Plug-in privileges are imported when GAX is used to import the GAX iWD Plug-in installation package.

## GAX iWD Plug-in Privileges

- `iWD Read Solution`—Read privilege for iWD Solution.
- `iWD Modify Solution`—Modify privilege for iWD Solution.
- `iWD Create Solution`—Create privilege for iWD Solution.
- `iWD Delete Solution`—Delete privilege for iWD Solution.
- `iWD Read Department`—Read privilege for iWD Department.
- `iWD Modify Department`—Modify privilege for iWD Department.
- `iWD Create Department`—Create privilege for iWD Department.
- `iWD Delete Department`—Delete privilege for iWD Department.
- `iWD Read Process`—Read privilege for iWD Process.
- `iWD Modify Process`—Modify privilege for iWD Process.
- `iWD Create Process`—Create privilege for iWD Process.
- `iWD Delete Process`—Delete privilege for iWD Process.
- `iWD Read Tenant`—Read privilege for Tenant.
- `iWD Read Capture Point`—Read privilege for Capture Point.

## RBAC-related Privileges

In order to apply full Role-Based Access Control to the iWD GAX Plug-in on the GAX dashboard, use

the following roles in iWD GAX Plug-in's template:

- IWD_READ_LOOKUP_TABLES—Read privilege for lookup tables
- IWD_MODIFY_LOOKUP_TABLES—Modify privilege for lookup tables
- IWD_CREATE_LOOKUP_TABLES—Create privilege for lookup tables
- IWD_DELETE_LOOKUP_TABLES—Delete privilege for lookup tables
- IWD_READ_DATAMART—Read privilege for Data Mart
- IWD_MODIFY_DATAMART—Modify privilege for Data Mart
- IWD_MONITOR_DATAMART—Monitor privilege for Data Mart
- IWD_MANAGE_DATAMART—Manage privilege for Data Mart

Users without the relevant permissions will not be able to view the iWD tile on the GAX dashboard.

## iWD Manager Privileges

- View Solutions—Viewing privilege for iWD Solutions
- Configure Ixn Custom Properties—Privilege to configure Interaction Custom Properties needed by iWD
- View Tasks for Departments—Viewing Tasks privilege for iWD Departments
- View Tasks for Processes—Viewing Tasks privilege for iWD Processes
- Modify Media Icons—Modification privilege for iWD Media Icons
- Create Media Icons—Creation privilege for iWD Media Icons
- Delete Media Icons—Deletion privilege for iWD Media Icons
- View Media Icons—Viewing privilege for iWD Media Icons
- Modify Filters—Modification privilege for iWD GTL Filters
- Create Filters—Creation privilege for iWD GTL Filters
- Delete Filters—Deletion privilege for iWD GTL Filters
- View Filters—View privilege for iWD GTL Filters. Required for displaying "'Filters'" tab in the left menu.
- View Global Task List—Viewing Global Task List page privilege
- Export Tasks—Exporting tasks privilege
- Cancel Tasks—Canceling tasks privilege
- Hold/Resume Tasks—Putting on hold or resuming tasks privilege
- Modify Tasks—Modification of tasks privilege
- View Tasks by Capture Point—Viewing tasks by capture point privilege
- View All Tasks—Viewing all tasks in solution privilege
- Open Rules Authoring—Privilege to open Rules Authoring link

- `Open iWD Web`—Privilege to open iWD Web link
- `Import Configuration`—Privilege to import configuration from an XML file
- `Export Configuration`—Privilege to export configuration to an XML file

## Important Information About Filtering Permissions

The following additional information is also important to note when you are filtering permissions:

- Create/Modify/Delete Filters privileges are pointless without the View Filters permission because it is required to show the "'Filters"' tab.
- Public filters can be changed from Private by any user that has Modify Filter permissions.
- The owner of a Private filter can always modify or delete that filter, even if they do not have Delete and/or Modify security role permissions.
- The username of the owner of any filter is displayed on the **Filter** configuration screen to make referencing easy.

# IWD Business Solution Configuration

> **Important**
> Configuration of a business solution is performed entirely in the iWD Plug-in for GAX, so this must be already installed.

## Business Structure

The Business Structure is a hierarchy of business units. Each Tenant can contain one or more Solutions as the first level of the hierarchy. Below Solutions are Departments. Below Departments are Processes. For example:

- East London Office—Solution (note that this meaning means the top node of a business structure, rather than the meaning of Solution in Genesys Configuration environment.)
  - Finance Department—Department
    - Accounts Payable—Process
    - Order Processing—Process

> **Warning**
> All node names have to be unique within the parent node. For example, moving Department D1 to another solution which already has a Department named D1 generates an error.

> **Warning**
> Only one business structure per Tenant is possible.

### Levels of a Business Structure for a Tenant

#### Solutions

Solutions are used for partitioning logical and physical resources for purposes of user access control and load partitioning (performance). Normally there will be one Solution per Tenant, though you can configure multiple solution instances per tenant, if necessary (for example, “Production” and “Test”).

A Solution in iWD represents a runtime environment, which is composed of the following:

- Runtime nodes—iWD runtime application instances that are within the Java application server in which services are being run
- Services—Services that enable iWD functionality, such as Data Mart, Statistics Adapter, and logging.
- Business logic—Primarily the configuration of iWD departments and processes.

The Solution level in a Business Structure corresponds to the Global level in Genesys Rules System for the processing logic of business rules.

### Departments

A Department represents an administrative unit within a Solution. A Solution can contain many Departments. The Department level of a Business Structure corresponds to the Department level in Genesys Rules System for the processing logic of business rules.

### Processes

A Process represents an administrative unit within a Department. A Department can contain many Processes. The Process level of a Business Structure corresponds to the Process level in Genesys Rules System for the processing logic of business rules.

## Display Options

### Filters and Constraints

Configuration Server respects tenancy permission settings. You can access only those objects that you have been granted permissions and privileges to access.

You can filter the contents of this list in two ways:

1. Type the name or partial name of an object in the **Quick Filter** field.
2. Click the cube icon to open the **Tenant Directory** filter panel. In this panel, click the Tenant that you want to select. Use the **Quick Filter** field in this panel to filter the Tenant list.

You can sort the items in the list by clicking the **Name** column. Clicking a second time reverses the sort order. You can add or remove columns by clicking **Select Columns**.

To select or de-select multiple objects at once, click **Select**.

### Data Fields

Each entry is shown with the following data fields:

- **Name**—The element's name.
- **Type**—`Solution`, `Department` or `Process`
- **ID**—The runtime ID of this element.
- **Contact Name**—Contact name for queries about this element.

- **Description**—Free-format text description of the element.

# Solutions

## To create a new Solution

To create a new Solution, do one of the following:

- If one or more Solution is already present, select one Solution and click **More**, then click **Clone**, then edit the Solution data fields.
- If one or more Solution is already present, Display the details of an existing Solution and click **Clone**, then edit the new Solution data fields.
- If no Solution is present, click the the **New** button (**+**), choose **Solution**, then edit the Solution data fields.

## Other Actions

From this context you can **Delete** this Solution.

> ### Warning
> 1. Deleting a Solution can have huge implications for the operation of a contact center. Do not undertake these without serious consideration.
> 2. If you delete a Solution, all related rules packages in GRS will be lost, and you will not be able to recreate new rules packages with the same name.

## Solution Details

### Descriptive Details

- **Solution Name**—The Solution name. Mandatory when you add a new Solution.
- **ID**—The ID of the Solution. Mandatory when you add a new Solution. The system will propose a default new Runtime ID.
- **Timezone**—Solution timezone. Use the drop-down list to change this.
- **First Day of Week**—The first day of the working week for this Solution. Use the drop-down list to change this.
- **Description**—Free-format text description of the Solution. Do not use the Runtime ID as a descriptor.

### Strategies

The following options allow customization of the strategies used by iWD in the current Solution.

- **Classification Strategy**—The name of the Classification strategy for this solution. Select one from the drop-down list.
- **Prioritization Strategy**—The name of the Prioritization strategy for this solution. Select one from the drop-down list.

### Queue Names

The following options allow customization of interaction queues used by iWD in the current Solution. You must define non-standard queue names when there are multiple iWD business processes configured in the same Tenant. Select the required queue names from the drop-down lists. If a value is missing, a default name is displayed for this queue.

- **Queue for canceled tasks**—Interaction queue for tasks canceled by a Global Task List user or by a capture point. Default values:
  - IRD—`iWD_Canceled`
  - Composer—`iwd_bp_comp.Main.iWD_Canceled`
- **Queue for captured tasks**—Interaction queue for tasks successfully processed by the Classification strategy. Default values:
  - IRD—`iWD_Captured`.
  - Composer—`iwd_bp_comp.Main.iWD_Captured`
- **Queue for completed tasks**—Interaction queue for tasks marked as completed by agents. Default values:
  - IRD—`iWD_Completed`
  - Composer—`iwd_bp_comp.Main.iWD_Completed`
- **Queue for error-held tasks**—Interaction queue for tasks that failed to be processed by the Classification or Prioritization strategies. Default values:
  - IRD—`iWD_ErrorHeld`
  - Composer—`iwd_bp_comp.Main.iWD_ErrorHeld`
- **Queue for new tasks**—Interaction queue recognized by iWD as an entry to the business process in this solution. Default values:
  - IRD—`iWD_New`
  - Composer—`iwd_bp_comp.Main.iWD_New`
- **Queue for queued tasks**—Interaction queue for tasks successfully processed by the Prioritization strategy. Default vlaues:
  - IRD—`iWD_Queued`
  - Composer—`iwd_bp_comp.Main.iWD_Queued`
- **Queue for rejected tasks**—Interaction queue for tasks rejected by the Classification strategy. Default values:
  - IRD—`iWD_Rejected`
  - Composer—`iwd_bp_comp.Main.iWD_Rejected`

## Interaction Server Settings

### Important

iWD Plugin for GAX will propagate the connection to the specified Interaction Server (details below) across iWD Manager Server application objects. The following parameters are set as the connection attributes of the Interaction Server connection in the iWD Manager Server application object:

- Port
- Connection Protocol
- Local Timeout
- Remote Timeout

- **Interaction Server**—The Interaction Server for this Solution. This can be an individual Interaction Server, Interaction Server Proxy or Interaction Server Proxy cluster. Please refer to Interaction Server Configuration for details. Use the drop-down list to make a selection. This drop-down list contains those Interaction Servers, Interaction Server Proxies and Interaction Server Proxy clusters which contain the Solution's parent Tenant on their Tenants list. Direct connection to Interaction Server clusters is not supported and therefore will not be presented in this list.
- **Port**—The connection port of the Interaction Server or Interaction Server cluster proxy. Use the drop-down list to change this. This drop-down list contains ports of the Interaction Server or Interaction Server cluster proxy chosen above.

### Warning

**For connections to a unique Interaction Server only**—The iWD Manager application has one common place for all Solutions (and Solutions in different tenants) where Interaction Server connection parameters (for a unique Interaction Server) are kept. If two Solutions are configured to use the same individual Interaction Server, the Interaction Server settings (that is, secure or non-secure) of the Solution that is configured *most recently* are the ones that the Interaction Server will use. It is preferable to ensure that both Solutions' settings are of the same type—either both secure, or both non-secure.
The affected parameters are:

- **Port**
- **Connection Protocol**
- **Protocol Timeout**
- **Local Timeout**
- **Event Buffer Size**
- **Remote Timeout**

- **Threads**

The only parameters to which this does not apply are **Attribute Filter Include/ Exclude**.

- **Connection Protocol**—The connection protocol of the Interaction Server. Use the drop-down list to change this.
- **Protocol Timeout**—The timeout configured for the connection protocol.
- **Local Timeout**—The timeout configured on the local server.
- **Event Buffer Size**—The maximum size in bytes of the event buffer.
- **Remote Timeout**—The timeout configured on the remote server.
- **Threads**—The number of threads available.
- **Attribute Filter Include**—Attributes included here will appear in the Custom Attributes displayed in the Global Task List in iWD Manager.
- **Attribute Filter Exclude**—Attributes excluded here will not appear in the Custom Attributes displayed in the Global Task List in iWD Manager.

### Warning

The JDBC URL *must* be provided in Interaction Server's DAP configuration object in release 9.0. Please refer to the Interaction Server Configuration page.

## History Node Settings

### Important

iWD Plugin for GAX will propagate the connection to the specified iWD History Node (details below) across iWD Manager Server application objects as well as the iWD Runtime Node associated with the given solution. The following parameters are set as the connection attributes of the iWD History Node connection in the iWD Manager Server and iWD Runtime Node application object:

- Port
- Connection Protocol
- Local Timeout
- Remote Timeout

- **History Node**—The name of the History Node for this Solution. This can be an individual History Node, or the name of a History Node cluster. Use the drop-down list to make a selection.
- **Connection Protocol**—The protocol of the connection to the History Node or cluster. Use the drop-down list to change this.
- **Port**—The connection port of the History Node or History Node cluster. Use the drop-down list to change this. This drop-down list contains ports of the History Node or History Node cluster.
- **Local Timeout**—The timeout configured on the local server.
- **Remote Timeout**—The timeout configured on the remote server.

> **Important**
>
> History Node and History Node clusters can only support one Tenant. For multi-tenant configurations, a separate History Node instance is needed for each Tenant.

## Migration

### Interaction custom properties and migration issues

This dialog enables you to:

- Add the necessary columns to tables in the Interaction Server and Interaction Server Event Log databases to support iWD.
- Create new iWD-related **Interaction Custom Properties**, which are a type of **Business Attribute**, in the Genesys configuration database.
- Create some options for the Interaction Server Event Log Database Access Point that are necessary to support iWD.
- Add `iWD_Completed` (IRD) or `iwd_bp_comp.Main.iWD_Completed` (Composer) or a customized business process queue name for completed tasks to the Interaction Server configuration.

If any of these updates was not previously done, this dialog will display a table of warnings informing you of the missing attributes or outdated database versions.

Click the **Configure** button to execute the configuration. A message will be displayed in the **Messages** column when the configuration has completed. A restart of Interaction Server is required if any configuration changes were made.

## Permissions Settings

Business Structure details include a **Permissions** tab on which users with the relevant permissions can view and edit permissions settings for all users of the selected node.

### Permissions Table

| Permission | Description |
|---|---|
| Read | Permission to read information and receive updates about the object. |
| Create | Permission to create objects in this folder. |
| Change | Permission to change the properties of the object. The Change permission is the same as allowing “Write” access. |
| Execute | Permission to perform a predefined action or set of actions with respect to the object. |
| Delete | Permission to delete the object. |
| Read Permissions | Permission to read the access control settings for the object. |
| Change Permissions | Permission to change the access control settings for the object. |
| Execute | Permission to perform a predefined action or set of actions with respect to this object. |
| Propagate | For container objects (such as Tenants). The `Propagate` check box controls whether to propagate this set of elementary permissions to the child objects. By default, the check box is selected). |

## Actions

- **Add Access Group**—Displays the **Select Access Group** panel from which you can select one of the available Access Groups to add to this node and for whom you can then configure permissions.
- **Add Person**—Displays the **Select Person** panel from which you can select one of the available Persons to add to this node and for whom you can then configure permissions.
- **Replace Recursively**—Enables you, upon confirmation, to remove permissions for all child objects of this container and replace them with the permissions defined in this container.

# Departments

## To create a new Department

To create a new Department, first choose the Solution to work with, then do do one of the following:

- If one or more Department is already present, select one Department and click **More**, then click **Clone**, then edit the Department data fields.
- If one or more Department is already present, display the details of an existing Department and click

**Clone**, then edit the new Department data fields.

- If no Department is present, click the the **New** button (**+**), choose Department, then edit the Department data fields.

## Other Actions

From here you can **Clone**, **Delete** or **Move** this Department. You can move the Department only to a Solution. Any Processes configured under it will also be moved. Runtime IDs are not moved in the Move function—you must create a new one for the Department and all its child Processes in its new Solution.

> ### Warning
> 1. Deleting or Moving a department can have huge implications for the operation of a contact center. Do not undertake these without serious consideration.
> 2. If you delete a Department or Process, any rules assigned to those objects will be inactivated and moved to the Solution level. This happens if you delete and re-create a Solution/Department/Process with the same name.

## Department Details

- **Department Name**—The department name. Mandatory when you add a new Department.
- **ID**—The department's Runtime ID. Mandatory when you add a new Department. The system will propose a default new Runtime ID.
- **Contact Name**—The contact name for the department, for informational purposes.
- **Contact Email**—The contact email for the department, for informational purposes.
- **Contact Phone**—The contact phone number for the department, for informational purposes.
- **Start Date**—The date on which the department becomes active. If left empty, the period start date is unconstrained.
- **End Date**—The last day that the department is active. If left empty, the period end date is unconstrained (that is, the department will be active infinitely).
- **Description**—Free-format text description of the Department.

## Department Attributes

Click **Add** to create new attributes.

- **Name**—The attribute name
- **Type**—Select from the drop-down list. Valid values are:
  - `Text`
  - `Percentage`
  - `Number`

  - `Date`
  - `Lookup Table`
- **Value**—The attribute value. If the type is a lookup table, then the value is set from the drop-down list.
- **Description**—Free-format text description of the attribute.

## Department Metrics

Click **Add** to create a set of user-defined metrics, for reporting purposes.

A key component of dashboards and reports is the comparison of actual metrics against target goals. Understanding the effectiveness or efficiency of organizations requires measuring performance against important goals that have been set by the organization. Targets can be associated with a number of objects, such as processes, departments, or tenants. For example, a work-time goal for a task will differ, based on its process; for example, orders will take longer than address changes. You can use metrics to measure this. Example:

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

- **Name**—The metric name
- **Type**—Select from the drop-down list. Valid values are:
  - `Text`
  - `Percentage`
  - `Number`
  - `Date`
  - `Lookup Table`
- **Value**—The attribute value. If the type is a lookup table, then the value is set from the drop-down list.
- **Description**—Free-format text description of the attribute.

## Permissions Settings

Business Structure details include a **Permissions** tab on which users with the relevant permissions can view and edit permissions settings for all users of the selected node.

### Permissions Table

| Permission | Description |
|---|---|
| Read | Permission to read information and receive updates about the object. |
| Create | Permission to create objects in this folder. |
| Change | Permission to change the properties of the object. The Change permission is the same as allowing “Write” access. |
| Execute | Permission to perform a predefined action or set of actions with respect to the object. |
| Delete | Permission to delete the object. |
| Read Permissions | Permission to read the access control settings for the object. |
| Change Permissions | Permission to change the access control settings for the object. |
| Execute | Permission to perform a predefined action or set of actions with respect to this object. |
| Propagate | For container objects (such as Tenants). The `Propagate` check box controls whether to propagate this set of elementary permissions to the child objects. By default, the check box is selected). |

### Actions

- **Add Access Group**—Displays the **Select Access Group** panel from which you can select one of the available Access Groups to add to this node and for whom you can then configure permissions.
- **Add Person**—Displays the **Select Person** panel from which you can select one of the available Persons to add to this node and for whom you can then configure permissions.
- **Replace Recursively**—Enables you, upon confirmation, to remove permissions for all child objects of this container and replace them with the permissions defined in this container.

# Processes

## To create a new Process

To create a new Process, first choose the Solution and Department to work with, then do one of the following:

- If one or more Process is already present, select one Process and click **More**, then click **Clone**, then edit the Process data fields.
- If one or more Process is already present, display the details of an existing Process and click **Clone**,

then edit the new Process data fields.

- If no Process is present, click the the **New** button (**+**), choose Process, then edit the Process data fields.

## Other Actions

From here you can **Clone**, **Delete** or **Move** this Process. You can move the Process only to a Department. Runtime IDs are not moved in the Move function—you must create a new one for the Process in its new Department.

> ### Warning
> 1. Deleting or Moving a Process can have huge implications for the operation of a contact center. Do not undertake these without serious consideration.
> 2. If you delete a Department or Process, any rules assigned to those objects will be inactivated and moved to the Solution level. This happens if you delete and re-create a Solution/Department/Process with the same name.

## Process Details

- **Process Name**—The Process name. Mandatory when you add a new Process.
- **ID**—The Runtime ID of the Process. Mandatory when you add a new Process. The system will propose a default new Runtime ID.
- **Contact Name**—The contact name for the process, for informational purposes.
- **Contact Email**—The contact email for the process, for informational purposes.
- **Contact Phone**—The contact phone number for the process, for informational purposes.
- **Start Date**—The date the process becomes active. The start date of the process cannot be earlier than the start date of the parent department.
- **End Date**—The last day that the process is active. If left empty, the period end date inherits the end date value of the parent department.
- **Description**—Free-format text description of the Process.

## Process Attributes

Click **Add** to create new attributes.

- **Name**—The attribute name
- **Type**—Select from the drop-down list. Valid values are:
  - `Text`
  - `Percentage`
  - `Number`
  - `Date`

    - `Lookup Table`
- **Value**—The attribute value. If the type is a lookup table, then the value is set from the drop-down list.
- **Description**—Free-format text description of the attribute.

## Process Metrics

Click **Add** to create new metrics.

A key component of dashboards and reports is the comparison of actual metrics against target goals. Understanding the effectiveness or efficiency of organizations requires measuring performance against important goals that have been set by the organization. Targets can be associated with a number of objects, such as processes, departments, or tenants. For example, a work-time goal for a task will differ, based on its process; for example, orders will take longer than address changes. You can use metrics to measure this. Example:

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

- **Name**—The attribute name
- **Type**—Select from the drop-down list. Valid values are:
    - `Text`
    - `Percentage`
    - `Number`
    - `Date`
    - `Lookup Table`
- **Value**—The attribute value.
- **Description**—Free-format text description of the attribute.

## Permissions Settings

Business Structure details include a **Permissions** tab on which users with the relevant permissions can view and edit permissions settings for all users of the selected node.

