CA Business Service Insight

Predefined Content Guide

8.2

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At http://ca.com/support, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

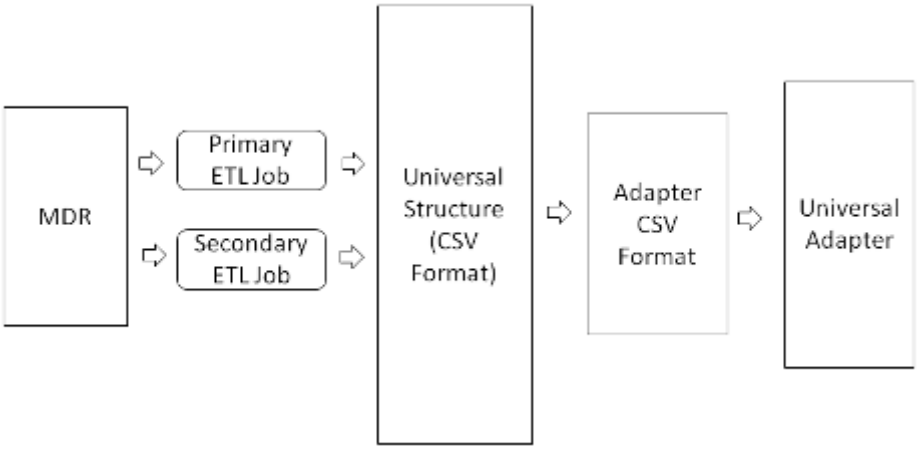
Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at http://ca.com/docs.

Contents

Chapter 1: Introduction



This guide discusses how to use CA Business Service Insight predefined content inherent components, template library structure, service level template, metrics, functions, business logic modules, and event types to generate live contract and reports.

This guide provides information about two types of packages:

- ITILv3 packages
- Functions packages

This section contains the following topics:

Target Audience

This guide is intended for:

- Service level managers who define service level agreements and implement them in the system, create service level templates and contract templates, and implement business logic formulas and adapters.
- Personnel required to view, generate and save reports and monitor whether key commitments are kept using the dashboard.
- Administrators that need to manage application users and permissions and carry out administration tasks such as define regional options, manage logs, install and unpack packages, define system preferences and manage the database and licenses.

Using This Guide

This guide is divided into the following sections:

- Introduction (see page 7) (this section), describes the benefits of predefined content and how CA Business Service Insight fulfills the ITIL concepts and processes with the ITILv3 predefined content.
- ITILv3 Content Reference (see page 13), provides descriptions for the major components in the content, for the structure of the predefined content packages, template library structure, service level templates, and the metrics.
- ITILv3 Implementation Notes (see page 25), discusses how to implement and create 'live' contracts and reports using predefined content. It provides step-by-step instructions for loading the predefined content package through committing a contract and generating reports.
- Functions Package Content Reference (see page 29), provides descriptions for the functions available in the predefined content package.
- Business Logic Modules (see page 33), provides detailed descriptions for the business logic modules and event types.

ITIL Processes Modeling Guidelines

ITIL (Information Technology Infrastructure Library) recognizes key disciplines in IT Service Management, including business perspective, service management, support and delivery, Information and Communication Technology (ICT) infrastructure management, and application management.

- The IT operation needs to understand the Business Perspective, so that appropriate services (Service Management) that align with business needs can be provided.
- The services are then constructed with an operational management component which generally has little from direct contact with clients (Service Delivery) and a Support function.
- IT Applications are created (or purchased and then implemented) and maintained.
- All of this runs on the ICT Infrastructure.
- Management of IT Security and providing appropriate systems access overlaps most of the other disciplines.

CA Business Service Insight and ITILv3 Predefined Packages

The Information Technology Infrastructure Library (ITIL) is a set of concepts and policies for managing information technology (IT) infrastructure, development and operations.

ITIL consists of a series of books giving guidance on the provision of quality IT services, and on the accommodation and environmental facilities needed to support IT.

ITIL is based on five core lifecycle titles:

- Service Strategy
- Service Design
- Service Transition
- Service Operation
- Continual Service Improvement

ITILv3 guidance takes a lifecycle approach, as illustrated below:

Key benefits of CA Business Service Insight predefined content include:

- Enabling faster implementations, using content based on best practices and ITIL standard
- Significantly reducing effort and time between kickoff and going live
- Expediting fast assimilation of ITIL processes

Functions Package

The functions package includes a set of functions that accelerate the definition of real-life quick metrics using the visual business logic workspace.

The package includes functions as well as business logic modules focused at the availability management and incident management domains. Also included are generic functions such as weighted average which complete the library of generic built-in functions.

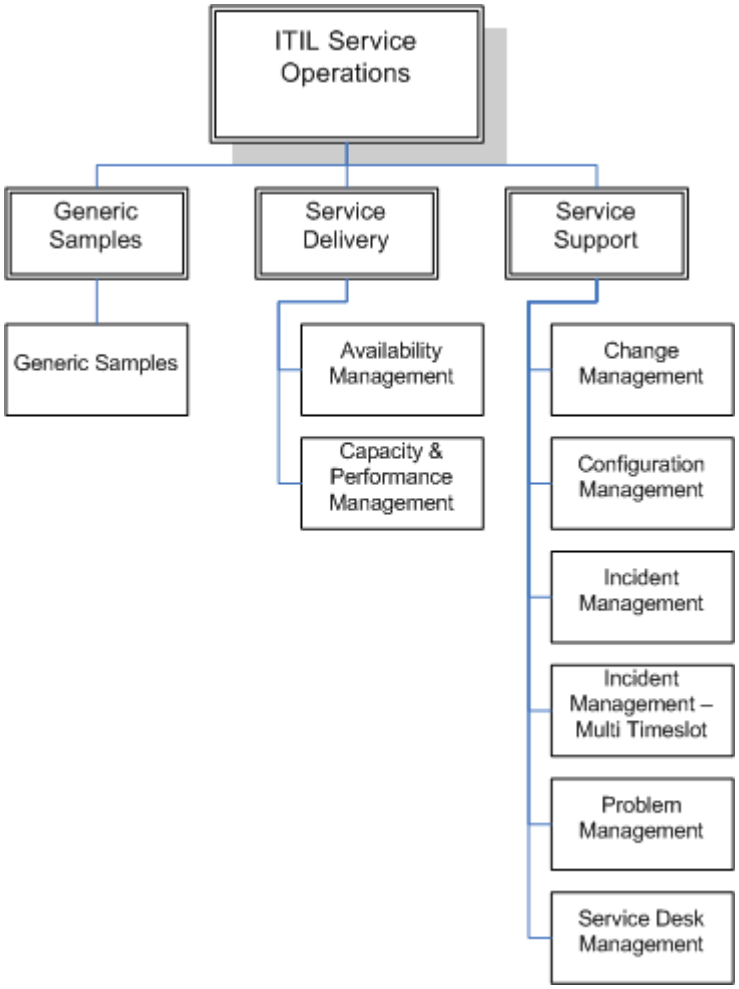
The package helps to speed-up the implementation, assure higher quality and standardization and alignment with ITILv3. Typical usage examples include:

- Availability Management - Percentage of Availability, Maximum Downtime, Number of Failures.
- Incident Management - Percent Resolution on Time, Percent Response on Time, Average Response Time.
- Generic Functions – Weighted Average, Number of Events within Threshold, Average Elapsed Time.

As the functions included in the package leverage on the CA Business Service Insight ITILv3 predefined content (which is offered under a separate license), the package is not part of the generic installation and requires separate import into a CA Business Service Insight environment.

Chapter 2: ITILv3 Content Reference

The predefined content package is structured as illustrated below:



ITIL Service Operations is a template library that reflects the phase of the ITSM Lifecycle responsible for 'business-as-usual' activities. Template Folders within this library focus on an organizations day-to-day activities and infrastructure used to deliver services.

ITIL Service Operations contains the following template folders:

- Generic Samples containing the following service level templates:
  - Generic Samples

    This service level template contains a metric that demonstrates how to use the Success Calculation Module.
- Service Delivery containing the following service level templates:
  - Availability Management
  - Capacity & Performance Management

    Each of these service level templates contains metrics that help in the management of the IT services themselves.
- Service Support containing the following service level templates:
  - Change Management
  - Configuration Management
  - Incident Management
  - Incident Management – Multi Timeslot
  - Problem Management
  - Service Desk Management

    Each of these service level templates contains metrics that help verify IT services are provided.

This section contains the following topics:

Generic Samples Template Folder

The Generic Samples template folder contains the following service level templates:

- Generic Samples

  This service level template contains metrics that demonstrate the use of advanced non-specific metrics.

  For the list of metrics see Generic Samples Metric List (see page 15).

Generic Samples Metric List

The following table lists the Generic Samples metrics.

Refer to Generic Business Logic Modules (see page 39) for in depth description of the business logic modules associated with this service level template.

Metric Name	Description
Success Calculation Sample	Counts successful events or calculates the percentage of successful events in all the events considered. Parameters enable filtering of irrelevant events, ignoring duplicate events, and ignoring events that are out of timeslot.

