Oracle® Universal Content Management
Content Tracker Administration Guide
10*g* Release 3 (10.1.3.3.0)

March 2007

Content Tracker Administration Guide, 10*g* Release 3 (10.1.3.3.0)

Contributing Authors: Deanna Burke

Contributors: Evan Suits

Table of Contents

C

Chapter 1: Introduction

Chapter 2: Operational Overview

Chapter 3: Using Content Tracker

Chapter 4: Report Generation

Chapter 5: Service Call Configuration

Appendix A: Configuring and Customizing Content Tracker

Appendix B: Troubleshooting

Appendix C: Third Party Licenses

Index

Chapter
1

INTRODUCTION

OVERVIEW

This section covers the following topics:

- ❖ About This Guide (page 1-1)
- ❖ What's New (page 1-5)
- ❖ Content Tracker Terminology (page 1-7)
- ❖ General Limitations (page 1-8)
- ❖ General Considerations (page 1-9)
- ❖ Conventions (page 1-9)

ABOUT THIS GUIDE

This section covers the following topics:

- ❖ Content Tracker Operational Summary (page 1-2)
- ❖ Audience (page 1-5)

Content Tracker Operational Summary

Content Tracker monitors activity on your Content Server instance, and records selected details of those activities. It then generates reports that may help you understand the ways in which your system is being used. This section covers the following topics:

- About the Components (page 1-2)
- About the Data Flow (page 1-3)

Note: This section provides a very brief overview about Content Tracker and Content Tracker Reports functionality. This summary provides a basic background about the components and should help prepare you for the more detailed information provided in Chapter 2 (*Operational Overview)*.

About the Components

Content Tracker and Content Tracker Reports are separate modules but work together to provide information about system usage. The information provided enables you to determine which content items are most frequently accessed and what content is most valuable to users or specific groups.

Understanding the consumption patterns of your organization’s content is essential to successful content management. This enables you to provide more appropriate, user-centric information more effectively. This section summarizes both components:

- Content Tracker Overview (page 1-2)
- Content Tracker Reports Overview (page 1-3)

Content Tracker Overview

Content Tracker monitors your system and records information about various activities. This information is collected from various sources, then merged and written to a set of tables in your Content Server database. You can customize Content Tracker to change or expand the types of information it collects. Content Tracker monitors activity from:

- **Content item accesses:**

  Content Tracker gathers information about content item usage. The data is obtained from Web filter log files, the Content Server database, and other external applications such as portals and Web sites. Content item access data includes dates, times, content IDs, current metadata, user names, and profile information about users.

- **Content Server services:**

  Content Tracker tracks all services that return content, as well as services that handle search requests. And, with a simple configuration change, Content Tracker can monitor literally any Content Server service, even custom services.

Content Tracker Reports Overview

After Content Tracker extracts data and populates applicable database repository tables, the information is available for report generation. Content Tracker Reports enables you to:

- **Generate reports:**

  Content Tracker Reports queries the tables created by Content Tracker and generates summary reports of various kinds of activities and the usage history of particular content items. The reports help you analyze specific groups of content or users based on metadata, file extensions, or user profiles. You can use the pre-defined reports that are provided, customize them to suit your installation, or use a compatible third-party reporting package.

- **Optimize content management practices:**

  You can also use the reported data for content retention management. That is, depending on the access frequency of particular content items during specific time intervals, you may decide to archive or delete some of the items. Similarly, applications can use the data to provide portlets with the top content for particular types of users.

About the Data Flow

Combined, the Content Tracker and Content Tracker Reports components perform three primary information processing functions:

- Data Recording (page 1-4)
- Data Reduction Process (page 1-5)
- Data Reporting (page 1-5)

Data Recording

Content Tracker records data from the following sources:

- **Web Server Filter Plugin:**

  When content is requested via a static URL, the web server filter plugin records certain details of the request and saves the information in one or more event log files. Event log files are organized according to the date on which the information was collected. The event log files are eventually used as input by the Content Tracker data reduction process.

- **Service Handler Filter:**

  Content Tracker has a list of services that it monitors. When one of these services is called, details of the service are copied and saved in the SctAccessLog table. You can change which services are monitored, and which details are recorded.

- **Content Tracker Logging Service:**

  Content Tracker supports a general purpose logging service that is a single-service call that can be used to log an event. It can be called directly via a URL, as an action in a service script, or from IdocScript.

- **Content Server Database Tables:**

  The Content Tracker data reduction process queries selected Content Server database tables. This is done primarily to obtain information about the names and accounts of users who were active during the reporting period.

- **Application API:**

  Content Tracker provides an interface by which other components and applications can be registered for tracking, and can have information about their activities recorded. For example, this interface allows cooperating applications, such as Site Studio, to log event information in real time.

Note: The Application API is included in the SctApplicationFilter.hda file. This interface is designed as a code to code call which does not involve a Content Server service. The Application API is not meant for general use. If you are building an application and are interested in using this interface, you should contact Consulting Services.

Data Reduction Process

The data reduction process gathers and merges the data obtained from the four data recording sources. Until this reduction process has finished, the data in the Content Tracker tables is incomplete. You will usually run the reduction once for each day's worth of data gathered. The reduction may be run manually, or may be scheduled to run automatically, usually during an off-peak period when the system load is light.

Data Reporting

Content Tracker Reports provides a set of reports that answer commonly asked questions about Content Server activity and usage. For example, you can determine which managed objects have been accessed most frequently, which searches are used most often, and which users have been most active. These reports are available directly, via the Content Tracker Report Generator main page, and indirectly as an action on the Content Information page. The reports, the underlying queries, and the output formatting are available for customization.

Audience

This administration guide is intended for system administrators who need to install and configure the Content Tracker and Content Tracker Reports components. System administrators will also use this guide to manage data acquisition, generate usage reports, and enhance Content Server functionality for end users. This guide assumes that the product has been installed correctly and that you are familiar with Content Server products and the architecture of the Content Server.

WHAT'S NEW

This section describes the new features and enhancements in the 10gR3 versions of Content Tracker and Content Tracker Reports components.

❖ **Custom Metadata Fields For Search Relevance Information:**

The snapshot function enables you to link activity metrics to custom metadata fields that can be populated with content item usage information. The data gathered by the activity metrics includes the date of the most recent access and the number of accesses in two distinct time periods. You can use the collected data in various ways. For example, you might want to order search results according to the most popular or most viewed content in the last week. For more information, see the Snapshot Tab (page 3-11).

- **Count Checkin Operations as Access Activity:**

  When a content item is checked in, the Last Access field in Content Server's DocMeta database table is initially empty. After a data reduction has been performed, the Last Access field is updated with the most recent date and time of access, or with the date and time of checkin if no accesses have occurred. An optional automatic load function allows you to update the last access activity metric for existing content to ensure that the Last Access field for content items is appropriately timestamped. For more information, see the Autoload check box (page 3-14) on the Snapshot Tab (page 3-11).

- **Greater Flexibility With Extended Service Logging:**

  The extended services logging function enables you to map and log data from most Content Server services to the combined output database table (SctAccessLog). This means that in addition to logging calls for services, you can now track specific data values that are relevant to those services. For more information, see the Services Tab (page 3-16) and the Extended Service Call Tracking Function (page 5-3).

- **Audit Trails for Failed User Authentications/Authorizations:**

  Content Tracker Reports now provides an auditing feature that enables you to monitor unsuccessful attempts to access the system or permission-protected content. Two reports are available that can help you analyze attempted security breaches that include failed user logons and unsuccessful attempts to access secure content items. This information is vital to ensure system and content security. For more information, see User Authentication/Authorization Monitoring and Auditing (page 4-21).

- **Filtered Report Results Based on User's Role/Account Permissions:**

  With this version of Content Tracker Reports, you can choose to generate reports in secure or non-secure mode. That is, you now have the option to filter the search results in reports based on the requesting user's role and account permissions. The same criteria that are used to limit Content Server search results can be used to restrict the content items that are included in the generated reports. For more information, see Security Checks and Query Results (page 4-23).

- **Include Both External and Internal User Access Data in Reports:**

  With this version of Content Tracker and Content Tracker Reports, the role and account information for externally authenticated users is recorded in the Content Server's UserSecurityAttributes database table. As a result, two pre-defined reports (Top Content Items by User Role and Users by User Role) include the content item access activity by external users. For more information, see External Users and Content Item Tracking (page A-10).

- **DataBinder Dump Facility to Debug Field Maps for Extended Service Logging:**

  If you plan to use the extended service logging function, the DataBinder dump facility is available to help you design and debug the field maps. This facility allows Content Tracker to write the DataBinder objects to a dump file and enables you to see what data is available when service events are recorded. For more information, see DataBinder Dump Facility (page B-4).

- **Web Site Access Activity (Site Studio):**

  With this version, there are pre-defined reports that you can use to generate and analyze Site Studio-specific data. One report summarizes content accesses from web site pages. Another report summarizes web site page visits. For more information, see Site Studio Web Site Activity Reporting (page 4-21).

Content Tracker Terminology

You should be familiar with the following terminology when using Content Tracker and Content Tracker Reports:

- **Data collection:** Gathering content access information programmatically and writing the information to event log files.
- **Data reduction:** Processing the information from data collection and merging it into a database table.
- **Data Engine Control Center:** The applet interface that provides access to the user-controlled functions of the Data Engine. The Data Engine Control Center is used to enable, schedule, and monitor data collection. It is also used to collect and manage data about user activity and service calls.
- **Collection:** Tab used to enable data collection.
- **Reduction:** Tab used to stop and start data reduction (that is, merging data into database tables).
- **Schedule:** Tab used to enable automatic data reduction.
- **Snapshot:** Tab used to enable activity metrics. Also, the word snapshot is used to denote historical information that is instantaneous in nature and specifies what user accessed a particular content item at a particular moment.

- **Services:** Tab used to add, configure, and edit Content Server service calls to be logged. It is also used to define the specific event details that are to be logged for a given service.
- **Service definitions:** The ResultSet structure in the service call configuration file (SctServiceFilter.hda) that contains entries to define each Content Server service call to be logged. The service definition ResultSet is named ServiceExtraInfo.
- **ServiceExtraInfo ResultSet:** See Service definition.
- **Service entry:** The entry in the service definition ResultSet (ServiceExtraInfo) that defines a Content Server service call to be logged. The ServiceExtraInfo ResultSet contains one service entry for each service to be logged.
- **Field map:** A secondary ResultSet in the service call configuration file (SctServiceFilter.hda) that defines the service call data and the specific location where the data is to be logged.
- **Top Content Items:** Most frequently accessed content items in the system.
- **Content Dashboard:** An HTML page that provides overview information about the access of a specific content item.

GENERAL LIMITATIONS

Content Tracker is supported on most hardware and networked configurations. There are, however, certain hardware and software combinations that require special consideration. Some known limitations include:

- An architecture using multi-node clusters installed in a single server are not supported for tracking and report generation. See Tracking Limitations in Single-Box Clusters (page 2-19).
- In some circumstances, exact access counts cannot be guaranteed for content that is requested through static URLs or by WebDAV. See Tracking Limitations with Static URLs and WebDav (page 2-19).
- When either Oracle or DB2 is used as the Content Server database, metadata values are case sensitive. See Oracle and DB2 Case Sensitivity (page 4-3).
- When Access Control Lists (ACLs) are enabled on a Content Server instance, the Content Tracker Reports secure mode does not work. See Access Control Lists and Content Tracker Reports Secure Mode (page 4-4).

- When Oracle is used as the Content Server database, some additional, customized file configurations are necessary if you want to use aliases to display column names. See Custom Report Queries and Oracle (page 4-16).

General Considerations

The following general considerations are applicable for the current version of the Content Tracker and Content Tracker Reports components:

- **Hanging Browser:**

  If Content Server happens to terminate while the Data Engine Control Center is running, the browser can also hang. To easily resolve this issue, close the hung browser window.

- **Local Time vs. GMT:**

  A new configuration parameter enables you to use local time instead of Greenwich Mean Time (GMT) to record user access times:

  - **SctUseGMT=true** configures Content Tracker to use GMT.
  - **SctUseGMT=false** configures Content Tracker to use local time. This is the default setting.

  If you are performing a new installation of Content Tracker and use the default setting for SctUseGMT, user accesses will be recorded in local time. If you are upgrading from an earlier version of Content Tracker and use the default setting for SctUseGMT, there will be a one-time retreat (or advance, depending on your location) in access times. Also, to accommodate the biannual daylight savings time changes, there will be discontinuities in recorded user access times (contingent on whether you use local time and your location).

Conventions

- The notation <*install_dir*>/ is used to refer to the location on your system where the product is installed.
- File names and file paths within text are indicated by the following convention: *config.cfg* file in the <*install_dir*>/config/ directory.

- Forward slashes / are used to separate the directory levels in a path name. This is true when referring to files on a local Windows file system or on a UNIX system. A forward slash will always appear after the end of a directory name.

The following symbols are used throughout this guide this help system:

Symbols	Description
💡	This is a **note**. It is used to bring special attention to information.
⚙	This is a **technical tip**. It is used to identify information that can be used to make your tasks easier.
❗	This is an **important** notice. It is used to identify a required step or required information.
✖	This is a **caution**. It is used to identify information that might cause loss of data or serious system problems.

Chapter

2

OPERATIONAL OVERVIEW

OVERVIEW

This chapter covers the following topics:

ABOUT CONTENT TRACKER

Content Tracker captures information regarding the consumption patterns of content items. Activity information is collected daily for both external and internal users. This includes tracking content accessed from Content Server by end users directly or via external applications such as portals and Web sites.

The information is gathered from Content Server Web filter log files, the Content Server database, and other external applications such as portals and Web sites. The specific data includes dates, times, content IDs, current metadata, user names, profile information about users, and Content Server service calls.

Once the data is collected, Content Tracker combines, analyzes and synthesizes the event information and loads the summarized activity into database tables. After reduction, this data becomes available for reporting purposes. You can use the Content Tracker Report Generator main page to produce reports that identify content usage trends. This will help you understand how your system is being used resulting in more successful content management.

Content Tracker Data Flow

Content Tracker collects event information such as dynamic and static content accesses and service calls. Several mechanisms are used to collect the data.

- Web Server Filter Plugin (page 2-5):

  Collects data values from static URLs and logs them in raw data files.

- Service Handler Filter (page 2-5):

  Examines Content Server service requests and writes certain details from them directly to the SctAccessLog table in real time. Only services listed in the SctServiceFilter.hda file are logged.

- Content Tracker Logging Service (page 2-6):

  Used to log event information generated by a suitably configured application.

This section covers the following topics:

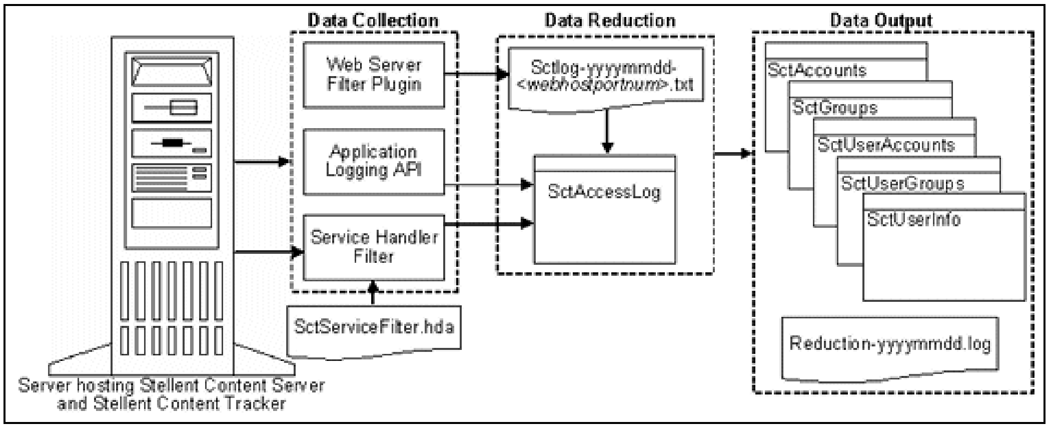
- Data Reduction Process (page 2-2)
- Data Reduction Process with Activity Metrics (page 2-3)

Data Reduction Process

During the data reduction process, the static URL information is extracted from the raw data files (see Content Tracker Event Logs (page 2-7) and combined with the service information already stored in the SctAccessLog table (see Combined Output Table (page 2-8).

This reduction process:

- Combines access information for static URL content access with service details.
- Summarizes information about user accounts that were active during the reporting period. This information is rolled up and written to the Content Tracker's user metadata database tables. See Metadata Capture (page 2-13).



Data Reduction Process with Activity Metrics

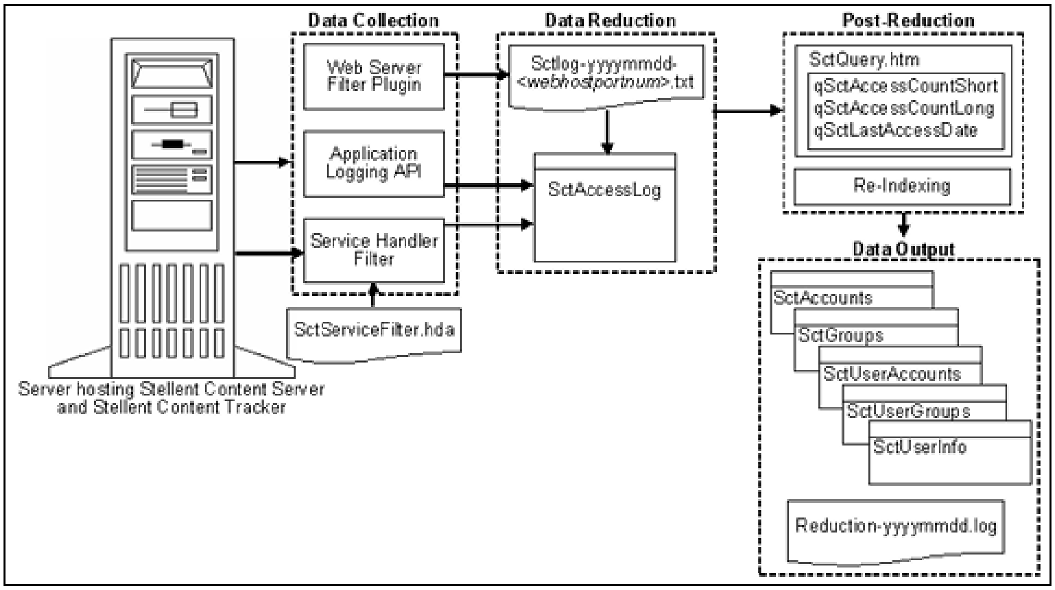
Content Tracker provides the option to selectively generate search relevancy data and store it in custom metadata fields. The snapshot function enables you to choose which activity metrics to activate. The logged data provides content item usage information that indicates the popularity of content items.

If you activate the snapshot function and activity metrics, the values in the custom metadata fields are updated following the reduction processing phase. When users access content items, the values of the applicable search relevance metadata fields change accordingly. Then, during the subsequent post-reduction step, Content Tracker uses applicable SQL queries to determine which content items were accessed during the reporting period.

Content Tracker updates the applicable database table metadata fields with the new values and initiates a re-indexing cycle. However, only the content items whose access count metadata values have changed are re-indexed. For more information about the snapshot function, the user interface screen, and activating the activity metrics, see Snapshot Tab (page 3-11). For more information about the activity metrics' SQL queries and how to customize them, see Activity Metrics SQL Queries (page A-8).

The post-reduction processing step is necessary to:

- Process and tabulate the activity metrics for each affected content item and load the data into the assigned custom metadata fields.
- Initiate a re-indexing cycle on the content items with changed activity metrics values. This ensures that the data is part of the search index and, consequently, accessible for selecting and ordering search results.



DATA COLLECTION

Content Tracker data collection includes collecting information from static URL references along with Content Server service call events. Both types of data are recorded in a combined output table (SctAccessLog). However, service calls are inserted into the log in real time whereas the static URL information must first undergo the reduction process (either manual or scheduled).

The mechanisms used to collect Content Tracker data include:

- Web Server Filter Plugin (page 2-5)
- Service Handler Filter (page 2-5)
- Content Tracker Logging Service (page 2-6)

Service Handler Filter

The Content Server service handler filter is the primary Content Tracker data collection mechanism. This filter makes it possible for Content Tracker to obtain information about dynamic content requests that come through the web server, and also about other types of Content Server activity, such as calls from applications. The service request details are obtained from the DataBinder that accompanies the service call, and the information is stored in the combined output table (SctAccessLog) in real time. For more information about the SctAccessLog table, see Combined Output Table (page 2-8).

There is a user-modifiable configuration file that is used to determine which Content Server service calls are logged. This file (SctServiceFilter.hda) uses a ResultSet structure that includes one service definition entry for each service to be logged. If you are using the extended service logging function, the SctServiceFilter.hda file also contains field maps that correspond to various service definition entries—see Services Tab (page 3-16). For more detailed information about configuring service calls using the service handler filter, see Chapter 5 (*Service Call Configuration)*.

Note: The ResultSet included in the SctServiceFilter.hda file is named ServiceExtraInfo. This ResultSet contains one or more service entries that define the services to be logged. To support the extended service logging function, additional ResultSets are used. These are called field map ResultSets. Each service that will have additional data values tracked must have a corresponding field map ResultSet in the SctServiceFilter.hda file. Field map ResultSets define the data fields, locations, and database destination columns for the related service.

Note: There should be no duplication or conflicts between services logged via the service handler filter and those logged via the Content Tracker logging service. If a service is named in the Content Tracker service handler filter file then such services are automatically logged so there is no need for the Content Tracker logging service to do this. However, Content Tracker makes no attempt to prevent such duplication.