### Permissions Table

| Permission | Description |
|---|---|
| Read | Permission to read information and receive updates about the object. |
| Create | Permission to create objects in this folder. |

| Permission | Description |
|---|---|
| Change | Permission to change the properties of the object. The Change permission is the same as allowing “Write” access. |
| Execute | Permission to perform a predefined action or set of actions with respect to the object. |
| Delete | Permission to delete the object. |
| Read Permissions | Permission to read the access control settings for the object. |
| Change Permissions | Permission to change the access control settings for the object. |
| Execute | Permission to perform a predefined action or set of actions with respect to this object. |
| Propagate | For container objects (such as Tenants). The Propagate check box controls whether to propagate this set of elementary permissions to the child objects. By default, the check box is selected). |

### Actions

- **Add Access Group**—Displays the **Select Access Group** panel from which you can select one of the available Access Groups to add to this node and for whom you can then configure permissions.
- **Add Person**—Displays the **Select Person** panel from which you can select one of the available Persons to add to this node and for whom you can then configure permissions.
- **Replace Recursively**—Enables you, upon confirmation, to remove permissions for all child objects of this container and replace them with the permissions defined in this container.

## Data Mart

> **Important**
> You cannot add a new Data Mart using this configuration component.

### Display Options

#### Filters and Constraints

Configuration Server respects tenancy permission settings. You can access only those objects that you have been granted permissions and privileges to access.

You can filter the contents of this list in two ways:

1. Type the name or partial name of an object in the **Quick Filter** field.
2. Click the cube icon to open the **Tenant Directory** filter panel. In this panel, click the Tenant that you want to select. Use the **Quick Filter** field in this panel to filter the Tenant list.

You can sort the items in the list by clicking a column head. Clicking a column head a second time reverses the sort order. You can add or remove columns by clicking **Select Columns**.

To select or de-select multiple objects at once, click **Select**.

### Data Fields

Each entry is shown with the following data fields:

- **Name**—Name of the solution hosting iWD Data Mart.

## General

- **Application**—Name of the iWD Runtime Node application attached to the selected Solution. There is a one-to-one relationship between Solutions requiring Data Mart and iWD Runtime Node applications. The iWD Runtime Node can be detached from the Solution by selecting a blank name in this field.
- **Host**—Host where the iWD Runtime Node is installed. Selectable from the list of hosts configured in GAX Configuration Manager.
- **Port**—Port assigned to iWD Runtime Node. Numeric field; the value must be between 1 and 65535 inclusive. The port must be unique within the host.
- **ETL Scripts Directory**—The directory on the server in which iWD Data Mart ETL scripts are stored. For example, the default path used when iWD Data Mart is installed is **C:\Program Files\GCTI\iWD Data Mart\etl**. **Note:** Unicode symbols in the path are not supported.
- **Configuration Server's Database Access Point**—The name of the Database Access Point associated with Configuration Server. Required for Data Mart's `Load Config` job.
- **Number of Threads**—Performance tuning: the size of the thread pool.
- **Ignored Dimensions**—Performance tuning: the list of dimensions that will be ignored by the Load Intraday job. One dimension per line.
- **Default Dimension Key**—Performance tuning: the default value which will be used for ignored dimensions' keys.
- **Clear Dimension Cache**—Enables or disables persistence of the dimension's cache between Load Intraday job runs. With value `true`, the dimension's cache is cleared when the Load Intraday job completes. With value `false` (default), the cache persists between Load Intraday job runs. Requires restart of the IWD Runtime Node to take effect.

> **Important**
>
> iWD History Node may receive reporting events, from Interaction Server, exceeding the predefined column length. By default, Interaction Server will truncate these values

while iWD Datamart will not. If iWD Datamart is failing on Load Intraday job it is up to the user to decide how data must be processed:

- The complete value must be preserved. In this case, increase the column length in both the databases, that is, Interaction Server's DB (interactions table) and iWD Datamart's DB.
- The truncated value is acceptable. Add the configuration option manually in the Runtime Node configuration object, this will be applied without a restart of iWD Datamart. On the **Options** tab, under the `settings` section add **truncate-columns** with its value set to `true`.

## Logging

The **Logging** tab configures internal logging capabilities within the iWD Runtime Node.

- **Log Level**—The Service log level. This should be set to `Info` unless otherwise instructed by Genesys Technical Support. The possible log levels are:
  - `Debug`—The most detailed informational events that are most useful in debugging an application.
  - `Info`—Informational messages that highlight the progress of the application.
  - `Warning`—Potentially harmful situations.
  - `Error`—Error events that might not affect the application's ability to run.
  - `Trace`—Turns on all logging.
  - `Off`—Turns off all logging.
- **Log Directory**—The directory in which the log files will be stored, for all services. If it starts with / (on Unix-based operating systems) or a drive letter (on Windows), an absolute path will be used; otherwise, the path is relative from the iWD Runtime Node installation directory.
  **Note:** It is strongly recommended that you only set the file path to a directory on a local machine, not a remote location such as a shared network drive. Logging to a remote location can severely impact performance.
- **Log Age**—Sets the number of days that log files should be kept in the system. A value of 0 disables this limit.
- **Log Size**—Sets a limit on the size of a single log file, in megabytes. A value of 0 disables this limit.
- **Log Files**—Sets a limit on the number of log files that are kept for this service, excluding the current log file. A value of 0 disables this limit.
- **Log to Console**—Determines (`true`/`false`, default = `false`) whether to log events to the console.
- **Enable Centralized Logging**—Check this checkbox to enable centralized logging to Message Server.

## Database

The **Database** tab defines a connection to a Data Mart database server. The configured database and user must exist in the database server. The user must have read/write permissions to the database.

- **Application**—The name of the Database Access Point application associated with Data Mart instance.
- **Database**—The name of the database. This is available only for MS SQL Server.
- **SID**—Oracle System ID of the database. The Oracle System ID (SID) is used to uniquely identify a particular database on a system. This is available only for Oracle database.
- **Server**—The database server. Selectable from list of configured hosts.
- **Port**—The TCP port number of the database server.
- **User Name** —The database user name.
- **Password**—The password for the database.
- **Auto-Sync**—The iWD Data Mart database will be initialized automatically the first time the Database Service and Kettle ETL Service are started. If the **Auto-Sync** option is selected, this initialization is automatic, and the Database Service will also check for updates to the iWD Data Mart database whenever a new version of iWD Data Mart is installed. When selected, the **Auto-Sync** option will also initialize ETL plug-ins.
- **JDBC URL**—Add a specific URL here for the Data Mart database used by iWD Data Mart.
- **JDBC Driver Class**—Should be set if you are going to use a custom JDBC driver different from the following default drivers:
  - MS SQL—`com.microsoft.sqlserver.jdbc.SQLServerDriver`
  - Oracle—`oracle.jdbc.OracleDriver`
  - PostgreSQL—`org.postgresql.Driver`

## Stat Server

The **Stat Server** tab configures the Statistics Adapter job and defines a connection to Genesys Stat Server. Statistics Adapter processes the statistical data created by the Aggregate Stats ETL job and writes stat-types and filters in the configuration for Genesys Stat Server. CCPulse+ requests iWD statistics from Stat Server, and reads the stat-types and filters from the Stat Server configuration.

### Important

Multiple Stat Servers could be specified manually via the Runtime Node configuration object. On the **Options** tab, provide a list of Stat Servers separated by semicolons for the following option:

- **[stat-server]/name**—<StatServer_1>;<StatServer_2>;...;<StatServer_N>

- **Application**—The Stat Server's application name. Selectable from list of installed Stat Server applications. Each Data Mart requires separate Stat Server instance.
- **Dimension Mapping**—Defines how statistical dimensions are mapped.
  - **Filter**—Dimensions are mapped to CCPulse+ filters.
  - **Virtual Queue**—Dimensions are mapped to Genesys virtual queues.
- **Virtual Queue Name**—Name of the Genesys virtual queue to which statistics are distributed.

Applicable only if Dimension Mapping is set to Virtual Queue.

- **Service Index**—Statistical service index for configuration options. This should be unique inside the set of indexes, assigned to statistical services served by the one instance of Genesys Stat Server.
- **Extension File Name**—Required to support a Genesys reporting environment with multiple instances of Stat Server Java Extensions. This is the name of the Stat Server Java extension jar file (**BPR_iWD_Extension.jar**). This file is saved to the Stat Server installation directory during installation of the iWD Stat Extensions. You can find the location of this file in Stat Server configuration options as the value of the **java-libraries-dir** option in the **[java-config]** section.
- **Extension Section Name**—Required to support a Genesys reporting environment with multiple instances of Stat Server Java Extensions. This property maps to the section name for the specific Stat Server Java Extension in the Stat Server configuration.
- **JDBC Driver Class**—Should be set if you are going to use a custom JDBC driver different from the one used by iWD Data Mart.
- **JDBC Driver JAR File**—The .jar file with the JDBC driver. The path is relative to the directory specified as **java-libraries-dir** in the Stat Server configuration.
- **JDBC URL**—Add a specific URL here for the Data Mart database used by Stat Server.

### Warning

The Statistics Adapter job does not check that the JDBC Driver Class and JDBC Driver JAR File values are consistent. It is the user's responsibility to enter a valid class name for this driver JAR.

### Important

JDBC URL and JDBC Driver options should be provided only when Stat Server's DBMS or JDBC driver differs from iWD Data Mart ones. Otherwise leave these options unchanged.

## Schedules

The **Schedules** tab configures execution schedule of three Data Mart job groups. The syntax follow standard CRON scheduling expression. For example, the following expression will cause the job to be executed every 15 minutes:

```
0 0,15,30,45 * * * ?
```

For more information about CRON scheduling, see [http://www.quartz-scheduler.org/documentation/quartz-2.1.x/tutorials/crontrigger Quartz Scheduler documentation]

- **Intraday**—The schedule for the Intraday job group: Load Config, Load Intraday, Aggregate Intraday, Aggregate Stats and Statistic Adapter. Typically scheduled to run every 15 minutes.
- **Historical**—The schedule for the Historical job group: Load Historical, Aggregate Historical and Maintain. Typically scheduled to run once a day, after midnight.

## Expirations

The **Expirations** tab configures the Maintain job, which deletes expired facts from Data Mart tables.

- **Record Details**—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.
- **Aggregation 15 min**—The number of days after which the data will be removed from 15-minute aggregation tables.

## Tenant Attributes

The **Tenant Attributes** tab enables selection of up to 5 of a tenant's custom attributes, that will be loaded into the CUSTOM_DIM dimension and associated to the tenant via the **CUSTOM_DIM_KEY** field.

- **Custom Attribute 1-5**—User-configured custom Tenant attributes, selectable from the list of Custom Attributes attached to the Tenant.

## Department Attributes

The **Department Attributes** tab allows selection of up to 5 of a departments's custom attributes that will be loaded into the CUSTOM_DIM dimension and associated to the departments via the **CUSTOM_DIM_KEY** field.

- **Custom Attribute 1-5**—User-configured custom Department attributes, selectable from the list of Custom Attributes attached to any Department within the Solution.

## Process Attributes

The **Process Attributes** tab allows selection of up to 5 of a process' custom attributes, that will be loaded into the CUSTOM_DIM dimension and associated to the processes via the **CUSTOM_DIM_KEY** field.

- **Custom Attribute 1-5**—User-configured custom Process attributes, selectable from the list of Custom Attributes attached to any Process within the Solution.

## Task Attributes

The **Task Attributes** tab defines up to 10 names of a task's custom attributes that will be loaded into the `task_fact` custom attribute fields (`CUSTOM_ATTRIBUTE 1-10`). Names must start with a letter, and only underscores and alphanumeric characters are supported.

## Dimension Mapping

The **Dimension Mapping** tab defines up to 5 comma-separated names of a task's custom attributes that will be loaded into the CUSTOM_DIM dimension and associated to the task via the **CUSTOM_DIM_KEY** field. Names must start with a letter, and only underscores and alphanumeric characters are supported.

# Lookup Tables

## Overview

You 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. Although business rules are managed in the Genesys Rules System, it is still possible to create rule parameters that use values from iWD Lookup Tables. Example: the **taskChannels** parameter in the iWD Standard Rules Template presents the user with a list of task channels that are read from an iWD Lookup Table.

The **taskChannels** parameter is configured as a database type rule parameter. The configuration of that parameter instructs the Genesys Rules Authoring Tool how to query the Configuration Server database to retrieve the values of the out-of-the-box iWD Lookup Table called **channels**. To create additional rule parameters that will retrieve the values from other Lookup Tables, you can make copies of the **taskChannels** parameter and modify the SQL query, changing the name of the Lookup Table from **channels** to the name of your Lookup Table.

## Display Options

### Filters and Constraints

Configuration Server respects tenancy permission settings. You can access only those objects that you have been granted permissions and privileges to access.

You can filter the contents of this list in two ways:

1. Type the name or partial name of an object in the **Quick Filter** field.
2. Click the cube icon to open the **Tenant Directory** filter panel. In this panel, click the Tenant that you want to select. Use the **Quick Filter** field in this panel to filter the Tenant list.

You can sort the items in the list by clicking a column head. Clicking a column head a second time reverses the sort order. You can add or remove columns by clicking **Select Columns**.

To select or de-select multiple objects at once, click **Select**.

### Data Fields

Each entry is shown with the following data fields:

- **Name**—The element's name.

## Actions

### To add a new Lookup Table

Either:

- From the List view, click **New** and complete the Lookup Table's details.

- Display the details of a Lookup Table and click **Clone**, then edit the details.

### To add a new key/label pair to a Lookup Table

Display the Lookup Table by selecting it, then click **Add** and complete the new details.

### Other Actions

From this context you can **Delete** or **Move** this Lookup Table. You can move the Lookup Table only to another Tenant. Runtime IDs are not moved in the Move function—you must create a new one for the Lookup Table in its new Tenant.

> **Warning**
> Deleting or Moving a Lookup Table can have huge implications for the operation of a contact center. Do not undertake these without serious consideration.

# Updating the Interaction Server databases and related configuration objects

## Purpose

To ensure iWD automatically runs the update scripts on the Interaction Server databases for compatibility with iWD 9.0, creates the necessary Business Attributes in Configuration Server, and configures the completed-queues option for Interaction Server.

## Prerequisites

- Interaction Server is correctly installed. (Refer to the eServices (Multimedia) 8.1 Deployment Guide).
- The installation of iWD components as outlined in this chapter is completed up to this point. In particular, iWD Plug-in for GAX must be installed.
- The JDBC connection URL is provided in Interaction Servers' DAP as described on the Interaction Server Configuration page.

## Procedure

1. Stop the Interaction Server application.
2. Log into iWD Plug-in for GAX.
3. Open **Business Structure**.
4. Navigate to your iWD tenant.
5. Navigate to your Solution from the navigation tree and select the **Migration** tab. The **Interaction custom properties and migration issues** table on the right side notifies you of the updates that must be made.
6. Press the **Configure** button.
7. Start the Interaction Server application.

### enable-revoke-from-agent

An Interaction Server configuration option, enable-revoke-from-agent, enables enhancements to task management in the Global Task List. Specifically, this option allows an `Assigned` interaction to be revoked from an employee desktop when the interaction is put on hold, canceled, or completed, from the Global Task List or through an integrated capture point.

# Working with Task Attributes and Interaction Properties

These topics describe the task and attribute properties that are supported in iWD 9.0.

## Task Attributes

Most of the iWD task attributes can be set when a task is created or updated through a Capture Point, although some, such as interaction ID, are set by iWD components or by Interaction Server. Most of the task attributes are displayed in the **Attributes** tab when a task is selected in the Global Task List. These attributes are maintained as pieces of attached data of the interaction, as it is stored in the Interaction Server's interactions database table. Some of the attributes are stored in independent columns in that database table, while others are stored in a binary (BLOB) format in a column in the interactions table called **flexible_properties**.

There are many reasons to update or access the data stored in the iWD task attributes, including:

- Setting the value of one or more task attributes as part of an iWD message such as `CreateTask`, when working with a capture point.
- Reading or updating task attributes in business rules.
- Using the data in the **Condition**, **Order**, and **Segmentation** tabs of Views in Genesys Business Processes.

> **Important**
>
> You cannot use properties with a `Timestamp` data type on the **Segmentation** tab.

- Reading or updating the data contained in task attributes within a routing strategy.
- Making the data available to an agent or knowledge worker desktop application, either to display to the agent or to facilitate a screen pop.
- Filtering the display of the Global Task List.

### Core Attributes

*Core attributes* describe the fundamentals of a task. These attributes are used in assembling tasks in the Global Task List, based on the business value and priority that are defined within iWD. Core attributes are either set automatically by iWD, or provided by the source system (through the Capture Point interface). The following are some iWD core task attributes:

- `activationDateTime`
- `assignedDateTime`
- `assignedToUser`

- businessValue
- captureId
- category
- completedDateTime
- dueDateTime
- expirationDateTime
- heldDateTime
- interactionid
- mediaType
- priority
- queue
- queueTarget
- queueType

### Extended Attributes

*Extended attributes* provide additional context about a task, enabling you to tailor the service-level agreement (SLA) rules for managing tasks on the Global Task List. They can also aid in customizing current-day and historical reporting. For example, use of several capture dates allows an organization to measure performance against the date and time at which an order or loan application was received by the source system or was submitted by the customer via a web form. The following are some extended task attributes:

- customerId
- customerSegment
- productSubtype
- productType
- requestedAgent
- requestedSkill
- resultCode
- sourceCreatedDateTime
- sourceDueDateTime
- sourceFirstCreatedDateTime
- sourceProcessSubtype
- sourceProcessType
- sourceTenant

## Task Attribute Mapping

| Task Attribute | Description | Column Name in Interactions Table | Type | Interaction Attached Data Key | Label in Global Task List | CP Direction |
|---|---|---|---|---|---|---|
| **Core Attributes** | | | | | | |
| Age | | • received_at<br>• moved_to_queue_at<br>• queue<br>• completed_at | String | • ReceivedAt<br>• MovedToQueueAt<br>• Queue<br>• CompletedAt | Age | Out |
| ActivationDateTime | | IWD_activationDateTime | Timestamp | IWD_activationDateTime | Activation D/T | In/Out |
| AssignedDateTime | | assigned_at | Timestamp | assigned_at | Assigned D/T | Out |
| AssignedToUser | | assigned_to | String | RTargetAgentSelected | Assigned To | Out |
| BrokerId | | Id | String | InteractionID | ID | Out |
| businessCalendarId | | IWD_businessCalendarId | String | IWD_businessCalendarId | Business Calendar ID | Out |

| Task Attribute | Description | Column Name in Interactions Table | Type | Interaction Attached Data Key | Label in Global Task List | CP Direction |
|---|---|---|---|---|---|---|
| BusinessValue | | IWD_businessValue | Integer | IWD_businessValue | Business Value | In/Out |
| CaptureID | | external_id | String | ExternalId | Capture ID | In/Out |
| CapturePointID | | IWD_capturePointId | String | IWD_capturePointId | Capture Point (value is shown instead of ID) | Out |
| Category | | IWD_category | String | IWD_Category | Category | In/Out |
| Channel | | IWD_channel | String | IWD_channel | Channel | In/Out |
| CompletedDateTime | | completed_at | Timestamp | CompletedAt | Completed D/T | Out |
| CreatedDateTime | | received_at | Timestamp | ReceivedAt | Created D/T | In/Out |
| DepartmentId | | IWD_departmentId | String | IWD_departmentId | Department (name is shown instead of ID) | Out |
| DueDateTime | | IWD_dueDateTime | Timestamp | IWD_dueDateTime | Task Due D/T | In/Out |
| ExpirationDateTime | | IWD_expirationDateTime | Timestamp | IWD_expirationDateTime | Expiration D/T | In/Out |
| HeldDateTime | | held_at | Timestamp | HeldAt | Held D/T | Out |
| InteractionID | | Id | String | InteractionID | ID | Out |
| mediaType | | media_type | String | MediaType | Media Type | In/Out |
| Moved to Queue D/T | | moved_to_queue_at | Timestamp | MovedToQueueAt | Moved to Queue D/T | Out |
| Priority | | priority | Integer | Priority | Priority | In/Out |
| ProcessId | | IWD_processId | String | IWD_processId | Process (name is shown instead of ID) | In/Out |
| Queue | | queue | String | Queue | Queue | Out |
| Queue Target | | • agent_id<br>• agent_group_id<br>• place_id<br>• place_group_id | String | • WorkbinAgentId<br>• WorkbinAgentGroupId<br>• WorkbinPlaceId<br>• WorkbinPlaceGroupId | Queue Target | Out |

| Task Attribute | Description | Column Name in Interactions Table | Type | Interaction Attached Data Key | Label in Global Task List | CP Direction |
|---|---|---|---|---|---|---|
| Queue Type | | • workbin<br>• agent_id<br>• agent_group_id<br>• place_id<br>• place_group_id | String | • Workbin<br>• WorkbinAgentId<br>• WorkbinAgentGroupId<br>• WorkbinPlaceId<br>• WorkbinPlaceGroupId | Queue Type | Out |
| ReprioritizeDateTime | | IWD_reprioritizeDateTime | DateTime | IWD_reprioritizeDateTime | Reprioritization D/T | Out |
| SolutionId | | IWD_solutionId | String | IWD_solutionId | | Out |
| Status | | • queue<br>• state<br>• is_locked | String | • Queue<br>• State<br>• IsHeld | Status | Out |
| TenantId | | IWD_tenantId | String | IWD_tenantId | Tenant | Out |
| **Extended Attributes** | | | | | | |
| CustomerSegment | | IWD_ext_customerSegment | String | IWD_ext_customerSegment | Customer Segment | In/Out |
| CustomerId | | IWD_ext_customerId | String | IWD_ext_customerId | Customer ID | In/Out |
| ProductType | | IWD_ext_productType | String | IWD_ext_productType | Product Type | In/Out |
| ProductSubtype | | IWD_ext_sourceProductSubtype | String | IWD_ext_productSubtype | Product Subtype | In/Out |
| RequestedAgent | | IWD_ext_requestedAgent | String | IWD_ext_requestedAgent | Requested Agent | In/Out |
| RequestedAgent Group | | IWD_ext_requestedAgentGroup | String | IWD_ext_requestedAgentGroup | Requested Agent Group | In/Out |
| RequestedPlace Group | | IWD_ext_requestedPlaceGroup | String | IWD_ext_requestedPlaceGroup | Requested Place Group | In/Out |
| RequestedSkill | | IWD_ext_requestedSkill | String | IWD_ext_requestedSkill | Requested Skill | In/Out |
| ResultCode | | IWD_ext_resultCode | String | IWD_ext_resultCode | Result Code | In/Out |
| SourceCreated Date/Time | | IWD_ext_sourceCreatedDateTime | DateTime | IWD_ext_sourceCreatedDateTime | TOS Created D/T | In/Out |
| SourceDue Date/Time | | IWD_ext_sourceDueDateTime | DateTime | IWD_ext_sourceDueDateTime | TOS Due D/T | In/Out |
| SourceFirstCreated Date/Time | | IWD_ext_sourceFirstCreatedDT | DateTime | IWD_ext_sourceFirstCreatedDT | TOS First Created D/T | In/Out |

| Task Attribute | Description | Column Name in Interactions Table | Type | Interaction Attached Data Key | Label in Global Task List | CP Direction |
|---|---|---|---|---|---|---|
| SourceProcessType | | IWD_ext_sourceProcessType | String | IWD_ext_sourceProcessType | TOS ProcessType | In/Out |
| SourceProcessSubtype | | IWD_ext_sourceProcessSubType | String | IWD_ext_sourceProcessSubtype | TOS Subprocess | In/Out |
| SourceTenant | | IWD_ext_sourceTenant | String | IWD_ext_sourceTenant | TOS Tenant | In/Out |
| **Custom Attributes** | | | | | | |
| Archive Destination | Archive Destination was maintained from iWD 8.0 and 8.1.0 releases. However, due to changes in task<br>archiving, it is no longer necessary to populate this attribute through business rules. | in flexible_properties | String | ArchiveDestination | Archive Destination | |
| IWD_isContactServer | Used for inner purposes of iWD BP | IWD_isContactServer | Integer | IWD_isContactServer | Is Contact Server | |
| IWD_isDone | Used for inner purposes of iWD BP | IWD_isDone | Integer | IWD_isDone | Is Done | |

# Configuring Custom Attributes

## Overview

When you capture a task from a source system and that task has custom attributes in it, you will need to configure the system properly so that each custom attribute is recognized. If a custom attribute is recognized, it:

- Appears on the Global Task List with a user-friendly label and can be used in advanced and custom filters.