Service Delivery Template Folder

Service delivery is the management of the IT services themselves. It involves a number of management practices to ensure that IT services are provided as agreed between the service provider and the customer. The processes included in this service level template are of longer term planning and improvement nature of the IT services.

The Service Delivery template folder contains the following service level templates:

- Availability Management

  This service level template contains metrics that ensure the practice of identifying levels of IT service availability for use in service level reviews with customers. The goal is to optimize the capability of delivering services to whom and when it is required cost effective and sustained level of availability that enable the business to satisfy business objectives.

  For the list of metrics see Availability Management Metric List (see page 16).

- Capacity & Performance Management

  This service level template contains metrics that are responsible for ensuring that IT processing and storage capacity provision match the evolving demands of the business in a cost effective and timely manner. Capacity management can be defined as business capacity management, service capacity management, and resource capacity management.

  For the list of metrics see Capacity & Performance Management Metric List (see page 17).

Availability Management Metric List

The following table lists the Availability Management metrics.

Refer to Availability Management Business Logic Modules (see page 35) for in depth description of the business logic modules associated with this service level template.

Metric Name	Description
Components number of failures (measured by monitoring tool)	Counts the number of times a service fails during the calculation period. This measurement assists with identifying problematic CI (Configuration item) to replace or maintain.
Max single failure time	Measures the maximum single failure time during the calculation period.
MTBF (Mean Time Between Failures)	The average elapsed time from the time the Service or supporting component is fully restored until the next occurrence of a failure to the same service or component.
MTBSI (Mean Time Between System Incidents)	Currently not implemented.
MTTR (Mean Time To Repair)	The average elapsed time from the occurrence of an incident to the resolution of the incident. The average time it takes to repair customer services.
Percent of service availability (based on monitoring tool events)	The available time out of the service agreed time up to the present based on the availability of all resources. Presents the current status of the Service.
Percent of service availability (based on monitoring tool events) - Advanced	The available time out of the service agreed time up to the present based on a threshold of available resources. Presents the current status of the Service.
Percent of service availability (correlates monitoring tool events and incidents)	Measures incident management system incidents reported by the customer and monitoring tool events.
Percent of Service availability with direct impact on users (based on incidents)	Measures Incident management system incidents reported by the customer.
Service number of customer impacting failures (correlates with incidents)	Measured by incident management system with incidents reported by the customer.
Service number of failures (measured by monitoring tool)	Measures the number of failures based on monitoring tool events.

Capacity & Performance Management Metric List

The following table lists the Capacity and Performance Management metrics.

Refer to Generic Business Logic Modules (see page 39) for in depth description of the business logic modules associated with this service level template.

Metric Name	Description
Average latency	Calculates average latency.
Average throughput	Calculates average throughput.
Customer satisfaction level	Overall customer satisfaction level.
Percent of CPU usage	Calculates the percent of CPU usage.
Percent of file storage usage	Calculates the percent of file storage.
Percent of memory usage	Calculates the percent of memory usage.
Percent of time CPU exceeds utilization	Percent of time that the CPU utilization exceeds a predefined threshold.
Percent of time file storage exceeds utilization	Percent of time that the file utilization exceeds a predefined threshold.
Percent of time memory exceeds utilization	Percent of time that the memory utilization exceeds a predefined threshold.
Percent of transactions completed on time	Percent of transactions completed within a predefined threshold.
Transaction average delivery time	Calculates the average transaction time.
Worst transaction time	Worst transaction time within the calculation period.

Service Support Template Folder

The Service Support template folder enables IT Services to be provided, and includes the processes of day-to-day operations that support the IT services.

The Service Support template folder contains the following service level templates:

- Change Management

  This service level template contains metrics that verify all changes to configuration items are carried out in a planned and authorized manner. Indicators related to the creation of request for changes, reviewing, prioritizing, planning and executing of maintenance processes.

  The objective of this service level template is to verify that standardized methods and procedures are used for efficient and prompt handling of all changes to controlled IT infrastructure, to reduce the number and impact of any related incidents upon service.

  For the list of metrics see Change Management Metric List (see page 19).

- Configuration Management

  This service level template contains metrics that are responsible for managing the service level template infrastructure and continuously aligning it with the (business) services.

  This service level template enables control of the infrastructure by monitoring and maintaining the information on all the resources needed to deliver services, it tracks all of the individual configuration items in an IT system which may be as simple as a single server, or as complex as the entire IT.

  For the list of metrics see Configuration Management Metric List (see page 20).

- Incident Management

  This service level template contains metrics that involve recording of incidents, initial support of incidents, the classification of incidents, the necessary investigation and deciding how to resolve the incident. The primary goal of the service level template is to restore normal service operations as quickly as possible and reduce the impact on business operations.

  For the list of metrics see Incident Management Metric List (see page 20).

- Incident Management – Multi Timeslot

  This service level template contains several metrics to handle incident management in a multi- location environment while each location differs by its timeslot and time zone.

  For the list of metrics see Incident Management – Multi Timeslot Metric List (see page 22).

- Problem Management

This service level template contains metrics that verify the resolution and prevention of incidents that affect the normal running of an organization's IT services. This includes ensuring that faults are corrected, preventing any recurrence of these faults, and the application of preventative maintenance to reduce the likelihood of these faults occurring in the first instance.

For the list of metrics see Problem Management Metric List (see page 22).

- Service Desk Management

  This service level template contains metrics that act as a single point of contact for all incoming calls to the IT department, provide a customer focused interface between the users and the IT to enable the efficient use of IT services.

  For the list of metrics see Service Desk Management Metric List (see page 23).

Change Management Metric List

The following table lists the Change Management metrics.

Refer to Incident Management Business Logic Modules (see page 50) for in depth description of the business logic modules associated with this service level template.

Metric Name	Description
Average change resolution time	Measures the average length of a change process. Measures the process efficiency.
Number of incidents caused by improper implementation of change	Currently not implemented.
Percent of changes reopened	Measures the rate of changes that were reopened. These measures reflect Change Management efficiency level.
Percent of changes resolved on time	Rate of changes resolved according to the customer expectations. Measures change success rate.
Percent of changes resolved on time, in business days	Rate of changes resolved according to the customer expectations measured in business days. Sensitive to change request location time slot and time zone.
Percent of changes resolved on time, on the same day	Rate of changes resolved on the same day, according to the customer expectations. Sensitive to change request location time slot and time zone.
Percent of changes responded on time	Rate of changes responded to, according to the customer expectations.

Metric Name	Description
Percent of incidents escalated to changes	Currently not implemented.
Total number of changes closed	Number of changes closed in pipeline.
Total number of changes in process	Number of changes currently in process.
Total number of changes opened	Number of changes in pipeline. Measures Change Management workload.

Configuration Management Metric List

The following table lists the Configuration Management metrics.

Refer to Generic Business Logic Modules (see page 39) for in depth description of the business logic modules associated with this service level template.

Metric Name	Description
Number CI tracked	Number of configuration items actually logged and tracked.
Number of incidents due to inaccurate CIs	Number of incidents caused by inaccurate configuration information.
Number of licenses not used	Number of software licenses that are not deployed. The aim is to minimize the cost of unused licenses.
Percent of CMDB completeness	Currently not implemented.
Percent of inaccurate CIs	Percent of configuration items which have different configuration compared to the logged information.

Incident Management Metric List

The following table lists the Incident Management metrics.

Refer to Incident Management Business Logic Modules (see page 50) for in depth description of the business logic modules associated with this service level template.

Metric Name	Description
Average resolution time for priority X incidents	Measures the average resolution time for incidents of a specified priority.

Metric Name	Description
Average response time for priority X incidents	Measures the average time to respond incidents of a specified priority.
Max resolution time for priority X incidents	Measures the maximum resolution time for incidents of a specified priority.
Max response time for priority X incidents	Measures the maximum response time for incidents of a specified priority.
Min resolution time for priority X incidents	Measures the minimum resolution time for incidents of a specified priority.
Min response time for priority X incidents	Measures the minimum response time for incidents of a specified priority.
Percent of incidents escalated	Currently not implemented.
Percent of incidents reopened for priority X incidents	Measures the rate of incidents that were reopened. These measures reflect Incident Management efficiency level.
Percent of priority X incidents resolved on time	Measures the rate of achieving the target resolution time for incidents based on the target resolution time per each priority set by the contract.
Percent of incidents resolved on time, in business days	Measures the rate of achieving the target resolution time (in business days) for incidents based on the target resolution time set by the contract.
Percent of incidents resolved on time, on the same day	Measures the rate of achieving the target resolution time (on the same day) for incidents based on the target resolution time set by the contract.
Percent of priority X incidents responded on time	Measures the rate of achieving the target response time for incidents based on the target response time per each priority set by the contract.
Total number of incidents closed	Counts the number of incidents that were closed during the calculation period. This measures helpdesk efficiency.
Total number of incidents in process	Counts the number of incidents that were processed during the calculation period. This measures helpdesk efficiency.
Total number of priority X incidents closed	Counts the number of incidents that were closed during the calculation period. This measures helpdesk efficiency.
Total number of priority X incidents opened	Counts the number of incidents that were opened during the calculation period. This measures helpdesk efficiency.

Note: When X appears in the metric name it means that there are multiple priorities.

