



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Best Practice Overview

intelligent Workload Distribution 8.5.0

3/14/2022

Table of Contents

iWD Best Practice Overview	3
Design Phase	4
Task Capture	5
Business Rules Configuration	7
Configuration	10
Task Distribution and Routing	13
Reporting	16

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 from release 8.5 onwards.

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.

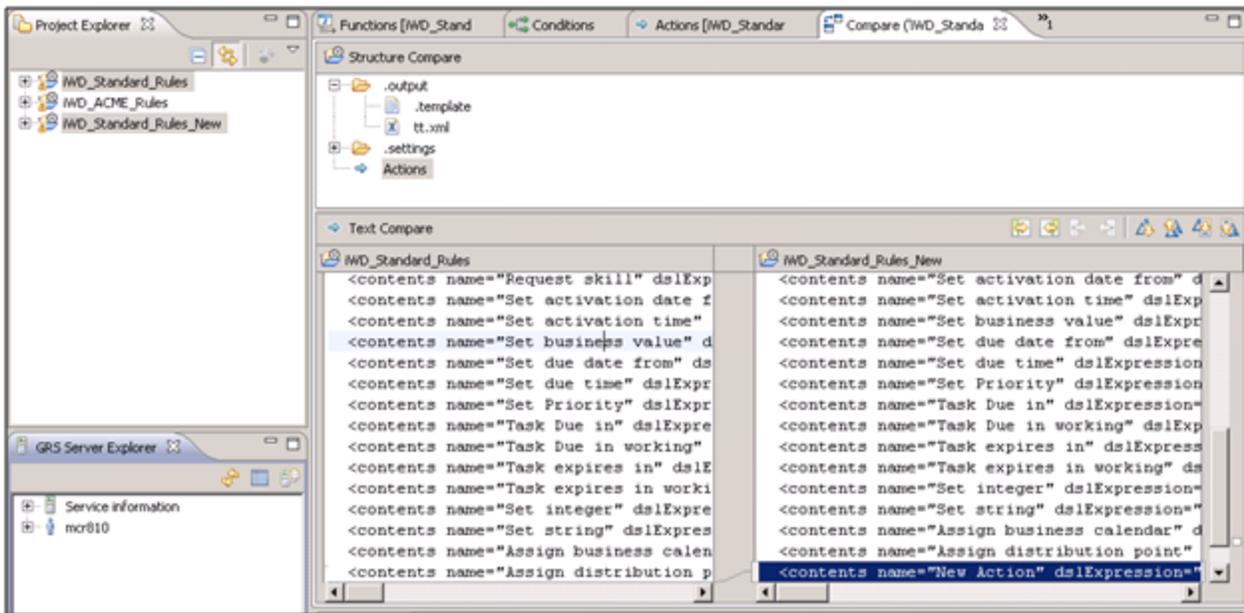
Business Rules Configuration

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 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.

increase priority							
		Add Condition		Add Action		Group	
Sections	Expression	Parameters					
When	Due time is in Priority is	18	to	72	hours		
		less than		500			
Then	Increase Priority	15					
	Reprioritize after	6	hours				

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

Use Event Log Filtering

You can configure which events will be stored in the Interaction Server Event Log database. Doing so can reduce the number of rows in that database for *active* tasks (tasks that have not yet expired and therefore, have not been purged by the iWD ETL Prune job). The events that are necessary for iWD Data Mart and the task's Event History to work properly are shown in the table called Interaction Server Events Required for Proper Functioning of iWD.

Interaction Server Events Required for Proper Functioning of iWD

Event identifier	Event name
104	EventPropertiesChanged
132	EventPartyAdded
133	EventPartyRemoved
161	EventProcessingStopped
162	EventPlacedInQueue
163	EventPlacedInWorkbin
193	EventExternalServiceRequested
194	EventExternalServiceResponded
200	EventHeld
201	EventResume

You can filter out all other events by configuring the Interaction Server Event Log Database Access Point application option `event-filter-by-id`. This option is configured in the `event-filtering` section. The value of the `event-filter-by-id` option will be a comma-separated list of the events you want to be logged. All other events will not be logged.

The list of events shown in the table above is appropriate if you are using the out-of-the-box iWD business process (IWDBP). The best way to analyze this is to turn off the `event-filter-by-id` option temporarily and to put one task through the entire business process, including reprioritization, handling at an end-user desktop, any re-queuing, and so on. Then, you can review the full list of events that have been captured by doing a query on the Event Log Database table, for a particular Interaction ID. If you do not require any custom reporting and have no need for the events other than those listed in the table above, you can apply the filter, as described.

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 that this type of load balancing and retries, 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.

Multiple Application Server Instances

It might be beneficial to set up the iWD Manager, iWD Runtime Node, and iWD Data Mart web applications on dedicated application server instances. In fact, Genesys recommends that you always use a dedicated application server instance for iWD Data Mart. However, having the other applications on dedicated instances can also improve scalability and availability.

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

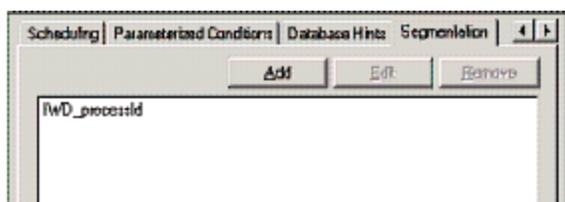
For information about the ScheduledAt property, see the topic “Setting the ScheduledAt Property” in the *Universal Routing 8.1 Business Process User's Guide*.



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

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 time out of 30 minutes before it goes to the red port. During this time, it is 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 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 you might resubmit the task into an error-handling strategy, which then sends the task back to the `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.