Web Server Filter Plugin

Managed content that is retrieved via a static URL does not usually invoke a Content Server service. Therefore, the Content Tracker web server filter plugin collects the access event details (static URL references) and records them in raw event logs (sctlog files). The information in these files requires an explicit reduction (either interactive or scheduled) before it is included in the combined output table (SctAccessLog) along with the service call data.

For more information about the sctlog files, see Content Tracker Event Logs (page 2-7). For more information about the SctAccessLog table, see Combined Output Table (page 2-8).

Content Tracker Logging Service

The Content Tracker logging service is a single-service call that may be called directly via a URL or as an action in a service script. It may also be called from IdocScript using the executeService() function. The calling application is responsible for setting any and all fields in the accompanying service DataBinder that need to be recorded, including the descriptive fields listed in the Content Tracker service filter configuration file (SctServiceFilter.hda). For more detailed information about configuring service calls using the Content Tracker logging service, see Chapter 5 (*Service Call Configuration)*.

Note: There should be no duplication or conflicts between services logged via the service handler filter and those logged via the Content Tracker logging service. If a service is named in the Content Tracker service handler filter file then such services are automatically logged so there is no need for the Content Tracker logging service to do it. However, Content Tracker will make no attempt to prevent such duplication.

Data Reduction

During Content Tracker data reduction, the static URL information captured by the web server filter plugin is merged and written into the output table (SctAccessLog) alongside the service call data. Additionally, at the time of the reduction, Content Tracker user metadata database tables are updated with information collected from the static URL accesses and from the service call event records gathered during the reporting time period.

For data reduction processing, Content Tracker uses:

- Content Tracker Event Logs (page 2-7)
- Combined Output Table (page 2-8)

Content Tracker Event Logs

When the Content Tracker web server filter plugin collects the access event details (static URL references), it records the information in raw event logs (sctLog files). The information in these files requires an explicit reduction (either interactive or scheduled) before it is included in the combined output table (SctAccessLog) along with the service call data.

Content Tracker supports multiple input files for different event log types and for configurations with more than one web server. For this reason, each web server filter plugin instance uses a unique tag as a filename suffix for the Content Tracker event logs. The unique identification suffix consists of the web server host name plus the server port number. The reduction process searches for and merges multiple raw event logs named sctLog-yyyymmdd-<*myhostmyport*>.txt. The raw event logs are processed individually.

This section covers the following topics:

❖ Recorded Usernames in Content Access Entries (page 2-7)

❖ File Storage after Reduction (page 2-8)

Recorded Usernames in Content Access Entries

Occasionally, in a raw event log entry, you may notice that Content Tracker does not capture a username for a content access event, even though the user is logged into Content Server. For example, a logged-in user performs a search, views the content information of an item, and clicks the Web location link. The raw event log entry includes the information except the username.

In this case, the item was access via a static URL request and, in general, the browser does not provide a username unless the web server asks it to send the user’s credentials. In particular, if the item is public content, the web server will not ask the browser to send user credentials, and the user accessing the URL will be unknown.

If you want Content Tracker to record the username for every document access, then you will need to configure your system such that a user login is required for every content item access. To do this, you must ensure that your content is not accessible to the guest role. In other words, if your content is not public, the user’s credentials will be required to access the items. This ensures that a username is recorded in the raw event log entry.

File Storage after Reduction

When raw data log files in the "new" cycle are reduced, the Data Engine moves the data files into the following subdirectories:

- <*install_dir*>/custom/ContentTracker/data/recent/yyyymmdd/

  The default number of data sets that the recent/ directory can hold is five sets (dates) of input data log files.

- <*install_dir*>/custom/ContentTracker/data/archive/yyyymmdd/

  The archive/ directory holds all input data log files that have been moved out of the "recent" cycle.

When raw data files are reduced, another file (*reduction_ts-yyyymmdd.txt)* is generated as a time stamp file. For more detailed information about reduction cycle states for raw data file processing, see Reduction Tab (page 3-4).

Combined Output Table

The SctAccessLog table contains entries for all static and dynamic content access event records. The rows in the table are tagged according to type:

- S indicates the records logged for service calls.
- W identifies the records logged for static URL requests.

The SctAccessLog table is organized using one line per event in the reporting period.

Note: By default, Content Tracker does not log accesses to GIF, JPG, JS, CSS, CAB, and CLASS file types. This means that Web activity involving GIFs, JPGs, JSs, CSSs, CABs, and CLASSs will not result in entries in the web server filter plugin event log files. Subsequently, entries for these file types will not be included in the combined output table (SctAccessLog) after data reduction.

To change the logging status of these file types, the desired file types must be enabled in the sct.cfg file located in the <*install_dir*>/custom/ContentTracker/resources/ directory. To enable logging of these file types, adjust the default setting for the SctIgnoreFileTypes (page A-3) configuration variable (gif,jpg,js,css). The default setting excludes these file types. To include one or more of these file types, delete each desired file type from the list. To ensure that these changes take effect, it is necessary to restart the web server and Content Server.

Note: The Content Tracker web server filter plugin cannot distinguish between URLs for user content and those used by the Content Server user interface. Therefore, it is possible that references to UI objects, such as client.cab, may appear in the static access logs. To eliminate these false positives, you may define a list of directory roots that are to be ignored by the Content Tracker filter plugin.

The list of directories is stored in the SctIgnoreDirectories (page A-3) configuration variable in the *sct.cfg* file located in the <*install_dir*>/custom/ContentTracker/resources/ directory. This list will eliminate most if not all of the user interface object references.

You can manually change the contents of the SctIgnoreDirectories value to list all the directories whose content should be ignored. You may want to change the default value:

- If you wish access to the UI objects to be logged along with user content.
- If you have a different view of which directories should be logged and which should be excluded from the logs.

The following table provides the information collected for each record in the SctAccessLog table.

Column Name	Type / Size	Field Definition
SctDateStamp	datetime	Local date when data collected - YYYYMMDD (contingent on customer location and time of day event occurs -- may differ from date recorded for eventDate); time set to 00:00:00 Data source: Internal
SctSequence	int / 8	Sequence unique to entry type Data source: Internal Cloned from: Revisions.dID
SctEntryType	char / 1	Entry type - “W” or “S” Data source: Internal Cloned from: Revisions.dReleaseState
eventDate	datetime	GMT time and date when request completed (date contingent on customer location and time of day event occurs -- may differ from date recorded for SctDateStamp)
SctParentSequence	integer	Sequence of outermost Service Event in tree, if any.
c_ip	varchar / 15	IP of client

Column Name	Type / Size	Field Definition
cs_username	varchar / 255	Cloned from: Revisions.dDocAuthor
cs_method	varchar / 10	“GET”
cs_uriStem	varchar / 255	Stem of URI
cs_uriQuery	varchar / [maxUrlLen]	Query portion, e.g. “IdcService=GET_FILE&dID=42...”
cs_host	varchar / 255	Content Server server name
cs_userAgent	varchar / 255	Client User Agent Ident
cs_cookie	varchar / [maxUrlLen]	Current cookie
cs_referer	varchar / [maxUrlLen]	URI leading to this request
sc_scs_dID	int / 8	dID Data source: from query or derived from URL (reverse lookup) Cloned from: Revisions.dID
sc_scs_dUser	varchar / 50	dUser Data source: Service DataBinder “dUser” Cloned from: Revisions.dDocAuthor
sc_scs_idcService	varchar / 255	Name of IdcService, e.g. GET_FILE Data source: from query or Service DataBinder “IdcService”
sc_scs_dDocName	varchar / 30	dDocName Data source: from query of Service DataBinder “dDocName” Cloned from: Revisions.dDocName
sc_scs_callingProduct	varchar / 255	Arbitrary identifier Data source: SctServiceFilter config file or Service DataBinder “sctCallingProduct”
sc_scs_eventType	varchar / 255	Arbitrary identifier Data source: SctServiceFilter config file or Service DataBinder “sctEventType”

Column Name	Type / Size	Field Definition
sc_scs_status	varchar / 10	Service execution status Data source: Service DataBinder "StatusCode"
sc_scs_reference	varchar / 255	"web", "native", "sdc_url" Values indicate the rendition of the accessed file; "web" = converted file (PDF), "native" = actual original file, and "sdc_url" = HTML. Data source: algorithmically from query parameters or ServiceFilter config file
comp_username	varchar / 50	Computed username. If a Service, obtained from UserData Service Object or HTTP_INTERNETUSER or REMOTE_USER or dUser. If a static URL, obtained from auth-user or internetuser.
comp_validRef	char / 1	"1" if the access was a Web reference (W), and ispromptlogin and isaccessdenied are both NULL, and the static URL exists at reduction time. Or, if the access was a service call (S) and the sc_scs_status field is NULL. "NULL" if the static URL did not exist at reduction time, or the user logon failed, or the logon succeeded but the user was not authorized to view the object. Indicates whether the referenced object exists and is available to the requesting user.
sc_scs_isPrompt	char / 1	"1" if true Data source: Plugin immediateResponseEvent field "ispromptlogin" Cloned from: Revisions.dReleaseState
sc_scs_ isAccessDenied	char / 1	"1" if true Data source: Plugin immediateResponseEvent field "isaccessdenied" Cloned from: Revisions.dReleaseState
sc_scs_inetUser	varchar / 50	Internet user name (if security problem) Data source: Plugin immediateResponseEvent field "internetuser" Cloned from: Revisions.dDocAuthor

Column Name	Type / Size	Field Definition
sc_scs_authUser	varchar / 50	Authorization user name (if security problem) Data source: Plugin immediateResponseEvent field “auth-user” Cloned from: Revisions.dDocAuthor
sc_scs_ inetPassword	varchar / 8	Internet password (if security problem) Data source: Plugin immediateResponseEvent field “internetpassword”
sc_scs_ serviceMsg	varchar / 255	Content Server service completion status Data source: Service DataBinder “StatusMessage”
extField_1 through extField_10	varchar / 255	General purpose columns to use with the extended service tracking function. In the field map ResultSets, the DataBinder fields are mapped to these columns.

DATA OUTPUT

Content Tracker snapshots user metadata as well as static URL accesses. It also logs service calls that are written in real time to the combined output table. Data reduction is necessary to process the static URL information and add it to the combined output table. Additionally, as a result of the data reduction process, the related Content Tracker user metadata database tables are updated with the information derived from processing the static URL data and the service call data.

The Content Tracker reduction process generates:

- Metadata Capture (page 2-13)
- Reduction Log Files (page 2-17)

Metadata Capture

In addition to static and dynamic content access request information, all metadata fields are accessible for use in reports generated by the Content Tracker Reports component. The logged metadata includes:

- Content Item Metadata (page 2-13)
- User Metadata (page 2-13)

Content Item Metadata

Rather than collecting content item metadata information, Content Tracker uses the standard Content Server metadata tables for content item metadata. This means that Content Tracker reports necessarily reflect current content item metadata. Therefore, if content item metadata has changed since a content item was accessed, any generated reports will reflect the changed metadata.

User Metadata

During the data reduction process, Content Tracker user metadata database tables are updated with information collected about users that were active during the reporting time period. These tables retain historically accurate user metadata. The names of the user metadata tables are formed from the root, which indicates the class of information contained, and an "Sct" prefix to distinguish the Content Tracker tables from native Content Server tables.

Two complete sets of user metadata database tables are created:

- Primary

  The Primary tables, named SctUserInfo, etc., contain the output for reduction data in the "new" and "recent" cycles.

- Archive

  The Archive tables, named SctUserInfoArchive, etc., contain output for reduction data in the "archive" cycle.

When a reduction data file is moved from "recent" to "archive," the associated table records are moved from the Primary table to the Archive table. This prevents excessive buildup of rows in the Primary tables, and ensures that queries performed against recent data will complete quickly. Rows in the Archive table will not be deleted. You may move them to alternate storage for your historical records, or you may delete them using any

SQL query tool. For more information about the reduction process and data cycles, see Reduction Tab (page 3-4).

Tech Tip: If you wish to delete all the rows in the Archive tables, you can simply delete the tables themselves. They will be recreated during the next Content Server restart.

Note: Reports are not run against archive data. Therefore, any data that has been demoted from ‘recent’ to ‘archive’ will not be included in the generated reports.

Updated user metadata tables include:

- SctAccounts Table (page 2-14)
- SctGroups Table (page 2-15)
- SctUserAccounts Table (page 2-15)
- SctUserGroups Table (page 2-16)
- SctUserInfo Table (page 2-16)

SctAccounts Table

The SctAccounts table contains a list of all accounts. The SctAccounts table is organized using one line for each account.

Field Name	Type / Size / Field Definition
SctDateStamp	datetime GMT day when data is collected
dDocAccount	varchar / 30 Name of a Content Server account

SctGroups Table

The SctGroups table contains a list of all user groups current at time of reduction. The SctGroups table is organized using one line per content item group.

Field Name	Type / Size / Field Definition
SctDateStamp	datetime GMT day when data is collected
dGroupName	varchar / 30 Name of a content item group

SctUserAccounts Table

The SctUserAccounts table contains entries for all the users who are listed in the SctUserInfo table and who are assigned accounts that are defined in the current instance. A separate entry exists for each user-account combination.

There is a special situation in which the group and account information of a user is not determined by Content Tracker. This occurs in a proxied configuration that has multiple proxy instances. When the current instance is a proxy, the group information for an active user who is defined in a different proxy is replaced by a single placeholder line in SctUserGroups for that user. This line contains the username and a "-" placeholder for the group. If at least one account is defined in the current instance, a similar entry is created in SctUserAccounts for any user who is defined in a different proxy.

The SctUserAccount table is organized using one line per Content Server user and user’s account.

Field Name	Type / Size / Field Definition
SctDateStamp	datetime GMT day when data is collected
dUserName	varchar / 100 Name of the user. If local to a proxy instance, is prefixed by the content server relative URL, e.g. cs_2/user1

Field Name	Type / Size / Field Definition
Account	varchar / 30 Name of account to which user has access. Placeholder in proxy-to-proxy configuration when current proxy instance has at least one account.

SctUserGroups Table

The SctUserGroups table references only those users who logged on during the data collection period. If ContentTracker is running in a proxied Content Server configuration, only groups that are defined in the current instance are listed. For example, a user named "joe" is defined in the master instance and has access to groups "Public" and "Plastics" in the master instance. If "joe" logs on to a proxy instance and the group "Plastics" is not defined in the proxy, only the association between "joe" and "Public" will appear in SctUserGroups.

The SctUserGroups table is organized using one line for each user's group for each user that is active during the reporting period.

Field Name	Type / Size / Field Definition
SctDateStamp	datetime GMT day when data is collected
dUserName	varchar / 100 Name of the user. If local to a proxy instance, is prefixed by the content server relative URL, e.g. cs_2/user1
dGroupName	varchar / 30 Name of group that the user has permission to access. No distinction is made for the type of access (R, RW, etc.).

SctUserInfo Table

The SctUserInfo table includes all users known to the current instance plus any additional users from a different instance who have logged on to the current instance during the data collection period. In a proxied configuration, users that are local to one instance are usually known (visible from the UserAdmin application) to other instances. (For this visibility to occur, Content Server instances must typically be restarted after local users have been added.) However, when a user is defined locally with the same name in two instances, only the local user is visible in each of these instances.

For example, the user "sysadmin" defined in the master is not the "sysadmin" user that appears in the UserAdmin application for a proxy. The proxy has its own "sysadmin" user who is defined locally. It is possible for these two different users to both log on during the same data collection period: the user from the master would log on as "sysadmin" and the user from the proxy would log on as something like "cs_2/sysadmin". In this case cs_2/ is the server relative URL that must be prepended to the proxy username. The userinfo file generated for this period will contain separate entries for "sysadmin" and "cs_2/sysadmin".

The SctUserInfo table is organized using one line per Content Server user.

Field Name	**Type / Size / Field Definition**
SctDateStamp	datetime GMT day when data is collected
dUserName	varchar / 100 Name of the user. If local to a proxy instance, is prefixed by the content server relative URL, e.g. cs_2/user1
dUserType	varchar / 30 Type of the user. Placeholder if user has no type

Reduction Log Files

When data reduction is run, the Content Tracker Data Engine generates a summary results log file in <*install_dir*>/custom/ContentTracker/log/. The reduction log files are named using the format reduction-yyyymmdd.log. The reduction logs may be useful to help diagnose errors that have occurred during the data reduction process. For more information about the raw event log files and their corresponding reduction logs, see Content Tracker Event Logs (page 2-7).

TRACKING LIMITATIONS

The current versions of Content Tracker and Content Tracker Reports have the following tracking limitations:

- Tracking Limitations in Single-Box Clusters (page 2-19)
- Tracking Limitations with Static URLs and WebDav (page 2-19)

Tracking Limitations in Single-Box Clusters

Currently, Content Tracker and Content Tracker Reports do not support multi-node clusters that are installed in a single server. This is true even though multiple network cards are installed and each cluster node has its own IP address. In this case, the Content Server instance for each cluster node can successfully bind its IntradocServerPort to its specific IP address.

Unfortunately, only one cluster node is able to bind its Incoming Provider ServerPort to its specified IP address. Consequently, all of the cluster nodes share and alternately use the same Incoming Provider ServerPort. As a result, the SctLock provider for Content Tracker can only track document accesses on one cluster node at a time.

Tracking Limitations with Static URLs and WebDav

Content Tracker is unable to guarantee exact access counts for content requested through static URLs or by WebDAV clients. The access counts determined by Content Tracker are generally correct, but there are specific, exceptional circumstances in which Tracker is unable to determine whether the content was actually delivered to the requesting user, or if it was, which specific revision of the content was delivered. The following cases may result in incorrect access counts:

- Missed Accesses for Content Repeatedly Requested via WebDAV (page 2-19)
- False Positive for Access by Saved (stale) Static URL (page 2-20)
- Wrong dID Reported for Access by Saved Static URL (page 2-21)

Missed Accesses for Content Repeatedly Requested via WebDAV

Scenario: User accesses a document via a WebDAV client, and then accesses the same document in the same manner later on. Only the first WebDAV request for the document is recorded. Access counts reported for such content will tend to be lower than actual.

Details: WebDAV clients typically use some form of object 'caching' to reduce the amount of network traffic. If a user requests a particular object, the client will first determine whether it already has a copy of the object in a local store. If it does not, the client will contact the server and negotiate a transfer. This transfer will be recorded as a COLLECTION_GET_FILE service request.

If the client already has a copy of the object, it will contact the server to determine whether the object has changed since the client local copy was obtained. If it has changed, then a new copy will be transferred and the COLLECTION_GET_FILE service details will be recorded.

If the client copy of the object is still current, then no transfer will take place, and the client will present the saved copy of the object to the user. In this case, the content access will not be counted even though the user appears to get a "new" copy of the original content.

False Positive for Access by Saved (stale) Static URL

Scenario: User saves a "Web Location" (URL) for a content file. The content is subsequently revised in such a way that the saved URL is no longer valid. The user then attempts to access the content via the (now stale) URL, and gets a "Page Cannot be Found" error (HTTP 404). Content Tracker may record this as a successful access even though the content was not actually delivered to the user. Access counts reported for such content will tend to be higher than actual.

Details: The "Web Location" of a content file is the means by which a user can access content via a "static URL". The specific file path in the URL is used in two, slightly different contexts: It is used by the web server to locate the content file in the Content Server repository, and it is also used by Content Tracker to determine the dID and dDocName of the content file during the data reduction process. The problem occurs when the content is revised in such a way that the web location for a given Content ID changes between the time the URL is saved and the time the access is attempted.

For example, if a Word document is checked in, and then revised to an XML equivalent, then the web location for the latest revision of the content will change from:

/stellent/groups/public/documents/adacct/xyzzy.doc
to:
/stellent/groups/public/documents/adacct/xyzzy.xml
where: "xyzzy" is the assigned Content ID.

The original revision is “renamed” as:
/stellent/groups/public/documents/adacct/xyzzy~1.doc

This means the original Web Location will no longer work as a static URL. The Content ID obtained from the original URL, however, will match the latest revision. Consequently, Content Tracker reports this as an access to Content ID "xyzzy", even though the web server was unable to deliver the requested file to the user.

Wrong dID Reported for Access by Saved Static URL

Scenario: User accesses content via the "Web Location" (URL). The content is then revised before the Content Tracker data reduction operation is performed. The user will be reported as seeing the latest revision, not the one that s/he actually saw. Access counts reported for such content will tend to be attributed to a newer revision than actual. You can minimize this effect by scheduling or running Content Tracker data reductions on a regular basis.

Details: This is related to False Positive for Access by Saved (stale) Static URL, described above. That is, the web server uses the entire web location, (e.g. /stellent/groups/public/documents/adacct/xyzzy.doc), to locate and deliver the content, while Content Tracker uses only the ContentID portion to determine the dID and dDocName values. Moreover, Content Tracker makes this determination during data reduction, not at the time the access actually occurs. Consequently, Content Tracker will report the user as having seen the revision current at the time of the reduction, not the one that was current at the time of the access.

There are some implications of this that are not immediately obvious, such as when the group and/or security of the revision are changed from the original. For example, if a user accesses "Public" Revision 1 of a document through a static URL, and the document is subsequently revised to Revision 2 and changed to "Secure" before the Content Tracker data reduction takes place, Tracker will report that the user saw the Secure version. This may also occur when the content file type changes. If the user accesses an original .xml version, which is then superseded by an entirely different .doc before the data reduction is performed, Tracker will report that the user saw the .doc revision, not the actual .xml.