- Can be properly populated in iWD Data Mart.
- Can be used in the Condition, Order and Segmentation tabs of Views in Genesys Business Processes (except for attributes with Timestamp data type).

If you want a task attribute inside the binary data in the **flexible_properties** column to be recognized, you must create an Interaction Custom Property that corresponds to that attribute.

## Prerequisites

- The custom attribute (referred as a Custom Interaction Property) is configured using the procedures in the eServices Interaction Properties Reference Manual.

> ### Important
> iWD treats both section and option names defined in the **Attributes Values** option as case sensitive.

## Procedure

1. In GAX, open **Business Attributes**, expand **Interaction Custom Properties** and open its **Attribute** values.
2. In your new attribute value, go to the **Annex** tab and create section **[iWD]**.
3. In the new **[iWD]** section, create an option called `type`, with value `string`, `int` or `date` for string, number or timestamp column respectively. This option affects how your attribute will be treated in **Global Task List** and **Filters** tabs in iWD Manager.
4. If required, configure user-friendly labels for any custom attribute, that will appear in the Global Task List's **Attributes** tab, in the list of **Advanced Filters**, and the list of attributes that are used when you are building custom filters. See the **Editing the iWD Manager's Resource File** tab on this page.

> ### Important
> If the `type` option is not specified, the attribute will be treated as `string`.

> ### Important
> There are specific columns in the **interactions** table that you should not change. Please refer to Chapter 8, "Interaction Properties", in the eServices 8.1 User's Guide.

## Editing the iWD Manager's Resource File

By default, the on-screen labels you will see on the Global Task List's **Attributes** tab of the **Task Details** panel for all custom task attributes will be the interaction user data keys. These labels may not be very user-friendly. For example they might include multiple words concatenated, or they might use underscore characters or have odd capitalization.

To configure user-friendly labels for custom task attributes by adding into a resources file, the labels that you want to see and that will be used by iWD Manager.

1. Stop your application server.
2. Navigate to **\config\evo\cmc\ui\resources**.
3. Create a new text file called **resources_custom.properties** (if it does not already exist).
4. In the **resources_custom.properties** file, add a line for each custom attribute for which you want to define an on-screen label, by using the following format: `TASK_attribute_key=Attribute Label` where: `attribute_key` matches the interaction user data key of the custom attribute. `Attribute Label` is the user-friendly string you want to display on the Global Task List. For example, if your custom attribute's user data key is `MyAttribute`, but you want it to display as `My Custom Attribute` on the Global Task List, your entry would look like this: `TASK_MyAttribute=My Custom Attribute.`

   > **Important**
   >
   > This label will be used in the following places in the Global Task List:
   >
   > - As the label for this attribute on the **Attributes** tab of the **Task Details** panel.
   > - In the **Select** columns to add drop-down list on the **Filters** page.
   > - As the column header if you add this custom attribute as a column in the task table when you are creating a custom filter.

5. In the same **resources_custom.properties** file that was used in the **Adding new database column(s) and Interaction Custom Property for each Custom Attribute procedure**, add a new line for each custom attribute, by using the following format: `FILTER_ATTR_attributedb_column_name=Attribute Label` where: `attributedb_column_name` matches the column name that you added to the interactions table. `Attribute Label` is the user-friendly string you want to display in the Global Task List. For example, if the name of the database column is `my_attribute`, this line in the **resources_custom.properties** file would look like this: `FILTER_ATTR_my_attribute=My Custom Attribute.`

   > **Important**
   >
   > This label will be used in the following places in the Global Task List:
   >
   > - In the **Advanced Filters** drop-down list.
   > - In the **Attributes** drop-down list that you use to add a custom attribute in a filter criterion, when you are creating or editing a Global Task List filter.

6. Restart the **iwd_manager** application on your application server for the changes to become effective.

# IWD Data Mart Services

If you have used the Genesys-recommended Genesys Administrator Extension installation procedure, iWD Services are automatically populated into the iWD Plug-in for GAX when the iWD Runtime Node is started. iWD Services are now configured in the iWD Plug-in for GAX. Please refer to **the iWD GAX Plug-in Help** for details about configuring iWD services. The list of services is as follows:

- Configuration Monitor
- Configuration Server Connector
- DM Initialize
- DM Load Config
- DM Load Intraday
- DM Aggregate Intraday
- DM Aggregate Stats
- Statistics Adapter
- DM Load Historical
- DM Aggregate Historical
- DM Maintain

Services are configured per Solution, and each Solution must be configured under a tenant in GAX. Therefore, before configuring services, you must configure a Tenant and Solution.

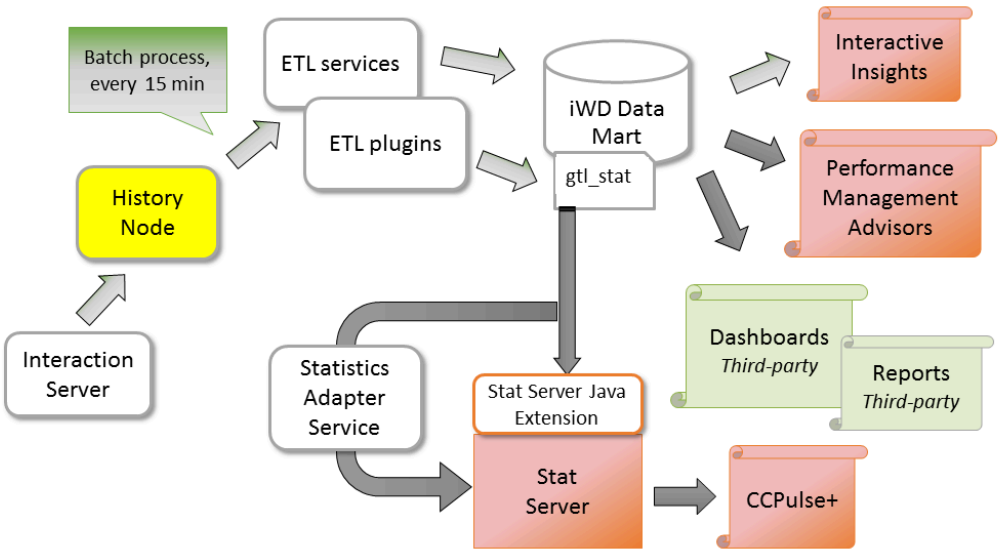
# IWD Reporting

## Overview

With an increasing number of choices in the marketplace and higher expectations of service quality, the ability to measure the efficiency and effectiveness of customer service delivery becomes a key component of success. iWD streamlines an often cumbersome reporting process through:

- *Cradle-to-grave* reporting from the time that a task enters the contact center until its completion.
- Consolidated reporting across the various systems that are involved in customer-service delivery: fax servers, workflow, customer-relationship management, and Genesys Customer Interaction Management.
- Reporting that is based on business context—with business process, customer segment, and product independent of channel, instead of being limited to interactions, queues, channels, and workflows.

The key to achieving the desired business results is having access to actionable business intelligence. Genesys iWD offers comprehensive reporting, providing management insight into business operation. It provides key indicators of performance both through current-day statistics and on an historical basis. The historical metrics are provided based on aggregates and measures that are populated by scheduled ETL processes, which extract data from the Genesys Interaction Server Event Log database and load it into the iWD Data Mart. This next figure provides a functional overview of iWD's reporting components. Third-party services can reference iWD statistics from the GTL_STAT table (GTL, for Global Task List) to display data in dashboards or within Genesys CCPulse+.




> **Important**
> Each iWD solution requires its own Data Mart.

## Database Objects

iWD Data Mart consists of the following database objects:

- Fine-grained fact tables—Store all attributes that are associated with tasks (**I_TASK_FACT/ H_TASK_FACT** tables), work-related events (**I_TASK_WORK_FACT/H_TASK_WORK_FACT** tables), when the task was assigned to one or more agents; and a full audit history of the task (**I_TASK_EVENT_FACT/H_TASK_EVENT_FACT** tables).

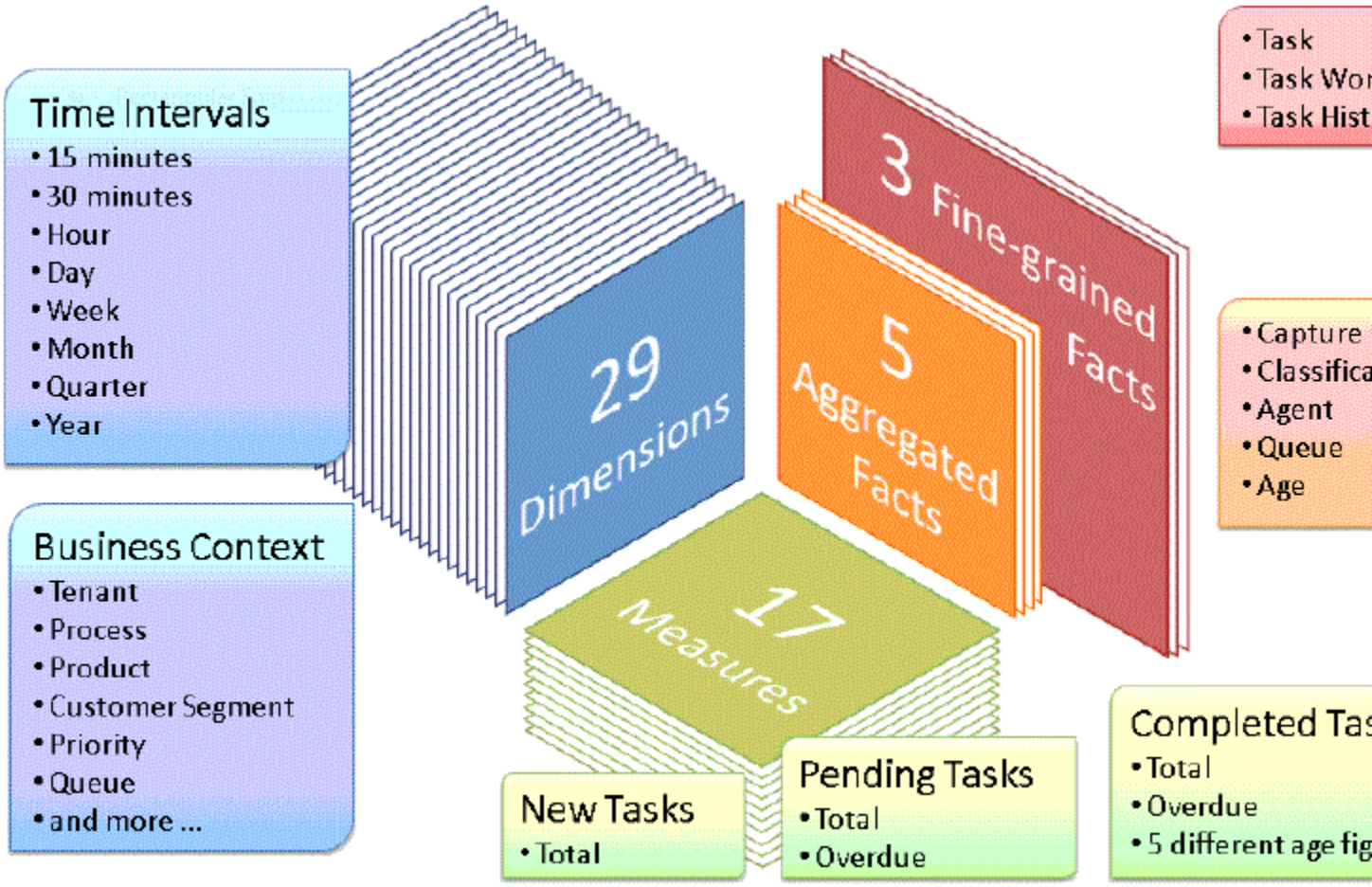
> **Important**
> The term *agent* refers to any resource, configured as a Person object in Configuration Server, that can handle tasks. (Within Genesys Administrator, Person objects appear as User objects within the interface.)

- Aggregated fact tables—Describe tasks in an iWD-oriented context across the various stages, or the

iWD life cycle of the task, from capture and classification to distribution to agent.

- Dimensions—Describe task attributes that are common across the fact tables in iWD Data Mart, such as iWD business process, priority, business value, and date and time. Fact tables link to these dimensions through keys.
- Measures—Represent numerical values (such as totals, durations, averages, minimums, and maximums) that are stored in aggregated fact tables across intraday and historical intervals. For example, the total number of completed tasks by 15-minute interval by an iWD process and business value would be captured within the **I_TASK_CAPT_FACT_15MIN** intraday table.

When they are connected to existing enterprise data marts, including Genesys Info Mart, analysts gain access to comprehensive views of the entire customer experience. Analytical reporting leverages existing business intelligence tools, such as those that are provided by Pentaho (which is an open-source product suite for business intelligence) or through a host of commercial products from Cognos or SAP Crystal Reports.




iWD Data Mart–Dimensions, Measures, and Facts

## Reporting Services

The Statistics Adapter Service allows performance of custom aggregations on data and sends the resulting statistics to Genesys Stat Server. Statistics then can be viewed in CCPulse+ or any other Stat Server client.

Historical reporting is enabled by a number of ETL jobs that transform and load iWD runtime data into a separate reporting database that is called the iWD Data Mart for a list of preconfigured ETL jobs). Essentially, the iWD Data Mart 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. Refer to the **iWD 9.0 Data Mart Reference Guide** for more information.

## Statistics for a Chosen Dimension Available Through Stat Server

The list of statistics coming from iWD Data Mart that are available in Stat Server is as follows:

- `GTL_ACTIVE`—Number of active tasks for each department and process configured in the system.
- `GTL_HELD`—Number of held tasks for each department and process configured in the system.
- `GTL_PENDING_15MIN`—Number of pending tasks, as of the trailing 15-minute interval, for each department and process.
- `GTL_OVERDUE_15MIN`—Number of overdue tasks, as of the trailing 15-minute interval, for each department and process.
- `GTL_NEW_15MIN`—Number of new tasks, as of the trailing 15 minute interval, for each department and for the solution.
- `GTL_NEW_30MIN`—Number of new tasks, as of the trailing 30 minute interval, for each department and for the solution.
- `GTL_NEW_60MIN`—Number of new tasks, as of the trailing 60 minute interval, for each department and for the solution.
- `GTL_COMPLETED_15MIN`—Number of completed tasks, as of the trailing 15 minute interval, for each department and for the solution.
- `GTL_COMPLETED_30MIN`—Number of completed tasks, as of the trailing 30 minute interval, for each department and for the solution.
- `GTL_COMPLETED_60MIN`— Number of completed tasks, as of the trailing 60 minute interval, for each department and for the solution.

In addition to the above statistics, a set of filters is created—for each Solution, Department and Process there is one filter created. The filter definition looks like this:

- Filter name—`GTL_<object type>_<object runtime id>`

### The object type can be one of these values:

- SLT
- DPT

- PRC

For example, the filter name for a Process with runtime ID T2_C3_P4 is:

```
T2_C3_P4: GTL_PRC_T2_C3_P4
```

- Filter definition—`PairExists("Dimension", "<object type>_<object runtime id>")`

For example, the filter definition for a Process with runtime id = T2_C3_P4 is:

```
PairExists("Dimension", "PRC_T2_C3_P4")
```

# Configuring CCPulse+ for iWD

## Purpose

In Genesys, iWD current-day 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.

## Prerequisites

- The iWD Data Mart database has been created, and the corresponding Database service has been configured and is running.
- The Aggregate Statistics ETL job is configured and running.
- Stat Server must be running, with the iWD Stat Extensions installed.
- The Statistics Adapter service must be configured and running.

## Procedure

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

> **Important**
>
> This is the virtual queue that was specified during manual installation of iWD. iWD Setup Utility is not supported in release 9.0

### Next Steps

To be able to view iWD current-day statistics in Genesys CCPulse+, it is necessary to create a CCPulse+ template. The following procedure explains how to create a CCPulse+ template.

# Creating a CCPulse+ template

## Purpose

To create a CCPulse+ template to use to view iWD current-day statistics.

## Procedure

1. Open the CCPulse+ template wizard by selecting **Template Wizard...** from the **Tools** menu.
2. 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**.
3. 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.
4. 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).
5. Click the **>>** button to add the selected statistic to the newly created statistic group.
6. In the **Requested Statistics** tree view, select the newly added statistic, and then click the **Properties** button.
7. In the **Statistic Properties** dialog box enter `Alias` (which is how the statistic is displayed in CCPulse+). Set the following values, then click **OK**:
   - **Insensitivity** to `1`
   - **Interval Type** to `Selection and GTLAggregated`
   - **Notification Mode** to `Time Based`
   - **Filter** to the filter type that represents the required iWD dimension
8. Add more statistics and statistic groups, if necessary, and then click the **Next** button.
9. In the **Graph** dialog box, adjust graph parameters, if necessary, and then click **Finish.**
10. 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.
11. In the **Real-Time Data Template** dialog box, select the previously created template from the list, and click **OK.**

# Configuring Pulse for iWD

## iWD Pulse Widgets Templates

iWD Runtime Node is distributed along with several out-of-the-box widget templates. They can be found in the ***<iWD_Runtime_Node_install_folder>*/pulse** folder. To import them into Pulse, please refer to this article.

## iWD Datamart New Task

### Widget template description

This widget template allows you to create a widget representing the number of new tasks for the last 15-, 30- and 60-minute intervals.

### Statistics used in the widget template

| Statistic | Description |
|---|---|
| GTL_NEW_15MIN | The number of new tasks received in the last 15-minute interval, for each solution and department. |
| GTL_NEW_30MIN | The number of new tasks received in the last 30-minute interval, for each solution and department. |
| GTL_NEW_60MIN | The number of new tasks received in the last 60-minute interval, for each solution and department. |

## iWD Datamart Completed Task

### Widget template description

This widget template allow you to create a widget representing the number of completed tasks for the last 15-, 30- and 60-minute intervals.

### Statistics used in the widget template

| Statistic | Description |
|---|---|
| GTL_COMPLETED_15MIN | The number of completed tasks received in the last 15-minute interval, for each solution and department. |
| GTL_COMPLETED_30MIN | The number of completed tasks received in the last 30-minute interval, for each solution and department. |
| GTL_COMPLETED_60MIN | The number of completed tasks received in the last 60-minute interval, for each solution and department. |

## iWD Datamart Active and Pending Task

### Widget template description

This widget template allow you to create a widget representing the number of

- Active and held tasks in the system.
- Overdue and pending tasks at the end of the given time interval.

### Statistics used in the widget template

| Statistic | Description |
|---|---|
| GTL_ACTIVE | The number of active tasks for each department and process configured in the system. |
| GTL_HELD | The number of held tasks for each department and process configured in the system. |
| GTL_PENDING_15MIN | The number of pending tasks (where the status is Queued, Assigned, or Held) at the end of the given time interval, for each department and process. |
| GTL_OVERDUE_15MIN | The number of pending overdue (due date < end of the given time interval) tasks (where the status is Queued, Assigned, or Held) at the end of the given time interval, for each department and process. |

## iWD Datamart New Completed 15 min

### Widget template description

This widget template allow you to create a widget representing the number of new and completed

tasks for the last 15 minutes.

### Statistics used in the widget template

| Statistic | Description |
|---|---|
| GTL_NEW_15MIN | The number of new tasks received in the last 15-minute interval, for each solution and department. |
| GTL_COMPLETED_15MIN | The number of completed tasks received in the last 15-minute interval, for each solution and department. |

## iWD Datamart New Completed 30 min

### Widget template description

This widget template allow you to create a widget representing the number of new and completed tasks for the last 30 minutes.