Incident Management – Multi Timeslot Metric List

The following table lists the Incident Management – Multi Timeslot metrics.

Refer to Incident Management Business Logic Modules (see page 50) for in depth description of the business logic modules associated with this service level template.

Metric Name	Description
Location Timeslot Sender	Sends timeslot enter and timeslot exit events based on the metric's time zone and timeslot.
Multi timeslot percent of incidents resolved on time	Rate of incidents resolved according to the customer expectations. Sensitive to change request location timeslot and time zone.
Multi timeslot percent of incidents resolved on time, in business days	Measures the rate of achieving the target resolution time (in business days) for incidents based on the target resolution time set by the contract. Sensitive to change request location time slot and time zone.
Multi timeslot percent of incidents resolved on time, on the same day	Measures the rate of achieving the target resolution time (on the same day) for incidents based on the target resolution time set by the contract. Sensitive to change request location time slot and time zone.
Timeslot Events Receiver (Dummy Metric)	Dummy metric which includes registration to timeslot enter and exit event types, to verify they are included as part of the package.

Problem Management Metric List

The following table lists the Problem Management metrics.

Refer to Incident Management Business Logic Modules (see page 50) for in depth description of the business logic modules associated with this service level template.

Metric Name	Description
Average problems resolution time	Measures how long the average problem resolution process takes. Measures the efficiency of the process.
Percent of incidents escalated to problems	Currently not implemented.

Metric Name	Description
Percent of problems reopened	Measures the rate of problems that were reopened, these measures impact the problem solution efficiency level.
Percent of problems resolved on time	Rate of problems resolved according to the customer expectations. Measures the problems success rate.
Percent of problems resolved on time, in business days	Rate of problems resolved according to the customer expectations in business days.
Percent of problems resolved on time, on the same day	Rate of problems resolved on the same day, according to the customer expectations.
Total number of problems closed	Number of problems closed in pipeline.
Total number of problems in process	Number of problems currently in process.
Total number of problems opened	Number of problems in pipeline. Measures Problem Management workload.

Service Desk Management Metric List

The following table lists the Service Desk Management metrics.

Refer to Service Desk Management Business Logic Modules (see page 59) for in depth description of the business logic modules associated with this service level template.

Metric Name	Description
Average call duration	Average call experience for customers.
Average waiting time (for abandoned calls)	Average time customers wait before call is abandoned
Average waiting time (for answered calls)	Average time customers wait for their calls to be answered
Percent of abandoned calls	Number of abandoned calls after the agreed waiting time.
Percent of calls answered within a predefined number of seconds	Calculates the percentage of calls from the total number of calls that were answered within the specified number of seconds.
Percent of calls answered in more than 60s	Rate of calls answered after a 60 second time frame.

Metric Name	Description
Percent of calls escalated to incident management	Rate of calls converted to incidents divided by the total number of calls.
Total number of calls	Total number of calls recorded.

Chapter 3: ITILv3 Implementation Notes

The predefined content package in CA Business Service Insight is implemented so as to create live contracts and reports.

The implementation workflow includes the following steps:

1. Loading the predefined content package.
    a. Upload the predefined content.
    b. Unpack the predefined content package.

    For detailed explanation on how to perform these steps, see the Packages section in the CA Business Service Insight User Guide.
2. Building a custom template library based on the predefined template library you unpacked (focusing on specifics necessary to fine tune to the customer environment).
    a. Create a custom template library based on the predefined content template library that was unpacked.
    b. Add a template folder.

    Note: You can arrange your template library in any way that best suits your business needs. However, we recommend that you do not alter the original ITIL Service Operations template library. This is a starting point for building your ITILv3 structure.
        a. Add a service level template.
        b. Modify any of the metric's details.
        c. Customize the service level template.

    For detailed explanation on how to perform these steps, see the Template Libraries section in the CA Business Service Insight User Guide.
3. Creating the contract.
    a. Create a contract based on the service level template you created.

    For detailed explanation on how to perform these steps, see the Template Libraries and the Contracts sections in the CA Business Service Insight User Guide.
4. Generating data using the predefined content event types.
    a. Before committing the new contract, ensure that the incoming data can interface with the predefined event types. Event types are included in predefined content but adapters are not. The reason that the adapters are not included is that data sources vary with each customer. When creating an adapter to interface with predefined content select the predefined event types and map the relevant fields in the input to the event type fields.
    b. Make sure your resource model is aligned with the registrations. If there are custom resource types and resource groups, be sure to associate them to the proper resources and update any metric registrations that use them. The predefined content is shipped by default with registrations to resources associated with contract party and service.

For detailed explanation on how to perform these steps, see the Adapters, Event Types and Resources sections in the CA Business Service Insight User Guide.

Chapter 4: Functions Package Content Reference

The CA Business Service Insight predefined content package includes additional functions on top of the generic ones that are released with CA Business Service Insight. This enables gaining more value from the quick metrics module, which streamlines the end-to-end process of creating reports on top of the data in a bottom-up approach.

The new functions are divided into the following service domains:

- Availability Management
- Incident Management
- Generic Formulas – Advanced
- Generic Formulas – Elapsed Time

Notes: To import the functions package you should:

Upload the package either manually to the relevant folder or using the upload functionality in the packages feature. After uploading the package is shown as available only in the import page and not in packages page.

Import the package using the import functionality.

This section contains the following topics:

Availability Management Functions List

The following table lists the Availability Management functions.

Function Name	Description
Percentage of Availability	Calculates the percentage of time when all components (resources) are available. The calculation is performed based on up/down events.
Maximum Downtime	Calculates the maximum downtime that occurred in a period. The calculation is performed based on up/down events.

Function Name	Description
MTBF (Mean Time between Failures)	Calculates the average time between failures. The calculation is performed based on up/down events.
Number of Failures	Calculates the number of failures in a period. The calculation is performed based on up/down events.
MTTR (Mean Time To Repair)	Calculates the average elapsed time from the occurrence of an incident to the resolution of the incident. The average time it takes to repair customer services. The calculation is performed based on up/down events.

Incident Management Functions List

The following table lists the Incident Management functions.

Function Name	Description
Percent Resolution on Time	Calculates the percentage of tickets resolved on time
Average Resolution Time	Calculates the average tickets resolution time
Maximum Resolution Time	Calculates the maximum time to resolve tickets
Minimum Resolution Time	Calculates the minimum time to resolve tickets
Percent Response on Time	Calculates the percentage of tickets responded on time
Average Response Time	Calculates the average tickets response time
Maximum Response Time	Calculates the maximum tickets response time
Minimum Response Time	Calculates the minimum tickets response time

Generic Formulas – Advanced Functions List

The following table lists the Generic Formulas - Advanced functions.

Function Name	Description
Weighted Average	Calculates the weighted average value of selected fields and weights
Number of events within Threshold	Counts the number of events where a selected value is within a specific threshold
% of events within Threshold	Calculates the percentage of events where a selected value is within a specific threshold
Percentage of value	Calculates the percentage of value out of another value

Function Name	Description
Count Unique	Counts the number of events with a unique value in a selected field

Generic Formulas – Elapsed Time Functions List

The following table lists the Generic Formulas - Elapsed Time functions.

Function Name	Description
Average Elapsed Time	Calculates the average elapsed time between two date fields
Maximum Elapsed Time	Calculates the maximum elapsed time between two date fields
Minimum Elapsed Time	Calculates the minimum elapsed time between two date fields
Percentage of Elapsed Time within Threshold	Calculates the percentage of events where elapsed time between two date fields is within threshold.

Chapter 5: Business Logic Modules

The following business logic modules are provided with the predefined content packages:

Domain	Module Name	Description
Availability Management	Availability and Failure Statistics	Calculates the availability of services or components. In addition provides a set of statistical calculations of availability and failures.
Generic Modules	Average Measurements	Calculates simple average and weighted average.
	Generic Calculation	Provides a set of generic calculations such as min, max, sum, percent of, etc.
	Success Calculation	Counts successful events using a user-defined success criteria.
Incident Management	Percent of Reopened Tickets	Calculates the number of reopened tickets in a calculation period.
	Resolution Time	Resolution time of an incident is calculated as the elapsed time between when the incident was opened until a final solution was provided for it.
	Response Time	The response time is calculated as the time passed from the opening of the incident until it was first handled.
	Timeslot Events Sender	Sends timeslot enter and timeslot exit events based on the metric's time zone and timeslot, so as to supply the timeslot sensitive metric with the location specific timeslot information.
	Total Number of Incidents	Counts the number of incidents received in the calculation period.
Service Desk Management	Call Center	Calculates statistics for the performance of a call center.

This section contains the following topics:

General Features

The following are general features that are generic to all modules.

- Debugging

  If the metric supplies a parameter called Debug and its value is Yes the module writes messages to the log. The messages are written in every event handler and in various strategic places in the module.

- Callbacks

  The Success Calculations module allows the extension of the basic functionality of the module by way of callbacks. The callbacks are supported in the following places:

  - Before Period Start
  - After period start
  - Before timeslot enter
  - After timeslot enter
  - Before event
  - After event
  - Before on load
  - After on load
  - Before result
  - After result
  - Period end
  - On metric changed
  - Timeslot exit
  - On registrations

  The callbacks that are performed before the events can return true in which case the rest of the execution of that event handler will be skipped. This allows the user to replace the functionality of the event handler with one of his/her own.