Chapter

3

USING CONTENT TRACKER

OVERVIEW

This section covers the following topics:

Concept

Interface

Tasks

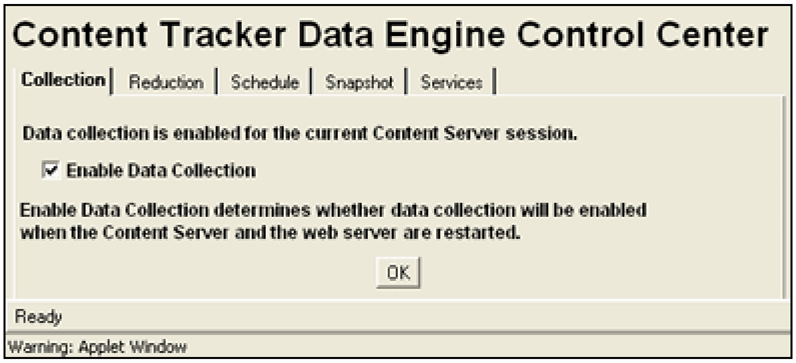
Data Engine Control Center

The Data Engine Control Center is the applet interface that provides access to the user-controlled functions of the Data Engine. You access the applet by clicking the Content Tracker Administration link on the Administration tray. Then you click the Data Engine Control Center icon on the resulting page.

Through the interface, you can:

- Enable and disable data collection.
- Start and stop data reduction.
- Monitor progress of data reduction.
- Delete data.
- Schedule data reduction to run automatically.
- Enable activity metrics and collect search relevance information about managed content accesses.

- ❖ Add, configure, and edit service calls to be logged.
- ❖ Define specific event details that are to be logged for a given service.



Content Tracker Data Engine Control Center

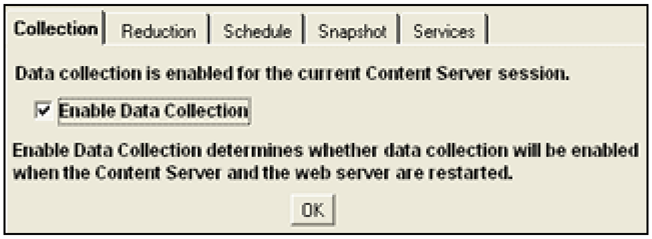
Data collection is enabled for the current Content Server session.

☑ Enable Data Collection

Enable Data Collection determines whether data collection will be enabled when the Content Server and the web server are restarted.

Collection Tab

Use the Collection tab to enable web traffic data collection for a Content Server session. When data collection is enabled, the Content Tracker Data Engine collects and writes raw data to logs located in the /data directory (<*install_dir*>/custom/ContentTracker/data/). These logs provide part of the input used in the reduction process and will accumulate as long as the collection process is properly enabled. Data accumulates whether or not the reduction process is performed.



Data collection is enabled for the current Content Server session.

☑ Enable Data Collection

Enable Data Collection determines whether data collection will be enabled when the Content Server and the web server are restarted.

Feature	**Description**
Enable Data Collection check box	Selecting this check box enables data collection for the current Content Server session.
OK button	Implements the current data collection setting.

Important: Changing the status of the check box does not immediately enable or disable data collection. The Content Server and the web server must be restarted before any changes will be in effect. Look carefully at the sentence above the check box to determine whether data collection is enabled.

- When enabled, the sentence reads "Data collection is enabled..."
- When disabled, the sentence reads "Data collection is not enabled..."

Reduction Tab

Use the Reduction tab to start and stop data reduction manually, to monitor progress of a data reduction operation, and to delete the raw data files from which the table rows are generated. During reduction, data is written to the appropriate tables and creates a log file that reflects the reduction process. These log files reside in the log/ directory (<*install_dir*>/custom/ContentTracker/log/).

Each line item on the Reduction tab is raw (input) data that is gathered and organized on a daily basis. The raw data is the unprocessed data collected from the web server filter plugin. This data is ultimately used as input to the Content Tracker reduction process.

The primary concepts of data reduction include:

- Data Reduction Cycles (page 3-4)
- Access Modes and Data Reduction (page 3-5)
- Reduction Sequence for Event Logs (page 3-5)

Data Reduction Cycles

Reduced table data is moved from the primary tables to the corresponding archive tables when the associated raw data is moved from ‘recent’ to ‘archive’ status. The primary tables contain the output for reduction data in the ‘new’ and ‘recent’ cycles and the archive tables contain output for reduction data in the ‘archive’ cycle.

Raw data is demoted from ‘new’ to ‘recent’ when the data is reduced and it is more than one day old. Thus, the ‘new’ cycle indicates that the data is for the current day or is unreduced data from previous dates. The ‘recent’ cycle indicates that the data is from yesterday or earlier and has been reduced.

Raw data is demoted to ‘archive’ (and the corresponding rows in the SctAccessLog table are moved to the SctAccessLogArchive table) when the number of ‘recent’ sets reaches a configured threshold number and a reduction process is run, either manually or via the scheduler. For more information about configuring the threshold number for ‘recent’ sets, see SctMaxRecentCount (page A-4). If a reduction process is never run, the raw data remains in the ‘recent’ cycle indefinitely.

Access Modes and Data Reduction

The way users access content items determines how those accesses are recorded in the SctAccessLog table. There are two basic user access modes: service accesses (viewing the actual native file) and static URL accesses (viewing the web location file). If content items are accessed via a service, the events are recorded in the SctAccessLog table in real time. In this case, the activity is recorded immediately and is not dependent on the reduction process.

However, if content items are accessed via static URLs, the web server filter plugin records the events in a static log file. During the data reduction process, the static log files for a specified date are gathered and the data is moved into the SctAccessLog table. In this case, if data reduction is not performed for a given date, there will be no static URL records in the SctAccessLog and no evidence that these accesses ever occurred.

Thus, the difference in the way static and service accesses are processed has implications with regard to interval counts—see Snapshot Tab (page 3-11). For example, a user might access a content item twice on Saturday: once via the web location file (static access) and once via the native file (service access). The service access is recorded in the SctAccessLog table but the web location access is not.

Then, if Sunday’s data is reduced, only the service access (not the static access) is included in the summaries of the short and long access count intervals. However, if Saturday’s data is also reduced, then both the service and static accesses are recorded in the SctAccessLog table and, subsequently, included in the short and long access intervals.

Reduction Sequence for Event Logs

Generally, data sets are reduced in chronological order to ensure that the information included in generated reports is as current as possible. In particular, the order in which the raw data log files are reduced determines what specific user access data is logged and counted. During reduction, the SctAccessLog and user metadata database tables are modified with data from the raw data files.

If you are using the snapshot function to gather search relevance information, then the metadata fields associated with the activated activity metrics are also updated during data reduction. The activity metrics use custom metadata fields that are included in Content Server's DocMeta database table. For more information, see Snapshot Tab (page 3-11).

The currentness of the information in the various database tables is dependent on the order in which you reduce the data sets. Content Tracker always changes the activity metrics values according to the applicable data in the reduction data set. Normally, data sets are reduced in calendar order to ensure that activity metrics will advance as expected. In fact, to ensure that data values are complete and current, you should perform data reduction on a daily basis.

Note: If the data sets are reduced out of order, re-reducing the current or most recent data set will correct the counts. However, it is always preferable to consistently reduce data in calendar order.

The following scenarios show how the reduction sequence affects the stored data.

Scenario 1:

Depending on how content items are accessed, if activity on certain days (such as Saturdays and Sundays) is never reduced, then accesses that occur on those days might never be logged or counted—see Access Modes and Data Reduction (page 3-5). Similarly, if a content item is accessed on Tuesday and reductions are done for Monday and Wednesday, the Tuesday access is might not be counted toward the last access of that content item.

Scenario 2:

If there was a significant increase in accesses in the last few days, and you reduce data from two weeks earlier, the long and short access metrics for content items will not reflect the recent activity. Instead, the interval values from two weeks earlier override today's values. Reducing the current or most recent data set will correct the counts.

Note: The reduction order does not adversely affect the Last Access date. The reduction process only changes the Last Access date if the most recent access in the reduction data set is more recent than the current Last Access value in Content Server's DocMeta database table.

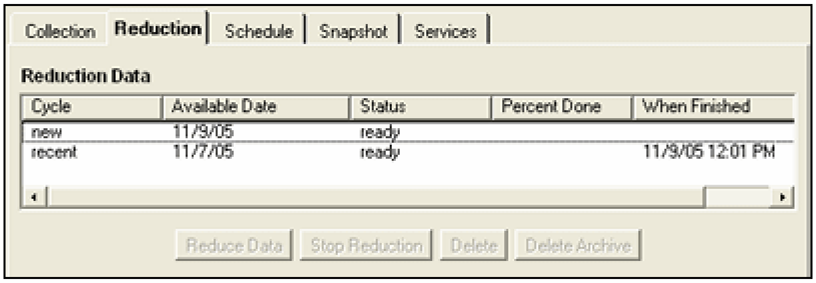
If you have reduced a recent data set and a particular content item had been accessed, the Last Access field is updated with the most recent access date in the reduction data set. If you then re-reduce an older data set, the older access date for this content item will not overwrite the current value.

For more information about long and short activity metrics, see Snapshot Tab (page 3-11) and Enable Short / Long Access Count updates check boxes and corresponding Fields / Intervals (page 3-15).

Scenario 3:

Reducing the data sets in an arbitrary order interferes with the demotion of “recent” data files to “archive” data files. The movement of the associated table records is based on the age—archive tables are intended to store the “oldest” data. If the data sets are reduced in random order, it is not apparent which data is the oldest.

For more information about recent and archive data files, see User Metadata (page 2-13), Data Reduction Cycles (page 3-4), and Cycle column (page 3-8).



Reduction Data

Cycle	Available Date	Status	Percent Done	When Finished
new	11/9/05	ready		
recent	11/7/05	ready		11/9/05 12:01 PM

Feature	**Description**
Cycle column	Shows the state of the input data. **new** = the input data for an available date has not been reduced. After input data has been reduced, the cycle changes to ‘recent.’ Exception: If input data for the current system date is reduced, the cycle continues to display the data as ‘new.’ **recent** = the input data has been reduced but has not been moved to archive. The number of recent sets is user configurable. The default number is five sets of input data that have been most recently run through data reduction will be displayed as ‘recent.’ **archive** = the input data has been reduced and has been moved out the cycle ‘recent.’ Data will remain in archive cycle until deleted.
Available Date column	Shows the date for which and on which the input data was collected.
Status column	Shows the status of the reduction data. **ready** = the input data is available to be reduced. **running** = the selected input data is being reduced. **archiving** = the input data is being moved from ‘recent’ to ‘archive’ cycle

<table>
<tr><th>Feature</th><th>Description</th></tr>
<tr><td>Percent Done column</td><td>Shows the progress of the data reduction process. Displays only while the input data is ‘running.’</td></tr>
<tr><td>When Finished column</td><td>Shows the date and time when the data reduction process was completed.</td></tr>
<tr><td>Reduce Data button</td><td>Pressing this button initiates the data reduction process on the selected input data.</td></tr>
<tr><td>Stop Reduction button</td><td>Pressing this button terminates a running data reduction process.</td></tr>
<tr><td>Delete button</td><td>Pressing this button deletes the selected input data.</td></tr>
<tr><td>Delete Archive button</td><td>Pressing this button deletes the input data that is in the ‘archive’ cycle.</td></tr>
<tr><td colspan="2"><b>Note:</b> The Delete and Delete Archive buttons enable users to eliminate raw data logs. However, there is no provision for deleting the reduced data records from either the primary or archive tables. Users must use a database utility to manage the primary and archive tables.
However, if the raw data is deleted using the Delete and Delete Archive buttons, any reduced data in the tables will not be affected.</td></tr>
</table>

Schedule Tab

Use the Schedule tab to enable automatic data reduction. Using the Schedule tab, reduction runs can be configured to run on a scheduled basis. A common usage scenario is to use the scheduler to periodically reduce the raw data. In this case, there would be a steady flow of raw data into the 'recent' and 'archive' repositories, and a similarly steady flow of reduced data from the primary tables to the archive tables. For additional information about raw data, data statuses and primary / archive tables, see Reduction Tab (page 3-4).

The following are key characteristics of the Content Tracker reduction process:

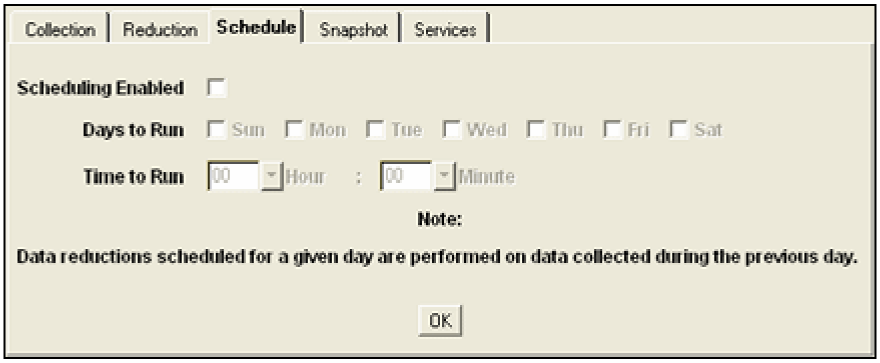
- If the Content Tracker Data Engine is disabled the day before a scheduled reduction run, no data is collected. If the Content Tracker Data Engine is enabled on the day of the scheduled reduction run, the scheduler will not run because no data is available.
- Data reductions scheduled for a given day are performed on data collected during the previous day. The previous day is defined as the 24-hour period beginning and ending at midnight (system time).

Important: Depending on various conditions of the system load, the following error may be issued if the scheduled reduction is set to run within a few minutes after midnight:

```
<date_time>: Cannot reduce data. A request is in progress to delete raw data
that was generated on this date.
```

If this message is issued, try scheduling the reduction run 5 or 10 minutes later.

Tech Tip: To conserve CPU resources, reduction runs can be scheduled for early morning hours when the system load is generally the lowest.



Scheduling Enabled ☐

Days to Run ☐ Sun ☐ Mon ☐ Tue ☐ Wed ☐ Thu ☐ Fri ☐ Sat

Time to Run 00 Hour : 00 Minute

Note:

Data reductions scheduled for a given day are performed on data collected during the previous day.

Feature	Description
Scheduling Enabled check box	Selecting this check box enables data reduction to run automatically.
Days to Run check boxes	Selecting one or more check boxes sets the days when the data reduction will run.
Time to Run fields	Used to select the hour and minute to set the time when the data reduction will run.
OK button	Saves the current reduction schedule settings.

Snapshot Tab

Use the Snapshot tab to selectively enable and automatically assign specific activity metrics to pre-defined, custom metadata fields. When activated, the activity metrics and corresponding metadata fields provide search relevance information about user accesses of content items. An optional automatic load function allows you to update the last access activity metric to ensure that checked-in content items are appropriately timestamped.

Content Tracker fills the search relevance custom metadata fields with content item usage information that indicates the popularity of particular content items. This information includes the date of the most recent access and the number of accesses in two distinct time intervals.

Users can apply the information generated from these activity metrics functions in various ways. You can selectively use the activity metrics to subsequently order search results based on content item popularity. For example, you might want to order search results according to which content items have been recently viewed or the most viewed in the last week.

If the snapshot function is activated, the values in the search relevance metadata fields are updated during a post-reduction step. During this processing step, Content Tracker uses SQL queries to determine which content items have changed activity metrics values. Content Tracker updates the applicable database tables with the new values and initiates a re-indexing cycle. However, only the content items that have changed metadata values are re-indexed. See Data Reduction Process with Activity Metrics (page 2-3).

Note: The Snapshot tab enables you to automatically activate the snapshot function and selectively enable each of the activity metrics. Each function that you activate must have a custom metadata field associated with it.

For more information, see Enabling the Snapshot function and the Activity Metrics Options (page 3-28) and Linking Activity Metrics Functions to Search Relevance Metadata Fields (page 3-29).

Alternatively, you can manually update the applicable configuration variables in Content Tracker's *sct.cfg* file. See Appendix A (*Configuring and Customizing Content Tracker*).

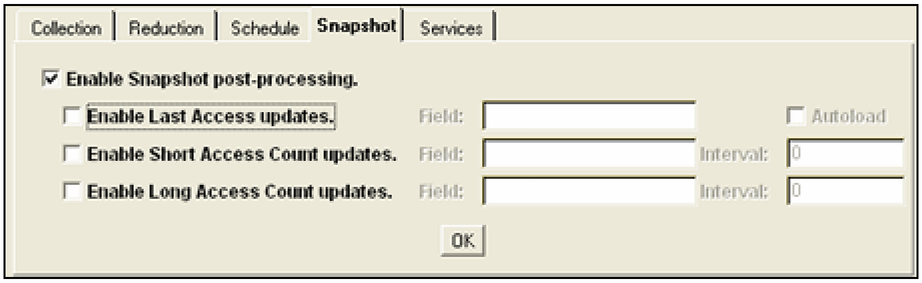
Important: Before you can link the activity metrics functions to custom metadata fields, they must already exist and must be of the correct type. The metadata field associated with the Last Access metric must be of type Date. The metadata fields associated with the Access Count metrics must be of type Integer. See Creating the Search Relevance Metadata Fields (page 3-26).

Caution: When you create custom metadata fields to use in conjunction with the activity metrics, you have the option to enable them for the search index. If the custom metadata fields are indexed (and searchable), the access values stored in them are more efficiently accessed. That is, indexed fields are more useful for selecting and/or ordering search results by relevance.

Indexing is expensive, particularly when full text search is enabled. The disadvantage of indexed metadata fields is that when the values in the search relevance metadata fields change, the affected content items must be re-indexed to update their values in the database table. Therefore, on a large instance with many content item accesses, updating the search relevance fields will adversely affect performance.

Alternatively, you can disable the indexing function of the custom metadata fields. In this case, it is possible to search on and find values for non-indexed metadata fields, but the search is more expensive.

Note: If re-indexing the affected content items degrades the performance too severely, you can optionally deactivate the Snapshot function. Unfortunately, this means that the activity metrics information will no longer be collected. As a result, you will be unable to order current search results by usage (for example, listing accessed content items in order of decreasing popularity).



Collection | Reduction | Schedule | **Snapshot** | Services

☑ Enable Snapshot post-processing.

□ Enable Last Access updates. Field: □ Autoload

□ Enable Short Access Count updates. Field: Interval: 0

□ Enable Long Access Count updates. Field: Interval: 0

OK

<table>
<tr><th>Feature</th><th>Description</th></tr>
<tr><td>Enable Snapshot post-processing check box</td><td>Selecting this check box activates the activity metrics functions and enables users to individually select and assign functions to applicable metadata fields. When you enable the snapshot function, the metadata auto-load occurs after the next reduction cycle. By default, the snapshot function is disabled.</td></tr>
<tr><td colspan="2"><b>Important:</b> Before you activate the snapshot function, you must decide which custom metadata fields will be associated with each of the enabled activity metrics. Also, the custom metadata fields must already exist and must be of the correct type.
The search relevance metadata field associated with the Last Access metric must be of type Date. The search relevance metadata fields associated with the Access Count metrics must be of type Integer. See Creating the Search Relevance Metadata Fields (page 3-26).</td></tr>
<tr><td>Enable Last Access updates check box and corresponding Field metadata field</td><td>Selecting this check box activates the associated search relevance metadata field and the Autoload check box. In the Field field, enter the internal name of the metadata field to be linked to this activity metric. For example, xLastAccess.</td></tr>
<tr><td colspan="2"><b>Note:</b> When content is checked in, the Last Access field in Content Server’s DocMeta database table is initially empty. After a data reduction has been performed, the Last Access field is updated with the most recent date and time of access, or with the date and time of checkin if no accesses have occurred. But, some applications need to have the Last Access field contain a valid value at all times. You can do this using the Default Value field, the Autoload option, or the Batch Loader. See Setting a Checkin Time Value for the Last Access Metadata Field (page 3-30).</td></tr>
</table>

<table>
<tr><th>Feature</th><th>Description</th></tr>
<tr><td colspan="2"><b>Note:</b> For the Last Access metric, Content Tracker only checks for accesses on the reduction date. As a result, there may be gaps in the record. For more information and examples, see Reduction Sequence for Event Logs (page 3-5).</td></tr>
<tr><td>Autoload check box</td><td>Selecting this check box and clicking the OK button invokes a query that, by default, uses the current date and time to populate any empty Last Access metadata fields in Content Server’s DocMeta database table. See Populating the Last Access Field Using the Autoload Option (page 3-32).</td></tr>
<tr><td colspan="2"><b>Important:</b> If you use the Autoload option, be aware of the following operational considerations:
❖ Autoload is primarily intended for use with applications that count checkin operations as an access activity. For more information, see Populating the Last Access Field Using the Using the Default Value (page 3-31)
❖ Running Autoload may affect every record in Content Server’s DocMeta database table. therefore, you should use this option sparingly.
❖ The only DocMeta records that are affected are those where the Last Access metadata field is empty (NULL).
❖ Autoload is persistent. The state of the Autoload check box is saved along with all the other Snapshot settings. To ensure that you do not inadvertently use this option, you should clear the Autoload check box and re-save your activity metrics field settings immediately after performing the autoload function.
❖ Content Tracker does not invoke Content Server’s indexer after Autoload completes the update. You must decide when to rebuild the collection.
❖ By default, the Autoload query sets the Last Access metadata field to the current date and time. However, you may customize the query to meet the needs of your application. See Customizing the Autoload Option SQL Query (page A-9).</td></tr>
</table>

Feature	Description
Enable Short / Long Access Count updates check boxes and corresponding Fields / Intervals	Selecting these check boxes activates the associated search relevance metadata fields and time interval fields. In the Field field(s), enter the internal name(s) of the metadata field(s) to be linked to the activity metric(s). For example, xShortAccess or xLongAccess. Specify the interval in days for the activity metric count(s).

Note: The two Access Count metrics differ only in the accounting period. For example, last 30 days versus last 90 days, last week versus last year, etc. Also, the time intervals specified in the activity metrics are independent of each other. For example, you can set the number of days in the first interval period (Short Access) to more than those in the second interval period (Long Access).

Note: Access counts are only tabulated for dates that have been reduced. If you do not reduce data for one or more days, the accesses on those days are not logged or counted. For more information and examples, see Reduction Sequence for Event Logs (page 3-5).