### Statistics used in the widget template

| Statistic | Description |
|---|---|
| GTL_NEW_30MIN | The number of new tasks received in the last 30-minute interval, for each solution and department. |
| GTL_COMPLETED_30MIN | The number of completed tasks received in the last 30-minute interval, for each solution and department. |

## iWD Datamart New Completed 60 min

### Widget template description

This widget template allows you to create a widget representing the number of new and completed tasks for the last 60 minutes.

### Statistics used in the widget template

| Statistic | Description |
|---|---|
| GTL_NEW_60MIN | The number of new tasks received in the last 60-minute interval, for each solution and department. |

| Statistic | Description |
|---|---|
| GTL_COMPLETED_60MIN | The number of completed tasks received in the last 60-minute interval, for each solution and department. |

## iWD Datamart Active vs Held

### Widget template description

This widget template allow you to create a widget representing the number of active and held tasks in the system.

### Statistics used in the widget template

| Statistic | Description |
|---|---|
| GTL_ACTIVE | The number of active tasks for each department and process configured in the system. |
| GTL_HELD | The number of held tasks for each department and process configured in the system. |

## iWD Datamart Overdue vs Pending

### Widget template description

This widget template allows you to create a widget representing the number of overdue and pending tasks at the end of the given time interval.

### Statistics used in the widget template

| Statistic | Description |
|---|---|
| GTL_PENDING_15MIN | The number of pending tasks (where the status is Queued, Assigned, or Held) at the end of the given time interval, for each department and process. |
| GTL_OVERDUE_15MIN | The number of pending overdue (due date < end of the given time interval) tasks (where the status is Queued, Assigned, or Held) at the end of the given time interval, for each department and process |

## Integrated Pulse Widget Templates

Genesys Pulse is distributed along with a set of predefined templates. There are templates available for creating reporting widgets for iWD Agent Activity and iWD Queue Activity. There's much more detailed information about how to work with templates and widgets in the template creation topic from the Pulse documentation.

You can also view a list of the available iWD Agent Activity and Queue Activity statistics in the Report Templates topic of the Pulse documentation.

## Pulse Dashboard Configuration

After importing widget templates you may want to create your own dashboard or wallboard. Please refer to this article to find out how to create dashboards and wallboards.

To add widgets based on imported or provided templates, please refer to this article.

# Configuring Data Mart for GCXI Reporting

The GCXI Microstrategy reporting solution for iWD Data Mart uses an abstract data model pre-built on the Microstrategy server. To make GCXI Reporting work properly, the underlying aggregate plug-in tables must be created in the iWD Data Mart database schema. Currently, the data model on GCXI is pre-built for the Classification, Capture, Queue and Agent plug-ins.

The process of creating the appropriate database objects is triggered automatically during start-up of iWD Runtime Node. For startup to complete correctly, it is necessary to verify that the appropriate plug-ins are enabled. To enable the Classification, Capture, Queue or Agent plug-ins, please use the information here.

# Configuring iWD for Multiple Business Processes

- For the IWD Business Process for IRD, see the **Working with the IWDBP for IRD** document for full details.
- For the IWD Business Process for Composer, see the **Working with the IWDBP for Composer** document for full details.

iWD supports multiple business processes (or, more generally, interaction workflows) in one tenant. For example, it is possible to create separate business processes for different media types. As a result, business processes may be simpler and easier to maintain. Independently created business processes define their own sets of interaction queues. iWD allows you to configure custom queues for each iWD solution. Each independent business process requires a separate iWD solution.

## Interaction Queues

iWD recognizes seven interaction queues. By default they are created by the standard iWD business process (IWDBP) and have the following names in the IWD Business Process for IRD:

- `iWD_New`
- `iWD_Captured`
- `iWD_Queued`
- `iWD_Completed`
- `iWD_Rejected`
- `iWD_Canceled`
- `iWD_ErrorHeld`

In the iWDBP for Composer they have the following names:

- `iwd_bp_comp.Main.iWD_New`
- `iwd_bp_comp.Main.iWD_Canceled`
- `iwd_bp_comp.Main.iWD_Captured`
- `iwd_bp_comp.Main.iWD_Completed`
- `iwd_bp_comp.Main.iWD_ErrorHeld`
- `iwd_bp_comp.Main.iWD_Queued`
- `iwd_bp_comp.Main.iWD_Rejected`

If there is more than one business process, customized queues must be configured for each solution in the iWD GAX Plug-in. The set of allowed queues is taken from all defined business processes. The

names of the chosen queues will then be used by both iWD Manager and iWD Data Mart instead of the default ones.

### Adding Custom Queue Names to Interaction Server

You must also ensure that the names of all customized queues for completed tasks are added to the list of queue names in Interaction Server in the completed-queues

## Filters

Pre-defined filters on the Global Task List have explicit queue names in their conditions. When custom queues are defined, it is necessary to update filters' criteria with generic queue names instead of explicit ones. For example, the filter criterion `Queue is iWD_Completed` or `Queue is iwd_bp_comp.Main.iWD_Completed` should be changed to `Queue is Completed`. After such a change the filter will work correctly in all solutions with defined custom queues for completed tasks.

The following filter criteria support generic queue names:

- `Queue is '{queue}'`,
- `Queue is not '{queue}'`.

When you choose one of these criteria in the **Filters** page of iWD Manager, a drop-down list appears in place of `'{queue}'`. There are seven generic queue names available on the list:

- `Canceled`
- `Captured`
- `Completed`
- `ErrorHeld`
- `New`
- `Queued`
- `Rejected`

## Integrated Capture Points

Integrated Capture Points' options must be set accordingly so that they can put new or modified interactions in the correct interaction queues. When an integrated Capture Point is connected with an iWD solution, its options are automatically synchronized with the solution. The following options are updated in Capture Points to work with a customized iWD business process:

### JMS Capture Point and File Capture Point

- `inbound-transformer-parameters`

    - CancelQueues
    - CompleteQueues
    - RestartQueues
- outbound-transformer-parameters
    - CancelQueues
    - CompleteQueues
    - ErrorHeldQueues
    - RejectQueues
    - RestartQueues

### Web Service Capture Point and Database Capture Point

- iwd-parameters
    - CancelQueues
    - CompleteQueues
    - ErrorHeldQueues
    - RejectQueues
    - RestartQueues

### All Capture Points

- default-values
    - Queue

The following mapping between configured queues and Capture Points' options is maintained.

| Capture Point Option | iWD Solution's Queue | Default Value in IRD | Default Value in Composer |
|---|---|---|---|
| default-values/Queue | New | iWD_New | iwd_bp_comp.Main.iWD_New |
| RestartQueues | New | iWD_New | iwd_bp_comp.Main.iWD_New |
| CompleteQueues | Completed | iWD_Completed | iwd_bp_comp.Main.iWD_Completed |
| RejectQueues | Rejected | iWD_Rejected | iwd_bp_comp.Main.iWD_Rejected |
| CancelQueues | Canceled | iWD_Canceled | iwd_bp_comp.Main.iWD_Canceled |
| ErrorHeldQueues | Error Held | iWD_ErrorHeld | iwd_bp_comp.Main.iWD_ErrorHeld |

The options are updated whenever a user changes any of the queues in the iWD Solution configuration in GAX. They are also modified when a user changes the assigned Solution in the Capture Point's configuration in GAX. If no Solution has been assigned to the Capture Point, the queue options can be set manually.

# Database Cluster Configuration

## Access to the Database Layer

Support for cluster level solutions like Oracle RAC or MSSQL Cluster requires configuration in configuration objects that access the database layer. Most of these configuration objects (except the regular one) provide an extension called JDBC  URL that must be used in cluster solutions like Oracle RAC or MSSQL Cluster. Objects and their configurations are listed below:

- The DAP in iWD Runtime Node that accesses the iWD Data Mart database (JDBC URL can be set up using:
  **GAX -> Configuration/iWD Plugin -> Datamart -> chosen solution -> Database field "JDBC URL"**)
- The DAP in iWD Runtime Node that accesses the Configuration Server database (JDBC URL can be configured manually by adding a `jdbc` section to the DAP options, with key `url` and a suitable value)

## Configuration for Oracle Support

Oracle 12c and Oracle 12c RAC are both supported in release 9.0.x. Configurations with and without Single Client Access Name (SCAN) are supported.

## JDBC URLs

There are different Oracle JDBC URLs depending on whether your configuration has SCAN enabled or disabled.

### SCAN enabled

The Oracle JDBC URL for the Data Mart DAP is:

```
jdbc:oracle:thin@//rac-scan:1521/orcl.abc.xyz.domain.com
```

### SCAN disabled

The Oracle JDBC URL for the Data Mart DAP is:

```
jdbc:oracle:thin@(DESCRIPTION=(LOAD_BALANCE=on)(ADDRESS=(PROTOCOL=TCP)(HOST=RACNode1)
(PORT=RACNodePort1))(ADDRESS=(PROTOCOL=TCP)(HOST=RACNode2)(PORT=RACNodePort1))
(CONNECT_DATA=(SERVICE_NAME=RACServiceName)))
```

## Limitations

- If RAC failover is forced when iWD Data Mart jobs are running, iWD Data Mart jobs will finish with an error. The next execution of iWD Data Mart jobs finish correctly without errors.
- The database initialization process for the iWD Data Mart database does not support failover. Initial configuration should be carried out made in a stable environment. If an error occurs during initialization the database must be dropped and created again from scratch.
- In this initial release of Oracle support, only one SCAN is supported, though there might be up to three in a given environment.

# Install and Configure iWD Web

## Background

Before iWD Web, capturing work items has always been handled either by out-of-the-box capture adapters like a Web service interface, XML and IBM WebSphere MQ adapters, or by Genesys eServices equivalents.

iWD Web is a web-based, easy-to-deploy workload management solution. It enables organizations to automate and optimize the distribution and handling of tasks and increase workforce efficiency. Purpose-built to deliver rapid time to value, it provides the capability to upload task lists without the need to connect to other source systems, such as CRM, ERP, and legacy systems, leveraging export functionality commonly available to business users.

Instead of having to rely on manual task distribution to your team via spreadsheets (and also CSV files,) or team members having to pick their next work item manually from a CRM, ERP, or legacy system, iWD Web enables you to create new lists with task definitions based on data that business or operations managers can upload from .CSV, .XLS and .XLSX files. You can use your CRM, ERP, or legacy system's export to spreadsheet/CSV file feature to download a list of work items that your team needs to work on and import that file to iWD Web, which reads and treats every line of your spreadsheet as a task to be analyzed and properly distributed to your available team members.

## Prerequisites

Before installing and starting iWD Web, you must have followed as a minimum steps 1 thru 10 as described in the Task Summary **here**.

# Installing iWD Web Manually on Windows

1. Make sure you have the IP for Windows available on your server. You must also have access to iWD Web templates.
2. Login to GAX.
3. Import the iWD Web templates (**iwdweb** and **iwdwebapp**).
4. Create an application based on the **iwdweb** template.

> **Important**
>
> iWD Web requires two applications. The first one is of `Server` type, and is used to store configuration options in Configuration Server as well as defining Interaction Server connections. The second one is of type `ThirdPartyApplication`. It is used to define user credentials/permissions for login operation in the iWD Web user interface. This step refers to the server application. The Server type is `ThirdPartyServer` in releases prior to 9.0.005 and `Genesys Generic Server` in releases from 9.0.005+.

5. Login to your Windows server.
6. Create an iWD Web log directory: **/GCTI/iWD**
7. Locate and double-click **setup.exe** in the **iWD Web** directory of the iWD DVD.
8. The iWD Web Installation Wizard opens. Click **Next** in the **Welcome** screen.
9. In the **Connection Parameters** to the Configuration Server screen, enter the login details to connect to Genesys Configuration Server and then click **Next**:
   **Host name**—The host of Genesys Configuration Server

   **Port**—The port that is used by Genesys Configuration Server

   **User name**—The user name of the Person (or User) as defined in Genesys Configuration Manager or Genesys Administrator.

   **Password**—The password that is associated with the Person (or User).
10. Choose the iWD Web application and click **Next**.
11. Choose the destination location for iWD Web.
12. Click **Next**.
13. In the **Ready to Install** screen, click **Install** to begin the installation of iWD Web.
14. When the installation has been completed, click **Finish**, then check the configuration information here.

# Installing IWD Web Manually on Linux

1. Make sure you have the IP for Linux available on your server. You must also have access to iWD Web templates.
2. Login to GAX.
3. Import the iWD Web templates (**iwdweb** and **iwdwebapp**).
4. Create an application based on the **iwdweb** template.

> **Important**
>
> iWD Web requires two applications. The first one is of `Server` type, and is used to store configuration options in Configuration Server as well as defining Interaction Server connections. The second one is of type `ThirdPartyApplication`. It is used to define user credentials/permissions for login operation in the iWD Web user interface. This step refers to the server application. The Server type is `ThirdPartyServer` in releases prior to 9.0.005 and `Genesys Generic Server` in releases from 9.0.005+.

5. Create a user who will be used to manage the iWD Web application (in the sample installer below, `genesys`).
6. Re-login using your iWD Web user credentials.
7. Create an iWD Web log directory, grant write permissions and pass its absolute path by one of the following methods:
   - Environment variable `IWDWEB_LOG_DIR`
     For example; `$ export IWDWEB_LOG_DIR=/var/log`
   - Command line argument: `--iwdweb.logging.dir`
     For example; `./iwd_web.sh --iwdweb.logging.dir=/var/log/ -app "iwdweb_srv"...`
   - Add an option to the iWD Web Server application object **iwdweb.logging.dir** in the **[boot]** section.
     For example; **[boot] iwdweb.logging.dir=/var/log**
   - If none of the above is done, the default location is used instead as follows; **/GCTI/iWD**
8. Execute the installer:

```
[genesys@xx-yyy-zzzzzz ip]$ pwd
/home/genesys/ip
[genesys@xx-yyy-zzzzzz ip]$ ls -la
total 45456
drwxrwxrwx.  2 genesys genesys     4096 Jul  8 12:50 .
drwx------. 38 genesys genesys     4096 Jul  8 12:50 ..
-rwxrwxr-x.  1 genesys genesys 42876663 Jul  8 12:50 data.tar.gz
-rwxrwxr-x.  1 genesys genesys     3527 Jul  8 12:50 genesys_silent.ini
-rwxrwxr-x.  1 genesys genesys    53716 Jul  8 12:50 gunzip
-rwxrwxr-x.  1 genesys genesys  2772389 Jul  8 12:50 installer.tar.gz
-rwxrwxr-x.  1 genesys genesys     5188 Jul  8 12:50 install.sh
-rwxrwxr-x.  1 genesys genesys     1322 Jul  8 12:50 ip_description.xml
-rwxrwxr-x.  1 genesys genesys     8114 Jul  8 12:50 iscript.tar.gz
-rwxrwxr-x.  1 genesys genesys    13539 Jul  8 12:50 ospatchlist.txt
-rwxrwxr-x.  1 genesys genesys     5102 Jul  8 12:50 read_me.html
-rwxrwxr-x.  1 genesys genesys   738069 Jul  8 12:50 tar
```

```
-rwxrwxr-x.  1 genesys genesys    36270 Jul  8 12:50 tar_gunzip_license.txt
[genesys@xx-yyy-zzzzzz ip]$ ./install.sh
************************************************
* Welcome to the Genesys 9.x Installation Script *
************************************************

Installing iWD Web, version 9.x.aaa.bb

Please enter the hostname or press enter for "xx-yyy-zzzzzz" =>

Please enter the following information about your Configuration Server:

Configuration Server Hostname =>pl-byd-iwd163
Network port =>2020
User name =>selenium
Password =>

Please choose which application to install:
1 : iWD Web_9.x.aaa.bb_xx-yyy-zzzzzz
=>1

Please enter full path of the destination directory for installation =>/genesys/
iWD_Web_9.x.aaa.bb

Extracting tarfile: data.tar.gz to directory: /genesys/iWD_Web_9.x.aaa.bb
iWD_Web.sh
iwdweb-9.x.aaa.bb.jar

Installation of iWD Web, version 9.x.aaa.bb has completed successfully.

[genesys@xx-yyy-zzzzzz ip]$ cd /genesys/iWD_Web_9.x.aaa.bb
[genesys@xx-yyy-zzzzzz iWD_Web_9.x.aaa.bb]$ ls -la
total 48048
drwxrwxrwx.  2 genesys genesys     4096 Jul  8 12:53 .
drwxr-xr-x. 55 genesys genesys     4096 Jul  8 12:53 ..
-rwxrwxr-x.  1 genesys genesys     1322 Jul  8 12:53 ip_description.xml
-rwxrwx---.  1 genesys genesys 49156325 Jul  3 19:18 iwdweb-9.x.aaa.bb.jar
-rwxr-x---.  1 genesys genesys     2392 Jul  8 12:53 iWD_Web.sh
-rwxrwxr-x.  1 genesys genesys    13539 Jul  8 12:53 ospatchlist.txt
-rwxrwxr-x.  1 genesys genesys     5102 Jul  8 12:53 read_me.html
[genesys@xx-yyy-zzzzzz iWD_Web_9.x.aaa.bb]$
```

9. When the installation has been completed, check the configuration information here.

# Creating and Configuring iWD Web in GAX

## Creating the iWD Web Application

1. Create a new iWD Web application (iWD Web app_9.x.aaa.bb_xxxyyyzzz) based on the `iwdwebapp` template.

> **Important**
> This should be of type ThirdPartyApplication.

2. On the **Connections** tab add the connection to the Configuration Server. If you have both primary and backup Configuration Servers, add a connection only to the primary.
3. Click **Save** to save the Application object.

## Configuring the iWD Web Server Application

1. Open the iWD Web Server Application which was created during the installation process.
2. On the **Connections** tab:
    1. Add a connection to the Interaction Server
    2. Add a connection to the Configuration Server. If you have both primary and backup Configuration Servers, add a connection only to the primary.

> **Important**
> iWD Web supports HA Configuration Server deployment. The primary Configuration Server must be up and running when iWD Web is started.

> **Important**
> iWD Web supports HA Interaction Server deployment. For the `default` profile, HA for Interaction Server API client is supported. For the `wscp` profile, HA for capture points is not supported.

> **Important**
> iWD Web supports HA Message Server deployment.

3. On the **Ports** tab add the port where iWD Web REST server will listen for connections. The **Ports** tab lists communication ports used by the clients of an application to connect to a server. To support specific high-availability configurations, more than one server can be registered on the same port within the same host. Otherwise, do not assign the port number to any other server on the same host. There could be two ports maximum; one secured, and one not secured. If there is more than one port of each type, only one of them will be used, the others ignored.
4. Make sure that the correct tenant is added to the **Tenants** tab.
5. Configure the application details on the **Application Options** tab:
    1. Configure the **boot** section:
        1. Set the `boot:iwdweb.config_server.client_app_name` property to the same as the iWD Web Application name (`iWD Web app_9.x.aaa.bb_xxxyyyzzz`).

> **Important**
> This is a ThirdPartyApplication type. If this application doesn't already exist, you must create it.

        2. Set the profile parameter in the **spring.profiles.active** option to either one of the following values:
            - `default, form-login`
            - `wscp, form-login`

           The first part of the parameter (`default` or `wscp`) determines how iWD Web will interact with Interaction Server:
           `default` – directly via the Genesys PSDK and Interaction Server API client. This value is used by default if the option is missing.

           `wscp` – through a Web Service Capture Point (already embedded in Interaction Server).

           The second part of the parameter sets the **Security** profile for which you must use the default `form-login` value to ensure Spring security supports authentication against Configuration Server using username/password credentials entered on a login form and to enable CSRF protection. This Security profile is also set by Spring Boot's **spring.profiles.active** option, so both the entered values must be present and separated with a comma.
        3. Configure any other iWD Web properties as required. See **here** for more details.
    2. If you selected `default` in the previous step:
        1. Set the `ixn:solution_id (runtime_id) (SLT1)`.
        2. Set the `ixn:capture_point_id` property. Use the iWD Web Capture Point name `iWD_Web_WSCP`.
        3. `ixn:department_id, ixn:process_id` may be used to set default values for Department and Process.

4. Configure any other required properties.

6. Click **Save** to save the Application object.

# Configuring High Availability

## Purpose

iWD Web is a web application and its multiple instances can be deployed in a load-balanced environment to support High Availability.

## Procedure

1. Create an application for every web server of iWD Web based on the iWD Web Server template.
2. Configure each application as described in Configuring the iWD Web Server Application. Use the same value for the **boot:iwdweb.config_server.client_app_name** option for all the Server applications.
3. Deploy an IP for each of the Server applications created in Step 1.
4. Set up some load-balancing software against the configured web servers of iWD Web. iWD Web supports any third-party load balancer that supports sticky sessions. You should configure session affinity (sticky sessions) based on JSESSIONID. The following is a sample configuration for Apache httpd deployed against two iWD Web servers running on the same host:

```
<Proxy balancer://iwd_web_balancer>
BalancerMember http://localhost:8090/iwd_web lbset=0
BalancerMember http://localhost:8091/iwd_web lbset=1
ProxySet lbmethod=bybusyness
</Proxy>
RewriteEngine On
RewriteRule ^/iwd_web/(.*) balancer://iwd_web_balancer/$1 [P] stickysession=JSESSIONID
ProxyPassReverse "/iwd_web" "balancer://iwd_web_balancer"
```

# Configure a Web Service Capture Point for iWD Web

> **Important**
>
> To configure automated upload in release 9, you cannot use a Web Service Capture Point. You must create an IWD Web capture point

.

## If you selected Spring profile `wscp`

If you chose the `wscp` value for the Spring profile, you need to configure a Web Service Capture Point (WSCP) for iWD Web to use. The quickest way to do this is to clone an existing WSCP and make configuration changes specific to your iWD Web application and environment. Or, you could create a new one in GAX using the information in this topic.