  In the Before result callback, the alternate value for Result is set in the callback function.

  In the After result callback, the calculated result (received as a parameter) is already set and therefore the information can be used in the callback.

Availability Management Business Logic Modules

The following sections describe the business logic modules used in the metrics common to availability and failure statistics.

Availability Management Assumptions

The following assumptions and basic behaviors are taken for all the modules under the Availability domain:

- The initial status of the Component, at the first calculation period, is Up, before any event is received to report on this component.
- There is a NONE status. When a resource status is unknown, the status up-to the NONE status is considered based upon the last resource real status.
- Whenever out of timeslot period is defined the exception period is handled in the same manner.
- In case the whole calculation period falls out of timeslot the result of the period is NULL.
- Forced Events: In some cases the information on the service availability from the monitoring tool is incorrect or not valid. For these cases, an additional event is defined that its status is forced over any event received prior to it. For example, if the Service was reported as Up and a Force Down event is received, the Service is considered as Down
- Incident Events: In some cases incident events reports for the availability status. For example "Incident – Open" for priority 1 ticket stands for DOWN status while "Incident – Resolved" for priority 1 ticket stands for UP status.
- When a new event arrives, the formula checks the status of the previous event. If the status of the previous event is down, the period between this event and the new one is considered as down.
- Events may be reported for the status of a specific component or device or globally for the service. In case both global and component events are reported, the global is considered as the real status.

Availability Management Event Types

The following Availability management event types provide the status for the availability of a specific device or service:

- Availability UP Event
- Availability DOWN Event
- Availability Force UP Event
- Availability Force DOWN Event
- Availability Force NONE Event

In addition to the above, incident events (as described in the Incident Management Module section) may be used to reflect the availability status of the service or of a device. For example Incident – Open may represent DOWN status while Incident – Resolved represents UP status.

Availability Event Type Structure

The following table shows the structure for all Availability event types:

#	Name	Field Type	Field Description	Required for Calculations
1	Component	String	Resource on which availability is reported.	N
2	AvailabilityVal	Float	% of Availability.	N

Availability and Failure Statistics Module

Percentage of Service/Component Availability Assumptions

Calculates the percentage of time the Service/Component was available in a certain period of time.

Availability calculations are done either for individual components or for a system. For an example of implementation of availability Business Logic, refer to Case Study 14: Time Accumulation Clock Handling under Appendix 2 in the Implementation Guide.

Availability is calculated as follows:

- For each calculation period the calculation method is:

  Service or Component Availability equals the Total AST (Agreed Service Time, calculated as the time from the beginning of the calculation period until the end of the calculation period excluding out of timeslot periods) minus the Actual Downtime during the agreed service time (calculated as the accumulation of all down time periods within the calculation period excluding out of timeslot periods) divided by the Total AST multiplied by 100%.

Availability for a system built out of a cluster of resources:

- It determines whether the system is available at a certain point of time based on the system logic, which is based on different types of load-balancing between the resources in the cluster.
- Component balancing: If at least the number of items based on the Threshold parameter is up, the system is up.

Failure Statistics Assumptions

Failure statistics is constructed to deliver several types of calculations in the Availability domain. These calculations are very similar in nature to Percentage of Service/Component Availability and therefore are implemented in the same module. The module produces the result of the calculation determined by the WhatToCalculate parameter.

- If downtime is reported more than once, it is counted as a single failure. In cases when the events are redundant, the downtime is counted once it initially started.
- Downtime is considered as a failure when falls within the agreed service time within the timeslot.

Module Interface

The following table lists the parameters.

Name	Description
WhatToCalculate	Determines the calculation result.

Threshold	The number (or percentage) of resources required to be UP for the system to be UP.
ThresholdUnit	The unit for the Threshold (Items or %).
Debug	Determine if in debug mode.

Module Dependencies

The following table lists the included modules.

Name	Description
Log Library (LL)	Library for enhanced logging functionality.
Parameter Kernel Library (PKL)	Library responsible for verifying and populating module parameters.
Parameters Overlay - Basic Action Library (POBAL)	Library responsible for verifying and populating some generic parameters.
Callback Kernel Library (CKL)	Library that enables the callback functionality.
Quick Metric Filtering (QM)	Filtering module required for Quick Metric functionality.
Quick Metric Mapping (QM)	Mapping module required for Quick Metric functionality.

Module Calculations (WhatToCalculate parameter)

- MAX – The maximum time period the system was unavailable during the calculation period.
- MTBF – Mean Time Between Failures. The average time between failures during the calculation period.
- MTTR – Mean Time To Recover. The average elapsed time from the occurrence of a failure to the resolution of the failure.
- NumOfFailures – The number of times the system was unavailable during the calculation period.
- Availability – The percentage of time the system was available during the calculation period.
- PercentOfDowntime - The percentage of time the system was unavailable during the calculation period.
- AvailabilityTime – The time that the system was available during the calculation period.
- Downtime - The time that the system was unavailable during the calculation period.

Generic Business Logic Modules

This section provides description for set of generic business logic modules that can be utilized in the calculation of several domains and types of calculations such as capacity and performance management, configuration management, customer satisfaction etc.

All configuration management metrics are utilizing the generic calculation module.

Capacity and performance management metrics and customer satisfaction metrics are utilizing the average measurements module.

The following are example of event types that are used as input for the generic modules:

Performance Event Type Structure

#	Name	Field Type	Field Description
1	Timestamp	Date	Time of measurement.
2	Component	String	Resource on which performance is reported.
3	CPU	String	% of CPU load.
4	Memory	Float	% of Memory utilization.
5	Throughput	Float	Average bytes/second.
6	Latency	Float	Average of seconds.

Transaction Event Type Structure

#	Name	Field Type	Field Description
1	Timestamp	Date	Time measurement.
2	Transaction	String	ID/Type of transaction conducted.
3	Component	String	Resource on which performance is reported.
4	Transaction Description	String	Additional information about the transaction.
5	Duration	Float	Response time of the transactions.

Customer Satisfaction Event Type Structure

#	Name	Field Type	Field Description
1	Customer	String	Customer Name.
2	SatisfactionLev	Integer	Customer Level of Satisfaction.

Configuration Event Type Structure

#	Name	Field Type	Field Description
1	Customer	String	Customer name.
2	NumberOfCIs	Integer	# of configuration items.
3	NumOfIncidents	Integer	# of incidents caused by improper configuration of assets.
4	NumOfDeviations	Integer	# of deviations identified between configuration repository and actual asset configurations.
5	NumOfUnAcctLic	Integer	# of licenses purchased and not accounted for in repository.

Average Measurements Module

The average measurements module contains formulas for standard average calculation. It was designed to be reusable with custom business logic modules.

Average Measurements Module Interface

The following table lists the parameters.

Name	Description
Valuefield	Specifies the integer value of the event field required for calculation.
WeightField (int)	The number indicates which event field contains the weight value included in calculations.
Debug	Determines if in debug mode.

Module Dependencies

The following table lists the included modules.

Name	Description
Log Library (LL)	Library for enhanced logging functionality.
Parameter Kernel Library (PKL)	Library responsible for verifying and populating module parameters.
Parameters Overlay - Basic Action Library (POBAL)	Library responsible for verifying and populating some generic parameters.
Callback Kernel Library (CKL)	Library that enables the callback functionality.

Quick Metric Filtering (QM)	Filtering module required for Quick Metric functionality.
Quick Metric Mapping (QM)	Mapping module required for Quick Metric functionality.

Module Calculations (WhatToCalculate parameter)

Metrics that use the Average Measurements module calculate a simple average or weighted average calculation (if relevant value field was reported as parameter). Therefore, there is no WhatToCalculate parameter.

Event Registration

Any event which includes numeric fields can be used for registration.

Count Unique Module

Counts the number of events with a unique value in a selected field.

Count Unique Module Interface

The following table lists the parameters.

Name	Description
ValueField	The name of the event field to check uniqueness.
Debug	Determines if we are in debug mode.

Module Dependencies

The following table lists the included modules.

Name	Description
Log Library (LL)	Library for enhanced logging functionality.
Parameter Kernel Library (PKL)	Library responsible for verifying and populating module parameters.
Parameters Overlay - Basic Action Library (POBAL)	Library responsible for verifying and populating some generic parameters.
Entity Counter Library (ECL)	Library used to keep track of entity counters.
Quick Metric Filtering (QM)	Filtering module required for Quick Metric functionality.
Quick Metric Mapping (QM)	Mapping module required for Quick Metric functionality.

Module Calculations (WhatToCalculate parameter)

Metrics that use the Count Unique module calculate a simple count unique therefore there is no WhatToCalculate parameter.

Event Registration

Any event with a string field that can be used to count uniqueness based on, can be used for registration.

Elapsed Time Module

Provides set of calculations for calculating elapsed time between two time fields while out of timeslot periods are excluded from calculation.

Elapsed Time Module Interface

The following table lists the parameters.