Important: You should not reduce data in random order because the Access Count metrics are affected by the reduction date order. For more information and examples, see Reduction Sequence for Event Logs (page 3-5).

Note: The fields on the Snapshot tab are case-sensitive. Therefore, it is important that all field values are spelled and capitalized correctly. Content Tracker uses the following error checks to validate each enabled activity metric field value:

- Content Tracker checks the DocMeta database table to ensure that the custom metadata field actually exists.
- Content Tracker ensures that the custom metadata field is of the correct type. That is, the Last Access metadata field must be of type Date and the Short / Long Access Count fields must be of type Integer.
- Content Tracker checks to explicitly exclude the dID metadata field.

Feature	Description
OK button	Saves the snapshot configuration and updates Content Tracker’s sct.cfg file with the current settings in the activity metrics fields. If the Autoload check box is selected, immediately after the activity metrics settings are saved, the Last Access field is updated in the DocMeta database table.

Important: If you make any changes to the configuration, all of the content items are updated during the next reduction cycle.

Services Tab

Use the Services tab to log additional Content Server service calls along with data values relevant to the associated services. The Services tab enables you to conveniently add or edit service entries and, optionally, their corresponding field map ResultSets in the service call configuration file (SctServiceFilter.hda). Every service that you want to be logged must have a service entry in the SctServiceFilter.hda file.

The service entries in the SctServiceFilter.hda file allow Content Tracker to gather event and usage information. The enabled services automatically log various general DataBinder fields, such as dUser and dDocName. Linking a field map ResultSet to a service entry enables you to use the extended service call tracking function. The field map ResultSet consists of a list of data field names, location names, and their associated general purpose table column names in the output database table (SctAccessLog).

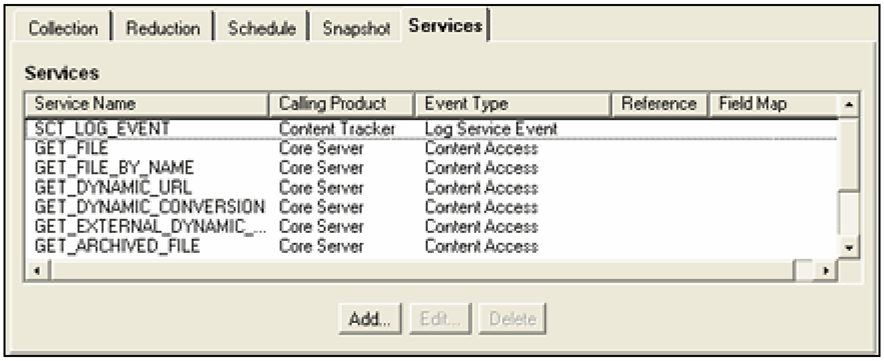
The SctAccessLog table provides additional general purpose columns for use with the extended service call tracking function. You can fill these columns with any data values you feel are appropriate for the associated service call. When you list the data field names in the field map ResultSet, you must also list the location name that is the source of the data field, and the table column name where the data is logged. Because the extended service tracking function logs and tracks specific data for a specific service call, you can generate customized reports for access and usage information.

Caution: In field map ResultSets, nothing prevents you from mapping data fields to existing, standard SctAccessLog table columns. The extended service mapping occurs after the standard field data values are collected. Consequently, you can override any of the standard table column fields.

For example, the service you are logging might carry a specific user name (such as, MyUserName=john) in a data field. You could use the extended tracking function to override the contents of the sc_scs_dUser column. In this case, you simply combine MyUserName and sc_scs_dUser and use them as the data field, location, and table column set in the field map ResultSet.

Therefore, it remains your responsibility to ensure that the data being logged is a reasonable fit with the SctAccessLog column type.

Note: For more information about the SctAccessLog table and the general purpose columns, see Combined Output Table (page 2-8). For more detailed information about the SctServiceFilter.hda file, the extended service call tracking function, and ResultSet configuration, see Chapter 5 (*Service Call Configuration)*.



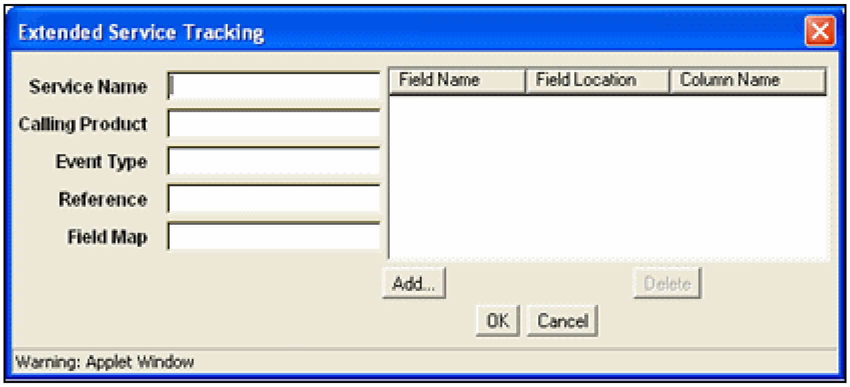
Services

Service Name	Calling Product	Event Type	Reference	Field Map
SCT_LOG_EVENT	Content Tracker	Log Service Event		
GET_FILE	Core Server	Content Access		
GET_FILE_BY_NAME	Core Server	Content Access		
GET_DYNAMIC_URL	Core Server	Content Access		
GET_DYNAMIC_CONVERSION	Core Server	Content Access		
GET_EXTERNAL_DYNAMIC_...	Core Server	Content Access		
GET_ARCHIVED_FILE	Core Server	Content Access		

Feature	Description
Services list	Provides the names and result set values of each service logged by Content Tracker.
Add button	Opens the Extended Services Tracking Screen (page 3-18).
Edit button	Opens the Extended Services Tracking Screen (page 3-18). Applicable fields are populated with the current result set values.
Delete button	Deletes the selected service.

Extended Services Tracking Screen

Use the Extended Services Tracking screen to configure (add, edit, or delete) the service entries contained in the ServiceExtraInfo ResultSet that is included in the SctServiceHandler.hda file. To access this screen, click the **Add** button on the Services Tab (page 3-16).

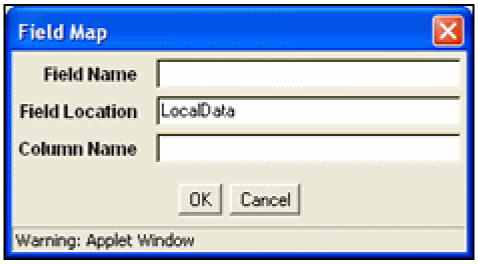


<table>
<tr><th>Feature</th><th>Description</th></tr>
<tr><td colspan="2"><b>Note:</b> Content Tracker does not perform error checking (such as field type or spelling verification) for the extended services tracking function in the Data Engine Control Center. Errors are not generated until you perform a reduction. The fields on the Extended Services Tracking screen are case-sensitive. Therefore, be careful to enter proper service entry field values—especially service call names. Ensure that all field values are spelled and capitalized correctly.</td></tr>
<tr><td colspan="2"><b>Note:</b> The ServiceExtraInfo ResultSet in the SctServiceHandler.hda file contains a service entry for each service to be logged. If you select an existing service entry from the Service Name field, the applicable fields will already be populated with the existing service entry’s field values. However, if you are adding a service, the new service entry will use the values you enter for the Service Name field, Calling Product field, Event Type field, Reference field, and Field Map field.
For more information about configuration requirements for service entries and field map ResultSets, see Service Call Configuration File Contents (page 5-5).</td></tr>
<tr><td>Service Name field</td><td>The name of the service to be logged. For example, GET_FILE. If the line for the service name is not included in the service entry, the service is not logged.</td></tr>
<tr><td>Calling Product field</td><td>An arbitrary string. It is generally set to “Core Server” for all standard Content Server entries.</td></tr>
<tr><td>Event Type field</td><td>An arbitrary string. It is generally set to “Content Access” for all standard Content Server entries.</td></tr>
<tr><td>Reference field</td><td>Used to set the sc_scs_reference field in the SctAccessLog table. If blank, the internal getReference logic is used.</td></tr>
</table>

Feature	Description
Field Map field	The name of the field map ResultSet that is added to the SctServiceFilter.hda file. This field is only required if you plan to use the extended service call tracking function. This function enables you to log arbitrary DataBinder field information to one or more of the general purpose columns in the SctAccessLog table. **Note:** To help you design your field map, a configuration variable can be set that writes out the object when the service is called. This enables you to see what data is available at the time the event is recorded. For more information, see SctDebugServiceBinderDumpEnabled (page B-3).
Field Name/Field Location/Column Name list	Lists each set of data field, location, and table column names associated with a field map ResultSet. The Field Map Screen (page 3-21) field values are used to populate this list.
Add button	Opens the Field Map Screen (page 3-21).
Delete button	Deletes the selected field map ResultSet.
OK button	Saves the field values and adds or updates the service entry to the ServiceExtraInfo ResultSet. Clicking OK will also add the field map ResultSet to the SctServiceFilter.hda file if you have created one—refer to Field Map Screen (page 3-21).
Cancel button	Closes the Extended Services Tracking screen without saving any changes.

Field Map Screen

Use the Field Map screen to configure the field map ResultSets that are linked to the service entries and included in the SctServiceHandler.hda file. To access this screen, click the **Add** button on the Extended Services Tracking Screen (page 3-18).



Field Name
Field Location: LocalData
Column Name

Feature	Description
Note: Content Tracker does not perform error checking (such as field type or spelling verification) for the extended services tracking function in the Data Engine Control Center. Errors are not generated until you perform a reduction. The fields on the Field Map screen are case-sensitive. Therefore, be careful to enter proper field and column names. Ensure that all field values are spelled and capitalized correctly.	
Note: To use the extended service call tracking function, you must map a service entry to a field map ResultSet in the SctServiceHandler.hda file. The value in the Field Map field on the Extended Services Tracking Screen (page 3-18) is the name of the field map ResultSet.	
Field Name field	The name of the data field in the Content Server service DataBinder whose data values are logged to a general purpose column in the SctAccessLog table. The target column is specified in the Column Name field.
Important: Values in field map ResultSets are not available for use as fields to be logged.	

Feature	Description
Field Location field	The section in the Content Server service DataBinder where the data field to be logged is located: Three values are supported: LocalData—the default value. Environment BinderResultSet—returns a comma-delimited string that contains all the values in the ResultSet. Due to space limitations in the SctAccessLog table, this value is useful for small ResultSets. The size is restricted to 255 characters, allowing for the comma’s, etc.
Tech Tip: To accommodate more than 255 characters, you can enlarge/redefine the SctAccessLog table columns using standard database tools. For example, if you open up extField_3 to 2047, then it will hold the equivalent amount of data. However, most databases have page size limitations that you need to consider. Also, there is the practical consideration in that SQL does not parse strings efficiently.	
Column Name field	The specific general purpose column in the SctAccessLog table where data values from a specified DataBinder field are logged. The data field is specified in the Field Name field.
OK button	Saves the values and adds the field map ResultSet to the SctServiceFilter.hda file.
Cancel button	Closes the Field Map screen without saving the list of DataBinder field names and their corresponding SctAccessLog column names.
Tech Tip: Content Tracker provides a debugging configuration variable that, if enabled, configures the service handler filter to write out the service DataBinder objects into dump files. These can be used a diagnostic tools when you are developing field map screens. The dump files enable you to see what data is available at the time the particular service events are recorded. For more information, see SctDebugServiceBinderDumpEnabled (page B-3).	

Accessing the Data Engine Control Center

The Data Engine Control Center is used to enable, schedule, and monitor data collection and reduction.

To access the Data Engine Control Center:

1. Open the Content Tracker Administration page:

   Administration tray—Content Tracker Administration.

2. Scroll down and click the **Data Engine Control Center** icon.

   The Content Tracker Data Engine Control Center interface is displayed.

Enabling or Disabling Data Collection

When data collection is enabled, Content Tracker logs web traffic activity on the content server. By default, the Enable Data Collection check box is selected on the Collection tab of the Data Engine Control Center. Selecting this check box enables data collection. Clearing this check box disables data collection.

To enable or disable data collection:

1. Open the **Data Engine Control Center**—see Accessing the Data Engine Control Center (page 3-23).
2. On the Collection tab, select (to enable collection) or clear (to disable collection) the **Enable Data Collection** check box.
3. Click **OK**.

Important: After you click OK, do not immediately exit the applet. You must wait until the Updated Data Collection state confirmation message displays. Occasionally, this may take a few seconds. If you exit the applet before the confirmation message displays, the requested change(s) may not occur.

4. After the Updated Data Collection state confirmation message displays, click **OK**.
5. Restart the Content Server.

Important: Look carefully at the text above the check box to determine whether data collection is enabled or disabled.

- When enabled, the text reads "Data collection is enabled..."
- When disabled, the text reads "Data collection is not enabled..."

Running Data Reduction Manually

To manually reduce data:

1. Open the **Data Engine Control Center**—see Accessing the Data Engine Control Center (page 3-23).
2. On the Reduction tab, click (to highlight) the set of input data to reduce.
3. Click the **Reduce Data** button.

   A confirmation dialog box is displayed.
4. Click **Yes** to reduce the data.

   The Status will change from ‘ready’ to ‘running,’ and the Percent Done will display the progress of data reduction. When data reduction is complete, the time stamp will be displayed in When Finished, and the Cycle will display ‘recent.‘

Note: If you choose to reduce the current date's data, the data will be reduced, but the Cycle will continue to display the data set as ‘new.’

Setting Data Reduction to Run Automatically

To set data reduction to run automatically:

1. Open the **Data Engine Control Center**—see Accessing the Data Engine Control Center (page 3-23).
2. On the Schedule tab, select the **Scheduling Enabled** check box.
3. Select check boxes for the days when data collection will occur.
4. Select the hour and minute when data collection will occur.
5. Click **OK**.

Important: After you click OK, do not immediately exit the applet. You must wait until the Updated reduction scheduling information confirmation message displays. Occasionally, this may take a few seconds. If you exit the applet before the confirmation message displays, the requested change(s) may not occur.

6. After the Updated reduction scheduling information confirmation message displays, click **OK**.

   Data will be reduced automatically on the day(s) and time that you selected.

Deleting Data Files

Data files can be deleted during two cycles:

- ❖ Deleting Data Files in Any Cycle (page 3-25)
- ❖ Deleting Data Files in ‘Archive’ Cycle (page 3-25)

Deleting Data Files in Any Cycle

Caution: Deletion of data is permanent and cannot be undone within Content Tracker or Content Server.

1. Open the **Data Engine Control Center**—see Accessing the Data Engine Control Center (page 3-23).
2. On the Reduction tab, click (to highlight) the set of input data to delete.
3. Click the **Delete** button.

   A confirmation dialog box is displayed.
4. Click **OK** to delete the data.

   The selected data set is deleted, and is no longer displayed in the window

Deleting Data Files in ‘Archive’ Cycle

Caution: Deletion of data is permanent and cannot be undone within Content Tracker or Content Server.

1. Open the **Data Engine Control Center**—see Accessing the Data Engine Control Center (page 3-23).
2. On the Reduction tab, click the **Delete Archive** button.

   A confirmation dialog box is displayed.
3. Click **OK** to delete the data.

   All data sets in ‘archive’ cycle are deleted, and are no longer displayed in the window

Creating the Search Relevance Metadata Fields

Before you can implement the snapshot function, you must decide which custom metadata fields will be associated with each of the enabled activity metrics. Also, the custom metadata fields must already exist and must be of the correct type. Depending on which activity metrics you plan to enable, you must create one or more custom metadata fields using the following procedures, as applicable:

- Creating the Custom Metadata Field for the Last Access Metric (page 3-26)
- Creating the Custom Metadata Fields for the Short and Long Access Count Metrics (page 3-27)

Creating the Custom Metadata Field for the Last Access Metric

To create custom metadata fields to assign to the last access field:

1. Open the Content Tracker Administration page:

   Administration tray—Content Tracker Administration

2. Click the **Configuration Manager** icon.

   The Configuration Manager interface is displayed.

3. On the Information Fields tab, click **Add**.

   The Add Custom Info Field screen is displayed.

4. Enter the name of the metadata field to be assigned to the Last Access metric. For example, LastAccess.
5. Click **OK**.

   The Add Custom Info Field screen is displayed.

6. Select **Date** from the Field Type drop-down menu.

Note: Normally, you do not need to enter a value in the Default Value field. However, if you do not enter a value for this field and there is no specified default value, then the Last Access field is not populated until a content item has been checked in and a data reduction run. Some applications, however, need to have the Last Access field contain a valid value at all times. In this case, you will need to enter a value in the Default Value field that will ensure that the Last Access field is populated with the date and time of the content checkin. For more information, see Populating the Last Access Field Using the Using the Default Value (page 3-31).

Note: Field Type with a value of Date is the only required attribute for the last access custom metadata field. However, if you want the last access custom metadata field to be searchable, you must ensure that the Enable for Search Index check box is selected.

Indexing this custom metadata field is optional, although indexing makes searches on this field more efficient. Furthermore, indexing allows you to query the accumulated search relevance statistics and generate useful data. For example, you can create a list of content items ordered by their popularity, etc.

For more information about the advantages and disadvantages of indexing the search relevance metadata fields, see Snapshot Tab (page 3-11).

7. Click **OK**.

   The custom metadata field is added to the Field Info list on the Information Fields tab.

8. Click **Update Database Design** to validate the current database and add the custom metadata field to the system.

Creating the Custom Metadata Fields for the Short and Long Access Count Metrics

To create custom metadata fields to assign to the short and long access fields:

1. Open the Content Tracker Administration page:

   Administration tray—Content Tracker Administration

2. Click the **Configuration Manager** icon.

   The Configuration Manager interface is displayed.

3. On the Information Fields tab, click **Add**.

   The Add Custom Info Field screen is displayed.

4. Enter the name of the metadata field to be assigned to the Short or Long Access Count metric. For example, ShortAccess or LongAccess.

5. Click **OK**.

   The Add Custom Info Field screen is displayed.

6. Select **Integer** from the Field Type drop-down menu.

Note: Field Type with a value of Integer is the only required attribute for the Short and Long Access Count custom metadata field. However, if you want the Short and Long Access Count custom metadata fields to be searchable, you must ensure that the Enable for Search Index check box is selected for both.

Indexing these custom metadata fields is optional, although indexing makes searched on these fields more efficient. Furthermore, indexing allows you to query the accumulated search relevance statistics and generate useful data. For example, you can create a list of content items ordered by their popularity, etc.

For more information about the advantages and disadvantages of indexing the search relevance metadata fields, see Snapshot Tab (page 3-11).

7. Click **OK**.

   The custom metadata field is added to the Field Info list on the Information Fields tab.

8. Click **Update Database Design** to validate the current database and add the custom metadata field to the system.

Enabling the Snapshot function and the Activity Metrics Options

By default, the snapshot function and activity metrics are disabled. To use these optional features, you must first enable the snapshot post-processing function which activates the activity metrics choices. Then, you can selectively enable the desired activity metrics and assign their preselected custom metadata fields.

To enable the snapshot function and activate the activity metrics:

1. Open the **Data Engine Control Center**—see Accessing the Data Engine Control Center (page 3-23).
2. Click the **Snapshot** tab.
3. Select the Enable Snapshot post-processing check box.

   The snapshot function is enabled and the activity metrics options are activated.

4. Click **OK**.

   A confirmation dialog box is displayed.

5. Click **OK**.

   The snapshot state and Content Tracker’s configuration file (sct.cfg) are updated.

Note: To verify the snapshot function and activity metrics have been enabled, you can access the Content Tracker's sct.cfg file in the following directory:

<*install_dir*>/custom/ContentTracker/resources/sct.cfg

Note: Optionally, you can manually enable the snapshot function and activate the activity metrics options. For more detailed information about the specific snapshot configuration variables and how to manually edit them, see Configuration Variables (page A-2) and Manually Setting Content Tracker Configuration Variables (page A-7), respectively.

Linking Activity Metrics Functions to Search Relevance Metadata Fields

After the activity metrics options have been activated, they must be individually selected to enable them. Enabling the activity metrics also activates their corresponding custom metadata fields.

To enable the activity metrics and activate their corresponding custom metadata fields:

1. Open the **Data Engine Control Center**—see Accessing the Data Engine Control Center (page 3-23).
2. Click the **Snapshot** tab.

   The snapshot function must be enabled; otherwise the activity metrics options are not activated. See Enabling the Snapshot function and the Activity Metrics Options (page 3-28).
3. Select one or more of the activity metric check boxes.

   Each selected activity metric is enabled and each corresponding custom metadata field is activated.
4. In the Field field, enter the internal name of the custom metadata field to be linked to the activity metric. For example, xLastAccess, xShortAccess, or xLongAccess.
5. For the Short and Long Access Counts, enter the applicable Interval amounts in days. For example, 7 days for the Short Access Count and 28 days for the Long Access Count.
6. Click **OK**.

   A confirmation dialog box is displayed.

7. Click **OK**.

   The snapshot state and Content Tracker’s configuration file (sct.cfg) are updated.

Note: Content Tracker performs minimal error checking on the activity metrics field names. Be aware that the fields on the Snapshot tab are case-sensitive. It is important that all field values are spelled and capitalized correctly. For more information about the specific Content Tracker error checks for the snapshot function, see Snapshot Tab (page 3-11).

Note: To verify that the activity metrics are linked to the appropriate custom metadata fields, you can access the Content Tracker’s sct.cfg file in the following directory:

<*install_dir*>/custom/ContentTracker/resources/sct.cfg

Note: Optionally, you can manually link the activity metrics to their respective custom metadata fields. For more detailed information about the specific activity metrics configuration variables and how to manually edit them, see Configuration Variables (page A-2) and Manually Setting Content Tracker Configuration Variables (page A-7), respectively.