Once a `wscp` is created, provide the URI to it as an application option of iWD Web Server application object: **[boot] iwdweb.wscp.uri = <WSCP_URI>**

For example, **<WSCP_URI>** may look like this:

```
http://<wscp_host>:<wscp_port>/Genesys/Interaction/<wscp_object_name>/WebServiceCapturePoint
```

## If you selected Spring profile `default`

Either create a new WSCP dedicated to iWD Web or clone an existing one and make the following configuration changes:

1. Change `settings:capture-point-type` to a dummy capture point type—for example, `X_webservice_X`.

> **Important**
>
> If the `default` profile is used, only the `runtime_id` property from the Capture Point is used. The dummy capture point that you create should not be used by any other applications, tools or clients. The iWD Web WSCP capture point `runtime_id` property is added to each task which is sent to Interaction Server. The `runtime_id` property allows iWD Manager to identify the originator of iWD Web tasks. Based on this property, Data Mart is also able to identify these tasks and calculate aggregates which are used to calculate statistics.

2. Make a note of the newly created capture point's `runtime_id`.

# iWD Web Configuration

iWD Web configuration options are described here.

# Automatic Upload of XLS/CSV Files

The upload engine uses default folder names for the iWD Web capture point:

- Inbound
- Error
- Processed
- Partially Processed

All of these can be renamed in the **Application Options** tab of the iWD Web capture point configuration object, if needed.

## Process Overview

1. The source system must be configured to put new files in the **Inbound** folder. The application does not support modification of files in the **Inbound** folder. If you need to make any changes to an input file, delete it from the folder, edit it, then put it back.
2. The uploader automatically picks up new files once their presence is detected. The files are processed then moved to the relevant folder (**Processed** / **Partially Processed** / **Error**).
   - If all tasks from the file were successfully submitted to Interaction Server, the file is moved to the **Processed** folder.
   - If parsing of the file fails (because of malformed input) or if Interaction Server rejects all tasks from the file, it is placed in the **Error** folder.
   - If a subset of tasks fails, the file is placed in the **Partially Processed** folder and extended with a **Submit Status** column containing error messages or submission IDs for failed/successful tasks. To resubmit these tasks, open the file and make edits, then replace it in the **Inbound** folder to await the next file pickup.
3. In addition to moving files, the uploader also lists automated upload task statuses on the **Jobs Lists** tab in iWD Web, in the same way as for manual uploads.

# Creating a Capture Point for IWD Web for Automated Task Upload

In order to use the automatic job upload feature in release 9.0, you must configure a special IWD Web capture point in GAX. This type of capture point is similar to a File capture point. In its configuration you specify:

- The locations/folders for:
  - Inbound tasks—The location to which the source system will write new XLS/XLSX/CSV files, to be fetched by the iWD Web importer.
  - Partially processed tasks—The location for files for which at least one record could not be processed.
  - Processed tasks—The location for files after they have been successfully processed by the iWD Web importer.
  - Error—The location for files containing a format error which has caused processing to fail.
- The frequency with which the Inbound folder will be scanned.

## Creating the iWD Capture Point application in GAX

### Import the iWD Capture Point template in GAX

1. Navigate to **Environment -> Application Templates**.
2. Click **New**.
3. Click **Import Application Template** in the top right corner.
4. Choose the correct **iWDCapturePoint.apd** file.
5. Click **OK**.

### Creating the iWD Capture Point Configuration Object in GAX

1. Navigate to **Environment -> Applications**.
2. Click **New**.
3. Select the correct the **iWD Web Capture Point** template.
4. Select the **Host** (**General** tab) and **Tenant** (**Tenant** tab).
5. In the **Application Options** tab, set the configuration options for the capture point as shown in the table below.
6. Open the iWD Web Server application and add the newly created iWD Capture Point(s) to the connections list.

## Capture Point Application Options

| Option Name | Changes Take Effect | Mandatory | Default Value | Type | Description |
|---|---|---|---|---|---|
| **[default-values]** | | | | | |
| InteractionType | After restart | Yes | Inbound | String | Interaction type |
| InteractionSubtype | After restart | Yes | InboundNew | String | Interaction subtype |
| IWD_solutionId | After restart | Yes | SLT1 | String | Solution ID |
| IWD_tenantId | After restart | Yes | SYSTEM | String | Tenant runtime ID |
| MediaType | After restart | Yes | Workitem | String | Interaction default media type |
| Queue | After restart | Yes | IWD_New | String | Default interaction queue |
| **[settings]** | | | | | |
| capture-point-type | After restart | Yes | iwdweb | String | A capture point is instantiated only if this option is set to `iwdweb`. |
| inbound-scan-interval | After restart | No | 10 | Integer | Defines (in seconds) how often the inbound directory is scanned for new files. Value range: `0` - Integer.MAX_VALUE (32-bit) |
| inbound-directory | After restart | Yes | | String | Defines the path to the inbound directory in the file system |
| error-directory | After restart | No | Error | String | Defines the path to the error directory. If it exists, it must be different from all other directories specified in the options. |
| partially- | After restart | No | PartiallyProcessed | String | Defines the |

| Option Name | Changes Take Effect | Mandatory | Default Value | Type | Description |
|---|---|---|---|---|---|
| processed-directory | | | | | path to the partially processed directory. If it exists, it must be different from all other directories specified in the options. |
| processed-directory | After restart | No | Processed | String | Defines the path to the processed directory. If it exists, it must be different from all other directories specified in the options. |
| move-non-compatible-from-inbound-default | After restart | No | true | Boolean | If set to `true`, non-compatible files from the **inbound** directory (files that do not have either .XLS, .XLSX or .CSV file extensions) are moved to the **Error** directory. If the **error-directory** option is not configured, or its value is empty, and the **move-non-compatible-from-inbound-default** option is set to `true`, non-compatible files will be deleted from the **inbound** directory. If set to `false`, non-compatible files in the **inbound** directory are |

| Option Name | Changes Take Effect | Mandatory | Default Value | Type | Description |
|---|---|---|---|---|---|
| | | | | | ignored. |
| skip-lines | After restart | No | `0` (zero) | Integer | Number of lines to skip before header lookup. Value range: `0` - Integer.MAX_VALUE (32-bit) |
| **[mapping]** | | | | | |
| <ATTRIBUTE>.name | After restart | Yes | N/A | String | Column name to attach the attribute to. |
| <ATTRIBUTE>.df | After restart | No | N/A | String | Date format for current attribute. Use JVM SimpleDateFormat pattern as specified in the official Oracle documentation. |
| <ATTRIBUTE>.value | After restart | No | N/A | String | Override file value with provided value only if cell value is empty . |

# Defining Custom Mapping for iWD Web Capture Point

To use custom column names or date formats and/or define default values for missing columns in your input files, do the following:

1. Navigate to **GAX > Application > *iWD Web Capture Point name (iwd web or iwd web capture point)* > Application Options**.
2. Edit the **[mapping]** section as follows:
   Each attribute may have up to three rows in the configuration table—All attributes are optional.
   - ATTRIBUTE_NAME.name—The column name in your input file. Must be defined if you want to create a default `ATTRIBUTE_NAME.value` or `ATTRIBUTE_NAME.df`.
   - ATTRIBUTE_NAME.value—The default value for the corresponding attribute if it is not present in the input file. Note that default values will be applied to all rows of all input documents regardless of whether they have the corresponding column. That is, if you define a default value—say, `customer_segment.value = "My Segment"`, every incoming task without `customer_segment` will be assigned this value as default.
   - ATTRIBUTE_NAME.df—Only applicable to DateTime fields. Allows the definition of a custom date format for input files. Such syntax is defined in the Oracle documentation in section "Date and Time Patterns". Also see "Examples" on the same page.
   - ATTRIBUTE_NAME—The corresponding Business Attributes/Interaction Custom Properties name (not to be confused with the Interaction Custom Properties display name). Please see Working with Task Attributes and Interaction Properties.

# IWD and the Genesys Rules System

These topics describe how iWD interoperates with the Genesys Rules System.

> **Tip**
>
> The following documents will be helpful:
>
> - **The Genesys Rules System Overview**
> - **The Genesys Rules System deployment summary**
>
> Unless otherwise stated, all the procedures described in these topics require a fully installed iWD Manager and Genesys Rules System as a prerequisite.

## IWD and GRS Overview

Genesys Rules System provides all the business rules functionality for the Genesys intelligent Workload Distribution (iWD) solution, a business application for dynamically prioritizing the distribution of work tasks to the people who are best suited to handle them.

For information about how to migrate your existing rule templates and rules, see the Genesys System Migration Guide.

Rule templates are created in the Genesys Rules Development Tool and the templates are published to the rules repository. Users then use the Genesys Rules Authoring Tool to create a rule package that incorporates one or more rule templates. The Rules Authoring Tool is also where users create new rule packages that incorporate rule templates, author rules inside the rule package based on the rule templates, validate the rules, and then deploy their rule package to the Genesys Rules Engine. iWD provides a Standard Rules Template for use with the Genesys Rules System, and the Genesys Rules Authoring Tool (GRAT) can be launched from iWD Manager without the need for separate user authentication.

Client applications such as the iWD business process (IWDBP) then make requests to the Genesys Rules Engine to have rules in the rule package evaluated at various decision points in a task's lifecycle.

The Genesys Rules System is not only used by Genesys iWD. It is also used by other Genesys solutions, including Genesys Conversation Manager. As such, there are some objects that must be properly configured for iWD, when working with the Genesys Rules System. These include:

- Configuring the proper rule template type when creating a new template
- Configuring the proper rule package type when creating a rule package
- Proper use of rule phases.

These objects are described in more detail in these topics.

# Rule Templates

Rule templates for iWD are created in the Genesys Rules Development Tool, which is an Eclipse plug-in that can either be installed into a standalone Eclipse application or can be installed into Genesys Composer.

Rule templates are used to define the building blocks that are used by rules authors to build rules for task classification and prioritization at the Global, Department, and Processes levels of the business structure of an iWD Solution.

Normally rule templates are created or modified by IT personnel. Defining new rule conditions or actions sometimes requires a basic knowledge of the Java programming language, although in many cases it is possible to review the syntax of existing rule conditions and actions as a guide.

> **Important**
>
> For more detail about rule template components, click here (opens in a new window).

All iWD rule templates include a fact called `_GRS_Environment`, which must not be modified. Unlike rule templates for other Genesys solutions, it is not necessary to create any additional facts for iWD rules.

> **Important**
>
> Rule conditions and actions are made up of a Language Expression and a Rule Language Mapping. The Language Expression defines what the rule condition or action does, in plain language. It is what the rules author (who is often a business user) will see when constructing a rule. The Rule Language Mapping defines the same Language Expression in Java code. This is the real code that is executed. Rule Language Mapping follows Drools syntax (which is Java with some minor extensions). Regarding available methods, there is a single variable available to rules called `$data`, which represents task attributes as a key-value collection. It is an instance of the `KeyValueCollection` class which is part of Platform SDK. For more information, see the Platform SDK Developer's Guide (release 7.6 or later).

# The iWD Standard Rules Template

## Overview

iWD comes with an out-of-the-box rule template called the iWD Standard Rules Template, that a user will import into the Genesys Rules Development Tool. This template is the foundation for the rules that are most frequently used in iWD deployments. The template can be used as an example when new rule conditions or actions are required for a particular iWD deployment.

Although the functionality defined in the iWD Standard Rules Template can be extended, Genesys recommends creating one or more custom rule templates for this purpose. One reason for this is that you can use Configuration Server object permissions to control which users can access specific rule templates. For example, you might have one custom rule template for all users of the Genesys Rules Authoring Tool, which includes common rule conditions and actions, to which all users should have access. Then you can have a second custom rule template that contains advanced rule conditions and actions, which you only make accessible to experienced rule authors.

> **Important**
>
> If Genesys provides an updated iWD Standard Rules Template, all custom changes to the previous version will be lost after importing of the updated template.

After you modify an existing rule template, or create a new one, the rule template is published to the Genesys Rules System rules repository. This makes the template available to users of GRAT, so the template can be added to a new rule package, or an existing rule package can be updated with the latest version of a template.

## Importing the iWD Standard Rules Template into the Genesys Rules Development tool and publishing it to the Rules repository

### Purpose

To import the iWD Standard Rules Template into the Genesys Rules Development Tool and then publish it to the Genesys Rules System rules repository. This will make the Standard Rules Template available to rules authors, from within the Genesys Rules Authoring Tool.

1. Launch Eclipse (or Composer).
2. Open the Template Development perspective (if it is not already open).
3. Navigate to **Window > Open Perspective > Other > Template Development**.

4. In the **Eclipse** menu, navigate to **File >Import**.
5. In the **Import** dialog box, select **Existing Projects into Workspace** from within the **General** folder.
6. Click **Next**.
7. Select the **Select root directory** radio button
8. Select **Browse** to browse to the iWD Standard Rules Template folder.

> **Tip**
> The iWD Standard Rules Template will be in the rule **Templates** subdirectory in the folder where the iWD Manager supporting files were installed on your file system. For example, **C:\Program Files\GCTI\iWD Manager\ruleTemplates**.

9. Click **Finish**.

   The iWD Standard Rules Template is now visible in the Eclipse (or Composer) **Project Explorer** view.

10. In the **Project Explorer** view, right-click **iWD_Standard_Rules project** and select **Publish...**.
11. Using the **Publish Template Wizard**, select the **iWD_Standard_Rules** template project.
12. Select the **Edit properties** link.
13. In the **Properties** dialog, under **Template Properties**, ensure that `Type = iWD`.
14. For the **Tenant** property, select the Configuration Server tenant to which this rule template will be associated.
15. Click **OK**.
16. Using the **Publish Template Wizard**, select the **iWD_Standard_Rules** template project.
17. Click **Finish**.

    The **Publishing** dialog, with a progress indicator, will display `Publish successful` after the template project has been successfully published to the Genesys Rules System repository.

> **Important**
> You cannot import the iWD Standard Rules Template project into GRDT if it already exists in the Eclipse workspace. Therefore, if you have a new version of the iWD Standard Rules Template to import, you must first delete the existing one. For this reason, Genesys strongly recommends that you use custom rule templates for new rule conditions or actions, rather than extending the iWD Standard Rules Template.

## Next Steps

If you do not plan to create any custom rule templates at this time, go to the Genesys Rules

Authoring Tool to create a rule package. See **Rule Authoring for IWD**.

To review information about the parameters, conditions, actions, and functions provided in the iWD Standard Rules Template, see the **iWD Standard Rules Template Contents** tab on this page.

# iWD Standard Rules Template Contents

## Conditions, Actions, Parameters, and Functions

The iWD Standard Rules Template defines the most commonly used rule conditions, actions, parameters, and functions.

### Conditions

| Condition | Explanation |
|---|---|
| Business value is `"businessValue_From" to "businessValue_To"` | If the business value of the task is between `businessValue_From` and `businessValue_To`, then... This rule condition is designed to test inclusive conditions. That is, the `"businessValue_From"` and `"businessValue_To"` conditions that are being checked are `">="` and `"<="`, respectively. This must be understood in order to use this rule condition effectively. |
| Capture point is `"capturePoint"` | If the capture point is `capturePoint`, then... This rule condition is designed to be used at the rule package (Global) level, as a way to classify tasks, based on the capture point from which they originated. This is in lieu of the Capture Point-level rules that were available in iWD 8.0 but are not available in the Genesys Rules System. This rule condition requires that the `capturePoint` rule parameter be properly configured so that GRAT can access the list of Capture Points from the iWD configuration database. |
| Channel is `"taskChannels"` | If the specified channel of the task is `taskChannels`, then... This rule condition requires that the `taskChannels` rule parameter be properly configured so that GRAT can access the list of task channels from the iWD confutation database. |
| Department is `"department"` | If the specified department of the task is `department`, then... |
| Due Time is in `"periodFrom" to "periodTo" "periodType"` | If the task due date/time is between `periodFrom` and `periodTo` specified time `periodType`, then... This rule condition is designed to test conditions that are not inclusive. That is, the `"periodFrom"` and `"periodTo"` conditions that are being checked are `">"` and `"<"`, respectively. This must be understood in order to use this rule condition effectively. |
| Integer `"attribute" "operator" "integerValue"` | When the value of a specified custom attribute of type `integer` is greater than/less than/equal to the specified `integerValue`, |

| Condition | Explanation |
|---|---|
| | then... |
| Is first prioritization | When the rules being applied are part of the initial prioritization step (the value of the task attribute `reprioritizeDateTime` is empty), then... |
| Is reprioritization | When the rules being applied are part of the reprioritization step (the value of the task attribute `reprioritizeDateTime` is not empty), then... |
| Is working day | When today is a working day, then... This condition is evaluated, based on the default business calendar assigned to the rule package or a business calendar that has been explicitly assigned to the task in a previous rule. A business calendar must be assigned to the rule or in a separate rule action before this action can be used. |
| Is working time | When the current time is working time, then... This condition is evaluated based on the default business calendar assigned to the rule package or a business calendar that has been explicitly assigned to the task in a previous rule. A business calendar must be assigned to the rule or in a separate rule action before this action can be used. |
| Media type is `"mediaType"` | When the specified media type of the task is `mediaType`, then... |
| No process selected | If process is not assigned to the task, then... |
| Priority is `"operator" "priority"` | If iWD priority of the task is equal to/not equal to/less than/less than or equal to/greater than/greater than or equal to p `riority`, then... |
| Process is `"process"` | If the task is assigned to `process`, then... |
| String `"attribute" equals "stringValue"` | When the value of a specified custom attribute of type `string` is equal to the specified `stringValue`, then... |
| Task is overdue | When the task is overdue, then... |
| Task overdue in `"period" "periodType"` | Checks if task will be overdue after a given time interval. |

## Actions

| Action | Explanation |
|---|---|
| Activate task in "period" "periodType" | Activate the task in the specified period of time, using time unit periodType . The activation date/time is used in the IWDBP business process to determine when the task should be moved from the iWD_Captured queue into the Prioritization routing strategy. |
| Activate task in "period" working "periodType" | Activate the task in the specified period of time, using time unit periodType, according to the task's business calendar (working days, working hours and so on). The activation date/time is used in the IWDBP business process to determine when the task should be moved from the iWD_Captured queue into the Prioritization routing strategy. A business calendar must be assigned to the rule or in a separate rule action before this action can be used. |
| Activate task beginning next working day | Activates the task at the beginning of the next business day. A business calendar must be assigned to the rule or in a separate rule action before this action can be used. |
| Activate task end working day | Activates the task at the end of the working day when the task was received. A business calendar must be assigned to the rule or in a separate rule action before this action can be used. |
| Assign business calendar "businessCalendar" | Assign the specified businessCalendar to the task. |
| Assign distribution point "distributionPoint" | Assign the specified distributionPoint to the task. This rule condition requires that the distributionPoint rule parameter be properly configured so that GRAT can access the list of distribution points from the iWD configuration database. |
| Assign iWD department "department" | Assign the task to the specified department. |
| Assign iWD process "process" | Assign the task to the specified process. |
| Assign WFM Activity "wfmActivity" | Assign the specified Genesys Workforce Management Activity called wfmActivity to the task. To ensure GRS can read objects from Genesys Workforce Management, you must configure the following object in the Configuration Manager under specific tenant's Scripts folder: Name — WFM Connection (or other descriptive name) Type — Data Collection<br><br>Under Annex tab, configure the following section: Section — wfm<br><br>Under this section, configure the following properties: wfmCfgServerApplName —configuration application name of the WFM Server wfmCfgServerUserName —User name that is used to connect to WFM Server wfmCfgServerPassword —Password that is used to connect to WFM Server wfmServerUrl —URL that is used to connect to WFM Server. The URL must have the following format: |

| Action | Explanation |
| --- | --- |
| | `http://<wfm_server_host_name>:<wfm_server_port_number>`<br><br>For example, if the WFM Server is running on a host named `WFMHost001` on port `7023`, the URL would be: `http://wfmhost001:7023` |
| Increase iWD priority `"priority"` | Increase the iWD priority of the task by the specified amount defined in `priority`. |
| Multiply business value `"multiplier"` | Multiply the business value of the task by the specified multiplier. |
| Reprioritize after `"period"` `"periodType"` | Reapply prioritization rules to the task after the specified period of time, using time unit `periodType` from the current time. |
| Reprioritize after `"period"` working `"periodType"` | Reapply prioritization rules to the task after the specified period of time, using time unit `periodType`, according to the task's business calendar (working days, working hours, and so on). A business calendar must be assigned to the rule package, or directly to the task, before this action can be used. |
| Request agent `"agent"` | Request a specific agent for the task. |
| Request skill `"skill"` | Request a specific skill for the task. |
| Request agent group `"agentGroup"` | Specify which `agentGroup` is required to process the task. |
| Request place group `"placeGroup"` | Specify which `placeGroup` is required to process the task. |
| Set activation date from `"customAttribute"` | Set the activation date/time of the task from the specified `customAttribute` of the task. The activation date/time is used in the IWDBP business process to determine when the task should be moved from the `iWD_Captured` queue into the Prioritization routing strategy. |
| Set activation time `"time"` | Set the time when the task will be activated. The activation date/time is used in the IWDBP business process to determine when the task should be moved from the `iWD_Captured` queue into the Prioritization routing strategy. The value for "time" must be entered by the rules author in UTC, because it will be attached to the task in the `IWD_activationDateTime` attribute. The value of this attribute will be displayed to users in the Global Task List based on their operating system's time zone settings. Entering time values in UTC is not intuitive for a business user, therefore, if there is a business reason to attach a specific activation time to a task, you might want to create a custom rule condition that displays the local time (considering the time zone of the business, for example) in the Language Expression, while assigning the corresponding value in UTC, in the Rule Language Mapping. Take care with date changes when specifying the activation time in a local time zone. For example, |

| Action | Explanation |
| --- | --- |
| | 23:00 in GMT-5 is 04:00 in UTC on the next day, so date should also be incremented by one day. |
| Set business value "businessValue" | Set business value of the task to the specified businessValue. |
| Set due date from "customAttribute" | Set the due date/time of the task from the specified customAttribute of the task. |
| Set due time "time" | Set the time when the task is due. The value for "time" must be entered by the rules author in UTC, because it will be attached to the task in the IWD_dueDateTime attribute. The value of this attribute will be displayed to users in the Global Task List, based on their operating system's time zone settings. Entering time values in UTC is not intuitive for a business user, therefore, if there is a business reason to attach a specific due time to a task, you might want to create a custom rule condition that displays the local time (considering the time zone of the business, for example) in the Language Expression, while assigning the corresponding value in UTC, in the Rule Language Mapping. Take care with date changes when specifying the activation time in a local time zone. For example, 23:00 in GMT-5 is 04:00 in UTC on the next day, so date should also be incremented by one day. |
| Set integer "attribute" value "integerValue" | Set the value of a specified custom attribute of type integer to the specified integerValue. |
| Set priority "priority" | Set the iWD priority of the task to the specified value priority. |
| Set string "attribute" value "stringValue" | Set the value of a specified custom attribute of type string to the specified stringValue. |
| Task Due in "period" "periodType" | The task is due after the specified period of time, using time unit periodType. |
| Task Due in "period" working "periodType" | The task is due after the specified period of time, using time unit periodType, according to the task's business calendar (working days, working hours, and so on). A business calendar must be assigned to the rule package, or directly to the task, before this action can be used. |
| Task expires in "period" "periodType" | Sets the task's expiration date/time after the specified period of time, using time unit periodType.<br><br>A task's expiration date/time is what is used in the IWDBP business process to determine when a task should be deleted from the Interaction Server database. |
| Task expires in "period" working "periodType" | Sets the task's expiration date/time after the specified period of time, using time unit periodType, according to the task's business calendar (working days, working hours, and so on). A |

| Action | Explanation |
| --- | --- |
| | task's expiration date/time is what is used in the IWDBP business process to determine when a task should be task should be deleted from the Interaction Server database.<br><br>**Note:** A business calendar must be assigned to the rule package or directly to the task, before this action can be used. |

