



Getting Started

This chapter contains the following topics:

- [About Cisco Prime Service Catalog Reporting Solution, page 1](#)
- [Reporting Workflow, page 2](#)
- [Reporting Architecture and Components , page 2](#)
- [Metrics and Attributes, page 5](#)

About Cisco Prime Service Catalog Reporting Solution

Cisco Prime Service Catalog comes with a dedicated reporting environment for business intelligence. Reporting allows you to execute reports, which are presented as tabular or graphical representations of selected data. Cisco Prime Service Catalog Reporting Solution offers two modules for business intelligence:

Reporting

The Reporting module is bundled with a basic Service Catalog license. The Reporting module provides a set of prebuilt reports on service design, organizational entities, and on transactions (requisitions and tasks) processed through Service Catalog. This module also provides a set of Key Performance Indicators (KPIs), charts which can be configured to appear on each user's reporting dashboard. The following sections discuss capabilities provided by the Reporting module:

Reporting allows you to execute reports, which are presented as tabular or graphical representations of selected data. High-level reports now include multiple levels so that you can drill-down into more detailed views.

Advanced Reporting

The Advanced Reporting module may be licensed as an add-on to a basic Service Catalog license. The Advanced Reporting module includes a data mart for Service Catalog, as well as tools for building simple queries and more complex reports based on data in the data mart.

Advanced Reporting provides complete analysis capabilities, user-defined reports, and prebuilt data marts, to allow for greater visibility and control over IT business management. Business value metrics provide in-depth insight into the quality of IT service and financial data.

You can also use Advanced Reporting to define and customize your own reporting content. Business analysts may author simple queries to create Ad-Hoc reports, while report developers may use the Advanced Reporting applications to design more detailed technical reports.

Reporting Workflow

Workflow for Reporting consists of various phases:

- 1 Purchase the licenses for reporting solution. For more information, see [Cisco Prime Service Catalog Ordering Guide](#).
- 2 An administrator installs the Reporting software module and integrates it with Prime Service Catalog application. For more information, see [Cisco Prime Service Catalog Installation and Upgrade Guide](#).
- 3 An administrator creates user roles and capabilities to access the Prime Service Catalog Reporting Solution and configure reporting components.
- 4 A service designer or a report designer ensures that the Ad-Hoc Reporting module allows the extraction to a data mart of requisition (form data) from dictionaries and services which their designers have designated as "Reportable". For more information, see [Designing Reports](#).
- 5 You can now generate reports based /on your requirements. If you are an end user interested in using the Reporting module to access and view reports, see [Using Reports](#).
- 6 Ensure that you perform regular maintenance operations.

Reporting Architecture and Components

Service Catalog comes with a dedicated reporting environment for business intelligence. The environment consists of multiple components, each of which is optimized for a particular task or set of users. These components are listed below. The environment consists of multiple components, each of which is optimized for a particular task or set of users. These components are listed below.

- **Prebuilt reports** are a set of production quality reports which provide an analysis of service, requisition, and task-level performance.
- **Key Performance Indicators (KPIs)** are a set of graphs which can be displayed on the application portal, to allow instant access to data on trends within the system.
- **Ad-Hoc Reports** allows business users to construct queries based not only on data available in the standard reports (regarding task, service, and requisition performance) but also on data from fields in custom-designed dictionaries included in the site's service forms.
- **Report Designer** allows business users to modify standard reports and to create new reports incorporating not only requisition, task, or service-related data but the service-form data (individual dictionaries and fields) that is customized for each service.
- The **Custom Java Provider** allows the Service Catalog person profile information to be used as a Cognos namespace. This provides integrated, Single Sign-On access to all reporting tools for all users registered in Service Catalog, with the reporting privileges assigned to them by the administrator.

These reporting capabilities are seamlessly integrated into the Service Catalog framework, but are implemented using tools from the IBM Cognos Series 10 Business Intelligence (BI) solution toolset. These tools are summarized below.