Setting a Checkin Time Value for the Last Access Metadata Field

The Last Access Date field is normally updated by Content Tracker when a managed object is requested by a user and a data reduction run. Therefore, the Last Access field in Content Server’s DocMeta database table may be empty (NULL) until the next data reduction is run.

However, some applications require that the date and time of content checkin be recorded immediately in the Last Access field. To accommodate this requirement, the Last Access field must be populated with an appropriate date and time value. Content Tracker provides several methods to populate the Last Access field:

- Populating the Last Access Field Using the Using the Default Value (page 3-31)
- Populating the Last Access Field Using the Autoload Option (page 3-32)
- Populating the Last Access Field for Batchloads and Archives (page 3-32)

Populating the Last Access Field Using the Using the Default Value

Normally, you do not need to enter a value in any field that has a Default value. However, if you do not enter a value for the Last Access field, and there is no specified default value, then the field is not populated when a content item is checked in. The checkin date or most recent access date is only recorded once a data reduction has been run.

To support the requirements for particular applications, you can use the Autoload option to backfill the Last Access field for existing content—see Populating the Last Access Field Using the Autoload Option (page 3-32). For all future content item checkins, you can configure the Last Access custom metadata field by setting the Default Value field.

The value you enter must be a function or an expression that will cause the field to be populated with the date and time of content checkin. This ensures that the current date and time is automatically entered into the Last Access field.

To populate the Last Access field using the Default Value field:

1. Open the Content Tracker Administration page:

   Administration tray—Content Tracker Administration

2. Click the **Configuration Manager** icon.

   The Configuration Manager interface is displayed.

3. On the Information Fields tab, select the custom metadata field that you have linked to the Last Access metric and click **Edit**.

   The Edit Custom Info Field screen is displayed.

Note: The Last Access custom metadata field must already exist. If not, you must create it and link it to the Last Access activity metric function. See Creating the Custom Metadata Field for the Last Access Metric (page 3-26)

4. In the Default Value field, enter an expression that will cause the field to be populated with the date and time of content checkin.

   For example, you could specify a default value of *<$dateCurrent()$>* to cause the Last Access field to be populated with the current checkin date and time.

5. Click **OK**.

   The Last Access custom metadata field is updated.

6. Backfill the Last Access field in for existing content—see Populating the Last Access Field Using the Autoload Option (page 3-32).

Populating the Last Access Field Using the Autoload Option

The Autoload option on the Snapshot tab allows you to retroactively replace NULL values in the Last Access field with the current date and time. The only DocMeta records that are affected using the Autoload option are those where the Last Access metadata field is empty (NULL).

To populate the Last Access field using the Autoload option:

1. Open the **Data Engine Control Center**—see Accessing the Data Engine Control Center (page 3-23).
2. Click the **Snapshot** tab.

   The snapshot function must be enabled; otherwise the activity metrics options are not activated. See Enabling the Snapshot function and the Activity Metrics Options (page 3-28).
3. Select the **Enable Last Access updates** check box.
4. Link the Last Access metric to the applicable custom metadata field—see Linking Activity Metrics Functions to Search Relevance Metadata Fields (page 3-29).
5. Select the **Autoload** check box.
6. Click **OK**.

   A confirmation dialog box is displayed and the current date and time are inserted into the applicable Last Access fields (those with NULL values) in Content Server's DocMeta database table.

Tech Tip: By default, the Autoload query sets the Last Access metadata field to the current date and time. However, you may customize the query to set the Last Access field to dCreateDate, dReleaseDate, or any other time that meets the needs of your application. See Customizing the Autoload Option SQL Query (page A-9).

Populating the Last Access Field for Batchloads and Archives

To ensure proper retention of archived and batchloaded content, you must set the Last Access field date for the import/insert. Otherwise, the access date for these content items will be NULL, and retention based on this field will fail.

Important: The Last Access date can be used in conjunction with Retention Manager to maintain retention schedules. Ensuring that this field is set properly during batchloads and archives is important for the success of the retention. Please consider carefully what date is most reflective of when the content was last accessed. For example, an import of 1998 data is probably better tagged with that date than the date you perform the import.

The name of the Last Access field is based on the name you specified in Configuration Manager—see Creating the Custom Metadata Field for the Last Access Metric (page 3-26). In the case of Last Access, xLastAccess would be used in the import/insert—see Enable Last Access updates check box and corresponding Field metadata field (page 3-13).

To populate the Last Access field using Content Server's Batch Loader:

1. Access the Batch Loader.
2. Create a file record that establishes an appropriate Last Access date. The following is an example of an applicable file record:

```
# This is a comment
Action=insert
dDocName=Sample1
dDocType=ADACCT
xLastAccess=5/1/1998
dDocTitle=Batch Load record insert example
dDocAuthor=sysadmin
dSecurityGroup=Public
primaryFile=links.doc
dInDate=8/15/2001
<<EOD>>
```

3. Run the Batch Loader to process the file record.

Note: Refer to the *Content Server Managing System Settings and Processes* guide for more detailed information.

Editing the Snapshot Configuration

To modify the current snapshot activity metrics settings:

1. Open the **Data Engine Control Center**—see Accessing the Data Engine Control Center (page 3-23).
2. Click the **Snapshot** tab.

   The snapshot function must be enabled; otherwise the activity metrics options are not activated. See Enabling the Snapshot function and the Activity Metrics Options (page 3-28).
3. Make the necessary changes in the activity metrics fields.
4. Click **OK**.

   A confirmation dialog box is displayed.
5. Click **OK**.

   The snapshot state and Content Tracker's configuration file (sct.cfg) are updated.

Note: Content Tracker performs minimal error checking on the activity metrics field names. Be aware that the fields on the Snapshot tab are case-sensitive. It is important that all field values are spelled and capitalized correctly. For more information about the specific Content Tracker error checks for the snapshot function, see Snapshot Tab (page 3-11).

Note: To verify the modified values of the snapshot and activity metrics configuration variables, you can access the Content Tracker's sct.cfg file in the following directory:

<*install_dir*>/custom/ContentTracker/resources/sct.cfg

Note: Optionally, you can manually edit the configuration settings for the snapshot activity metrics. For more detailed information about the specific activity metrics configuration variables and how to manually edit them, see Configuration Variables (page A-2) and Manually Setting Content Tracker Configuration Variables (page A-7), respectively.

Adding/Editing Service Entries

To add or edit a service:

1. Open the **Data Engine Control Center**—see Accessing the Data Engine Control Center (page 3-23).
2. Click the **Services** tab.
3. Click **Add** to create a new service entry.

   Or, select an existing service entry from the Service Name list and click **Edit**.

   The Extended Service Tracking screen is displayed. The fields are empty if you are adding a new service entry.

   If you are editing an existing service entry, the fields are populated with those values. In this case, the Service Name field is deactivated.
4. Enter or modify the applicable field values (except in the Field Map field).

   If you want to link this service entry to a field map ResultSet, enter the applicable name in the Field Map field. Then, see the procedure for Adding Field Map ResultSets and Linking Them to Service Entries (page 3-36).
5. Click **OK**.

   A confirmation dialog box is displayed.
6. Click **OK**.

   The Extended Service Tracking screen closes and the Services tab on the Data Engine Control Center is displayed.

   If you added a new service entry, it is included in the Services list. If you edited an existing service entry, the updated field values are included in the Services list.

   The services state and Content Tracker’s SctServiceFilter.hda are updated.

Note: Content Tracker does not perform error checking (such as field type or spelling verification) for the extended services tracking function in the Data Engine Control Center. Errors are not generated until you perform a reduction. These fields are case-sensitive.Therefore, if you are adding new services or editing existing services, be careful to enter the proper service call names. Ensure that all field values are spelled and capitalized correctly.

Note: To verify that the service entry's values are added to the SctServiceFilter.hda file or that the existing service entry's values are properly modified, you can access Content Tracker's SctServiceFilter.hda file in the following directory:

<*install_dir*>/custom/ContentTracker/resources/SctServiceFilter.hda

Note: Optionally, you can manually add or edit services. For more detailed information about service entries in the SctServiceFilter.hda file and how to manually edit them, see About the Service Call Configuration File (page 5-2) and Manually Editing the SctServiceFilter.hda File (page 5-9), respectively.

Adding Field Map ResultSets and Linking Them to Service Entries

To implement the extended service call tracking function, you need to link service entries to field map ResultSets in the SctServiceFilter.hda file.

To add a field map ResultSet and link it to a service entry:

1. Open the **Data Engine Control Center**—see Accessing the Data Engine Control Center (page 3-23).
2. Click the **Services** tab.
3. Select the desired service entry from the Service Name list.

   If you need to add a new service entry, see Steps 3 through 6 in the procedure for Adding/Editing Service Entries (page 3-35).
4. Click **Edit**.

   The Extended Service Tracking screen is displayed and the fields are populated with the selected service entry's values. In this case, the Service Name field is deactivated. If necessary, you can edit this service entry's values now in addition to adding the field map ResultSet.

   If this service is already linked to a field map ResultSet, the name is listed in the Field Map field and one or more data field, location, and table column set are listed in the Field Name, Field Location, and Column Name fields. If you want to edit or delete existing data field, location, and table column sets, see the procedure for Editing Field Map ResultSets (page 3-38).

5. If the selected service is already linked to a field map ResultSet, skip this step. However, if the selected service is not linked to a secondary ResultSet, the Field Map field is empty. Enter the name of the field map ResultSet.
6. Click **Add**.

   The Field Map screen is displayed.
7. Enter the appropriate values in the fields.
8. Click **OK**.

   The Field Map screen closes and the values are added to the Field Name and Column Name fields.

Note: If you need to add more than one data field, location and table column set, repeat Steps 6 through 8 as necessary.

9. Click **OK**.

   A confirmation dialog box is displayed.

   The Extended Service Tracking screen closes and the Services tab on the Data Engine Control Center is displayed.

   The services state and Content Tracker's SctServiceFilter.hda file are updated.
10. Click **OK**.

Note: Content Tracker does not perform error checking (such as field type or spelling verification) for the extended services tracking function in the Data Engine Control Center. Errors are not generated until you perform a reduction. These fields are case-sensitive. Therefore, if you are adding new field map ResultSets or editing existing field map ResultSets, be careful to enter the proper DataBinder field names and SctAccessLog table column names. Ensure that all field values are spelled and capitalized correctly.

Note: To verify that the field map ResultSet values are added to the service call configuration file or that the values are properly modified, you can access the Content Tracker's SctServiceFilter.hda file in the following directory:

<*install_dir*>/custom/ContentTracker/resources/SctServiceFilter.hda

Note: Optionally, you can manually add field map ResultSets and manually link them to service entries. For more detailed information about service entries and field map ResultSets in the SctServiceFilter.hda file and how to manually edit them, see About the Service Call Configuration File (page 5-2) and Manually Editing the SctServiceFilter.hda File (page 5-9), respectively.

Editing Field Map ResultSets

To edit a field map ResultSet:

1. Open the **Data Engine Control Center**—see Accessing the Data Engine Control Center (page 3-23).

2. Click the **Services** tab.

3. Select the desired service entry from the Service Name list.

4. Click **Edit**.

   The Extended Service Tracking screen is displayed and the fields are populated with the selected service entry’s values. In this case, the Service Name field is deactivated. If necessary, you can also edit the other field values for this service entry in addition to editing the field map ResultSet.

5. To edit the secondary ResultSet, you can either:

   ❖ Add one or more data field, location, and table column sets. See Steps 6 through 8 in the procedure for Adding Field Map ResultSets and Linking Them to Service Entries (page 3-36).

   ❖ Delete one or more data field, location, and table column sets as follows:

      a. Select the field, location, and table column set to be deleted.

      b. Click **Delete**.

6. Click **OK**.

   A confirmation dialog box is displayed.

   Click **OK**.

   The Extended Service Tracking screen closes and the Services tab on the Data Engine Control Center is displayed. The services state and Content Tracker’s SctServiceFilter.hda file are updated.

Note: Content Tracker does not perform error checking (such as field type or spelling verification) for the extended services tracking function in the Data Engine Control Center. Errors are not generated until you perform a reduction. These fields are case-sensitive. Therefore, if you are editing field map ResultSets by adding one or more data field, location, and table column sets, be careful to enter the proper data field names, location names, and SctAccessLog table column names. Ensure that all field values are spelled and capitalized correctly.

Note: To verify the modified values of the data field, location, and table column sets in field map ResultSets, you can access the Content Tracker's SctServiceFilter.hda file in the following directory:

<*install_dir*>/custom/ContentTracker/resources/SctServiceFilter.hda

Note: Optionally, you can manually modify the values of the data field, location, and table column sets in field map ResultSets. For more detailed information about field map ResultSets in the SctServiceFilter.hda file and how to manually edit them, see About the Service Call Configuration File (page 5-2) and Manually Editing the SctServiceFilter.hda File (page 5-9), respectively.

Deleting Service Entries

To delete a service:

1. Open the **Data Engine Control Center**—see Accessing the Data Engine Control Center (page 3-23).
2. Click the **Services** tab.
3. In the Services list, select the service entry that you want to delete.

4. Click **Delete**.

   You are asked to verify your request to delete service logging for this service entry.

5. Click **Yes**.

   The selected service entry is deleted from the Services list and removed from the SctServiceFilter.hda file.

Note: To verify that the service entry has been deleted, you can access the Content Tracker's SctServiceFilter.hda file in the following directory:

<*install_dir*>/custom/ContentTracker/resources/SctServiceFilter.hda

Note: Optionally, you can manually delete specific service entries. For more information, see Manually Editing the SctServiceFilter.hda File (page 5-9).

Deleting Field Map ResultSets

To delete a field map ResultSet:

1. Open the **Data Engine Control Center**—see Accessing the Data Engine Control Center (page 3-23).
2. Click the **Services** tab.
3. In the Services list, select the service entry that is linked to the field map ResultSet that you want to delete.
4. Click **Edit**.

   The Extended Service Tracking screen is displayed and the fields are populated with the selected service entry's values.

5. Remove the field map ResultSet name from the Field Map field.
6. Select a data field, location, and table column set and click **Delete**.

   The data field, location, and table column set is removed from the list. Repeat this step for each data field, location, and table column set (as necessary).

7. Click **OK**.

   The field map ResultSet is removed from the SctServiceFilter.hda file. It is no longer linked to the service entry.

Note: To verify that the field map ResultSet has been deleted, you can access the Content Tracker's SctServiceFilter.hda file in the following directory:

<*install_dir*>/custom/ContentTracker/resources/SctServiceFilter.hda

Note: Optionally, you can manually delete field map ResultSets. For more information, see Manually Editing the SctServiceFilter.hda File (page 5-9).

Chapter 4

REPORT GENERATION

OVERVIEW

This section covers the following topics:

Concepts

Interface

Tasks

About Content Tracker Reports

Content Tracker Reports uses the captured and reduced data to generate reports that outline the usage history of particular pieces of content. You can use the pre-defined reports that are provided or create custom queries for the information to be tracked. Optionally, you can use any external commercial reporting tool—External Report Generator (page 4-34).

Reports can be derived from a variety of criteria, including specific users, groups of users, and any set of content that can be defined by a query or group of metadata values. Based on the variables in the system (such as number of users, amount of content, metadata count, etc.), Content Tracker Reports enables hundreds of key metrics to be included in reports. Specialized reports enable you to understand and disclose which content is most relevant to users.

GENERAL CONSIDERATIONS

This section covers the following topics:

- Oracle and DB2 Case Sensitivity (page 4-3)
- Access Control Lists and Content Tracker Reports Secure Mode (page 4-4)
- Compatibility of Existing SQL Reports: Content Tracker Reports 7.0 or earlier (page 4-4)

Oracle and DB2 Case Sensitivity

If Oracle or DB2 is used as the Content Server database, metadata values are case sensitive and care must be used when entering the content metadata values in the applicable query report criteria. As a result, depending on how a value is entered in the corresponding field, Content Tracker Reports may not return all possible matching files.

With either an Oracle or DB2 Content Server database, values must be entered exactly as they are entered in the Content Server. Therefore, depending on the lettering structure of the values in the Content Server, the values entered in the query metadata fields will need to be entered in all lowercase letters, all uppercase letters or mixed-case letters. Otherwise, Content Tracker Reports will not return all of the matching files.

For example, if the content type in the Oracle or DB2 Content Server database is *AdAcc* but the user enters it in the query field as *adacc, ADACC,* or *Adacc*, Content Tracker Reports will not return any results. In this case, the content type metadata value must be entered using mixed-case letters. This is true for all of the metadata fields in each of the pre-defined query reports.

Access Control Lists and Content Tracker Reports Secure Mode

The security checks preference variable (SctrEnableSecurityChecks) is set when you install the Content Tracker Reports component. Essentially, this preference variable enables you to select one of two security modes: secure and non-secure. The security checks preference provides the option to employ individual user role and account information to restrict the visibility of content item information in report results.

This means that you control what content items (and, subsequently, the metadata) that users can see in their generated reports. Ideally, users should not be able to see anything through Content Tracker Reports that they couldn't find via a Content Server search. Therefore, if you select the secure mode, the information in any generated report will be filtered based on the user’s role and account privileges.

However, if you have enabled Access Control Lists (ACLs) on your Content Server instance, the secure mode option in Content Tracker Reports does not work. During installation, you must leave the security checks preference check box blank. This means that on an ACL-based system, the secure mode must be disabled. In this case, it is possible for users other than a system administrator to see information about content items that they would not otherwise be authorized to access and view.

Note: For more detailed information about the security checks installation preference and how it will affect the report queries and report results, see Security Checks and Query Results (page 4-23). For specific information about the installation preference prompt and the two security checks options, refer to the *Content Tracker Installation Guide*.

Compatibility of Existing SQL Reports: Content Tracker Reports 7.0 or earlier

Content Tracker 7.5 and later employ major architectural revisions of both the Content Tracker and the Content Tracker Reports components. Because of the significant schema and design changes to the components, there is no feasible way to guarantee that SQL report queries used with 7.0 and earlier versions of Content Tracker and Content Tracker Reports are compatible with the 7.5 and later versions.

However, the following list details the changes that were made and the reasons for them. This information should ensure that most users will experience few problems in adapting their queries to accommodate the new architectural and schema changes.

- Access information gathering has shifted from the web server to Content Server which makes Content Tracker independent of web server type. This eliminates the need for Content Tracker to depend on the web server to write an event log in a particular format.
- As a result of eliminating web server dependencies, installing the Content Tracker and Content Tracker Reports components is greatly simplified. Furthermore, the number and type of supported server configurations has expanded considerably. For example, Content Tracker now supports installations with multiple web servers and installations in which the web server(s) does not have direct access to the Content Server storage volume.
- Content Tracker no longer logs content item metadata so the content metadata tables are no longer generated. The standard Content Server metadata tables are used for content item metadata. This reduces the processing that needs to occur. It also ensures that all metadata is accessible to Content Tracker and that metadata values are up-to-the-minute current and accurate. Additionally, eliminating redundant output tables also stops redundant content metadata logging.
- Content Tracker continues to log user metadata and the applicable Content Tracker metadata tables in the database are updated during the reduction process.This ensures historically accurate user metadata.
- Static URL references are collected and logged by the web server filter plugin and Content Server service calls are logged by the service handler filter. Both types of event details are recorded in a table (SctAccessLog) which replaces the SctEnhanced table. used in Content Tracker v7.0 and earlier.
- The structural changes in Content Tracker v7.5 and later have increased the scope and improved the granularity of managed content usage reporting. This is due to eliminating the use of web server log files, expanding the collection of both static URL accesses and Content Server service calls, and ensuring that Content Tracker has access to all metadata values (both user and content item).

  Instead of reporting on page/object accesses at the URL level, Content Tracker can now track activity at the template/fragment level. This enables users to see how the various content item elements are being used. This means that Content Tracker can perform more interesting analysis on Site Studio sites and other applications that don’t use the Content Server’s native interface.

- Search results can incorporate specific data about user accesses. Custom metadata fields are used to record user access activity. Tracking results included in reports can reflect the popularity of certain content items or provide access data during specific time periods. Access tracking can include both internal and external users.
- The security checks feature optionally enforces user role and account privilege verification. When enabled, the security checking function filters the data based on role/account permissions of the user requesting the report. Consequently, different users may see different results after generating the same report. Otherwise, default SQL report queries generate reports with identical data for all users without regard for their user/account memberships.

Pre-Defined Query Reports

This section covers the following topics:

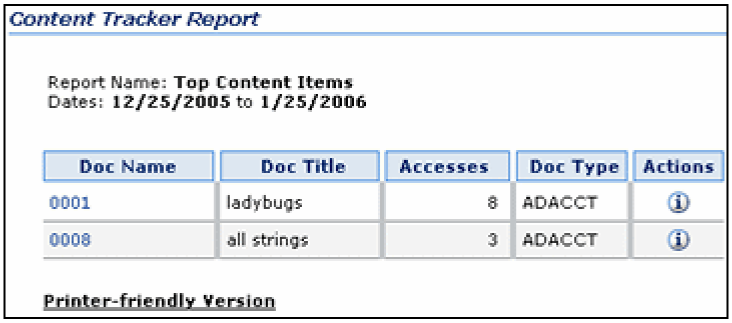
- Default Report Format (page 4-7)
- Content Dashboard Feature (page 4-8)
- Drill Down Report Feature (page 4-9)
- Content Tracker Report Generator Main Page (page 4-10)
- Generating Reports (page 4-13)
- Accessing Drill Down Reports (page 4-14)
- Accessing Reports from the Information Page (page 4-14)
- Viewing Separated Access Results (page 4-15)
- Viewing Combined Access Results (page 4-15)

Default Report Format

Each report produced using the Content Tracker Report Generator main page has the same general format and visual layout. The following is the Top Content Items report that is selected by default when the Content Tracker Report Generator main page is accessed. The information provided by the reports is extracted from the reduced data in the SctAccessLog database table and other Content Server database tables, as necessary.