## Parameters

| Parameter | Description |
| --- | --- |
| `agent` | Presents you with a list of agents' user names that are read from the Genesys Configuration Server database. This list is dynamic; it changes as agents are added or removed. The agents displayed will be based on the access control of the Genesys Rules Authoring Tool user. |
| `agentGroup` | Presents you with a list of Agent Groups that are read from the Genesys Configuration Server database. This list is dynamic; it changes as agents are added or removed. The agent groups displayed will be based on the access control of the Genesys Rules Authoring Tool user. |
| `archive` | Presents you with an enumerated list of possible archive destinations for expired tasks. This parameter uses an `Enum`, which is included in the Standard Rules Template, called `archive`.<br><br>Note: This action has been maintained from iWD 8.0 and 8.1.0 but it is no longer necessary to use it, due to changes in the way task archiving works in the out-of-box iWD business process (IWDBP) in iWD 8.1.1. |
| `attribute` | Enables you to enter text that represents the name of a task attribute. |
| `businessCalendar` | Presents you with a list of business calendars that are defined in the rule package. This list is dynamic; it changes as business calendars are added or removed. |
| `businessValue` | Enables you to enter the numeric value that represents business value. |
| `businessValue_From` | Enables you to enter a numeric value that represents the lower boundary of business value. |
| `businessValue_To` | Enables you to enter a numeric value that represents the upper boundary of business value. |

| Parameter | Description |
|---|---|
| capturePoint | Presents you with a list of Capture Points that is generated dynamically by reading the iWD configuration database. |
| customAttribute | Enables you to enter text that represents the name of a task custom attribute. |
| department | Presents you with a list of departments that are defined in the iWD Solution. This list is dynamic; it changes as departments are added or removed. |
| diffByPeriod | Returns the difference between the two dates in the specified time units (minutes, hours, days). |
| difWorkingDays | Returns the difference between two dates in working days. |
| difWorkingHours | Returns the difference between two dates in working hours. |
| difWorkingMinutes | Returns the difference between two dates in working minutes. |
| integerValue | Enables you to enter an integer value for use in rule conditions or actions that evaluate the value of task custom attributes. |
| mediaType | Presents you with a list of media types that are defined in the Genesys Configuration Server under the tenant. This list is dynamic; it changes as media type attributes get added or removed. |
| multiplier | Enables you enter to enter a numeric value by which some other parameter (such as priority ) will be multiplied in a rule. |
| operator | Provides an enumerated list of operators equal to, not equal to, less than, less than or equal to, greater than, and greater than or equal to, that can be used to compare values of custom attributes of type integer, to values specified in a rule condition. This parameter uses an Enum, which is included in the Standard Rules Template, called operator. |
| period | Enables you to enter a numeric value. Combined with the period type, it gives the actual value of the time period. |
| periodFrom | Enables you to enter a numeric value that represents the start point of a period, in time units, according to period type. |
| periodTo | Enables you to enter the numeric value that represents the end point of a period, in time units, according to period type. |

| Parameter | Description |
|---|---|
| periodType | Presents an enumerated list of types of time periods, such as days, hours or minutes. This parameter uses an Enum, which is included in the Standard Rules Template, called periodType. |
| placeGroup | Presents you with a list of place group names that are read from the Genesys Configuration Server database. This list is dynamic; it changes as place groups are added or removed. The place groups displayed will be based on the access control of the Genesys Rules Authoring Tool user. |
| priority | Enables you to enter the numeric value that represents iWD priority. |
| process | Presents you with a list of processes that are defined in the iWD Solution. This list is dynamic; it changes as processes are added or removed. |
| skill | Presents you with a list of skills that are read from the Genesys Configuration Server database. This list is dynamic; it changes as skills are added or removed. The skills displayed will be based on the access control of the Genesys Rules Authoring Tool user. |
| stringValue | Enables you to enter a string value for use in rule conditions or actions that evaluate the value of task custom attributes. |
| taskChannels | Presents the user with a list of task channels, read from an iWD Lookup Table. |
| time | Enables you to enter a time value that is used in various rule conditions and actions. |
| wfmActivity | Presents a list of WFM activities, retrieved dynamically from the WFM Server. |

## Functions

| Function | Description |
|---|---|
| adjustWorkingDate | Adds or subtracts a given time interval from the given date according to the current business calendar. |
| compareDate | Compares the value of two dates. For a list of possible returned values, see comments within the function. |
| compareInteger | Compares the value of two integers, taking as inputs the two |

| Function | Description |
|---|---|
| | integers and a comparison operator. Returns `true` or `false`. |
| `getCurrentCalendar` | Returns the current business calendar ID. |
| `getCurrentDT` | Returns the current date and time, in UTC. |
| `getDTValue` | Gets the value of a task attribute as date/time. |
| `getIntValue` | Gets the value of a task attribute as an integer. |
| getPeriodDTFrom | Adds or subtracts a given time interval from the given date, regardless of the business calendar. |
| `getStringValue` | Gets the value of a task attribute as a string. |
| `getWFMActivity` | Retrieves WFM Activity. |
| `getWFMActivityContext` | Retrieves WFM Activity context (Business Unit or Site). |
| `increaseIntegerValue` | Increases the value of an integer by adding to it. |
| `isNull` | Is `true` if a given task attribute value is `Null`, otherwise, the value is `false`. |
| `isProcess` | Is `true` if a given task is assigned to a given process, otherwise, the value is `false`. |
| `isWorkingDay` | Returns `true` if the current day is the working day. |
| `isWorkingTime` | Returns `true` if the current time is the working time. |
| `multiplyIntegerValue` | Increases the value of an integer by multiplying it by some factor. |
| `notNull` | Is `true` if a task attribute value is not `Null`, otherwise, the value is `false`. |
| `setDepartment` | Sets the department for a task. |
| `setDepartmentAndProcess` | Sets the department and process of the given task from a given |

| Function | Description |
| --- | --- |
| | string, in `department\|process` format. |
| `setDTValue` | Sets the value of a task attribute as date/time. |
| `setIntValue` | Sets the value of a task attribute as an integer. |
| `setProcess` | Sets the process for a task. |
| `setStringValue` | Sets the value of a task attribute as a string. |
| `setTime` | Sets the time portion of a given date/time value from a string value, in `hh:mm` format. |

# Custom Templates

## Creating a Custom Template

### Procedure

1. Start the Genesys Rules Development Tool.
2. To open the **Template Development** perspective (if not already opened); go to **Window > Open Perspective > Template Development.**

> **Tip**
> If the **Template Development** perspective does not appear in the list, select **Other** and then, select the **Template Development** perspective.

3. Select **File > New > Rule Template Project**.

> **Tip**
> If the **Rule Template Project** type is not visible, select **Other** and then, within the Genesys Rules System folder in the **New** dialog, select **Rule Template Project**.

4. Click **Next**.
5. On the **Rule Template Project** screen of the **New Rule Template Project** dialog, enter a name for the template project and select the Eclipse workspace where the project files will be stored.
6. Click **Next**.
7. On the **Template Properties** screen, for the **Type** property, select `iWD`.
8. For the **Tenant** property, select the Configuration Server tenant to which this rule template will be associated.

> **Tip**
> Optionally, you can provide a description for the rule template project.

9. Click `Finish`.

The custom rule template is displayed in the **Eclipse Project Explorer** view.

### Next Steps

After parameters, conditions, actions, and functions are created as necessary for the custom rule template, you must publish it to the Genesys Rules System rules repository. See steps 8-15 of **Importing the iWD Standard Rules Template into the Genesys Rules Development tool and publishing it to the Rules repository**.

> **Tip**
>
> For more detail about rule template components click here or go to the *Genesys Rules System 8.1 Development Tool Help*.

## Conditions and Actions

### Language Expression Syntax

If a rule action Language Expression contains one or more parameters, the expression must terminate with one of those parameters. For example, the following Language Expression for a rule action generates an error:

```
Set Priority "{priority}" and update
```

whereas, the following Language Expression is valid:

```
Set and Update Priority "{priority}"
```

### Using iWD Lookup Tables in Rules

The Genesys Rules System offers several ways to create rule parameters that will present an enumerated list of items to the rules author as a drop-down list. However, there are certain situations, in which you might want to use a Lookup Table that has been configured in iWD Manager, as the basis for a dynamic rule parameter. This might be valuable because Lookup Tables provide user-friendly name/label pairs that can be changed easily by a business user, while some other types of dynamic parameters, such as List Objects in Configuration Server, cannot be edited easily by a business user.

If you want to use a Lookup Table in a rule parameter, it will be a parameter of type `database`, where the database being read is the iWD configuration database. The best way to create a new rule parameter that will look up values from an iWD Manager Lookup Table is to use an existing parameter as an example. For example, in the iWD Standard Rules Template, you can look at the `taskChannels` rule parameter.

We can see that the SQL query that is executed will read data from several tables in the iWD configuration database:

```
SELECT cmc_lookup_attr_entry.valueKey,cmc_lookup_attr_entry.valueLabel
FROM ((cmc_lookup_attr_entry INNER JOIN cmc_lookup_attr_type
ON cmc_lookup_attr_entry.attributeTypeId=cmc_lookup_attr_type.id)
INNER JOIN cmc_tenant
ON cmc_lookup_attr_type.tenantId=cmc_tenant.id)
WHERE cmc_lookup_attr_type.name='Products'
AND cmc_tenant.name='ABCD';
```

In this example, the only parts of the query that would need to change are the name of the Lookup Table, which is referenced in the query as `cmc_lookup_attr_type.name`, and the name of the tenant, which is referenced in the query as `cmc_tenant.name`. If the Lookup Table is configured under the System tenant, then the `cmc_tenant.name` would be set to `System` in the query.

## Operational Parameters

Operational parameters provide another user-friendly way to make rules dynamic, enabling a business user to change a dynamic value that will be referenced in a rule, without having to change the rule itself. Operational parameters are a special type of parameter that is created by users through Genesys Administrator Extension and, when deployed, are stored as options of `Transaction` objects of the type `List` in the Genesys Configuration Server database. At rule execution time, when the Genesys Rules Engine evaluates a rule that contains an operational parameter, it obtains the current value of the associated `Transaction` object option from Configuration Server. The rule developer determines from which `Transaction` object, and which option of that object, the value of the operational parameter should be fetched, and the rule author uses this parameter as part of a condition or action.

For example, an operational parameter might be called `waitTimeThreshold`. If a caller is waiting longer than this threshold for an available agent, perhaps some other action will be performed. Instead of specifying a value for the threshold in the rule like the following:

```
When
Caller's wait time is greater than 30 seconds
Then
Offer a callback
```

the rule author could specify:

```
When
Caller's wait time is greater than {waitTimeThreshold}
Then
Offer a callback
```

The value of `{waitTimeThreshold}` can be changed at any time by a user that is using Genesys Administrator Extension and this action will have an immediate effect without the user having to modify and redeploy a rule package.

# Rule Authoring for iWD

Rule authoring for iWD is done through the Genesys Rules Authoring Tool (GRAT). This section describes general information about using GRAT for iWD, and how to use it for creating decision tables, linear rules, and business calendars.

## Login Permissions

The User permissions to the various capabilities of the GRAT are controlled by Role-Based Access Control, which is configured through Genesys Administrator (if you are not using Genesys Administrator, you will have full access to the GRAT).

See **Role-Based Access Control** in the Genesys Rules System Deployment Guide.

## Configuring iWD Tenant and User for GRAT Access

To complete the necessary configuration of an iWD managed tenant and an iWD user's security role, so that the Rules Authoring link appears on the iWD Manager navigation bar, properly launch the Genesys Rules Authoring Tool web application, and log in the user, complete the following procedure.

### Procedure

1. Log in to iWD Manager as a user with full administrative privileges.
2. From the **Tenant** drop-down list, select your iWD-managed tenant.
3. Select the **Profile** link.
4. Under Genesys Rules Authoring Tool URL, enter a valid URL for the Genesys Rules Authoring Tool.
5. Test this URL in a separate browser window to ensure that it brings up the login screen.

   The default syntax for the URL should be:

   `http://<host>:<port>/genesys-rules-authoring`

   where;

   - `<host>`—Is your application server host name.
   - `<port>`—Is the listening port of your application server (usually `8080` by default).
   - `genesys-rules-authoring`—Is the name of the Genesys Rules Authoring Tool web application that is deployed on your application server. (This will be the default name of that web application unless you have modified it.)

6. Click **Save**.
7. Select the **Security Policy** link.
8. Select the name of an existing Security Role, to which you want to give access to the Genesys Rules Authoring Tool, or create a new one.
9. Scroll down to the **Application Permissions** section.
10. Under the **Run** column, check the checkbox for the Rules Authoring permission.
11. Click **Save**.

## Configuring GRAT Access Control to IWDBP

When you launch GRAT from iWD Manager, the active tenant will be the Configuration Server tenant that is linked to the iWD managed tenant you were working with in iWD Manager. The GRAT navigation tree will display the iWD Solutions that are configured under the tenant. If you don't see any Solutions in the navigation tree, but you have one or more Solutions configured in iWD Manager, you must ensure that the user has permission to the appropriate folders under the Business Structure folder in Configuration Server.

> **Important**
>
> The Business Structure folder is created during the migration process from an earlier version of iWD to iWD 9.0.

### Procedure

To configure access control for the Genesys Rules Authoring Tool user, to allow him/her to see the appropriate iWD Solutions when logged into GRAT, do the following:

1. Log in to Genesys Administrator or Genesys Configuration Manager.
2. Navigate to the tenant that maps to the iWD managed tenant with which you are working.
3. Locate the Business Structure folder (in Genesys Administrator, go to **Environment > Business Units/Sites**).
4. In the Business Structure folder, locate the iWD Solution folder to which you want the Genesys Rules Authoring Tool user to have access.
5. If you are using Genesys Configuration Server:
   a. On the **Security** tab of that iWD Solution, configure at least Read access for the Genesys Rules Authoring Tool user, or for at least one Configuration Server Access Group to which that user belongs.

   If you are using Genesys Administrator:

   a. Highlight the iWD Solution folder and select **Edit**.
   b. Select the **Permissions** tab.

c. Use either **Add Access Group** or **Add User** to configure at least Read access to this iWD Solution, for the necessary Genesys Rules Authoring Tool user or his Access Group(s).

> **Important**
>
> When you add the user or Access Group, you can decide whether or not to propagate the permissions. If you do not propagate the permissions, you will need to explicitly add permissions to the Department(s) and Process(es) under the iWD Solution, to which you want the user or access group to have access.

4. To grant access to any additional iWD Solutions, repeat Steps 4 to 6.

# Creating a New Rules Package

## Summary

The rule package is the parent object for all the business rules for an iWD Solution.

1. Log in to iWD Manager and launch the Genesys Rules Authoring Tool.
2. In the navigation tree, expand the Solution with which you want to work.
3. Select **New Rule Package**.
4. On the right side of the screen, fill in the properties for the rule package, such as.
   - **Package Name**—Used internally and primarily in the Configuration Server List Object that will be read in the iWD business process, so the business process knows which rule package should be evaluated by the Genesys Rules Engine. There are some reserved keywords that must not be used when you name the rule package. See Creating Rule Packages topic in the GRAT Help.
   - **Business Name**—Enter any user-friendly name you wish to provide to identify the rule package.
   - **Package Type**—Select `iWD`. Selecting iWD will ensure that the list of available rule templates of type `iWD` will be displayed.
   - **Rule Package** (Optional)—Enter a description for the rule package.
   - **Template** section—Select one or more rule templates. This will determine which collection of rule actions and conditions will be made available to the business rules authors who are responsible for creating rules in this package.
5. Save the rule package.

   The new rule package will appear in the navigation tree on the left side of the pane, displaying all the Departments and Processes underneath the package, based on the access control of the user.

## Next Steps

After a rule package has been created, you can start creating business rules. For the descriptions of

the various types of rules and rule objects, see the Rules Overview tab on this page.

> **Important**
>
> The list of rule templates that are available to the person creating the rule package will depend on that user's access to the Script objects that represent each template. In Configuration Server, under each tenant, there is a **Script** folder that contains a subfolder called **Template Access Control**. In that folder, there is a script of type `Data Collection`, for each rule template published to the rules repository. The access control defined on the **Security** tab of that Script object will determine which users and access groups can use that template in a rule package.

## Rule Levels and Types

In the Genesys Rules Authoring Tool, there are three levels at which business rules can be created:

- Rule Package (referred to as Global Rules)
- Department
- Process

When the appropriate node is selected on the rule package tree, you can then select the **Rules** tab to view or edit the rules for that level of the business structure. Rules are presented in a list, with an associated phase. The order of the rules is relevant, because they will be evaluated, within a particular phase, in the same order as they appear. You can change the order of rules by clicking the up and down buttons. The logic of a particular rule can be expressed as either a linear rule or a decision table.

Any iWD extended or custom attribute can be read or updated by business rule conditions or actions, respectively.

### Global Rules

Rules that are created at the rule-package level are also known as *Global Rules*. Global rules enable you to specify rules that will apply to the entire iWD Solution. For example, they enable you to configure rules that classify or prioritize all tasks globally, instead of at a lower level of the business structure. Global rules are applied before any other rules.

This means that each rule phase (classification and prioritization) is triggered from within the IWDBP business process in the following sequence:

- Global rules
- Department rules
- Process rules

> **Important**
>
> A classification rule must be configured that assigns the task to a Process, or the Process can be assigned directly in the `createTask` message when the task is captured by a capture point. If a Process is not assigned to the task in either of these two methods, the task status will be changed to `Error Held`. For example, a department can be assigned in a global rule, followed by assigning a process at the department level. Alternatively, a process might be directly assigned in a global rule. If a process is assigned in both global rules and department rules, based on the same conditions, the department rules will override.

## Capture Point Rules

You can use the rule condition `Capture Point is…` from the iWD Standard Rules Template, at the Global Rules level. Classification rules are not created at the Capture Point level.

## Linear Rules

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 for a simple business case:

1. In Genesys Rules Authoring Tool, expand the rule package and select a node at which you want to create the rule.
2. On the right side of Genesys Rules Authoring Tool panel, on the **Rules** tab, click **New Linear Rule**.
3. Enter a **Name** for the rule that identifies it.
4. Optionally, enter a **Description** for the rule.
5. Select the **Phase** in which to apply the rule.

   You can choose `classification` or `prioritization`.
6. Optionally, assign a **Calendar** (business calendar) to the rule.
7. If required, set the **Start Date** and **End Date**.

   > **Important**
   >
   > If **Start Date** and **End Date** are left empty the rule activation period is unconstrained.
8. From the **Add Condition** combo box, select one or several conditions for the rule.

   By default, conditions are concatenated by using the logical`AND` operator.

However, you can select other functions (for example, "or" or "and not") from the Add Condition list, and you can also select multiple conditions and use the Group (or Ungroup) functions.

> **Important**
>
> 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. Genesys strongly recommends using a custom rule package for all new rule actions or conditions.

9. Select one or several actions for the rule from the **Add Action** combo box.
10. To save the specified rule, click **Save**.

    You can optionally enter a check-in comment, which will be available later on the **Audit Trail** tab.

After you have created a linear rule, you can create additional linear rules or decision tables, or deploy your rule package.

> **Important**
>
> The available conditions and actions that are presented to the rules author is driven by the set of rule templates to which the user has access. This depends on the user's permissions to the Script objects that represent each template. In Configuration Server, under each tenant, there is a **Script** folder that contains a subfolder called **Template Access Control**. In that folder, there is a script of type Data Collection, for each rule template published to the rules repository. The access control that is defined on the **Security** tab of that Script object will determine which users can access which rule templates, which will control the final list of rule actions and conditions presented.

## Decision Tables

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 rule for a complex business case:

1. On the **Rules** tab, click **New Decision Table**.
2. Specify the rule **Name**, **Description**, **Phase**, **Calendar**, **Start Date**, and **End Date**, if required.
3. From the **Add Condition** combo box, select zero or more conditions for the rule.
4. From the *Add Action*' combo box, select one or more actions for the rule.
5. To add a new row to the decision table, select the green plus (**+**) icon to the right of the right-most rule

action.

6. Enter the required parameters for each rule condition and action.
7. Optionally, enter a **Name** for the row.
8. To add a new row to the decision table, select the green plus (**+**) icon to the right of the right-most rule action. until you have set all of the required cases.

   The result is a table in which the columns represent rule conditions and actions and the rows contain real conditions and action parameter values.

   > **Important**
   >
   > To delete a condition, action, or rule, click the delete icon (-), which is located on the right side of the column (for conditions and actions), or on the right side of the row (for rules).