- **IBM Cognos 10 Connect** allows Service Catalog to display Cognos reports and other reporting objects.
- **IBM Cognos 10** provides production-quality reports to business users and nontechnical users. Such users can use Cognos 10 capabilities to print reports or save output in formats suitable of Office or other applications.
- **IBM Cognos Query Studio** (presented to users as “Ad-Hoc Reports”) allows users to perform ad-hoc queries on the reporting data.
- **IBM Cognos Report Studio** (presented to users as “Report Designer”) allows users to modify existing reports or produce new reports. It is recommended for technical users or power users, who can understand the relationships among different types of data stored in the Service Catalog database.
- **IBM Cognos Data Manager** is the tool used by Cisco engineers to produce the programs that extract data from the transactional database and load it into the dedicated reporting environment.
- **IBM Cognos Framework Manager** is used by Cisco engineers to define the data level and business level views of the information which are available to end users for ad-hoc reports and queries.
- **IBM Cognos Event Studio** defines events, triggered by the value of data in the reporting database. When an event occurs, users can be notified by means such as e-mail or running an exception report. The application does not come with any preconfigured events, but allows Advanced Reporting users to define their own events, based on contents of the data mart.

Reporting Architecture

This section describes the architecture of the Service Catalog reporting solution. It can be read by anyone curious about how the Cognos components are used and the role each plays in the solution. It can safely be skipped by those interested primarily in how to use the Service Catalog reporting solution to run the prebuilt reports supplied in the Standard Package or in generating their own ad-hoc reports or queries.

Data Mart Architecture

In general, it is not best practice to allow users to run reports in the same environment in which a transactional system such as Service Catalog is operating. The resource requirements for running reports, ad-hoc queries, and other in-depth analyses are vastly different from resource requirements for running a transactional system that responds acceptably to online users. Therefore, data that forms the basis for reports and in-depth analyses is typically extracted from the transactional system and loaded into an environment dedicated and optimized for reporting.

The Service Catalog dedicated reporting environment consists of two packages, whose contents and usage are explained in detail in the rest of this chapter.

Standard Reports Data Package

The Standard Reports Package supports the prebuilt reports and KPIs. A variety of output options provide information on task-, service-, and requisition-based measures and trends. In addition, reports on the structure of the Service Catalog data, including persons, organizations, service teams, and service groups, are available.

All data used in the prebuilt reports is also available in the Custom Reports package. Over time this package is merged with the Custom Reports package.

Custom Reports Data Package

The Custom Reports Data Package provides a dimensional model which allows ad-hoc reports on task, service and requisition-related data. In addition to the measures and attributes available in the Standard Reports package, data entered by users into the customized service forms configured at each site is available. This “form data reporting” (FDR) provides visibility into all attributes of all dictionaries and services which the service designers have designated as “Reportable”.

Standard Reports Package

The Standard Reports Package consists of a set prebuilt reports and key performance indicators (KPIs) that are supplied with Service Catalog and the database tables required to support generation of these reporting objects. These prebuilt objects meet many business unit requirements for the reports generated from operational data.

Database Tables

The database which supports the Standard Reports Package contains both detail tables and summary tables.

The detail tables provide a denormalized view of the database. Each table provides all the data that appears on a corresponding report. This means that running each report is optimized, so no report needs to access related data in lookup tables. It also means, however, that these tables *cannot* be combined in a report with other tables, since there are no relationships between the tables: each table is a complete, denormalized view of one type of fact about the OLTP system, to the specified level of detail. Further, no drill-up or drill-down, to different levels of detail, is possible.

The summary tables contain aggregated or summarized data. Presummarizing data eliminates processing delays that would otherwise occur if summary reports needed to aggregate data in real-time, in response to a report request. As for the detail tables, each summary table should be used only for its dedicated reports or KPIs—no summary tables can be joined to other tables to support ad-hoc reporting requirements.