Note: Only users that physically open a content item are included in the Content Tracker Report Generator's compiled results. The opened content item can be the web location file (the absolute path to the content item), an HTML version (by using Dynamic Converter), or the actual native file. Users that open only the Content Information page are not included in the tracked data.

Note: There is generally a one-day delay from the time that a user accesses a content item until that information is included in the Content Tracker Report Generator's access history results. The information must first be accumulated by Content Tracker and then undergo a data reduction cycle. Thus, the content item access history results are derived from the reduced data in the SctAccessLog and other Content Server database tables. Manually reducing the data immediately updates the database tables and, subsequently, the generated query reports will also display the updated information. For more information about the reduction process, see Reduction Tab (page 3-4).



Content Tracker Report

Report Name: **Top Content Items**
Dates: **12/25/2005** to **1/25/2006**

Doc Name	Doc Title	Accesses	Doc Type	Actions
0001	ladybugs	8	ADACCT	ⓘ
0008	all strings	3	ADACCT	ⓘ

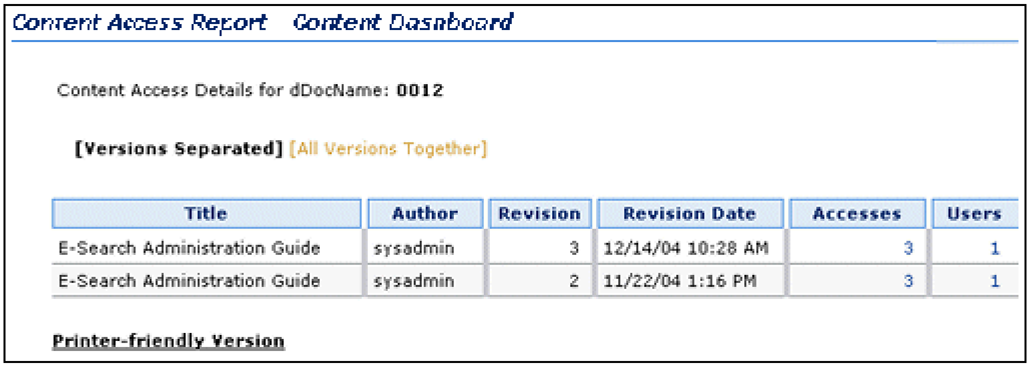
Printer-friendly Version

Field	Description
Report Name field	The name of the selected query report.
Dates field	The dates entered in the Start Date and End Date fields. If you did not enter specific dates, the default dates are used for the query.

Field	Description
Results table columns	Provide the relevant information for the selected report.
Printer-friendly Version link	Opens a new browser window and displays the report without the navigation trays.

Content Dashboard Feature

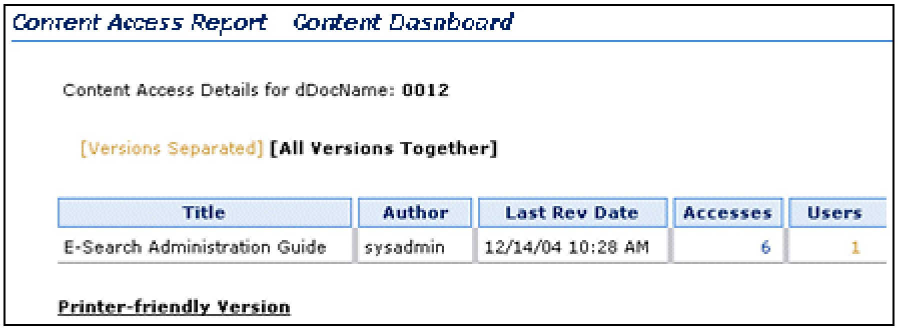
When a generated query report contains an active link to a specific content item, clicking the link displays the corresponding Content Dashboard. The content dashboard in the following screen capture shows that two versions of a particular content item were each accessed three times. In this view, the revision access results are shown individually.



Content Access Details for dDocName: 0012

Title	Author	Revision	Revision Date	Accesses	Users
E-Search Administration Guide	sysadmin	3	12/14/04 10:28 AM	3	1
E-Search Administration Guide	sysadmin	2	11/22/04 1:16 PM	3	1

If you click the All Versions Together link on the Content Dashboard, the access results for both versions are combined.

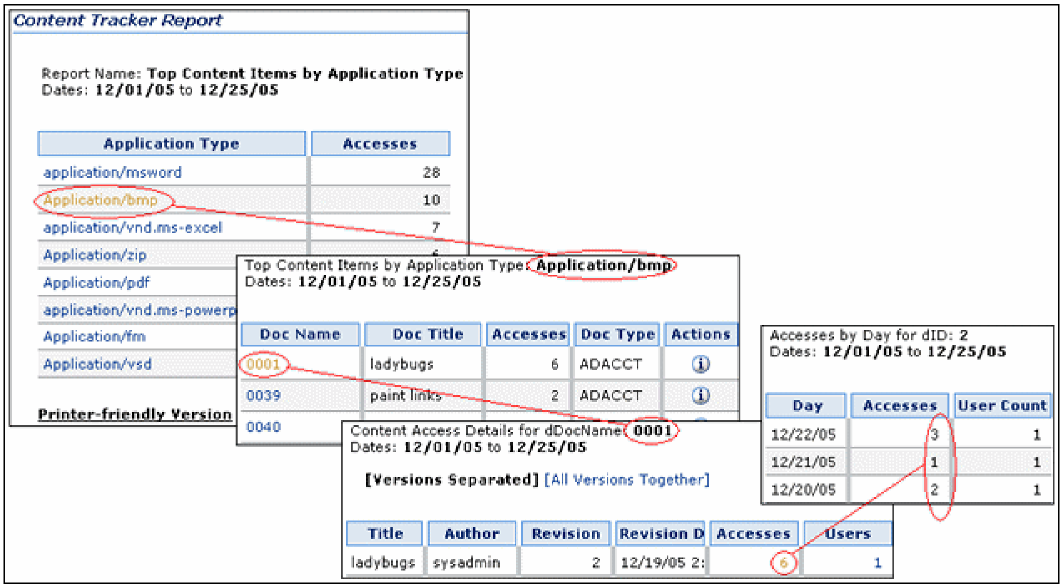


Content Access Details for dDocName: 0012

Title	Author	Last Rev Date	Accesses	Users
E-Search Administration Guide	sysadmin	12/14/04 10:28 AM	6	1

Drill Down Report Feature

There are various levels of report results that are generated for each pre-defined report. Depending on the search criteria you enter on the Content Tracker Report Generator main page, the results are filtered accordingly. The top level reports are summary reports and provide very general information. You can use the links on the top level reports to drill down to more specific information.



Content Tracker Report

Report Name: **Top Content Items by Application Type**
Dates: **12/01/05** to **12/25/05**

Application Type	Accesses
application/msword	28
Application/bmp	10
application/vnd.ms-excel	7
Application/zip	6
Application/pdf	
application/vnd.ms-powerp	
Application/fm	
Application/vsd	

Printer-friendly Version

Top Content Items by Application Type: **Application/bmp**
Dates: **12/01/05** to **12/25/05**

Doc Name	Doc Title	Accesses	Doc Type	Actions
0001	ladybugs	6	ADACCT	ⓘ
0039	paint links	2	ADACCT	ⓘ
0040				

Content Access Details for dDocName: **0001**
Dates: **12/01/05** to **12/25/05**

[Versions Separated] [All Versions Together]

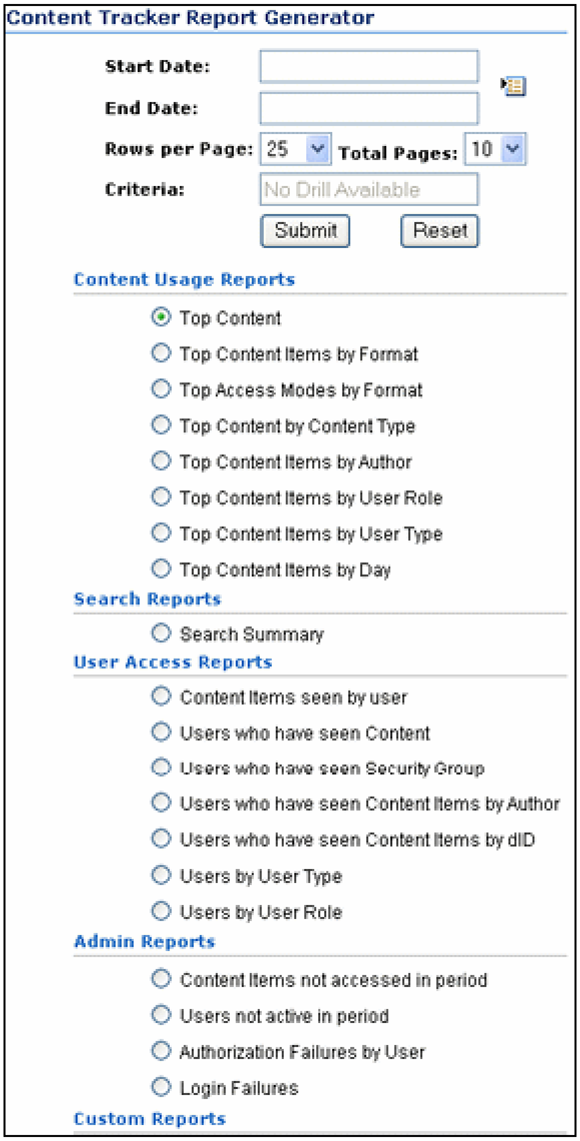
Title	Author	Revision	Revision D	Accesses	Users
ladybugs	sysadmin	2	12/19/05 2:	6	1

Accesses by Day for dID: **2**
Dates: **12/01/05** to **12/25/05**

Day	Accesses	User Count
12/22/05	3	1
12/21/05	1	1
12/20/05	2	1

Content Tracker Report Generator Main Page

The Content Tracker Reports component provides pre-defined queries organized into several main categories. These pre-defined reports are designed to answer the most common questions about system activity. The individual reports provide drill-downs reports based on applicable criteria. The Content Tracker Report Generator main screen is accessed by clicking the Content Tracker Reports link in the Administration tray.



Content Tracker Report Generator

Start Date:

End Date:

Rows per Page: 25 Total Pages: 10

Criteria: No Drill Available

Content Usage Reports

● Top Content
○ Top Content Items by Format
○ Top Access Modes by Format
○ Top Content by Content Type
○ Top Content Items by Author
○ Top Content Items by User Role
○ Top Content Items by User Type
○ Top Content Items by Day

Search Reports

○ Search Summary

User Access Reports

○ Content Items seen by user
○ Users who have seen Content
○ Users who have seen Security Group
○ Users who have seen Content Items by Author
○ Users who have seen Content Items by dID
○ Users by User Type
○ Users by User Role

Admin Reports

○ Content Items not accessed in period
○ Users not active in period
○ Authorization Failures by User
○ Login Failures

Custom Reports

Field	Description
Start Date field	Designates the beginning of a specific time period for records to be searched.
End Date field	Designates the end of a specific time period for records to be searched.
Date drop-down menu	**Yesterday**—Enters the date of the previous day into the Start Date and today's date into the End Date fields. **Latest Week**—Enters the beginning and ending dates of the previous week into the Start Date and End Date fields. **Latest Month**—Enters the beginning and ending dates of the previous month into the Start Date and End Date fields. **Latest Year**—Enters the beginning and ending dates of the previous year into the Start Date and End Date fields.
Rows per Page field	Specifies the number of results rows each page of the report can contain.
Total Pages field	Specifies the maximum number of results pages the report can contain.
Criteria field	Filters the search results and immediately accesses the applicable drill-down report. For example, if you are searching top content items by author, entering a specific author in the criteria field displays the drill-down report listing the content items authored by that user. Leaving this field blank causes the top level query to be executed, which provides a list of authors on the system.
Submit button	Generates and displays the selected report type.
Content Item Usage Reports	
Top Content	Lists the most frequently accessed content items in the system.
Top Content Items by Format	Lists the most frequently accessed content items by application type (i.e., pdf or txt).
Top Access Modes by Format	Lists the most frequently accessed content item application types (i.e., pdf or txt) by access mode.

Field	Description
Top Content by Content Type	Lists the most frequently accessed content items by content item type.
Top Content Items by Author	Lists the most frequently accessed content items by author.
Top Content Items by User Role	Lists the most frequently accessed content items by user role.
Top Content Items by User Type	Lists the most frequently accessed content items by user type.
Top Content Items by Day	Lists the most frequently accessed content items for each day.
Search Reports	
Search Summary	Lists the types of searches that have been done and the search criteria.
User Access Reports	
Content Items seen by user	Lists, by specific users, the number of content items most frequently opened by the indicated user.
Users who have seen Content	Lists, by specific content item, the users who have accessed the content item.
Users who have seen Security Group	Lists, by specific users, the content items most frequently accessed in specific security groups.
Users who have seen Content Items by Author	Lists, by specific users, the content items most frequently accessed that have been authored (checked in) by specific users.
Users who have seen Content Items by dID	Lists, by specific users, the content items most frequently accessed based on the internal content item identification numbers.

Field	Description
Users by User Type	Lists, by specific users, the content items most frequently accessed based on the user type.
Users by User Role	Lists, by specific users, the content items most frequently accessed based on the user role.
Admin Reports	
Content Items not accessed in period	Lists the content items that have not been accessed followed by the content items most frequently accessed.
Users not active in period	Lists the users that have not accessed content items followed by users that have accessed content items.
Authorization Failures by User	Lists the users that tried unsuccessfully to access content items to which they did not have authorization privileges.
Login Failures	Lists the users that tried unsuccessfully to log in to the system.
Custom Reports	
custom report name	Generates a customized report based on user-defined search queries.

Generating Reports

To generate a pre-defined or custom report:

1. Open the Content Tracker Report Generator main page by clicking the **Content Tracker Reports** link in the Administration tray.
2. Select the radio button of the desired report type.
3. Enter any desired search and filtering criteria in the applicable fields.
4. Click **Submit**.

   The selected report type is displayed.

Accessing Drill Down Reports

To access one or more drill down reports:

1. Generate a pre-defined or custom report. See Generating Reports (page 4-13).
2. After generating a pre-defined report, each line item result contains an active drill down report link. Click on the desired link.

   The selected drill-down report is displayed.

Note: Some reports contain multiple levels of drill down reports. For example, the Top Content Items report contains a DocName drill down report link. Clicking this link generates another report that displays the applicable content access details for the selected content item. In this report, two additional drill down reports are available: one for Accesses and another for Users.

Accessing Reports from the Information Page

The Access History Report for any content item can be generated from the Information page of that content item as follows:

1. Search for a content item and click the associated Info icon.

   The Content Information page is displayed.
2. Select **View Access History Report** from the Global Actions list.

   The most current Content Access Report for the content item is displayed.
3. On the Content Access Report, click the live **Accesses** link.

   The most current Accesses by Day report for the content item is displayed.
4. On the Content Access Report, click the live **Users** link.

   The most current Accesses by User report for the content item is displayed.

Viewing Separated Access Results

By default, the access results for multiple versions of a single content item are displayed individually on the Content Dashboard. To see the separated access results view of the Content Dashboard report:

1. Generate a content item-based query report from the Content Tracker Report Generator main page. See Generating Reports (page 4-13). For example, select Top Content (page 4-11) from the Content Items Usage Reports list (on the Content Tracker Report Generator main page) to generate the applicable report.
2. Select a content item from the results report and click the content identification number listed in the DocName column.

   The Content Dashboard for the selected content item is displayed. By default, this view shows the access results for each revision of the selected content item that was accessed. For more information, see Content Dashboard Feature (page 4-8).

Viewing Combined Access Results

To see the combined access results view of the content dashboard report:

1. Generate a content item-based query report from the Content Tracker Report Generator main page. See Generating Reports (page 4-13). For example, select Top Content (page 4-11) from the Content Items Usage Reports list (on the Content Tracker Report Generator main page) to generate the applicable report.
2. Select a content item from the results report and click the content identification number listed in the DocName column.

   The Content Dashboard for the selected content item is displayed.
3. Click the **All Versions Together** link.

   The resulting content dashboard view shows the combined access results for both versions.

CUSTOM QUERY REPORTS

In addition to the sample reports provided with Content Tracker Reports, you also have the option to create custom queries to track information.

This section covers the following topics:

- Considerations (page 4-16)
- Creating Custom Report Queries: Example (page 4-17)
- Custom Report Query Display Results (page 4-19)

Considerations

Before you begin creating your custom report query, you should be aware of some issues that may affect how you design your query. These issues include:

- Custom Report Queries and Oracle (page 4-16)
- Custom Report Queries and Extended Service Tracking (page 4-17)

Custom Report Queries and Oracle

If you are using Oracle and aliases to display the column names in the generated report, you must add the aliases to the following file:

<*install_dir*>/shared/config/resources/upper_clmns_map.htm

Example:

If your column headers are:

Name
Access_Date_GMT

Then, you must enter the following lines to the upper_clmns_map.htm file:

```
<tr>
<td>NAME</td>
<td>Name</td>
</tr>
<tr>
<td>ACCESS_DATE_GMT</td>
<td>Access_Date_GMT</td>
</tr>
```

Custom Report Queries and Extended Service Tracking

If you are using the extended service tracking function, you must be aware of what data values are written to specific columns in the SctAccessLog table before designing your SQL queries. In particular, you must be aware that the name of the service will always be logged to the sc_scs_idcService column. Therefore, you should include it as a qualifier in any query that uses the contents of the extended fields.

For more information about the extended service tracking function, see Services Tab (page 3-16) and About the Service Call Configuration File (page 5-2).

Creating Custom Report Queries: Example

This section provides an example that demonstrates how to create a non-secure custom report query. This particular query generates a report that lists users and their personal attributes. The data is derived from the Content Server's Users database table.

Note: The example in this section uses a non-secure query. Therefore, the generated report results can be viewed by any user regardless of their role and account privileges. All of the reports are generated using either non-secure of secure queries. The query selection is dependent on the security mode. For more detailed information about the optional security checks preference variable, see Security Checks and Query Results (page 4-23). If you want to create a secure report query, see Creating Secure Report Queries (page 4-32).

To create the custom users report:

1. Design your SQL report query.
2. Enter the custom report query into the query file of Content Tracker Reports:
   a. In a text editor, open the *contenttrackerreports_query.htm* file:

      <*install_dir*>/custom/ContentTrackerReports/resources/contenttrackerreports_query.htm

   b. Enter the custom report name, number of columns, and the source database table.

      For example, the following excerpt from the query file illustrates that the custom query report will extract the information from all columns in the Users database table.

```
<tr>
   <td>qCustomUsers</td>
   <td>
SELECT *
FROM Users
   </td>
   </tr>
```

3. Enter a link to the custom report in the Content Tracker Report Generator main page file:

   a. Open the following directory:

      <*install_dir*>/custom/ContentTrackerReports/templates

   b. In a text editor, open the following file:

      contenttrackerreports_main_page.htm

   c. Enter the attributes to display the link on the Content Tracker Report Generator main page.

      For example, the following excerpt from the main page file illustrates that the custom report link is presented as a selectable radio button and is listed as “Custom Users Report” on the page—see the Custom Report Link (page 4-19).

```
<h4 class=xuiSubheading>Custom Reports</h4>
<table width=80% border=0>
<tr>
   <td> <span class="tableEntry"><input type="radio"
   name="radiobutton" value="qCustomUsers">
   Custom Users Report </span></td>
</tr>
</table>
```

4. Enter the formatting requirements in the template resource file of Content Tracker Reports:

   a. Open the following directory:

      <*install_dir*>/custom/ContentTrackerReports/resources

   b. In a text editor, open the following file:

      contenttrackerreports_template_resource.htm

      To view the resulting custom report format, see the Generated Custom Report (page 4-20).

c. Enter the display features to use for the generated custom report as well as any desired drill-down reports—see the Drill-Down Report (page 4-20).

For example, the following excerpt from the template resource file illustrates that in addition to the link listing, the report title is “Deanna’s First Report” and a drill-down report is provided that is based on the content items seen by user report.

```
<!-- Custom Template -->
<@dynamichtml qCustomUsers_vars@>
   <$reportWidth = “100%”$>
   <$title = “<i>Content Access Report</i>”$>
   <$reportTitle=”Deanna’s First Report”$>
   <$column1Width=”35%”$>
   <$column0Drill=”qSctrDocsSeenByUser_Drill”$>
<@end@>
```

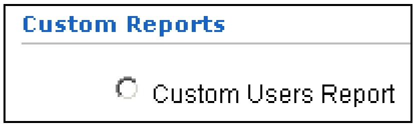
5. Restart the Content Server to apply the changes.