9. To save the specified rule, click **Save**.
10. Optionally, enter a check-in comment, which will be available later on the **Audit Trail** tab.

After you have created a decision table, you can create additional decision tables or linear rules, or deploy your rule package.

> **Important**
>
> The available conditions and actions that are presented to the rules author is driven by the set of rule templates to which the user has access. This depends on the user's permissions to the Scriptobjects that represent each template. In Configuration Server, under each tenant, there is a **Script** folder that contains a subfolder called **Template Access Control**. In that folder, there is a script of type `Data Collection`, for each rule template published to the rules repository. The access control that is defined on the **Security** tab of that Script object will determine which users can access which rule templates, which will control the final list of rule actions and conditions presented.

# Business Calendars

A business calendar is a set of rules that define working days and hours, and 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 can be assigned once (for example, at the Global Rule level), or can be assigned dynamically in a rule when needed.

To create new (or to manage existing) business calendars, launch the Genesys Rules Authoring Tool and expand a rule package. Select the **Business Calendars** entry in the tree.

# List of Solution's Business Calendar

The right side of the Genesys Rules Authoring Tool is split horizontally into two panes. The upper displays a list of a rule package's business calendars. The **New Calendar** button that is below this list is used to create a new business calendar. You can delete business calendars by clicking the delete button that is on the right side of the business calendar in the business calendar list. When a calendar list is selected, the lower pane displays the attributes of the selected business calendar.

## Business Calendar Attributes

Business calendars consist of a set of standard mandatory attributes and optional business calendar rules. See **Business Calendars** in the Genesys Rules Authoring Tool Help.

## Using Business Calendars in iWD Rules

After business calendars are defined, you can use them in rules. Business calendars must be assigned to a task before any business calendar-related calculations can be performed on task values. A Calendar can be assigned at the parent rule level, or to an individual rule. Only one calendar can be assigned to a task at a time, so a calendar can be assigned by one rule and then, overwritten by a later rule.

**iWD Rules Conditions and Actions Using Business Calendars**

| Condition/ Action | Parameters | Description |
|---|---|---|
| Assign business calendar | `{businessCalendar}` | Assigns a business calendar to a task. A business calendar must be assigned to a task, before any business calendar-related calculations can be performed on task values. A drop-down list displays a list of business calendars that are defined for the rule package. |
| Is Working Day | N/A | Calculates whether the current date/time is a working day, according to the assigned business calendar. |
| Is Working Time | N/A | Calculates whether the current date/time is working time, according to the assigned business calendar. |
| Reprioritize after | `{period} 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} 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} working {periodType}` | Sets a task's activation 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.* |
| Task expires in | `{period} working {periodType}` | Sets a task's expiration 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 |

| Condition/ Action | Parameters | Description |
|---|---|---|
| | | value, and {periodType} specifies working minutes, hours, or days. |

## Working with the businessCalendarService object

Business calendar logic is exposed to rule expressions via the businessCalendarService object. Each method of this object accepts business calendar ID as the first parameter. Business calendar ID is usually stored in businessCalendarId attribute of the task and can be accessed from rules expressions like this:

```
getStringValue("IWD_businessCalendarId", $data)
```

| businessCalendarService Method | Description | Example(s) |
|---|---|---|
| Date add(String businessCalendarName, Date addToThis, String type, int amount) | Adds working time to the timestamp according to the business calendar. Type parameter should be either "minutes", "hours" or "days". Amount parameter represents number of time units of the given type. It can also be negative. The method returns the resulting timestamp. When adding days, timestamp will be set to end of working day, for example when adding 1 day it will result in the end of the next working day. | businessCalendarService.add(getStringValue("IWD_busines $data), getDTValue("IWD_createdDateTime", $data), "hours", 4)); // get timestamp of 4 working hours after the task was created |
| boolean dayIsWorkingDay(String businessCalendarName, Date day) | Returns true if the day of the given timestamp is a working day (has any working hours). | businessCalendarService.dayIsWorkingDay(getStringValue( $data), evo.common.TimeHelper.getUTCTime()); // is today a working day<br>businessCalendarService.dayIsWorkingDay(getStringValue("IWD_businessCalenda getDTValue("IWD_dueDateTime",$data)); // does IWD_dueDateTime fall on a working day? |
| boolean timeIsWorkingTime(String businessCalendarName, Date time) | Returns true if the given timestamp is a working time. | businessCalendarService.timeIsWorkingTime(getStringValu $data), evo.common.TimeHelper.getUTCTime()); // is |

| businessCalendarService Method | Description | Example(s) |
|---|---|---|
| | | it working time right now<br><br>businessCalendarService.timeIsWorkingTime(getStringValue("IWD_businessCalend<br>getDTValue("IWD_dueDateTime",$data)); // does<br>IWD_dueDateTime fall within working hours? |
| int diffWorkingDays(String businessCalendarName, Date date1, Date date2) | Returns the number of working days between two given timestamps. | businessCalendarService.diffWorkingDays(getStringValue(<br>$data), getDTValue("IWD_createdDateTime",<br>$data),<br>evo.common.TimeHelper.getUTCTime()); //<br>how many working days have passed since<br>creation of the task? |
| int diffWorkingHours(String businessCalendarName, Date date1, Date date2) | Returns the number of working hours between two given timestamps. | businessCalendarService.diffWorkingHours(getStringValue<br>$data), getDTValue("IWD_createdDateTime",<br>$data),<br>evo.common.TimeHelper.getUTCTime()); //<br>how many working hours have passed since<br>creation of the task? |
| int diffWorkingMinutes(String businessCalendarName, Date date1, Date date2) | Returns the number of working minutes between two given timestamps | businessCalendarService.diffWorkingMinutes(getStringVal<br>$data), getDTValue("IWD_createdDateTime",<br>$data),<br>evo.common.TimeHelper.getUTCTime()); //<br>how many working minutes have passed since<br>creation of the task? |
| Date beginningOfWorkingDay(String businessCalendarName, Date utcDate) | Returns timestamp that contains opening hours for the given date. The result will be a composite timestamp, where the date part is taken from the given parameter, but the time part contains the beginning of working hours for the given date. If date is not a working day, a null value will be returned. | businessCalendarService.beginningOfWorkingDay(getString<br>$data),<br>evo.common.TimeHelper.getUTCTime()); //<br>get today's opening hours |
| Date endOfWorkingDay(String businessCalendarName, Date utcDate) | Returns timestamp that contains closing hours for the given date. The result will be a composite timestamp, where the date part is taken from the given parameter, but the time part contains the ond of working hours for the given date. If date is | businessCalendarService.endOfWorkingDay(getStringValue(<br>$data),<br>evo.common.TimeHelper.getUTCTime()); //<br>get today's closing hours |

| businessCalendarService Method | Description | Example(s) |
|---|---|---|
| | not a working day, a null value will be returned. | |

**How to retrieve today's opening hours?**

```
businessCalendarService.beginningOfWorkingDay(getStringValue("IWD_businessCalendarId", $data), evo.common.TimeHelper.getUTCTime());
```

**How to retrieve today's closing hours?**

```
businessCalendarService.endOfWorkingDay(getStringValue("IWD_businessCalendarId", $data), evo.common.TimeHelper.getUTCTime());
```

**How many hours left till closing hours?**

```
businessCalendarService.diffWorkingHours(getStringValue("IWD_businessCalendarId", $data), evo.common.TimeHelper.getUTCTime(),
 businessCalendarService.endOfWorkingDay(getStringValue("IWD_businessCalendarId", $data), evo.common.TimeHelper.getUTCTime()));
```

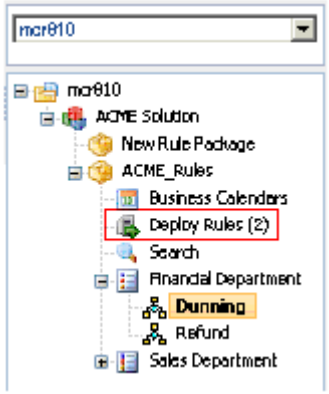
## Rule Deployment

Once you have created all the necessary rules in a rule package, it is time to deploy the rule package to a single instance of the Genesys Rules Engine or a cluster of GREs. Once the package is deployed, it can be invoked by a client such as the iWD business process (IWDBP).

Rule package deployment is done through the Genesys Rules Authoring Tool, and is independent of the iWD Solution deployment in iWD Manager. Thus, you can deploy a new version of a rule package for an iWD Solution without having to redeploy the entire Solution.

When a rule is created or edited and it has not been deployed, there is a checkmark in the **Pending Deployment** column of the rule.

In addition, when a rule package requires deployment or redeployment, there will be a visual indication next to the **Deploy Rules** node in the **Genesys Rules Authoring Tool** navigation tree, under the rule package itself.




For more information about rule package deployment, including scheduled deployment and deployment history, and snapshots, see Deploying/Undeploying Rules Packages (new document).

# Rule Evaluation by the IWD Business Process

## Rule Evaluation

After a rule package has been deployed to a Genesys Rules Engine, it can be invoked by a client. The standard client that invokes the Genesys Rules Engine is the iWD business process (IWDBP). For information about other ways to invoke rule packages, see the Genesys Rules System Deployment Guide.

> **Important**
> There are two separate business processes, one for Genesys IRD and one for Genesys Composer/ORS.

It is important to understand how rules are evaluated, when an evaluation is requested from the IWDBP business process. IWDBP has the following two routing strategies that invoke rule evaluation from the Genesys Rules Engine.

- Classification—Calls the Genesys Rules Engine and requests an evaluation of all rules for the iWD Solution's associated rule package that belong to the Classification phase.
- Prioritization—Calls the Genesys Rules Engine and requests an evaluation of all rules that belong to the Prioritization phase.

### Associating the iWD Solution with a Rule Package and Rules Engine

To ensure the IWDBP business process invokes the correct Genesys Rules Engine and the correct rule package, two List Objects must be properly configured:

- Iwd_Esp_List
- Iwd_Package_List

> **Important**
> Make sure that both of these List Objects are correctly configured, otherwise IWDBP will not work.

## Iwd_Esp_List

The IWDBP uses the data from the Iwd_Esp_List List Object to correlate the IWD_SolutionId to the name of the Genesys Rules Engine (or cluster).

> **Important**
>
> Before 8.5.1, the IWDBP used the Business Context Management Service (BCMS)—an External Service Protocol (ESP)— to communicate with the Genesys Rules Engine. Since 8.5.1, BCMS functionality has moved into the Genesys Rules Engine.

Iwd_Esp_List is also used, optionally, to correlate the IWD_SolutionId to the name of a Genesys Universal Contact Server (UCS) that is connected to Interaction Server. If this association is configured in Iwd_Esp_List, the business logic in IWDBP will update the interaction history in the UCS database to mark the interaction as done (the value of the **Status** column in the **Interaction** table in the UCS database will be set to 3) when the interaction enters one of these queues:

- iWD_Completed (or iwd_bp_comp.Main.iWD_Completed for Composer/ORS)
- iWD_Canceled (or iwd_bp_comp.Main.iWD_Canceled for Composer/ORS)
- iWD_Rejected (or iwd_bp_comp.Main.iWD_Rejected for Composer/ORS)

## Iwd_Package_List

The Iwd_Package_List List Object is used to correlate the IWD Solution ID (IWD_SolutionId) to the name of the rule package that will be evaluated when requests are made to the Genesys Rules Engine from the IWDBP business process.

Iwd_Package_List contains a single list called, RulePackageList. Create a new key/value pair for each iWD Solution that you have configured under your Configuration Server tenant, where the key or option is the IWD Solution ID and the value is the Package Name of the rules package. For multiple rule packages, specify each Package Name separated by a comma—for example; `rule.package1, rule.package2, rule.package3`

> **Important**
>
> The user configures two names for a rule package in GRAT. There is the **Package Name** and the **Business Name**. The value that the user enters in the RulePackageList list, must reference the Package Name (not the Business Name) of the rule package.

# Rule Evaluation Order

Within all the rules that are configured for a particular phase, the Genesys Rules Engine will evaluate rules in this order:

- Global rules (package-level)
- Department rules
- Process rules

To ensure the Genesys Rules Engine performs its evaluation in this order, the **sequential-mode** option (in the **[settings]** section in the Genesys Rules Engine Application) must be set to `false`. This is the default setting.

To ensure the Genesys Rules Engine evaluates all rules within a particular phase, within a particular node of the business structure hierarchy, the **group-by-level** option (in the **[settings]** section in the Genesys Rules Authoring Tool server Application) must be set to `true`. This is the default setting.

Within the rules that are defined at a particular node of the business structure hierarchy (for example, `Department`), the Genesys Rules Engine will follow the salience (order) that is defined in the rule package. This is controlled by the rules author, who can move rules up and down in the Genesys Rules Authoring Tool. In the example in Controlling Rule Evaluation Order (below), Rule-265 will be evaluated prior to Rule-294. If the rules author preferred Rule-294 to be evaluated first, they could move that rule up in the evaluation order by using the arrow button on the right side of the rule.

## Controlling Rule Evaluation Order

Within a set of rules for a particular phase, at a particular node in the business structure hierarchy, an action of one rule can influence a condition in a subsequent rule. To do this, it is necessary to include the `update($data)` parameter; in the `Rule Language Mapping` of the rule action or as a separate rule action. See the following example.

{{NoteFormat|Rules authors can change the sequence of rows in a decision table to permit more sophisticated outcomes. So, you can specify whether the the rows in a decision table are processed top-down or bottom-up by configuring an option (from GRAT 8.5.0) called evaluate-decision-table-rows-top-down.

### Example: Setting Rules at a Specific Phase

If you had one classification rule at the **Department** level that set the priority of a task to `100` and a second classification rule at the **Department** level that will increase the priority by `90`, but only if the priority has already been set to `100`:

The first classification rule would look like this:

```
WHEN
Priority is not equal to 100
THEN
Set Priority to 100
Update
```

The second classification rule would look like this:

```
WHEN
Priority is equal to 100
THEN
Increase Priority 90
```

In the first classification rule, it is important to specify that the priority be set to `100` only when it is not already equal to `100`. This is to avoid a problem where the Genesys Rules Engine might get into a loop during rule evaluation.

Also in the first classification rule, there is a rule action called `Update`. The Rule Language Mapping for this rule action is: `update($data);`

You could also modify the Rule Language Mapping of the `Set Priority` rule action, to add `update($data);` to the end of the Rule Language Mapping. For example:

```
setIntValue("Priority", {priority}, $data); update ($data);
```

In the latter case, however, you would want to create a new version of the `Set Priority` rule action and give it a different Language Expression to help differentiate it, such as `Set extended Priority`.

# Customer and Partner Translation

## Extracting localization files from iWD Manager

### Prerequisites

- iWD Manager installed in base (en-US) localization—without Genesys-provided localization package.

### Procedure

1. Locate the **iwd_manager.jar** file within the iWD Manager installation directory.
2. Extract the **iwd-login.json** and **iwd-manager.json** files from the **iwd_manager.jar\gui\i18n** folder and the **resources.properties** file from the **iwd_manager.jar\WEB-INF\classes\evo\cmc\ui\resources** folder to a temporary location. File **iwd_manager.jar** is a zip file and can be opened with 7zip or the WinZip utility on Windows and the unzip utility on Linux.

## Extracting localization files from iWD GAX Plugin

### Prerequisites

- Genesys Administration Extension package installed
- iWD GAX Plugin installed within Genesys Administration Extension

### Procedure

1. Locate the **gax-iwd.jar** file within the Genesys Administration Extension installation directory.
2. Extract the **web/nls/lk.js** file from **gax-iwd.jar** to a temporary location. File **gax-iwd.jar** is a zip file and can be opened with 7zip or the WinZip utility on Windows and the unzip utility on Linux.

## Extracting localization files from iWD Web

### Prerequisites

- iWD Web installed in base (en-US) localization—without Genesys-provided localization package.

### Procedure

1. Locate the **iwdweb.jar** file within the iWD Web installation directory.
2. Extract the **en-US.json** file from the **iwdweb.jar\static\i18n** folder and the **ValidationMessages.properties** file from the **iwdweb.jar\validation** folder to a temporary location. File **iwdweb.jar** is a zip file and can be opened with 7zip or the WinZip utility on Windows and the unzip utility on Linux.

## Translating the properties localization file

### Prerequisites

- JDK 8 installed.
- The **resources.properties** file extracted from the appropriate iWD Manager and saved in a temporary location.
- The **ValidationMessages.properties** file extracted from the appropriate iWD Web and saved in a temporary location.

### Procedure

1. Rename the **.properties** file to **.utf8**—for example, **resources.properties** to **resources.utf8**.
2. Translate the files. Each line in the file contains a key and value separated by the '=' character. Only the value part needs to be translated—key and '=' character must remain unchanged. Any text in braces {} must remain unchanged. The file should be saved in UTF-8 encoding.
3. Use the native2ascii utility located in JDK 8 to convert **.utf8** files into **.properties** files. Execute the command **native2ascii resources.utf8 resources.properties -encoding UTF-8** from the command line.

## Translating JS and JSON localization files

### Prerequisites

- JS and JSON localization files from iWD Manager, iWD Web and iWD Plug-in for GAX extracted and saved in a temporary location.

### Procedure

1. Only text in quotes needs to be translated. Any text in braces {} is a parameter and must be left unchanged.
2. The file must be saved in UTF-8 encoding.

# Installing translated files in iWD Manager

## Prerequisities

- iWD Manager package is installed.
- Translated **resources.properties**, **iwd-login.json** and **iwd-manager.json** files are available in a temporary location.

## Procedure

1. Update files **\WEB-INF\classes\evo\cmc\ui\resources\resources.properties**, **\gui\i18n\iwd-login.json** and **\gui\i18n\iwd-manager.json** within the **iwd_manager.jar** file with those from the temporary location. File **iwd_manager.jar** is a zip file and can be updated with 7zip or the WinZip utility on Windows and the zip utility on Linux.
2. Restart iWD Manager.

# Installing the translated lk.js in iWD Plug-in for GAX

## Prerequisites

- Genesys Administration Extension package installed.
- Appropriate localization package for Genesys Administration Extension installed (either Genesys- or Customer-provided).
- iWD Plug-in for GAX installed within Genesys Administration Extension.
- Translated **lk.js** file is available in a temporary location

## Procedure

1. Locate the **gax-iwd.jar** file within the Genesys Administration Extension installation directory.
2. Update the **web/nls/lk.js** file in **gax-iwd.jar** with the one from the temporary location. File **gax-iwd.jar** is a zip file and can be updated with 7zip or the WinZip utility on Windows and the zip utility on Linux.
3. Restart Genesys Administration Extension.

# Installing translated files in iWD Web

## Prerequisities

- iWD Web package is installed.
- Translated **ValidationMessages.properties**, **en-US.json** files are available in a temporary location.

## Procedure

1. Rename the **ValidationMessages.properties** file to **ValidationMessages_*<languagecode></languagecode>*.properties** and the **en-US.json** file to ***<languagecode></languagecode>***'''.json **where *<languagecode></languagecode>* means your desired language.**
2. Copy the JSON file prepared in the temporary location into the **\static\i18n** directory and the properties file into the **\validation** directory within **iwdweb.jar**. File **iwdweb.jar** is a zip file and can be opened with 7zip or the WinZip utility on Windows and the unzip utility on Linux.
3. Restart iWD Web.

# High Availability for iWD Components

## Component summary

For definitions of Hot and Warm Standby used below, please see the Application Redundancy page.

| Component | HA Status | Comment |
|---|---|---|
| **iWD Components** | | |
| iWD History Node | N+1 | See Installing an iWD History Node Cluster. |
| iWD Runtime Node | Cold standby | See High Availability for Runtime Node |
| iWD Plugin for GAX | See GAX below. | |
| iWD Manager | N+1 | Session is not shared between servers. See Configuring High Availability. |
| iWD Web | N+1 | Session is not shared between servers. See Configuring High Availability. |
| **Non-Genesys Components** | | |
| JMS | Vendor-specific | Please refer to your JMS MQ vendor's documentation for details on High Availability configuration.<br>For example, when using ActiveMQ as a JMS provider, the JMS Event Logger DAP should contain the following value under its Application Options to connect either with a local JMS broker or a remote one:<br>[logger-settings] / jms-provider-url=failover:(tcp://localhost:61616,tcp://remotehost:61616) |
| **Other Genesys Components** | | |
| IXN | Warm standby | • See the eServices Deployment Guide.<br>• See Interaction Server Cluster for cluster configuration. Please note that cluster configuration does not mean all Interaction Servers in the cluster have access to all interactions. |

| Component | HA Status | Comment |
|---|---|---|
| Interaction Server Integrated Capture Points | Warm standby | See the eServices Multi-tenancy and Load Balancing Guide. |
| GRE | N+1 | See the Genesys Rules System Deployment Guide. |
| ORS | Warm standby | See the Orchestration Server Deployment Guide |
| URS | Warm and hot standby | See the URS Deployment Guide, downloadable from here. |
| GAX | N+1 | See the Genesys Administrator Extension Deployment Guide. |
| UCS | Warm standby | See the eServices Deployment Guide. |
| SS | Warm and hot standby | See the Stat Server Deployment Guide. |
| Pulse | Hot standby | See the Genesys Pulse Deployment Guide. |
| Genesys Management Framework | | See the Genesys Management Framework Deployment Guide. |

## Important

The following components included in the iWD Solution are not runtime components and hence are not included in the table above:

- WDE
- GCXI
- IRD/Composer
- GRAT/GRDT.

# IWD Troubleshooting

## Support for RHEL 64-bit Platforms

Installers support installation of Red Hat Linux 64-bit platforms only when RHEL standard compatibility packs for 32-bit platforms have been installed. These packages have generic names like:

- `compat-glibc*`
- `compat-libstdc++*`
- `libstdc++* '`

You might also need to ensure that the `JAVA_HOME` variable points to the correct JDK.

## Blocked Lookup Tables

When a configuration transaction object already exists with its alias attribute set to `Iwd_Lookup_Tables_List`, it is not possible to create a new Lookup Table due to a constraint violation of the uniqueness of the alias attribute value. An error message is displayed to inform users about this violation.