Name	Description
WhatToCalculate	Determines the calculation result of the calculation period.
TimeUnit	Determines in which unit the result of time is presented (seconds, minutes, hours or days).
TargetTime	The threshold in which each event is compared to in order to calculate the percentage of events within this threshold.
Debug	Determines if in debug mode.

Module Dependencies

The following table lists the included modules.

Name	Description
Log Library (LL)	Library for enhanced logging functionality.
Parameter Kernel Library (PKL)	Library responsible for verifying and populating module parameters.
Parameters Overlay - Basic Action Library (POBAL)	Library responsible for verifying and populating some generic parameters.
Aggregation Library (AL)	Library used to produce results for common calculations.
Quick Metric Filtering (QM)	Filtering module required for Quick Metric functionality.

Quick Metric Mapping (QM)	Mapping module required for Quick Metric functionality.

Module Calculations (WhatToCalculate parameter)

- %WITHINTHRESHOLD/PERCENT – percentage of events that the elapsed time between the two date fields is less than the threshold.
- CNTWITHINTHRESHOLD – counts the number of events that the elapsed time between the two date fields is less than the threshold.
- AVG – Average elapsed time
- MAX – Maximum elapsed time
- MIN – Minimum elapsed time

Event Registration

Any event with two date fields (named Date1, Date2) can be used for registration.

Generic Calculation Module

The Generic Calculation Module provides generic calculations such as min, max, percent for fields provided as input.

Generic Calculation Module Interface

The following table lists the parameters.

Name	Description
WhatToCalculate	Determines the calculation result of the calculation period.
Valuefield	Specifies the integer sequence value of the event field required for calculation.
Valuefield2	Specifies the integer sequence value of the event field required for calculation. Used only when WhatToCalculate is set to ADVANCEDPERCENT.
Debug	Determines if in debug mode.
Threshold (int)	Specifies the calculation threshold.

Module Dependencies

The following table lists the included modules.

Name	Description
Log Library (LL)	Library for enhanced logging functionality.

Parameter Kernel Library (PKL)	Library responsible for verifying and populating module parameters.
Parameters Overlay - Basic Action Library (POBAL)	Library responsible for verifying and populating some generic parameters.
Aggregation Library (AL)	Library used to produce results for common calculations.
Callback Kernel Library (CKL)	Library that enables the callback functionality.
Quick Metric Filtering (QM)	Filtering module required for Quick Metric functionality.
Quick Metric Mapping (QM)	Mapping module required for Quick Metric functionality.

Module Calculations (WhatToCalculate parameter)

- AdvancedPercent – Calculates the percentage of the sum of one value field from the sum of a second value field.
- %WITHINTHRESHOLD/PERCENT – Calculates the percentage of events in which the field provided in the ValueField parameter is below the threshold.
- CNTWITHINTHRESHOLD – counts the number of events in which the field provided in the ValueField parameter is below the threshold.
- AVG - Average of the field provided in the ValueField parameter.
- MAX - Maximum of the field provided in the ValueField parameter.
- MIN - Minimum of the field provided in the ValueField parameter.
- COUNT - Count the number of events.
- SUM – Sum of the field provided in the ValueField parameter.

Event Registration

Any event that includes numeric fields can be used for registration.

Success Calculations Module

This module receives events and checks those events using success-criteria. It can perform the following calculations:

- Count – Count the number of events that fulfill the success criteria during the tracking period.
- Percent-Of Success – Count the percent of the events that fulfill the success criteria out of the total number of received events during the tracking period.

Example of Usage

An event type that handles support calls exists, and one of the fields is the call durations. This module can be used to calculate the percent of calls that lasted more than 30 minutes.

General Features

The following are general features of the Success Calculations module:

- Flexible behavior based on parameters

  The Success Calculations module has a set of parameters that configures its behavior.

  For every parameter there is a default value so that if a parameter is not defined in the metric, this default value is assumed. This allows for the creation of a simple metric that assumes default behavior without the need to create each and every parameter that is supported by the module.

- Simple versus advanced parameter settings:

  Users can define simple comparisons. If such comparisons are not enough, and users want to give the module maximal flexibility, it is possible to configure the module's behavior using complex expressions.

- Filtering

  The Success Calculations module supports filtering. This is accomplished using an optional table parameter called FilteringCriteria where one can encode a certain criteria (such as field X equals 3) and all events which do not meet that criteria are filtered out and not handled by the module. The exact structure of the table parameter is described in Criteria Parameter Structure (see page 47). If the metric does not supply this parameter then all the received events are handled.

- Distinctness

  The Success Calculations module supports distinctness. This means that the metric can supply a parameter that contains a field identifier. If the module receives several events with the same value in that field name during the same period of time, only the first of those events is handled. If the user does not supply this parameter or the parameter is empty then all the events are handled.

- Timeslot awareness

  The Success Calculations module can be aware of timeslots or not based on a parameter value. This is accomplished using an optional parameter called IgnoreTimeslots. If the value of this parameter is Yes, the module ignores all events received outside of timeslot hours. If the parameter is No, or the parameter is not defined the metric handles all events.

Criteria Parameter Structure

There are two parameters that define criteria in the Success Calculations module: the filter criteria and the success criteria. The structure of these parameters is identical.

The function is called with the current event details (i.e. the eventDetails parameter from the event handler) so as to base the success test on the details of the event being handled. This table parameter includes all the information needed to decide whether the current event is to be considered a success or not.

The return value is either true or false, based on the calculated result.

No entries in this table mean that all entries are considered successful.

The success criteria is calculated based on a comparison of a single or multiple fields to specific values, or based on the evaluation of expressions. Multiple criteria lines are joined based on user defined logical relations.

For example, if a successful event is defined as one that has a value in the field result that is between 3 and 6, the user would supply two lines, using the and relationship, and ask for:

```
result > 3 and result < 6
```

The format of the table is as follows:

Row #	Argument 1	Operator	Argument 2	Inter-row logical link
1				
2				
…				

The table fields are:

- Argument 1 – This field should be filled with the name of a field in the event details from which a value is to be extracted. When the Eval operator is used, this field is ignored, and may be left empty. This field can also contain the following:
  - @Resource – This returns the name of the event's resource (eventDetails.Resource).
  - @EventType – This returns the event's event type (eventDetails.EventType).
  - @SenderMetric – This returns the details of the metric which sent the event reusability event being handled (eventDetails.Sender).
- Operator – This field should be filled with the operator with which to compare arguments 1 and 2. This field may be one of the following values:
  - <
  - >
  - <=
  - >=
  - <> (not equal)
  - Eval

  When any of the normal comparators are used, the system takes the content of the field specified in argument 1, and compares it to the value in argument 2, using the selected comparator.

  If the Eval comparator is specified, the argument1 field is ignored. Instead, the content of argument 2 is taken and evaluated directly as an expression.
- Argument 2 – This field should be filled with the value to be compared to the content of the field specified in Argument 1. For example, if any value greater than 5 in the field result is considered a success, the string result is placed in argument 1, > in the operator field and the value 5 in the argument 2 field.

  When Eval is specified in the operator field, this field must contain an expression. An expression is a term in Visual Basic that can be evaluated as to whether it is true or false. For example, and evaluation term can be used to compare the results field to the value 5, as done above, by specifying the following term (all of it goes in Argument 2):

  ```
  eventDetails("result") > 5
  ```

  In this example, the term being evaluated is a comparison between the content of the result field and the value 5. Of course much more complex terms can be evaluated as well. A more complex example is the following term:

  ```
  Tools.NetTime(eventDetails("startDate"),eventDetails("endDate")) >
  eventDetails("MaxTimeSpan")
  ```

  In this example, the term is a comparison of the result of the net time method in the tools object, based in the values from two fields, compared to the value taken from a third field.

- Inter-row logical link – This field should be filled with the logical relationship between the comparison of the current line and that of the next one. The user can implement multiple comparisons. Each comparison or evaluation term is placed in a separate line in the table. This field defines how the results of the comparison relate to each other. Comparison relationships do not follow normal logical priority rules. Instead, they are always evaluated in the order they appear in the table. Legal values for this field are only and, or. When the table contains only one line, this field is ignored and may be left empty.

Success Calculation Module Interface

The following table lists the parameters.

Name	Description
IgnoreTimeslots	(optional, string [True/False]) - whether to handle events that occur outside of timeslot.
Debug	(optional, string [True/False]) - whether to print debugging comments to the log.
AggregationType	(optional, string [SUCCESSCOUNT,PERCENTOFSUCCESS]) - Whether to calculate the percent of success or the count of successful entries.
SuccessCriteria	(mandatory, table)- How to determine if an event is to be considered a success or a failure
FilteringCriteria	(optional, table) - How to determine if an event is to be handled or filtered out.
EventIdFieldName	(optional, string) - The name of the field to us as an event identifier for repeat event recognition.

Module Dependencies

The following table lists the included modules.

Name	Description
Log Library (LL)	Library for enhanced logging functionality.
Parameters Kernel Library (PKL)	Library responsible for verifying and populating module parameters.
Aggregation Library (AL)	Library used to produce results for common calculations.
Callback Kernel Library (CKL)	Library that enables the callback functionality.
Parameters Overlay - Basic Actions Library (POBAL)	Library that contains specific parameter loading functions and calls the parameters kernel library for the actual loading of the information.