Custom Report Query Display Results

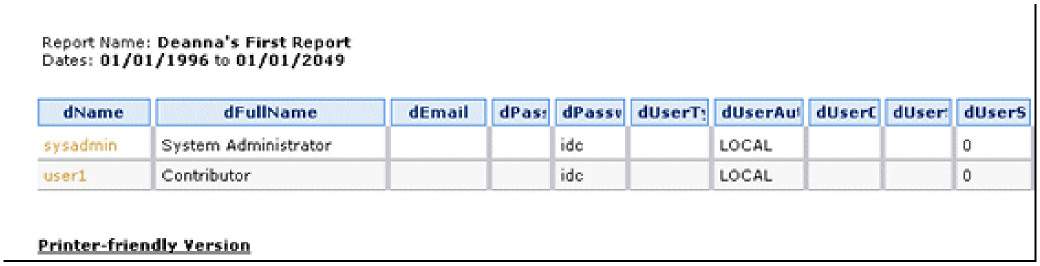
After you have successfully added the custom report query to the report query file, you can use it and view the resulting:

- Custom Report Link (page 4-19)
- Generated Custom Report (page 4-20)
- Drill-Down Report (page 4-20)

Custom Report Link



Generated Custom Report

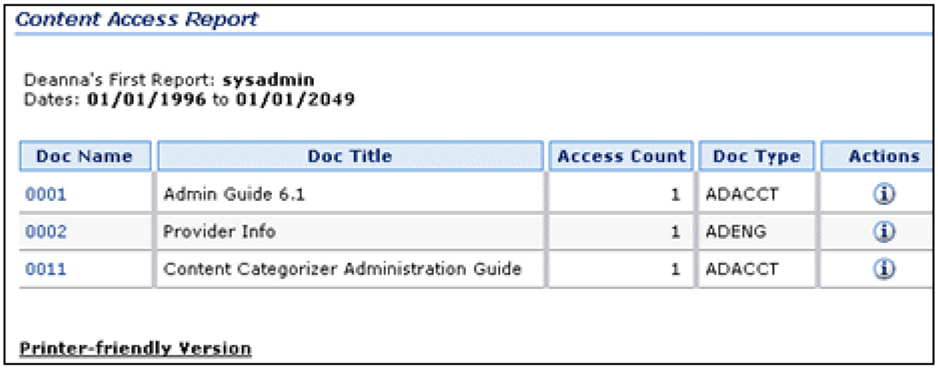


Report Name: **Deanna's First Report**
Dates: **01/01/1996** to **01/01/2049**

dName	dFullName	dEmail	dPas:	dPassv	dUserT:	dUserAu	dUserC	dUser:	dUserS
sysadmin	System Administrator			idc		LOCAL			0
user1	Contributor			idc		LOCAL			0

Printer-friendly Version

Drill-Down Report



Content Access Report

Deanna's First Report: **sysadmin**
Dates: **01/01/1996** to **01/01/2049**

Doc Name	Doc Title	Access Count	Doc Type	Actions
0001	Admin Guide 6.1	1	ADACCT	ⓘ
0002	Provider Info	1	ADENG	ⓘ
0011	Content Categorizer Administration Guide	1	ADACCT	ⓘ

Printer-friendly Version

SUPPLEMENTARY REPORT FEATURES

This section covers the following topics:

- User Authentication/Authorization Monitoring and Auditing (page 4-21)
- Site Studio Web Site Activity Reporting (page 4-21)

User Authentication/Authorization Monitoring and Auditing

Content Tracker Reports provides an auditing feature that enables you to monitor unsuccessful attempts to access the system or permission-protected content items. Two reports are available that can help you analyze attempted security breaches that include failed user logons and unsuccessful attempts to access secure content items. This information is essential to safeguard system and content security as well as to maintain proper maintain audit trails and records.

The available auditing reports include:

- **Authorization Failures by User (page 4-13)**

  This report provides access authorization denial information that includes user names and their IP addresses. Although these users have system access privileges, their role/account memberships may restrict them from accessing particular content items (such as access to payroll content).

- **Login Failures (page 4-13)**

  This report provides login/authentication failure information that includes user names and their IP addresses. The logged data does not distinguish between external, internal, and global users because, without a successful login, it is impossible to differentiate user types.

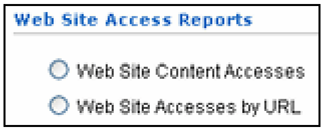
Site Studio Web Site Activity Reporting

If you are using Site Studio, then Content Tracker is automatically configured to track Site Studio activity. Content Tracker Reports uses the logged data to generate the pre-defined reports that summarize the Web site access results. Content Tracker Reports support for Site Studio-specific activity includes:

- Main Page Site Studio Report Links (page 4-22)
- Site Studio Pre-Defined Reports (page 4-22)

Main Page Site Studio Report Links

The Site Studio-specific Web access reports are included on the Content Tracker Report Generator main page if you have installed Site Studio. They are displayed following the User Access Reports (page 4-12) pre-defined report group.



Web Site Access Reports

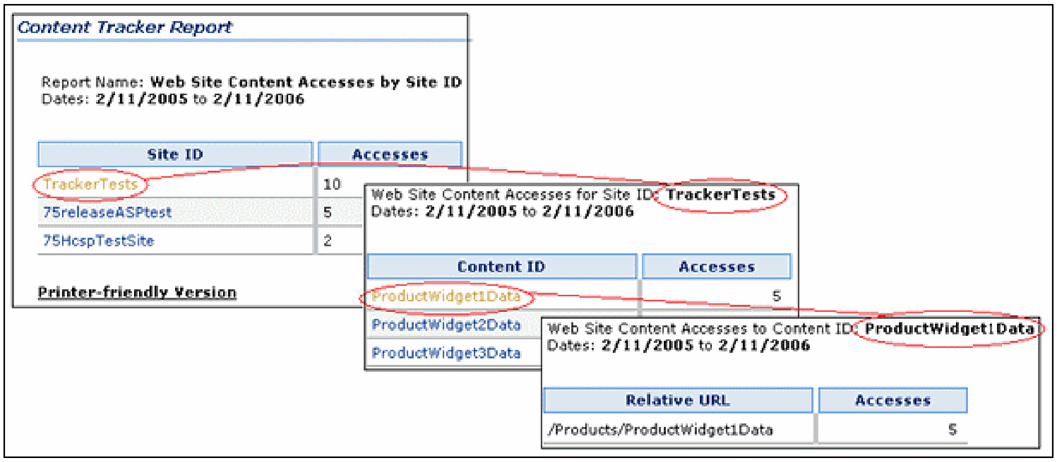
○ Web Site Content Accesses
○ Web Site Accesses by URL

Site Studio Pre-Defined Reports

The Site Studio pre-defined reports use the default Content Tracker Reports formatting and provide drill-down report capabilities. The top level reports for both are summary reports that use Site ID and Accesses as their general criteria. The drill-down reports provide the relevant statistics.

❖ **Web Site Content Accesses**

This report is ID based at the top level and in subsequent drill-down reports, the results are listed by Content ID and Relative URL.The information shows what URLs are being used to access a Web site. However, there are cases where many different URLs will actually display the same page. Therefore, the results of this report also provide the total number of hits on the nodes, regardless of how the user got there.



Content Tracker Report

Report Name: **Web Site Content Accesses by Site ID**
Dates: **2/11/2005** to **2/11/2006**

Site ID	Accesses
TrackerTests	10
75releaseASPtest	5
75HcspTestSite	2

Printer-friendly Version

Web Site Content Accesses for Site ID: **TrackerTests**
Dates: **2/11/2005** to **2/11/2006**

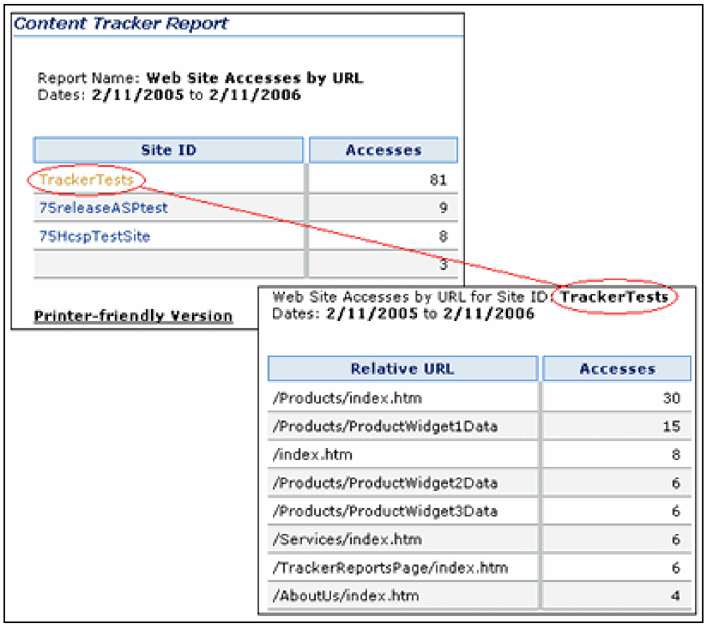
Content ID	Accesses
ProductWidget1Data	5
ProductWidget2Data	
ProductWidget3Data	

Web Site Content Accesses to Content ID: **ProductWidget1Data**
Dates: **2/11/2005** to **2/11/2006**

Relative URL	Accesses
/Products/ProductWidget1Data	5

❖ Web Site Accesses by URL

This report provides summaries of the site-relative URLs and the relevant activity sums.



Content Tracker Report

Report Name: **Web Site Accesses by URL**
Dates: **2/11/2005** to **2/11/2006**

Site ID	Accesses
TrackerTests	81
75releaseASPtest	9
75HcspTestSite	8
	3

Printer-friendly Version

Web Site Accesses by URL for Site ID: **TrackerTests**
Dates: **2/11/2005** to **2/11/2006**

Relative URL	Accesses
/Products/index.htm	30
/Products/ProductWidget1Data	15
/index.htm	8
/Products/ProductWidget2Data	6
/Products/ProductWidget3Data	6
/Services/index.htm	6
/TrackerReportsPage/index.htm	6
/AboutUs/index.htm	4

SECURITY CHECKS AND QUERY RESULTS

During the installation process for Content Tracker Reports, you have the option to employ individual user role and account information to restrict the visibility of content item information in report results. This means that you control what content items (and, subsequently, the metadata) that users can see in their generated reports. Ideally, users should not be able to see anything through Content Tracker Reports that they couldn't find via a Content Server search.

Caution: If you have enabled Access Control Lists (ACLs) on your Content Server instance, the secure mode option in Content Tracker Reports does not work. For more information, see Access Control Lists and Content Tracker Reports Secure Mode (page 4-4).

This section covers the following topics:

- Security Checks Preference Variable (page 4-24)
- Report Queries and Security Modes (page 4-26)
- Establishing the Security Mode (page 4-28)
- Changing the Security Checks Preference Setting (page 4-30)

Security Checks Preference Variable

The security checks preference variable (SctrEnableSecurityChecks) is set when you install the Content Tracker Reports component. Essentially, this preference variable enables you to select one of two security modes: secure and non-secure. The secure mode cares which user is running the report queries and the non-secure mode does not.

Note: During installation, you decide which mode is used by selecting the security checks checkbox or leaving it blank. For information about the security checks preference variable and installing the Content Tracker Reports component, see the *Content Tracker Installation Guide*. After installation, you also have the option to change the setting using the Component Manager—see Changing the Security Checks Preference Setting (page 4-30).

This section covers the following topics:

- Values for the Security Checks Preference Variable (page 4-24)
- Security Mode Examples (page 4-25)

Values for the Security Checks Preference Variable

The values for the security checks preference variable include:

- `SctrEnableSecurityChecks=True` (selected checkbox) enables the security checks installation preference and configures Content Tracker Reports to operate in secure mode.

  In secure mode, the same security criteria (role and account qualifications) that are used to limit Content Server search results are also applied to the Content Tracker Report Generator’s queries and the generated reports. Thus, it is possible that two different users running the Top Content Items report may see different results. See the Secure mode example: (page 4-25).

- **`SctrEnableSecurityChecks=False`** (blank checkbox) disables the security checks installation preference and configures Content Tracker Reports to operate in non-secure mode. This is the default setting.

  In non-secure mode, the additional role and account criteria used to restrict Content Server search results are not applied to Content Tracker Report Generator's queries and the generated reports. Thus, it is possible for a user other than a system administrator to see information about content items that they would not be authorized to access and view. See the Non-secure mode example: (page 4-25).

Security Mode Examples

A user might have admin, contributor, guest, and sysmanager privileges (a semi-admin user) but does not have the proper role/account membership to see a particular content item (such as the payroll report). The assigned privileges allow this user to access the Content Server Admin page, and therefore, the Content Tracker Report Generator main page. However, when this user performs a a standard search in Content Server, the results page would not reveal that the payroll report exists.

If the security checks preference variable is enabled, Content Tracker Reports enforces the same role/account membership checks. Then, depending on the user requesting a specific report, the role/account matching activity determines what content item usage data is included.

As demonstrated in the following examples, the report results generated for a specific user (the semi-admin user described above) are contingent upon whether the preference variable is enabled or not.

- **Secure mode example:**

  When the security checks preference is enabled, Content Tracker Reports is running in secure mode and checks for role/account matches. In this case, the semi-admin user is not entitled to retrieve and view confidential data. Due to the restrictions associated with this user's role/account privileges, the payroll content item remains completely invisible. The data is not included in report results and the user is unaware of its existence.

- **Non-secure mode example:**

  When the security checks preference is disabled, Content Tracker Reports is running in non-secure mode and does not check for role/account matches. In this case, although the semi-admin user is not entitled to access or view the payroll report, some confidential information associated with the payroll content item can nevertheless be retrieved.

At the very least, the user can discover the payroll report’s existence and view some of its metadata The danger in this situation depends on what kind of information the metadata contains. In some cases, even knowing the content item exists could be a serious breach of security.

Note: This kind of security breach is not limited to semi-admin users. For example, a non-privileged user (that is, someone not ordinarily authorized to view a particular content item on a search results page) might gain access to the Content Tracker Report Generator main page. This could occur either by reaching the Admin page or by guessing a URL. In this case, the user would see a report containing some of the metadata describing the prohibited content item.

Report Queries and Security Modes

The contenttrackerreports_query.htm file contains all the Content Tracker Report Generator’s queries that produce the pre-defined and custom reports. To support non-secure and secure modes, this file basically contains two sets of queries. One set cares which user is running the query (secure mode) and the other set does not care (non-secure mode). The security checks preference setting determines which set of queries is used—see Security Checks Preference Variable (page 4-24).

This section covers the following topics:

- ❖ Pre-Defined Reports and Security Modes (page 4-27)
- ❖ Custom Reports and Security Modes (page 4-27)

Note: For localization support, the word “document” was changed to “content item” in the pre-defined report names. However, the corresponding report queries still include an abbreviation for the word document (doc). The report query names have not been changed in the contenttrackerreports_query.htm file.

For example, the “Top Content Items” report is one of the pre-defined reports listed on the Content Tracker Report Generator main page. The corresponding report queries in the contenttrackerreports_query.htm file use the pre-existing naming conventions:

- qSctrTopDocs (non-secure version)
- qSctrTopDocs_SEC (secure version)

Pre-Defined Reports and Security Modes

Almost all the pre-defined report queries have both secure and non-secure forms included in the contenttrackerreports_query.htm file. Generally, if the search results of a query can be affected by user role and account privileges, then secure variants of the non-secure queries are included. And, if the security checks preference variable is enabled, then the secure forms of queries take precedence and are executed instead of the corresponding non-secure queries.

It is not possible to selectively enable or disable the security checks preference variable for individual report queries. However, it is possible to manage secure and non-secure queries by customizing the contenttrackerreports_query.htm file. In effect, you can disable security checks (account matching) for a particular query by deleting or renaming the secure form of the query. Thus, if the security checks preference variable is enabled, but a secure form of a given query is not found in the contenttrackerreports_query.htm file, then the non-secure form of the query is used to generate the report.

For more information about using security checks for a particular pre-defined query, see Enabling/Disabling Security Checks for Report Queries (page 4-31). For more information about collectively enabling or disabling security checks for all report queries, see Changing the Security Checks Preference Setting (page 4-30).

Custom Reports and Security Modes

In addition to the pre-defined reports, you can also create custom reports that are based on search queries tailored to your particular needs. In addition to creating custom reports, you can also selectively implement security checks for them. That is, if you want security checks performed for your new custom report, then you can include both the non-secure and secure forms of the query in the contenttrackerreports_query.htm file.

For example, you can add a custom report with both query forms. If the non-secure query name is qMyTopTwenty, then the secure query name would be qMyTopTwenty_SEC. If the security checks preference variable is enabled, the report is generated using the secure query (qMyTopTwenty_SEC). If the security checks preference variable is not enabled, the report is generated using the non-secure query (qMyTopTwenty).

Note: The secure form of a custom query should follow the specific pattern of the existing secure queries in the contenttrackerreports_query.htm file. For more information, see Creating Secure Report Queries (page 4-32).

For more information about using security checks for a particular custom query, see Enabling/Disabling Security Checks for Report Queries (page 4-31). For more information about collectively enabling or disabling security checks for all report queries, see Changing the Security Checks Preference Setting (page 4-30).

Establishing the Security Mode

To generate a requested report, Content Tracker Reports must select and execute the applicable non-secure or secure query.

This section covers the following topics:

- Query Type Selection Process (page 4-28)
- Example: Report Query Selection (page 4-29)

Query Type Selection Process

Content Tracker Reports chooses a report query based on the following process:

- When a user submits a report request, the name of that report query is fed to a dedicated Content Tracker Reports service.
- The Content Tracker Reports service enforces the security checks setting as follows:
  - **If the security checks preference is disabled:**

    Content Tracker Reports is running in non-secure mode and does not perform role/account matching (user role and account privilege verification). The Content Tracker Reports service searches for the non-secure version of the query and uses it to generate the requested report. It is irrelevant whether there is a secure version of the report query.

    In non-secure mode, only non-secure queries are used to generate reports. As a result, all users see the same report results regardless of their individual role and account memberships.
  - **If the security checks preference is enabled:**

    Content Tracker Reports is running in secure mode and performs role/account matching (user role and account privilege verification).

    To begin processing:

    The Content Tracker Reports service appends the “_SEC” suffix to the submitted query name and searches the contenttrackerreports_query.htm file for this variant of the requested query.

During the search:

- If the secure form of the query is found, then it is used to generate the requested report.

  This means that the security checks to enforce role/account matching are performed and the query results are limited by the role and account privileges of the user requesting the report. Accordingly, different users may see different data results.

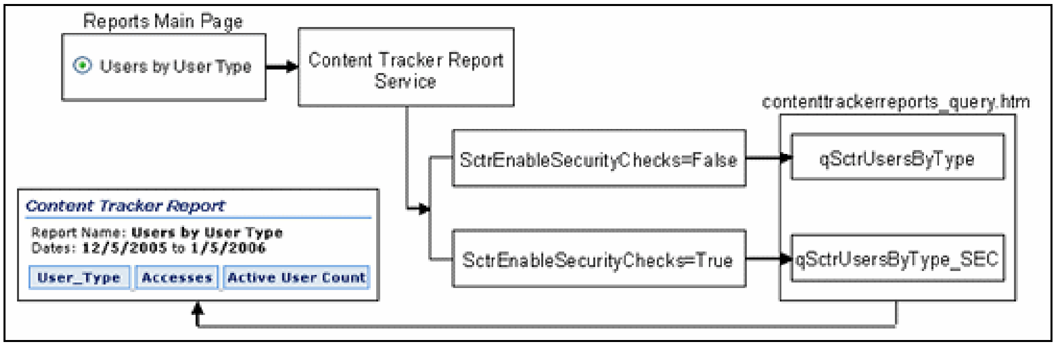
- If the secure form of the query is not found, then the non-secure variant is used.

  This actually produces the same result as if the security checks preference was disabled. This means, role/account permissions are not authenticated and the content item data is not filtered. Consequently, the results included in reports are identical for all users. It is possible for users without proper permissions to view confidential information.

Example: Report Query Selection

When a user requests the Users by User Type report from the Content Tracker Report Generator Main Page (page 4-10):

1. The report query name (qSctrUsersByType) is passed to the Content Tracker Reports service.
2. The Content Tracker Reports service evaluates the request based on the security checks preference variable:
   a. If security checks are disabled (set to false), then the service finds the qSctrUsersByType query in the contenttrackerreports_query.htm file.
   b. If security checks are enabled (set to true), then the service adds a security suffix to the query name (qSctrUsersByType_SEC) and searches for this variant in the contenttrackerreports_query.htm file.
3. Depending on the security checks status, Content Tracker Reports uses the applicable query to generate the Users by User Type report.


Reports Main Page
Users by User Type
Content Tracker Report Service
contenttrackerreports_query.htm
SctrEnableSecurityChecks=False
qSctrUsersByType
SctrEnableSecurityChecks=True
qSctrUsersByType_SEC
Content Tracker Report
Report Name: Users by User Type
Dates: 12/5/2005 to 1/5/2006
User_Type
Accesses
Active User Count

Changing the Security Checks Preference Setting

Optionally, you can manually enable or disable the ScrtEnableSecurityChecks preference setting:

1. Log in to Content Server as an administrator.
2. Select **Admin Server** from the Administration menu.

   The Content Admin Server page is displayed.
3. Click the name of the Content Server instance whose security checks preference setting will be changed.

   The Content Admin Server <*instance_name*> page is displayed.
4. Click **Component Manager**.

   The Component Manager page is displayed.
5. In the Update Component configuration field, select **ContentTrackerReports** from the drop-down list.
6. Click **Update**.

   The Update Component Configuration page is displayed.
7. In the ScrtEnableSecurityChecks preference field, enter the new setting (true or false).
8. Click **Update**.

   Content Tracker Reports is successfully updated with the new setting and is effective immediately. You do not need to restart Content Server.

Customizing Report Query Security

In secure mode, Content Tracker Reports always gives priority to the secure forms of queries. This means that if a a secure form of a query is found in the contenttrackerreports_query.htm file, then it is used to generate the report instead of the corresponding non-secure query.

It is not possible to selectively enable or disable the security checks preference variable for individual report queries. However, it is possible to manage secure and non-secure queries by customizing the contenttrackerreports_query.htm file. Depending on your security requirements for report data, you may optionally want to customize the report query file.

Customizing the report query file involves:

- Selectively enabling or disabling security checks (account matching) for specific report queries.
- Creating one or more non-secure custom report queries and, depending on the security requirements of the information, selectively including the corresponding secure version.

This section covers the following topics:

- Enabling/Disabling Security Checks for Report Queries (page 4-31)
- Creating Secure Report Queries (page 4-32)
- Examples of Non-Secure and Secure Report Query Versions (page 4-33)

Enabling/Disabling Security Checks for Report Queries

If the security checks preference variable is enabled, and a secure version of a query exists in the contenttrackerreports_query.htm file, then Content Tracker Reports will use the secure query to generate the requested report. However, you may decide that certain reports do not need to be generated using security checks. Accordingly, you can selectively disable the secure version of any report query.