Prebuilt Reports

The prebuilt reports that are included in the Standard Reports Package are created using the Report Studio tool and included in the Reporting module. The default report display format is set to HTML; but this delivery format, as well as other runtime parameters, can be modified when the reports are run. If the reporting solution includes Report Designer, users are able to view the definitions of the prebuilt reports and, if desired, modify the definition or create new reports to better meet corporate requirements.

Service KPIs

The service Key Performance Indicators are generated using JFreechart API's (JFreechart is not part of the Cognos product suite, but is open source software). The charts are generated on demand, by reading from the summary data tables created for each KPI.

Service Catalog Data Mart – Advanced Reporting

The Advanced Reporting module allows users to build custom reports and queries from data in the Service Catalog data mart, capturing data from Service Catalog.

Service Catalog Data Mart

The Service Catalog Data Mart is based on the Custom Reports Data Model package. This package gives users access to a data mart which includes service, task, requisition and effort-related information, such as the task performer or the duration of a task. The custom reports package differs from the standard package in important ways:

- The custom reports package does not include any prebuilt report or KPIs. It is meant strictly for ad-hoc reports and queries.
- The custom reports package allows access to form-based data, data derived from the dictionaries and attributes which are displayed and entered on user-configured service forms.
- The custom reports package is organized as a dimensional model, a flexible data model with the relationships among the different types of data, which encourages building ad-hoc reports and queries. This model is described in detail in [Creating Custom Reports](#).

Metrics and Attributes

A metric is a numeric quantity which can be aggregated. It is a measure which provides data you can use to identify and analyze IT service ordering, delivery trends, and problems.

Metrics enable you to:

- Monitor Service Catalog activity, resource consumption, and service delivery performance
- Perform root cause analysis to understand and resolve service delivery performance issues

Metrics and attributes in the data mart are derived using the computations explained in the table below.

Table 1: Metrics and Attributes Table

Metric	Description
Service Volume	Calculated as the total number of service requests within a defined time period. Depending upon the particular report, this may refer specifically to completed requests, or to requests in a variety of other states, such as ongoing or cancelled.
Service Cost	Derived from the price which is configured for a particular service. Price * Units and which may be dynamically adjusted on a request-by-request basis.

Metric	Description
Service Start Date or Time	STARTEDATE and STARTEDDATETIME for a service are initially set to the time when the customer submits the service request. STARTDATE and STARTEDDATIME values are updated as soon as authorizations are completed and the delivery moment begins. This means that before completion of authorizations STARTEDATE (TIME) refers to the time the request was submitted; after completion of authorizations it refers to the data and time the delivery moment began. All computations regarding standard compliance and task due dates use the actual delivery plan start date.
Task Volume	The total number of Authorization tasks or Delivery Plan tasks. Calculated as the total number of Authorization tasks or Delivery Plan tasks.
On-Time %	<p>A task is determined to be on time by comparing the time it was completed (CompletedDateTime) to the time it was scheduled to be completed (ScheduledDateTime). Duration (actual and default) is not used directly in this computation, although, of course, it was originally used to compute the due date and time.</p> <ul style="list-style-type: none"> • The percent of services or tasks completed on-time, on or before their calculated due date. • When a customer clicks the Order button, Service Catalog calculates the Due Dates for all tasks. • The task Due Date is calculated based on the duration configured for that task. This is the Operating Level Agreement (OLA) for that task. • The service Due Date is the date on which the last task in the delivery plan for that service is due. • This is the Service Level Agreement (SLA) for that service. • A delivery plan can have multiple late tasks, and still achieve its SLA.
Standard Compliance %	The percent of services or tasks completed within their configured duration.