Distinct Entity Library (DEL)	Library responsible for telling the caller whether a specified entity has already been seen.
Filtering Criteria Evaluation Library (FCEL)	Library responsible for filtering events according to user specifications.
Success Criteria Evaluation Library (SCEL)	Library responsible for the decision as to whether the current event is to be considered a success or a failure.

Incident Management Business Logic Modules

The following sections describe the generic business logic modules used in the metrics common to helpdesk services. These rules are applicable to calculate metrics of incident management, problem management, and change management.

Incident management modules calculations are based on a certain incident life cycle. The life cycle of an incident is reflected in the different statuses a single incident receives.

The following table lists the incident statuses, the meaning of each of the incident statuses, and their impact on calculations. The incident life cycle is applied to both problems and changes.

Ticket Status	Description
Open	Generated once in a life cycle of an incident, it is created when the ticket is first registered in the helpdesk system.
Reopen	Can be generated once only after Close status. A reopen event is considered toward calculations as a new incident.
InProcess	Generated once in a life cycle of an incident. Generated when work starts on the ticket and someone is actually investigating and working in order to find a solution.
Pending	Generated when any activity situation starts that is considered to be not included in the total resolution time period such as: waiting for the customer's response\input, waiting for equipment from supplier or any third party dependency. Can be generated multiple times for a single incident. In order to stop pending time for an incident, InProcess event should be received again.
Resolved	Generated once in a life cycle of an incident. Generated when a solution is provided for an incident and was communicated to the customer.
Closed	Generated once in a life cycle of an incident.

If an incident has changed its priority, a Close event should be triggered in the change date and an Open event with the new priority should be triggered.

Incident Management Event Types

The following incident management event types represent the status of the incident in the incident life cycle for each incident that is managed:

- Incident - Open
- Incident - ReOpen
- Incident - In process
- Incident - Pending
- Incident - Resolved
- Incident - Closed

Change Management Event Types

The following change management event types represent the status of the change request in the life cycle for each change that is managed:

- Change - Open
- Change - ReOpen
- Change - In process
- Change - Pending
- Change - Resolved
- Change - Closed

Problem Management Event Types

The following problem management event types represent the status of problem in the life cycle for each problem that is managed:

- Problem - Open
- Problem - ReOpen
- Problem - In process
- Problem - Pending
- Problem - Resolved
- Problem - Closed

Incident/Change/Problem Event Type Structure

All incident events have identical structures, as listed in the following table:

#	Name	Field Type	Field Description	Required for Calculations
1	ID	String	Identifier of the Incident	Y
2	Priority	String	There are three priority levels allowed	N
3	Customer	String	Customer ID	N
4	Service	String	Service ID	N
5	LocationName	String	Location Name for multi-location environment	N

The following Service Domains utilize this event structure:

- Incident Management
- Problem Management
- Change Management

Percent Of Reopened Tickets Module

This module calculates the number of reopened tickets in a calculation period. The percentage is calculated by counting the number of Priority X ReOpen ticket events.

Module Interface

The following table lists the parameters.

Name	Description
WhatToCalculate	Determines the resulting output at the end of the period.
Debug	Determines if in debug mode.

Module Dependencies

The following table lists the included modules.

Name	Description
Log Library (LL)	Library for enhanced logging functionality.
Entity Counter Library (ECL)	Library used to keep track of entity counters.
Parameter Kernel Library (PKL)	Library responsible for verifying and populating module parameters.

Parameters Overlay - Basic Action Library (POBAL)	Library responsible for verifying and populating some generic parameters.
Callback Kernel Library (CKL)	Library that enables the callback functionality.
Quick Metric Filtering (QM)	Filtering module required for Quick Metric functionality.
Quick Metric Mapping (QM)	Mapping module required for Quick Metric functionality.

Module Calculations (WhatToCalculate parameter)

- Percent – Calculates the percentage of tickets reopened for all incidents that were closed during the calculation period.
- Count – Calculates the number of tickets reopened for all incidents that were closed during the calculation period.

Event Registration

- See the list of event types under Incident Management Business Logic Modules (see page 50).

Resolution Time Module

All metrics that calculate a result based on the resolution time of the incidents/problems/changes use this module and use the same calculation method.

Resolution time of an incident is calculated as the elapsed time between when the incident was opened until a final solution was provided for it. The time in which the incident spent in pending status is removed from the resolution time. Also the time out of the time slot is removed from the resolution time.

Calculation Assumptions

- Incident receives a single closed event. Any event after the closed event is considered as an event for a different/new incident.
- In cases where the incident did not have an open event, usually because it started before the effective date of the contract, counting the resolution time starts from the beginning of the contract.
- If the first event on an incident is not an open event, the task is marked as stopped.
- Closed is the last event that can be received on a case.
- Time out of a timeslot or exception time is not counted towards resolution time.
- If an incident has more than one open event, the first one is taken as it's opening time.
- If there is a calculation period in which no tickets were closed, the result for that period is null.
- Incident is considered towards calculation in the period it was closed (not in the period it is resolved).

Module Interface

The following table lists the parameters.

Name	Description
TimeUnit	This is the Threshold time unit. For example, when the threshold is 3 days, the time unit is days.
TargetTime	The resolution time threshold in which each incident is compared to in order to calculate the percentage of incidents resolved within this threshold. For example, when the threshold is 3 days, the target time is days.
WhatToCalculate	Determines the resulting output at the end of the period. For example, when the threshold is 3 days, the target time is days.
Debug	Determines if in debug mode.

Module Dependencies

The following table lists the included modules.

Name	Description
Log Library (LL)	Library for enhanced logging functionality.
Time Counter Library (TCL)	Library used to keep track of entity time counters.
Aggregation Library (AL)	Library used to produce results for common calculations.
Parameter Kernel Library (PKL)	Library responsible for verifying and populating module parameters.
Parameters Overlay - Basic Action Library (POBAL)	Library responsible for verifying and populating some generic parameters.
Callback Kernel Library (CKL)	Library that enables the callback functionality.
Quick Metric Filtering (QM)	Filtering module required for Quick Metric functionality.
Quick Metric Mapping (QM)	Mapping module required for Quick Metric functionality.

Module Calculations (WhatToCalculate parameter)

- Avg - Calculates the average resolution time for all incidents that were closed during the calculation period.
- %WithinThreshold/Percent – Calculates the percentage of incidents with a resolution time within a threshold, for incidents closed during the calculation period.
- CNTWithinThreshold – Calculates the number of incidents with a resolution time within a threshold, for incidents closed during the calculation period.
- Min - Determines the closed incident with the shortest resolution time during the calculation period.
- Max - Determines the closed incident with the longest resolution time during the calculation period.

Event Registration

- See the list of event types under Incident Management Business Logic Modules (see page 50).

Response Time Module

All metrics that calculate a result based on the response time of the incidents use the same calculation method for calculating the response time of each of the incidents.

The response time is calculated as the time passed from the opening of the incident until it was handled. The time in which the incident spent in pending status is removed from the response time. Also the time out of the time slot is removed from the response time.

Module Interface

The following table lists the parameters.

Name	Description
TimeUnit	This is the Threshold time unit. For example, when the threshold is 3 days, the time unit is days.
TargetTime	The resolution time threshold in which each incident is compared to in order to calculate the percentage of incidents resolved within this threshold. For example, when the threshold is 3 days, the time unit is days.
WhatToCalculate	Determines the resulting output at the end of the period. For example, when the threshold is 3 days, the target time is days.
Debug	Determines if in debug mode.

Module Dependencies

The following table lists the included modules.

Name	Description
Log Library (LL)	Library for enhanced logging functionality.
Time Counter Library (TCL)	Library used to keep track of entity time counters.
Aggregation Library (AL)	Library used to produce results for common calculations.
Parameter Kernel Library (PKL)	Library responsible for verifying and populating module parameters.
Parameters Overlay - Basic Action Library (POBAL)	Library responsible for verifying and populating some generic parameters.
Callback Kernel Library (CKL)	Library that enables the callback functionality.
Quick Metric Filtering (QM)	Filtering module required for Quick Metric functionality.

Quick Metric Mapping (QM)	Mapping module required for Quick Metric functionality.

Module Calculations (WhatToCalculate parameter)

- Avg – Calculates the average response time for all incidents that were closed during the calculation period.
- %WithinThreshold – Calculates the percentage of incidents with a response time within a threshold, for incidents closed during the calculation period.
- Min – Determines the closed incident with the shortest response time during the calculation period.
- Max – Determines the closed incident with the longest response time during the calculation period.

Event Registration

- See the list of event types under Incident Management Business Logic Modules (see page 50).

Timeslot Events Sender Module

Since the Resolution Time business logic module does not take into account the timeslots of the customers the events came from, the Timeslot Events Sender business logic module and Resolution By Timeslots Extension library have been created. If your company is multinational and you are handling tickets that you need to resolve within a certain number of days in the customer's timeframe, you need to take the location of the customer who originated the ticket into account. The resolution time must take into account the time zone and timeslots of the specific location, which may be different from the location of the metric that does the actual calculation.