If this type of error arises, before checking privileges, do the following:

1. Check whether any object with a different on-screen name and the alias `Iwd_Lookup_Tables_List` already exists.
2. If so, back it up then remove and recreate the entire whole Lookup Table from scratch via the GAX interface.

## Error when Creating or Updating a Big Lookup Table

Lookup tables have a limit of 1000 entries each. This constraint was implemented to avoid connection timeouts between GAX and Configuration Server. It is still possible that this problem might occur in a customer environment when the connection has very low quality. The workaround for this problem is to edit the Application options of the GAX application and increase the value of the **confserv_timeout** property in the **[General]** section. If the **confserv_timeout** option is not present, a default value of `30` seconds is applied.

# Support for RHEL 64-bit Platforms

Installers support installation of Red Hat Linux 64-bit platforms only when RHEL standard compatibility packs for 32-bit platforms have been installed. These packages have generic names like:

> **Important**
>
> The 32-bit compatibility packages are no longer available for RHEL 8. So, these packages must be taken from the RHEL 7 distributive files.

- `compat-glibc*`
- `compat-libstdc++*`
- `libstdc++* '`

You might also need to ensure that the `JAVA_HOME` variable points to the correct JDK.

# Blocked Lookup Tables

When a configuration transaction object already exists with its alias attribute set to `Iwd_Lookup_Tables_List`, it is not possible to create a new Lookup Table due to a constraint violation of the uniqueness of the alias attribute value. An error message is displayed to inform users about this violation.

If this type of error arises, before checking privileges, do the following:

1. Check whether any object with a different on-screen name and the alias `Iwd_Lookup_Tables_List` already exists.
2. If so, back it up then remove and recreate the entire whole Lookup Table from scratch via the GAX interface.

# iWD Best Practice Overview

This group of topics provides some best practice guidelines to use when you are planning, deploying, and configuring intelligent Workload Distribution (iWD):

- Design Phase
- Task Capture
- Business Rules Configuration
- Configuration
- Task Distribution and Routing
- Reporting

# Design Phase

## Review Messaging Between the Source System and iWD

It is important to understand how the source system works. In particular, which triggers on the source system will be used to generate iWD messages (such as, `CreateTask` or `UpdateTask`). The involvement of all stakeholders—business analysts and technical personnel, representing both the source system and the iWD business process—is key to a successful outcome.

## Map Out Complete Task Flow or State Model

During the design phase, it is important to document the end-to-end message flows from the source system all the way to the agent desktop or toolbar, that describe messaging for each scenario that might be encountered. Ladder diagrams that show the various actors in the system can be very useful for this exercise.

Also, ask the question, “When is a task considered completed from a business perspective?” The answer affects both reporting and distribution. For example: A task enters `iWD_Queued` and is distributed to and handled by an employee and then, is passed on (for example, it is completed or transferred by the employee or deferred for later completion). In which of these scenarios do you want the task to be considered completed from the iWD state model perspective? Completed in this sense means completed by the Department and Process.

Do you want the task to be completed only when it is completed within the source system? This can allow for true end-to-end reporting that captures when the task is created in the source system until the time it is completed in the source system, plus the time spent on the task by any iWD-enabled employees. This discussion becomes particularly relevant when you design the triggers on the source system for the iWD operations such as `CreateTask`, `UpdateTask`, and `RestartTask`.

## Don’t Forget About the Desktop

There might be times when task-related logic needs to be implemented at the employee desktop, or toolbar. For example, if an `UpdateTask` message is received while the task is assigned to an employee, the desktop might need to react in different ways, depending on the content of the update message. In those cases, it is important to include the desktop or toolbar developer during the design phase when various scenarios are discussed and documented.

# Task Capture

## Selecting the Appropriate Capture Point

The capture point you select will depend on the capabilities of the source system from which you are capturing tasks. In some cases, there will be multiple options you will be selecting from. Therefore, it is useful to know the capabilities and limitations of the various capture points.

When possible, you should use the JMS Integrated Capture Point if the JMS (Java Message Service) enterprise messaging service is supported by your source system. This capture point is bi-directional, supporting an input queue and an output queue. Since it uses an enterprise messaging system, it is more reliable than other types of communication. For example, if iWD has a notification to provide to the source system, that notification will be placed in a JMS message queue by the JMS Integrated Capture Point. If there is a loss of communication between the JMS Provider and the source system, the notification message will remain in queue until communication is restored.

Legacy capture points are no longer supported.

The only other bi-directional capture point is the Database Capture Point, which is very flexible. However, to leverage the bi-directionality, it is necessary to update tables on the source system database. In some environments, this will not be possible.

The Web Service Integrated Capture Point is an excellent choice for integration with any source system that has the capability of invoking SOAP/HTTP messages from within its workflow. The message set of the Web Service Integrated Capture Point is very straightforward and is easy to integrate. The SOAP payload, in XML format, is easy to understand and is logically organized.

While the Web Service Integrated Capture Point is not bi-directional, you can use the Web Service Capture Point from the source system to request the latest status for a particular task, including the current values of all the task attributes. This could be done from the source system just before taking certain actions on the source system, to ensure that the latest updates that might have occurred on the Genesys side, can be propagated to the source system.

The XML file capture point is a good option when it is being integrated with legacy host systems that do not have support for web services or modern messaging systems. In most cases, these systems can generate flat files that consist of lists of attributes for each work item in its database. These flat files could be converted to an XML format and then read by the iWD XML File Integrated Capture Point.

## Proper Use of Extended Attributes

Where possible, use iWD extended task attributes to map the source system's data fields that you intend to use in business rules. This is preferable to the use of custom attributes, because the extended attributes are out-of-the-box. That is, no additional configuration is required to use them. Columns are already included in the Interaction Server databases for these attributes and their associated Interaction Custom Properties are defined in Configuration Server. In addition, they are

better supported in iWD Data Mart because some of these extended attributes are already dimensions in the iWD Data Mart database and can be queried out-of-the-box.

## Optimizing the Use of Custom Attributes

Do not capture unnecessary custom attributes from the source system. This increases the size of the attached data and the Interaction Server and Event Log databases. Include a custom attribute only within the `CreateTask` message, if you plan to use the attribute in one or more of the following ways:

- In business rules
- In a routing strategy
- To display to a user through the Global Task List
- To display to a user through an *agent desktop,* such as Genesys Interaction Workspace
- To use for reporting
- To display data to an employee on the desktop, consider the desktop on which the employee will access work on the task. If the employee is working directly on the source system desktop to fulfill the task, then it might not be necessary to capture attributes, if only to provide this data to the employee.

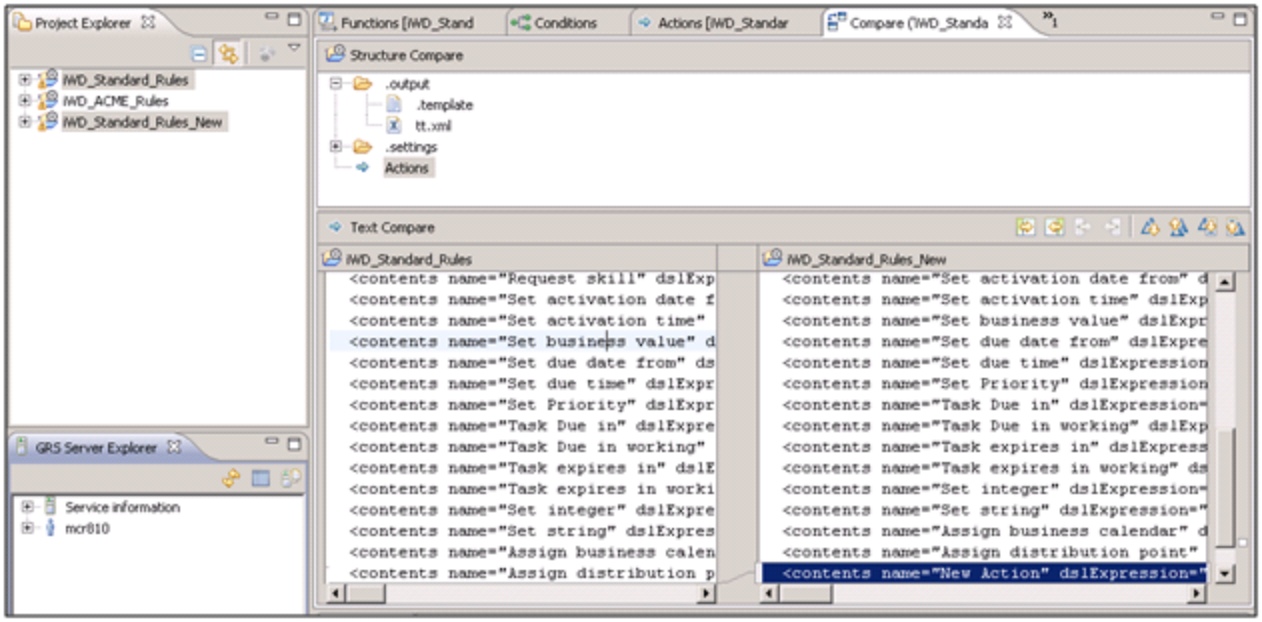
# Business Rules Configuration

## Use Custom Rule Templates

It is tempting to simply augment the iWD Standard Rules Template to meet your specific business requirements. However, it is a best practice to create one or more custom rule templates to add new rule conditions and actions that you require to meet your business requirements. There are several a few advantages to this approach, such as:

- Genesys might release new versions of the iWD Standard Rules Template from time to time. Importing a new version of the Standard Rules Template into the Genesys Rules Development Tool requires that you delete (or rename) the existing version. Therefore, any custom rule conditions or actions that you added to the Standard Rules Template would be lost.
- By keeping the iWD Standard Rules Template intact, it allows you to associate it with the Environment tenant in Configuration Server. In a multi-tenant environment, this enables a common, standard set of rule conditions and actions that can be accessed by all tenants.
- Access control can be applied to rule templates, because each template is represented by a separate Script object in Configuration Server. Therefore, multiple rule templates can be created, segregating different types of rule conditions and actions that will be accessed by different types of users or by different functional areas of the business. Normally, you will want the Standard Rules Template to be accessible by all users whereas, you might have other templates where basic rule conditions and actions are in one template and more advanced conditions and actions in a separate template. You can then use access control on the associated Script objects to determine which users will be able to access specific templates through the Genesys Rules Authoring Tool.
- If you do modify the iWD Standard Rules Template, Eclipse provides a way to compare your modified template project with the original version that is included in the Genesys iWD Manager Installation Package. Just rename your modified version of the template project, and then import the original Genesys version. Select both projects in the **Project Explorer**, right-click, and choose **Compare With Each Other**. A **Compare** view will display the differences in the Actions, Conditions, Functions, and Parameters. In the example below, one new Action has been added to the **iWD_Standard_Rules_New** project.

### Compare Modified Template with Original (Genesys) Version



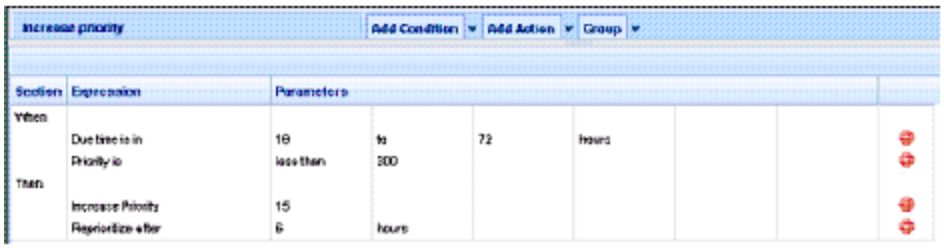

Compare Modified Template

## Design of Rules Hierarchy

It is useful to create business rules at different levels of the business structure (for example, Global Rules, Department-level, Process-level), rather than putting all rules at the Global level. Not only does this configuration make troubleshooting simpler, it also enables you to provide access control to specific sets of rules. Moreover, it enables you to set default rules. For example, at the Global Rules level you might set a default priority or a default due date for all tasks that meet specific criteria. You can then override those defaults at a lower level of the business hierarchy, based on various conditions.

## Use Prioritization Ranges

It can be very useful to define priority ranges for different types of tasks. Priority ranges define the minimum and maximum priorities that can be assigned to any type of tasks.These tasks are enforced when the tasks are assigned their initial priorities and when they are reprioritized over time. For example, in a blended environment you might reserve priorities 501+ for voice calls, whereas 401-500 are for the highest-value off-phone tasks, 301-400 are for the next highest-value, and so on. Through business rules, you can ensure that a particular type of task's priority never exceeds an upper ceiling. You can do this in your prioritization or reprioritization rule by adding a rule condition, such as `Priority is {operator} {priority}`, where `{operator} = less than`.



Upper Limit on Rule Priority

By maintaining tasks within predefined priority ranges, it is easier to troubleshoot anything that might happen with these tasks, and easier to predict what will actually be routed to an employee when they become available for work.

## Do Not Reprioritize Interactions Frequently

It is important to reprioritize tasks at reasonable time intervals. That is, if a task is not due for 3 days, and your business normally operates with a significant task backlog, it does not make sense to reprioritize that task every 15 minutes. That will just consume unnecessary resources that can put a stress on the system, when there are tens of thousands, or hundreds of thousands of tasks in queue. Therefore, plan your reprioritization intervals so that tasks that are not due for several days are only reprioritized once, or a few times per day, versus tasks that are due within the current working day (or a shorter time frame). Those tasks could be reprioritized once per hour.

The appropriate reprioritization intervals should be based on an analysis of your backlog and how soon you expect to work through it. Remember that you can set up different reprioritization intervals, based not only on when a task is due, but on any other criteria as well. The criteria might include department, process, business value, current priority, any custom attribute, or combination of custom attributes. It can be an extremely helpful exercise to graph out the different types of tasks and the way in which you expect the reprioritization of the tasks to occur over time. You can start by putting this data into an Excel spreadsheet, and then, within Excel, automatically generate a line chart. Put each task in a different color. The more intersecting lines you have on your graph, the more confusion you can expect to have when you put the system into operation.

# Configuration

## Align Business Structure and Business Requirements

In some businesses, the way you define Departments and Processes in iWD will directly align with how the business views distribution and reporting.

In other cases, consider aligning Departments and Processes with your reporting requirements and use Genesys skills to align with distribution. This is the recommended approach because the Departments and Processes can then be used as input in the Data Mart plug-ins—that is, the pre-defined attributes of Department and Process can be used to support the reporting metrics and dimensions. This makes it easier to provide statistics from a business point of view.

## Consider Using Multiple iWD Tenants

Consider configuring more than one iWD managed tenant, where each tenant aligns to a different business unit. This allows you to configure dedicated custom attributes in iWD Data Mart for each business unit. It also reduces the amount of data iWD Data Mart has to process from the Interaction Server Event Log database. This means you will need to set up multiple iWD Data Mart instances, but this configuration is more scalable.

> **Important**
>
> Using tenants is a recommended solution, but it is possible to use different solutions, because iWD Data Mart is a per-solution entity.

## Load Balance GRE in High Volume Deployments

If your iWD solution has particularly high volumes or uses frequent reprioritization, it might be useful to set up a cluster of Genesys Rules Engines (GRE) in a load-balanced configuration. Consider updating the out-of-the-box IWDBP business process to add a subroutine with this type of load balancing, with multiple runtime nodes within the solution. You can make the number of retry attempts configurable as a strategy variable or within a List Object so the value can be modified without changing the strategy itself.

# Task Distribution and Routing

This section includes information about best practices to consider when you are planning and configuring the task distribution and routing components of your iWD solution.

## Use Scheduling in the Queue Views

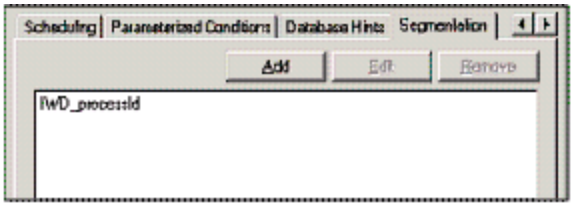
Queue views define the criteria that must be met for a task to be submitted from a queue to a Distribution routing strategy. Using the **Scheduling** tab can prevent a task from bouncing between Interaction Server and Universal Routing Server (URS), especially when there are no agents logged in to handle tasks. In this case, the **ScheduledAt** attribute can be used to reschedule submission of tasks back to URS.

> **Important**
>
> For information about the **ScheduledAt** property, see the topic "Setting the ScheduledAt Property" in the *Universal Routing 8.1 Business Process User's Guide.*


***Scheduling Tab—URS***

## Use Segmented Views to Keep Agents Busy

Consider using segmentation on the **Queue** view from the iWD_Queued queue (or custom interaction queue in your iWD business process) to the **Distribution** routing strategy. Segmenting interactions, based on iWD Process (IWD_processId), or by skill might be a good idea. This ensures that all types of work (tasks) will be submitted for distribution even if these tasks (in a specific process) have low priorities. There might be specific agents that are dedicated to manage or handle these low priority tasks. However, if they have low priorities, they might never be submitted to a distribution strategy.




***Segmentation Tab***

### Segmentation Feature

Segmentation ensures that all types of work (tasks) will be submitted for distribution even if the tasks (in a specific process) have low priorities. Segmentation also ensures that all agents are busy by distributing tasks in each segment separately, thus reducing agents' idle time. Each agent group can have its own task queue, and each segment can have its own limit.

Segmentation settings appear in the **ToDistribute** view in the **Distribution** strategy— now, the **Distribution** strategy can identify an interaction attribute (**IWD_Segment**) in those settings and use its value to route the interaction to the correct agent/agent group.

There is more information in these documents:

- IWD BP for IRD
- IWDBP for Composer/ORS

Scheduling will not be used in this example.

## Use Triggers in the Routing Strategy

There are times when a task is updated by the source system—for example, an agent who is not iWD-enabled has modified the task or an external system has modified the task. This task update might impact the classification, prioritization, or distribution of the task. Setting triggers on the **Distribution** strategy to react to specific changes in attached data elements enables you to reclassify, reprioritize, and redistribute tasks, as required.

There is a function in IRD/URS 8.x called **SetUpdateTrigger[]** which enables you to specify an attached data key that will be monitored for dynamic changes while an interaction is sitting in the **Target** block, waiting for an available agent.

### Example: Specifying Attached Data Key

An interaction is waiting for an available agent in a **Target** block in the **Distribution** strategy, with a timeout of `30` minutes before it goes to the red port. During this time, it possible that the source system will send an **UpdateTask** message (for example, if another agent pulled the task manually from the source system). If the **SetUpdateTrigger[]** function is enabled in the strategy for a specific attached data element, and then the value is updated, the interaction goes to the red port automatically, enabling you to evaluate the **UpdateTask** message and take control of the interaction in the strategy.

If **SetUpdateTrigger[]** is not set up in the **Distribution** routing strategy to handle this, then the interaction might be stuck in the **Target** block and could get distributed to other agents, which could cause a loss of synchronization between the source system and IWD. A recommended approach is to create a custom attribute such as **iwdAction** in the **SetUpdateTrigger[]** function, with possible values of `CreateTask`, `UpdateTask`, and so on. This attribute would be set by the source system in any `CreateTask` or `UpdateTask` messages that would be generated from the source system. After evaluating the message within the IWDBP business process, its value can be set to `CLEARED`.

Apart from using the **iwdAction** custom attribute in the **SetUpdateTrigger[]** function, this attribute can also be used in most of the **Interaction Queue** or **Workbin** views to determine if the task has been updated (by using the `UpdateTask` message) while sitting in those Queues or Workbins. This can be done by creating a new View from these Queues and Workbins called, for example, `BackEnd Update`, with this condition:

```
iwdAction !='CLEARED'
```

If this condition is met, the interaction can be submitted to a routing strategy that evaluates the update task message and performs the appropriate actions, such as distributing it to a specific agent, applying rules, or ignoring the update and sending it back its original location.

## Avoid Looping in Strategies

It is important to prevent looping within your routing strategies. It can place a lot of load on the Genesys Rules Engine (GRE) if reclassification and reprioritisation are occurring over and over again. If these requests fail because GRE is down, or a rule is not applied successfully due to a syntax error that cannot be caught during design time, consider taking one of the following actions:

- Use scheduling to delay the next attempt at rules evaluation.
- Place the task into the iWD_ErrorHeld queue (IRD business process) or iwd_bp_comp.Main.iWD_ErrorHeld (Composer business process) to stop a *bad* task from impacting the overall system. For example, you might check the value of the **IWD_processId** attribute to verify that it has been classified correctly—that is, it is not `NULL`. From the iWD_ErrorHeld queue or iwd_bp_comp.Main.iWD_ErrorHeld queue you might resubmit the task into an error-handling strategy, which then sends the task back to the iWD_New queue or iwd_bp_comp.Main.iWD_New queue or performs other integrity checking on the task's attached data.

## Consider Pull Versus Push Task Distribution

Many businesses like iWD's push-based model of task distribution. It helps avoid the common problem of employees picking the easiest task to work on next, and to ensure that employees are always working on the highest-priority task. However, it is likely that some business workflows will require a pull (workbin) model. So consider the role that Agent Group Workbins might play in your solution.

Also consider that agents might need to hold on to a task for a period of time if they cannot complete it immediately. This might require the agent to open and close the task multiple times before finally completing, or otherwise dispositioning, the task. In this case, you will probably want to use personal Agent Workbins. In both cases, you will need to consider the reporting and distribution implications.

# Reporting

## iWD Statistics in CCPulse+

You can display performance statistics that are related to iWD Departments and Processes, through CCPulse+. These statistics are sourced from the iWD Data Mart. No agent performance metrics from the iWD Data Mart are fed to Stat Server and therefore, to CCPulse+. However, you can use the standard Stat Server and CCPulse+ agent metrics such as Average Handle Time, Total Login Time, Total Talk Time, and so on. All of these agent status-related statistics will be applicable to iWD work item handling as well. For Average Handle Time, you can apply filters, including filtering by media type or any other attached data key-value pair such as iWD Department or Process.