Metric	Description
Task Duration	<p>Elapsed time, in hours, from start to finish, for an Authorization task or a Delivery Plan task.</p> <p>Task or a Delivery Plan task. Task duration is computed in work hours, based on the performer's calendar. Duration in reports is displayed in hours.</p> <p>Three measures of task duration are available in the task-based fact tables:</p> <ul style="list-style-type: none"> • ActualDuration measures the elapsed number of work hours, according to the customer's calendar, that it took for the task to be performed. • PerformerActualDuration measures the elapsed number of work hours, according to the performer's calendar, for the task to be performed. • DefaultDuration is the number specified by the service designer which designates the amount of time a task should take. <p>Assume, for example:</p> <ul style="list-style-type: none"> • The performer's calendar specifies 9-hour work days • The task became active at 9 am on a Monday (a work day) • The performer finished the task at 10 am the following day <p>In this case, the PerformerActualDuration would be 10 hours.</p>
Planned Effort	Derived from the amount of time configured in the Effort field for a particular Plan Task; measured in hours.
Actual Effort	Calculated from the amount of time (Effort Entry) entered by the task performer for a particular task.
Planning Effectiveness	Calculated as the percent of tasks which were completed within the configured amount of effort hours
Task Standard Compliance %	<p>A task is determined to be in compliance with its Operating Level Agreement (OLA) by comparing the performer actual duration (number of work hours between the time it was started and the time it was marked as completed) to the default duration designated for the task. The task is in compliance (the StandardComplianceFlag is 1) if the actual duration is less than the default duration.</p> <p>The percent of services or tasks completed within their configured duration. Standard Compliance % is calculated using either standard duration or task duration.</p> <p>Use for root cause analysis, to isolate on-time performance from elapsed-time performance.</p> <p>Based on the completion dates of previous tasks, it is possible for an individual task to violate its OLA, yet still be performed within its planned duration.</p>

Metric	Description
Service Standard Compliance %	<p>A service is determined to be in compliance with its Service Level Agreement (SLA) by totaling the actual duration of all component tasks and comparing this total to the default duration designated for the service. (Default duration is the Standard Duration which is configured by the service designer on the General Tab for the service). The service is in compliance (the StandardComplianceFlag is 1) if the actual duration is less than the default duration.</p> <p>The service default duration is entered and maintained manually and not validated against the component tasks of the service. Service designers should review the default duration, to ensure that it correctly reflects the default workflow, in terms of tasks completed and their durations, anticipated for the service.</p>
Task Count	<p>On the individual task level, this value is 1 or 0. For example, if the metric is called Completed On-Time Task Count, the count would be 1 for a task that was completed on-time. Tasks that are still ongoing have a value of 0. When you roll up a group of tasks the counts are summed up so you get a total count of the tasks that were completed on-time.</p>
Completed On-Time Percentage or Standard Compliance Percentage	<p>These percentages are provided at the individual task level with a value of 0 or 100. When you roll up a group of tasks, the percentages are averaged to provide an overall on-time or standard compliance percentage of the group of tasks.</p>
Late Open Task Count	<p>This count is assigned at the individual task level. If a task is late and is still in ongoing status it has a value of 1. Otherwise, it has a value of 0. You can roll up the counts from a group of tasks to see how many of them are late and still ongoing. The counts are summed up by default in the report. A high number of Late Open Task Count typically indicates a delivery process problem that leads to customer dissatisfaction.</p>
Rescheduling a task	<p>When a task is rescheduled:</p> <ul style="list-style-type: none"> • The new/rescheduled due date for the task is used for on-time calculations for that task. • There is no effect on the due date/on-time calculation for the service. • The new duration for the task is used in the calculation of standard compliance for the task. • The new duration for the task is not used in the calculation of standard compliance for the service.

Measures of Service Volume: Who is the Customer?

In the way that many customers' services are implemented, the name of the person who is the end customer of a service delivery is stored in the order form for the service. This information is not accessible in the Standard Reports Package and its prebuilt reports. So there are two options for registering the customer of the service delivery:

- Many reports refer to the CUSTOMERFIRSTNAME and CUSTOMERLASTNAME fields. If the end customer of a service is stored on the order form, these fields refer to the person who requested the service, and not necessarily the person for whom the service was performed. However, it is your only option for identifying an individual person as the customer.
- The ORGANIZATIONALUNITNAME field refers to the organizational unit that would be billed for the service delivery, if costs were being allocated. This is another way of thinking about who the customer is.

You can choose a service delivery option based on your customer.