This module sends timeslot enter and timeslot exit events, using event reusability. To implement this module, the user must create a metric that includes it. Each sending metric should be specific to a location, and the name of the location should be part of the name of the metric. The metric must have the timeslot and time zone relevant to the location. It must also populate the single parameter called LocationId. The content of this parameter is the name of the location, as it will appear in the event details of the events relevant for this location.

Module Interface

The following table lists the parameter.

Name	Description
LocationId	The unique name of the location for the metric.

Module Dependencies

The following table lists the included modules.

Name	Description
Parameters Kernel Library (PKL)	Library responsible for verifying and populating module parameters.
Callback Kernel Library (CKL)	Library that enables the callback functionality.

Total Number of Incidents Module

This module counts the number of events received in the calculation period. Any type of events that should be counted can be used with this module. For example, if it is required to count the number of incidents that were opened during the calculation period, then the registration should be for Open events. If the requirement is for counting the number of incidents that were closed during the period, the registration should be for closed events. The formula uses only a field called ID from the event. Therefore, any event with this value field can be used to be counted.

Calculation Assumptions

- All incidents that were received during the calculation period are counted
- Incidents that were received out of the timeslot period are counted as well
- Incident is counted once according to the incident ID. If several events from the same event type are received with a single incident ID, it is counted as a single occurrence

Module Interface

The following table lists the parameters.

Name	Description
WhatToCalculate	Determines the resulting output at the end of the period.
Debug	Determines if in debug mode.

Module Dependencies

The following table lists the included modules.

Name	Description
Log Library (LL)	Library for enhanced logging functionality.
Time Counter Library (TCL)	Library used to keep track of entity time counters.
Aggregation Library (AL)	Library used to produce results for common calculations.

Parameter Kernel Library (PKL)	Library responsible for verifying and populating module parameters.
Parameters Overlay - Basic Action Library (POBAL)	Library responsible for verifying and populating some generic parameters.
Callback Kernel Library (CKL)	Library that enables the callback functionality.
Quick Metric Filtering (QM)	Filtering module required for Quick Metric functionality.
Quick Metric Mapping (QM)	Mapping module required for Quick Metric functionality.

Module Calculations (WhatToCalculate parameter)

COUNT - Performs a simple count of entities dependent on the selected event types.

Event Registration

The event type registration is according to the events that should be counted with a restriction that the event type holds a value field of ID.

For example, in order to calculate any priority opened events, the registration is for Incident - Open event type.

Within X Business Days Module

This module contains the functionality needed to handle events, calculate results and manage periods for calculating the number of days it took to resolve all the tickets for a specified period. It supplies the functionality for both Within X Days calculations as well as Same Day/Next Day functionality.

The resolution time of each ticket is calculated as the time passed between when the ticket was opened until a final solution was provided for it. The time that the ticket spent in pending status is not counted. The time out of timeslot is also not counted.

Module Interface

The following table lists the parameters.

Name	Description
WhatToCalculate	Determines the method of calculation.
Debug	Determines if in debug mode.
DaysThreshold	This is the resolution threshold in days.

CountWholeDays	Determines whether the calculation takes the open time into account, or whether any second within timeslot in a day counts as an entire day.

Module Dependencies

The following table lists the included modules.

Name	Description
Log Library (LL)	Library for enhanced logging functionality.
Business Day Counting Library (BDCL)	Library that counts business days in ticket lifecycles.
Parameters Overlay - Basic Action Library (POBAL)	Library that contains generic parameters used by multiple libraries such as the logging parameter.
Parameters Overlay - Business Days Library (POBDL)	Library that contains parameters specific to the business day counting functionality.
Callback Kernel Library (CKL)	Library that enables the callback functionality.
Quick Metric Filtering (QM)	Filtering module required for Quick Metric functionality.
Quick Metric Mapping (QM)	Mapping module required for Quick Metric functionality.

Event Registration

- See the list of event types under Incident Management Business Logic Modules (see page 50).

Service Desk Management Business Logic Modules

The following sections describe the business logic modules used in the metrics common to Call Center services.

Service Desk Management Event Types

The Call Center service desk management event types provide call information for a particular center or operator.

Call Center Event Type Structure

The following table shows the structure of the Call event type:

#	Name	Description
1	EventTimestamp	Timestamp.

#	Name	Description
2	CallCenter	Name of call center/operator (could be locations or different providers or locations).
3	TotalCalls	Total number of calls offered in the call center at that time stamp.
4	Answered	Total number of calls connected to an operator (agent) who answered.
5	Abandoned	Total number of calls that were not attended. Total number of calls which abandoned before being served.
6	AnsweredWithinX Seconds_Count1	A number indicating the number of calls answered within the time limit described by the corresponding bucket (limit 1).
7	AnsweredWithinX Seconds_Limit1	A number indicating the number of seconds during which calls are answered (limit 1).
8	AnsweredWithinX Seconds_Count2	A number indicating the number of calls answered within the time limit described by the corresponding bucket (limit 2).
9	AnsweredWithinX Seconds_Limit2	A number indicating the number of seconds during which calls are answered (limit 2).
10	AnsweredWithinX Seconds_Count3	A number indicating the number of calls answered within the time limit described by the corresponding bucket (limit 3).
11	AnsweredWithinX Seconds_Limit3	A number indicating the number of seconds during which calls are answered (limit 3).
12	CallsAfter60s	Number of calls answered by an agent. It is a function of the time spent since the call was processed by the routing mechanism until the agent answered (equivalent to: waiting time + ringing time).
13	AbandBefore10s	Total number of calls abandoned before being served.
14	AbandBefore30s	Total number of calls abandoned before being served.
15	AbandBefore60s	Total number of calls abandoned before being served.
16	AbandAfter60s	Total number of calls abandoned before being served.
17	AvgDurCallProc	Average duration of call processing of call served. Call processing is calculated between the agent answering the call and the end of wrap-up time.

#	Name	Description
18	AvgWaitAnsCalls	Average time spent in queue by calls which were connected to a resource. The abandoned calls and calls redirected due to waiting timer overflow are not taken into account.
19	EscalatedCalls	% of Calls that an incident was opened for them in the Incident management system.

Call Center Module

The data arrives from the event source already aggregated in the form of summaries defined by the field names. The module allows to turn the summaries into percentages by dividing the specific requested summary (e.g. the number of calls dropped) by the total number of calls.

All the formulas that calculate the percentage of calls perform the same type of calculation. However, they use the appropriate summary for the calculation being performed by selecting the appropriate field from the event.

Events that have a timestamp that is out of the timeslot are ignored.

For calculating the Calls Answered Within X Seconds functionality, the module uses a dynamic system based on three pairs of fields: a field that contains the size of the bucket or seconds count, and a field that contains the result or summary for that count. This way, the number of seconds is not part of the field name and any seconds count can be handled. To use this dynamic system, two parameters are used, one to turn this functionality on and another that tells the metric which bucket to use. The metric then looks for a bucket of the specified size and uses the call count in the corresponding field (e.g. if the bucket was found in the AnsweredWithinXSeconds_Limit2 field, then the AnsweredWithinXSeconds_Count2 field is used for the value). The functionality assumes that there is always one and only one field with the specified bucket size. If it is not found, an error occurs.

The event may have all of the buckets or only some of them occupied. Each metric requires a different bucket size, and this size must be present in the event. This means that up to three different bucket sizes may be accommodated at the same time in each event.

Module Interface

The following table lists the parameters.

Name	Description
CallType	Used by the formula to select the relevant field in the event type for the accumulations of calls. The available options are: ANSWERED, ABANDONED, ESCALATED, OVERALL, CALLSAFTER60S, ABANDBEFORE10S, ABANDBEFORE30S, ABANDBEFORE60S, ABANDAFTER60S, PERCENTESCCALLS
WhatToCalculate	Determines which formula to use for calculation. The available options are: Percent, Count, WaitingTime, ProcessingTime. See below for details.
Debug	Determines if in debug mode and if debug messages need to be sent to the log. The available options are: True or False.

RespondBucketLimit	Used by the formula to select the relevant field in the event type for the Bucket Limit. The value is numeric and should be a count of seconds.
AnsweredWithinTarget	Tells the system whether to use the AnsweredWithin logic or not. If it is set to True, the CallType parameter is ignored. The available options are: True or False.
TimeUnit	Tells the system what time unit to calculate with.

Module Dependencies

The following table lists the included modules.

Name	Description
Log Library (LL)	Library for enhanced logging functionality.
Parameter Overlay - Call Center Library (POCCL)	Library that contains parameters specific to the call center functionality.
Parameter Overlay - Basic Action Library (POBAL)	Library that contains generic parameters used by multiple libraries such as the logging parameter.
Callback Kernel Library (CKL)	Library that enables the callback functionality.
Quick Metric Filtering (QM)	Filtering module required for Quick Metric functionality
Quick Metric Mapping (QM)	Mapping module required for Quick Metric functionality.