Measures of Task Delivery Performance

This section describes how to measure the performance of individuals.

Measuring the Performance of Individuals

Tasks can be assigned to individuals or assigned to queues from which individuals can draw work. The data that is available in the Standard Reports Package and prebuilt reports (specifically the fields PERFORMERID, PERFORMERFIRSTNAME, PERFORMERLASTNAME, PERFORMEROUID and PERFORMEROUNAME in DM_SERVICETASKDETAIL and DM_AUTHORIZATIONTASKDETAIL) can refer to a person under one of three conditions:

- The task has been assigned directly to that individual, and no other person has ever worked on it.
- The task has been assigned to a queue, and the performer is the only person who has ever worked on it.
- The task has been assigned to a queue, and the performer is the last person who worked on it, but there may have been others. In this case, the performance of the person listed in the report is not their own, but is also affected by all the people who previously touched the task. So it is not a fair measure of the performance of the individual named.

The first condition can be distinguished from the data in the Standard Reports Package. However, the second and third cannot. This means that reports that attempt to measure the performance of individuals will only fairly represent their performance if service teams have very clear and consistent rules about not sharing responsibility for tasks. For this reason, a disclaimer appears on all prebuilt reports that measure the performance of individuals.

Which Duration to Use?

Prebuilt reports about the duration of tasks performed during service delivery use the PERFORMERACTUALDURATION measure. This measure takes the performer's working calendar into account, so that weekends and other nonworking hours are not counted against their time working on the task. Conversely, the CUSTOMERDURATIONOFSERVICE measure makes the calculation taking the customer's calendar into account, making it an inaccurate measure of the performer's work.

When to Use Date vs. Duration-Based Measures

There are two ways to assess whether the delivery team is performing their tasks well. Both are valid, but they have different meanings and uses:

- **Is the task late, relative to the promise (Due Date) made to the customer?** To make this assessment, reports include the TASKONTIMEFLAG, TASKDUEDATE and TASKCOMPLETEDDATE measures. This is good for determining whether a queue or service team is meeting its promises to customers in terms of absolute dates. It is *not*, however, a good measure of the performance of individuals. Assume that performing a service requires three tasks, done by Persons A, B and C. If you are measuring the performance of Person C, their tasks could be late because Person A or B was late. So this measure is used only to assess the customer service of whole teams or queues.
- **Has the task been performed within the standard time we plan for this kind of task?** To make this assessment, standard reports use the TASKSTDCOMPLIANCEFLAG, TASKPROJECTEDDURATION and PERFORMERACTUALDURATION measures. This allows you to compare the actual duration spent on performing the task with the standard planned duration for this kind of task. This measure is more appropriate for assessing the performance of individual team members, because focusing on the duration isolates the task performance from any upstream effects: it is entirely possible for a task to be completed late according to its due dates, but with a duration that is equal to or less than the standard duration. That indicates a person who is performing their own work well, but falling behind due to upstream effects.

A solid, customer-focused measurement regime requires BOTH metrics. If you base everything on task duration, you are in effect saying “Who cares what we promised to the customer, as long as we are meeting our internal standards?” On the other hand, focusing only on the dates provides an inaccurate picture of team members’ performance, which does not allow you to make the effective resourcing decisions you need to improve your performance for customers.

Attributes

The dimensional attributes used in the report packages are derived from the data maintained in the Service Catalog OLTP database. Attributes are summarized below.

Table 2: Attributes from OLTP database

Attributes	Description
Customer*	The person who is the recipient of the service, either when it is ordered directly by the same person or when the service is ordered on behalf of the customer, and the home organizational unit of that customer
Billed*	The organization which is billed for the service. This is the same as the customer for the service unless a different organization is chosen as the Bill To: Customer when the ordered is reviewed (after a request has been saved, but before it has been submitted.) Only an organizational unit that has been designated as Billable can be chosen as the Bill To: Customer.
Performer*	The person (and corresponding home OU) who performed the task. See Measuring the Performance of Individuals , on page 9 for more information.