Module Calculations (WhatToCalculate parameter)

- Percent - Calculates the percentage of calls from a certain type out of the total number of calls for the calculation period. For example, "% of Abandoned Calls" would be calculated as ∑(Number of abandoned Calls)/∑(Number of Calls)*100
- Count - Calculates the total number of calls for the calculation period.
- WaitingTime - Calculates the average on waiting time, calculation formula: ∑(Events waiting time)/∑((Number of Events).
- ProcessingTime - Calculates the average on processing time, calculation formula: ∑(Events processing time)/∑((Number of Events).

Event Registration

All metrics that use the Call Center module are registered with the Call Center event type.

Library Modules

The predefined content package includes also a set of library modules that provide a set of utilities for the use of the full business logic modules.

The following is the list of libraries:

Library Name	Description
Aggregation Library	Library used to produce results for common calculations.
Business Day Counting Library	Library that does the actual manipulation of timestamps to figure out when a business day starts and ends. For more information, see Business Day Counting Library (see page 68).
Callback kernel Library	Library that enables the callback functionality whereby the function is called if it is implemented. If it is not implemented, it is not called and does not cause errors.
Criteria Evaluation Kernel Library	Library that does the actual criteria evaluation, for either the filtering or the success libraries.
Distinct Entity Library	Library responsible for telling the caller whether a specified entity has already been seen.
Entity Counter Library	Library used to keep track of entity counters.
Filtering Criteria Evaluation Library	Library responsible for filtering events according to user specifications. The event handler calls this functionality for each event to decide if the event is to be handled or not.
Log Library	Library for enhanced logging functionality.
Parameters Kernel Library	Library that contains the core handling of parameter reading. The function within is given all the information needed to load a parameter, including default value, allowed values and whether the parameter is a table. It supplies robust access to metric parameter information.
Parameter Library	Library responsible for verifying and populating module parameters.
Parameters Overlay - Basic Action Library	Library that contains specific parameter loading functions and calls the parameters kernel library for the actual loading of the information.
Resolution By Timeslots Extension Library	Adds the functionality of the location specific timeslot to the resolution module. For more information, see Resolution By Timeslots Extension Library.

Success Criteria Evaluation Library	Library responsible for the decision as to whether the current event is to be considered a success or a failure. It calls the Criteria Evaluation Library. The event handler calls this function with every event that has passed filtering, to determine if the event is to be considered successful.
Time Counter Library	Library used to keep track of entity time counters.
Within X Business Days By Timeslots Extension Library	Library that extends the Within X Business Days module, giving it the ability to handle multiple source timeslots. For more information, see Within X Business Days By Timeslots Extension Library (see page 68).

Resolution By Timeslots Extension Library

This module adds the functionality of the location specific timeslot to the resolution module. To enable this functionality the user must include the extension library, as well as the resolution module in the metric. The extended library has two new parameters that need to be populated.

- RegistrationList - This is a table parameter. The table has one column called LocationId. This parameter needs to contain a list of the names of all the metrics that send the timeslot enter and timeslot exit events for the locations relevant to the metric (as created from the Timeslot Events Sender module). The list must contain the names of the metrics that send events and not the names of the location. For example, if the name of the metrics has the prefix Location – , then the list should contain values such as Location – France and Location – Spain. This list is used to register and receive the timeslot enter and timeslot exit events for the specified locations.
- EventLocationId - This parameter tells the system how to find the location information within the details of each event it handles. This can be done in one of three ways:
  - The name of the event type field containing the location information. In this case the content of the parameter is the name of the corresponding field.
  - Event specific information not contained in the event details. To access this information, the first character must be an @:
    - @resource – The name of the resource that sent the event.
    - @eventtype – The name of the event type that sent the event.
    - @sendermetric – The name of the metric that sent the event.
  - The user can supply any expression that can be evaluated in VBS. This can be used to access more than one field or to access the custom attribute information of the sending resource. To use this functionality, the first character must be an '=' (equal sign). For example, to extract the custom attribute of the resource that the event was sent on, we would use:
    `=eventDetails.CustomAttribute("MyAttribute")`

    This would use the value of the myAttribute custom attribute of the resource that sent the event as the location identifier of the event.

Within X Business Days By Timeslots Extension Library

This module extends the Within X Business Days module, giving it the ability to handle multiple source timeslots. To enable this functionality the user must include the extension library, as well as the resolution module in the metric. The extended library has two new parameters that need to be populated.

- RegistrationList - This is a table parameter. The table has one column called LocationId. This parameter needs to contain a list of the names of all the metrics that send the timeslot enter and timeslot exit events for the locations relevant to the metric (as created from the Timeslot Events Sender module). The list must contain the names of the metrics that send events and not the names of the location. For example, if the name of the metrics has the prefix Location – , then the list should contain values such as Location – France and Location – Spain. This list is used to register and receive the timeslot enter and timeslot exit events for the specified locations.
- EventLocationId - This parameter tells the system how to find the location information within the details of each event it handles. This can be done in one of three ways:
  - The name of the event type field containing the location information. In this case the content of the parameter is the name of the corresponding field.
  - Event specific information not contained in the event details. To access this information, the first character must be an @:
    - @resource – The name of the resource that sent the event.
    - @eventtype – The name of the event type that sent the event.
    - @sendermetric – The name of the metric that sent the event.
  - The user can supply any expression that can be evaluated in VBS. This can be used to access more than one field or to access the custom attribute information of the sending resource. To use this functionality, the first character must be an '=' (equal sign). For example, to extract the custom attribute of the resource that the event was sent on, we would use:
    `=eventDetails.CustomAttribute("MyAttribute")`

    This would use the value of the myAttribute custom attribute of the resource that sent the event as the location identifier of the event.

Business Day Counting Library

This module does the actual calculation of timestamps to figure out when a business day starts and ends. Two different logics are supplied:

- Within X days – This takes the opening time of a ticket into account when counting days. For example, if the ticket is opened at 14:00 and closed at 13:00 the next business day, this is counted as a single day. If the ticket were to close at 15:00 on the next business day, that would count as two days.
- Same day, Next day – This ignores ticket open time. Any second in the day that has an active timeslot (and is thus considered a business day) makes that day count as a whole day. For example, a ticket that was opened on the last second of the timeslot of day 1 makes that day count as a whole day. When a ticket opened after the timeslot of day 1, that day is not counted. If the ticket is marked same day, the ticket must close before the end of that day. Note that which day is determined by the first 'within timeslot' second after the ticket is opened. A ticket that is opened after today's timeslot exit is only considered as active the next day, so a same day ticket would have to be closed tomorrow end of day, if it was opened after today's timeslot exit A ticket that is closed after midnight of the opening day is counted as two days.

Both business days logics are based on the same modules and differ only in their parameter settings.

- Single timeslot – This functionality is based the single timeslot/time-zone of the calculating metric itself. You need to implement the following parameters:
  - WhatToCalculate – This parameter allows you to decide if you want to get the number of tickets within the threshold, or whether you want to get a percentage of tickets within the threshold from the total tickets (PERCENT/CNTWITHINTHRESHOLD).
  - CountWholeDays – This parameter determines if you're after the Same/Next day functionality (TRUE), or the Within X business days functionality (FALSE).
  - DaysThreshold – This tells the system how many days are to be considered a success. For same day functionality, the first day is counted as well. Thus same day = 1, next day = 2. For Within X days, the number is more intuitive, so Within 3 days, the value needs to be 3.

    Note: The Same/Next day functionality is not actually limited to these two values, but can be implemented for example by the end of the third day.
- Multiple timeslots – In addition to the explanation above about single timeslot functionality, the new functionality supports multiple timeslot events. This support comes through an additional extension library called Within X Business Days By Timeslots Extension Library. This library is connected to the metric itself. It adds the timeslots functionality and requires an extra pair of parameters:
  - RegistrationList –A table parameter with a single field called LocationId. This table is filled with the list of location timeslot events sending metrics that have to be located in the same contract.

- EventLocationId – The definition of what the system needs to do to determine an incoming event's location.

Also, to enable this functionality, a new event type was added, to allow for the handling of daylight savings time changes across ticket lifecycles.

This event type is called LocationDstChangeNotification and has the following specifications:

- LocationId as string.
- DistanceFromUtc as integer.

Module Dependencies

The following table lists the included modules.

Name	Description
Log Library (LL)	Library for enhanced logging functionality.
Parameters Kernel Library (PKL)	Library responsible for verifying and populating module parameters.

Chapter 6: Introduction

This document contains the release notes for the Predefined Content. The following chapters describe the release in detail and provide other information that supplements the main documentation.

This release improves quality, reliability, performance and adds new functionality.

Chapter 7: Installation and Upgrade

For detailed installation instructions, see the CA Business Service Insight Predefined Content Guide document.

Chapter 8: Miscellaneous Enhancements

No new enhancement were made to the predefined content in this release.

Chapter 9: Known Issues

No issues were found in the predefined content in this release.

Chapter 10: Introduction

This document contains the release notes for the Predefined Content. The following chapters describe the release in detail and provide other information that supplements the main documentation.

This release improves quality, reliability, performance and adds new functionality.

Chapter 11: Installation and Upgrade

For detailed installation instructions, see the CA Business Service Insight Predefined Content Guide document.

Chapter 12: Miscellaneous Enhancements

No new enhancement were made to the predefined content in this release.

Chapter 13: Known Issues

No issues were found in the predefined content in this release.

Index

W