To disable security checks (account matching) for particular report queries:

1. In a text editor, open the contenttrackerreports_query.htm file:

   <*install_dir*>/custom/ContentTrackerReports/resources/contenttrackerreports_query.htm

2. Locate the secure version of the query that you want to disable.

3. Rename the query. For example, if you want to disable the qSctrUsersByType_SEC query, you can add the suffix "_disabled" to the query name:

   ```
   qSctrUsersByType_SEC_disabled
   ```

   Renaming the query ensures that the Content Tracker Reports service can not find the secure query in the contenttrackerreports_query.htm file. Instead, the non-secure version (qSctrUsersByType) will be used.

Note: Renaming a secure query is a temporary disabling solution. Later, if you decide that you prefer to use the secure version of a query, you can easily re-enable it by restoring its original name.

Alternatively, you can delete the secure version of the query. However, if you subsequently reconsider, you would need to recreate the entire secure version of the query.

4. Save and close the contenttrackerreports_query.htm file.
5. Restart the Content Server to apply the changes.

Creating Secure Report Queries

For most of the pre-defined report queries, there are both non-secure and secure versions in the contenttrackerreports_query.htm file. Optionally, you can create a secure version for any query that does not currently have one. In particular, this includes any non-secure custom queries that you have added.

To create a secure version of a non-secure report query:

1. In a text editor, open the contenttrackerreports_query.htm file:

   <*install_dir*>/custom/ContentTrackerReports/resources/
   contenttrackerreports_query.htm

2. Locate the query for which you want to create a secure version. For consistency, you should add your secure query immediately following the corresponding non-secure version.
3. Design your secure SQL report query: It might be helpful to review Step 2 in the Creating Custom Report Queries: Example (page 4-17) procedure.
4. Adjust your query to ensure that it follows the pattern of the existing secure queries:
   a. In the FROM clause, include the Revisions table.
   b. In the WHERE clause, include the %SCTR_SECURITY_CLAUSE% token. This acts as a placeholder for the WHERE clause that the Content Tracker Reports service inserts.

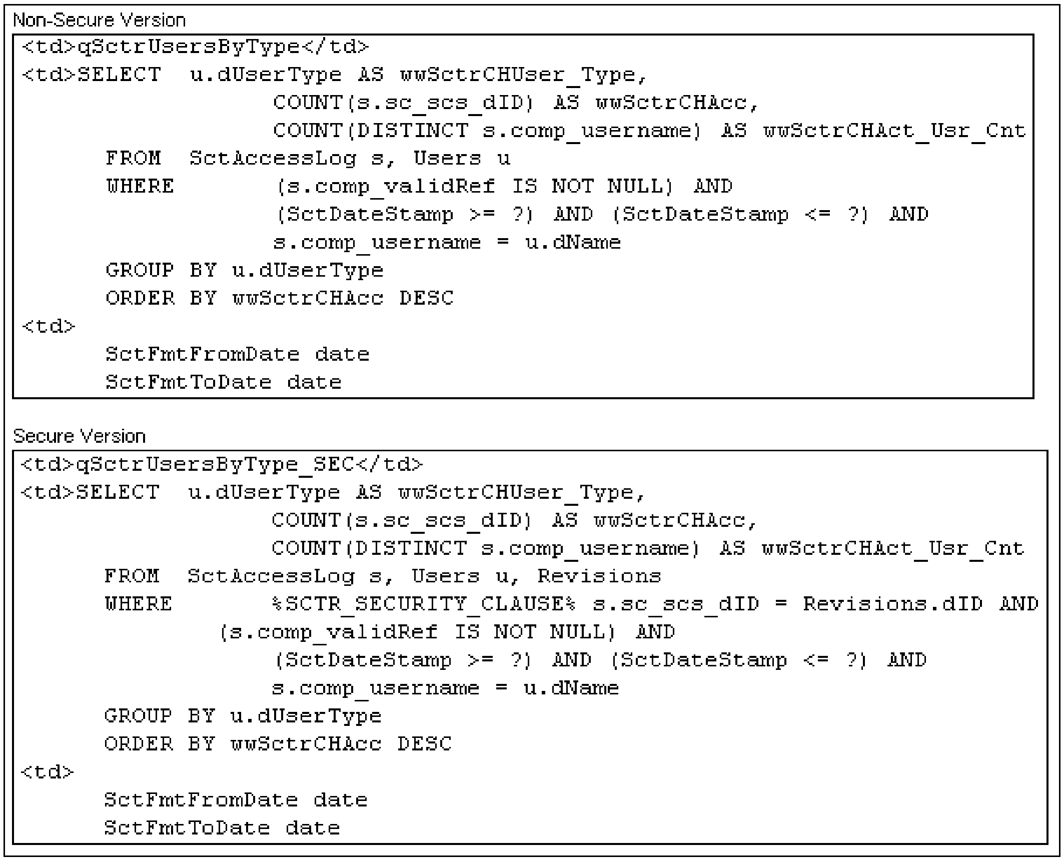
c. Complete the query following the established pattern in the existing secure queries.

   The Examples of Non-Secure and Secure Report Query Versions (page 4-33) illustrates a typical report query pairing.

5. Save and close the contenttrackerreports_query.htm file.

6. Restart the Content Server to apply the changes.

Examples of Non-Secure and Secure Report Query Versions



Non-Secure Version

```
<td>qSctrUsersByType</td>
<td>SELECT  u.dUserType AS wwSctrCHUser_Type,
                  COUNT(s.sc_scs_dID) AS wwSctrCHAcc,
                  COUNT(DISTINCT s.comp_username) AS wwSctrCHAct_Usr_Cnt
      FROM  SctAccessLog s, Users u
      WHERE       (s.comp_validRef IS NOT NULL) AND
                  (SctDateStamp >= ?) AND (SctDateStamp <= ?) AND
                  s.comp_username = u.dName
      GROUP BY u.dUserType
      ORDER BY wwSctrCHAcc DESC
<td>
      SctFmtFromDate date
      SctFmtToDate date
```

Secure Version

```
<td>qSctrUsersByType_SEC</td>
<td>SELECT  u.dUserType AS wwSctrCHUser_Type,
                  COUNT(s.sc_scs_dID) AS wwSctrCHAcc,
                  COUNT(DISTINCT s.comp_username) AS wwSctrCHAct_Usr_Cnt
      FROM  SctAccessLog s, Users u, Revisions
      WHERE       %SCTR_SECURITY_CLAUSE% s.sc_scs_dID = Revisions.dID AND
              (s.comp_validRef IS NOT NULL) AND
                  (SctDateStamp >= ?) AND (SctDateStamp <= ?) AND
                  s.comp_username = u.dName
      GROUP BY u.dUserType
      ORDER BY wwSctrCHAcc DESC
<td>
      SctFmtFromDate date
      SctFmtToDate date
```

EXTERNAL REPORT GENERATOR

Commercial report generation tools can be used to produce basic text reports or more sophisticated graphics such as bar graphs or pie charts from the data collected by Content Tracker. This section provides some general guidelines about connecting a third-party product to the stored Content Server database tables to generate custom reports.

Note: This guide assumes that users have a comprehensive working knowledge of or competent familiarity with the external reporting tool they are using to create custom reports. For this reason, this section is intentionally written to provide only very basic guidelines that can be applicable to most commercially available reporting products.

Using an External Report Generator

To generate custom reports from an external reporting tool:

1. Open the external reporting tool application
2. Set up an ODBC connection (if appropriate) to the Content Server database.
3. Select the database tables that you want to use in your report.
4. Link together the selected tables based on key IDs or fields that are common within the files. Ideally, each selected table could be linked using the same key ID or field if it is common to each table.
5. Choose and integrate the desired fields from each table into the report form. In most cases, the fields can be selected, dragged, and dropped onto the form.

   In this step, you design the customized report. The specific fields that you select will display as columns on the final, basic text report that the external reporting application generates.
6. Optionally, you may want to create custom parameters and/or criteria if the external reporting application supports these options.

   For example, one type of custom parameter would allow you to either have queried information hard-coded into the final report or use a prompt to obtain input directly form the end user. Additionally, creating specific sort criteria can strategically restrict and optimize the aggregate data included in the final report.
7. Specify the sorting order of the selected fields and format the final report output.
8. Preview the final report (optional).

9. Check the report into a delivery mechanism.

   Generally, the final report can be formatted and delivered as web-viewable pages or as a printable file. The external reporting application can also use the data results to create attractive graphics such as bar graphs or pie charts.

   Additionally, the saved file can be imported into other products such as Microsoft Excel or Word files.

Chapter

5

SERVICE CALL CONFIGURATION

OVERVIEW

This section covers the following topics:

Concepts

Tasks

About the Service Call Configuration File

The Content Tracker service handler filter makes it possible to gather information about Content Server activity other than content requests. Service request details are collected by the service handler filter and stored in the SctAccessLog table in real time. The details are obtained from the DataBinder that accompanies the service call. For a Content Server service call to be logged, it must have an entry in the service call configuration file (SctServiceFilter.hda).

The SctServiceFilter.hda file is a user-modifiable configuration file that is used to limit the number of service calls that are logged. This enables you to selectively control which services will be logged. Additionally, you can optionally expand the data logging function for any service call included in the SctServiceFilter.hda file. That is, you can also log and track data values of specific DataBinder fields that are relevant to a particular service. See Extended Service Call Tracking Function (page 5-3).

Note: Service tracking is limited to top-level services that are called via the server socket port. Sub-services, or services that are called internally, cannot be tracked.

Note: The purpose of the SctServiceFilter.hda file is to define which parts of Content Server are of particular interest to users. If a Content Server service is not listed in the SctServiceFilter.hda file, it is ignored by Content Tracker. Additionally, if a service is not listed in this file, it can only be logged by the Content Tracker logging service. See About the Content Tracker Logging Service (page 5-10).

Note: There are two ways to make changes to the SctServiceFilter.hda file. You can add new services and edit the existing service call parameters in the file from the Data Engine Control Center—see Services Tab (page 3-16). Or, you can manually edit the SctServiceFilter.hda file—see Manually Editing the SctServiceFilter.hda File (page 5-9).

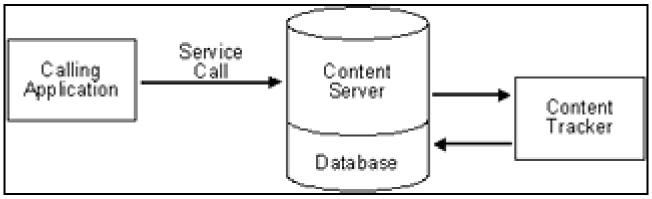
Tech Tip: You can control the services that you want to log by including or excluding them from the SctServiceFilter.hda file. This is an effective method to control logging for particular services or for all services. Also, the extended service call tracking function enables you to customize the type of data that is logged for a specific service.

General Service Call Logging

Services listed in the SctServiceFilter.hda file are detected by the Content Tracker service handler filter and the values of selected data fields are captured. Content Tracker then logs the named service calls. The information along with the timestamps, etc. are written dynamically into the SctAccessLog table.

For each enabled service, Content Tracker automatically logs certain standard DataBinder fields, such as dUser, dDocName, etc.Also, DataBinder fields associated with the extended service call tracking function are logged to the general purpose columns in the SctAccessLog table.

Data is inserted into the SctAccessLog table in real time using Content Tracker-specific services sequence numbers and a type designation of “S” for service. (“W” designations indicate static URL event types). Manual and/or scheduled reductions are only required to process the static URL access information gathered by the web server filter plugin. See Web Server Filter Plugin (page 2-5).



Extended Service Call Tracking Function

The extended service call tracking function enables you to log Content Server service calls and, optionally, supplement this information by also logging relevant data values from one or more additional DataBinder fields other than the standard DataBinder fields logged by each configured service call. This functionality is supported using:

- Service Call ResultSet Combinations (page 5-3)
- General Purpose Columns in the Output Table (page 5-4)

Service Call ResultSet Combinations

Each service that Content Tracker logs must have an entry in the ServiceExtraInfo ResultSet that is contained in the SctServiceFilter.hda file. These entries automatically log various standard DataBinder fields, such as dUser and dDocName. However, the service-related data logged by Content Tracker can be expanded by logging and tracking relevant data values from supplementary DataBinder fields.

The extended service call tracking function is implemented by linking the entries in the ServicesExtraInfo ResultSet to field map ResultSets. Each field map ResultSet contains one or more sets of data field names, the source location, and the destination table column name in the SctAccessLog table. This grouping allows you to select data fields that are relevant to the associated service call and have the data values logged into the specified column in the SctAccessLog table.

Note: Since more than one expanded service can be logged using the extended tracking function, the contents of the general purpose columns in the SctAccessLog table cannot be properly interpreted without knowing which service is being logged. The service name is always logged in the sc_scs_idcService (page 2-10) column. Your queries should match this column with the desired service name.

Caution: In field map ResultSets, nothing prevents you from mapping data fields to existing, standard SctAccessLog table columns. The extended service mapping occurs after the standard field data values are collected. Consequently, you can override any of the standard table column fields.

For example, the service you are logging might carry a specific user name (such as, MyUserName=john) in a data field. You could use the extended tracking function to override the contents of the sc_scs_dUser column. In this case, you simply combine MyUserName and sc_scs_dUser as the data field, location, and table column set in the field map ResultSet.

Therefore, it remains your responsibility to ensure that the data being logged is a reasonable fit with the SctAccessLog column type.

Note: For examples of linked service entries and ResultSets, see Linked Service Entries and Field Map ResultSets (page 5-8). For more information about the contents of the SctAccessLog table and the general purpose columns that are intended to be mapped to data fields, see the Combined Output Table (page 2-8). For more information about the service call user interface, see the Services Tab (page 3-16).

General Purpose Columns in the Output Table

In the field map ResultSets for extended service tracking, you must map the DataBinder fields to columns in the SctAccessLog table. The general purpose columns (extField_1 through extField_10) are available for mapping. These columns may be filled with any data values you consider appropriate for logging and tracking for a particular service. It is recommended and expected that you use these columns to avoid overwriting the standard table columns.

Tech Tip: The name of the service will always be logged to the sc_scs_idcService column. Therefore, you should include it as a qualifier in any query that uses the contents of the extended fields. For more information about custom reports that include specific SQL queries involving SctAccessLog table columns, see Creating Custom Report Queries: Example (page 4-17).

Service Call Configuration File Contents

The initial contents of the service call configuration file (SctServiceFilter.hda) are the commonly used content access, search, and user authentication services native to Content Server. This file contains a ResultSet structure with one entry for each service to be logged. Optionally, to support the extended service call tracking function, this file may also include field map ResultSets that are linked to the service entries contained in the ServiceExtraInfo ResultSet.

You can add new entries and/or edit existing entries in the SctServiceFilter.hda file with the Services user interface accessed through the Data Engine Control Center. Or, you can optionally change entries in the file manually. See the Services Tab (page 3-16) or Manually Editing the SctServiceFilter.hda File (page 5-9).

Note: You can review the set of initial services that Content Tracker logs into the SctAccessLog table by accessing the SctServiceFilter.hda file in the following directory:

<*install_dir*>/custom/ContentTracker/resources/SctServiceFilter.hda

For more detailed information about these services or any others that you may want to include in the service call configuration file, see the *Content Server Services Reference Guide*.

The following table provides details of the service call configuration file result set schema. The values are copied directly to the corresponding columns in the SctAccessLog table.

Feature	**Description**
ServiceExtraInfo ResultSet Contents:	
Service Name (sctServiceName)	The name of the service to be logged. For example, GET_FILE. If no row is present in the ResultSet for a given service, the service will not be logged.
Calling Product (sctCallingProduct)	An arbitrary string. It is generally set to “Core Server” for all standard Content Server entries.

Feature	Description
Event Type (sctEventType)	An arbitrary string. It is generally set to “Content Access” for all standard Content Server entries.
Reference (sctReference)	Used to set the sc_scs_reference field in the SctAccessLog table. If blank, the internal getReference logic is used.
Field Map (sctFieldMap)	The name of the field map ResultSet that is added to the SctServiceFilter.hda file. This field is only required if you plan to use the extended service call tracking function. This function enables you to log DataBinder field information to one or more of the general purpose columns in the SctAccessLog table.
Field Map ResultSet Contents:	
Field Map Link	The name of the field map ResultSet. **Note:** To help you create your field map, a configuration variable can be set that writes out the service DataBinder object. This enables you to see what data is available at the time the event is recorded. For more information, see SctDebugServiceBinderDumpEnabled (page B-3).
DataBinder Field (dataFieldName)	The name of the DataBinder field name whose data values are logged to a general purpose column in the SctAccessLog table. See also Field Name field (page 3-21).
Data Location (dataLocation)	The section in the Content Server service DataBinder where the field to be logged is located.See also Field Location field (page 3-22).
Access Log Column (accessLogColumnName)	The specific general purpose column in the SctAccessLog table where data values from a specified DataBinder field are logged. See also Column Name field (page 3-22).

Note: The fields copied from the DataBinder and inserted into the SctAccessLog table include: dID, dDocName, IdcService, dUser, SctCallingProduct, SctEventType, and SctReference. If the values for the latter three fields are included in a service's entry in the SctServiceFilter.hda file, they will override the corresponding values in the data field.

Tech Tip: Adding desired service calls to the SctServiceFilter.hda file and using this method to log specific activity allows you the advantage of providing values for the CallingProduct, EventType, and Reference fields. The assigned values are copied directly to the corresponding columns in the in the SctAccessLog table.

Note: There should be no duplication or conflicts between services logged via the service handler filter and those logged via the Content Tracker logging service. If a service is named in the Content Tracker service handler filter file then such services are automatically logged so there is no need for the Content Tracker logging service to do it.

ResultSet Examples

The default SctServiceFilter.hda file includes various common service calls. These are structured into:

- ServiceExtraInfo ResultSet Entries (page 5-7)
- Linked Service Entries and Field Map ResultSets (page 5-8)

Note: You can review the initial set of services that Content Tracker logs into the SctAccessLog table along with the service entries and field map ResultSets by accessing the SctServiceFilter.hda file in the following directory:

<*install_dir*>/custom/ContentTracker/resources/SctServiceFilter.hda

For more detailed information about these services or any others that you may want to include in the service call configuration file, see the *Content Server Services Reference Guide*.

ServiceExtraInfo ResultSet Entries

The following list provides examples of several service entries contained in the SctServiceFilter.hda file's ServiceExtraInfo ResultSet.

-
  ```
  GET_FILE_BY_NAME
  Core Server
  Content Access
  ```

- GET_DYNAMIC_URL
  Core Server
  Content Access
- GET_DYNAMIC_CONVERSION
  Core Server
  Content Access
- GET_EXTERNAL_DYNAMIC_CONVERSION
  Core Server
  Content Access
- GET_ARCHIVED_FILE
  Core Server
  Content Access
- COLLECTION_GET_FILE
  Folders
  Content Access

Linked Service Entries and Field Map ResultSets

The following table lists several examples of service entries that are linked to field map ResultSets. These examples, or other similar ones, are included in the initial SctServiceFilter.hda file.

Service Entries	Field Map ResultSets
GET_SEARCH_RESULTS Core Server Search SearchFieldMap	@ResultSet SearchFieldMap 3 dataFieldName 6 255 dataLocation 6 255 accessLogColumnName 6 255 MiniSearchText LocalData extField_1 TranslatedQueryText LocalData extField_2 • • • IsSavedQuery LocalData extField_7 @end
PNE_GET_SEARCH_RESULTS Core Server Search SearchFieldMap	

Service Entries	Field Map ResultSets
GET_FILE Core Server Content Access GetFileFieldMap	@ResultSet GetFileFieldMap 3 dataFieldName 6 255 dataLocation 6 255 accessLogColumnName 6 255 RevisionSelectionMethod LocalData extField_1 Rendition LocalData extField_2 @end

Manually Editing the SctServiceFilter.hda File

To add or change entries in the SctServiceFilter.hda file:

1. In a text editor, open the SctServiceFilter.hda file:

   <*install_dir*>/custom/ContentTracker/resources/SctServiceFilter.hda

2. Edit an existing entry or add a new service entry. For example, to add the GET_FILE_FORM service, enter the following service entry to the ServiceExtraInfo ResultSet in the file:

   ```
   GET_FORM_FILE
   Threaded Discussion
   Content Access
   <optional_reference_value>
   <optional_field_map_link_value>
   ```

   where:

   the *optional_field_map_link_value* is used if you are implementing the extended service call tracking function. In this case, you must also add or edit the corresponding field map ResultSet. Otherwise, if you are implementing extended service tracking, skip Step 3.

3. If you use extended service tracking, you must add or edit the corresponding field map ResultSet. For example, to add the SS_GET_PAGE service and track additional data field values, enter the following service entry and corresponding field map ResultSets to the file:

Service Entry	Field Map ResultSet
SS_GET_PAGE Site Studio Web Hierarchy Access web SSGetPageFieldMap	@ResultSet SSGetPageFieldMap 3 dataFieldName 6 255 dataLocation 6 255 accessLogColumnName 6 255 *<DataBinder_field_name>* *<data_field_location_name>* *<access_log_column_name>* @end
Note: Include as many sets of DataBinder field, location, and table column names as necessary.	

4. Save and close the file.
5. Restart the Content Server to apply the new definitions.

Note: Search request events are logged into the SctAccessLog table in real time and do not need to be reduced.

Note: Optionally, you can add or edit services with the user interface included in the Data Engine Control Center. For more information, see the Data Engine Control Center (page 3-2) and the Services Tab (page 3-16).

About the Content Tracker Logging Service

The Content Tracker logging service is a single service call (SCT_LOG_EVENT) that allows an application to log a single event to the SctAccessLog table. The service may be called directly via a URL or as an action in a service script. It may also be called from IdocScript using the executeService() function. The calling application is responsible for setting any and all fields in the service DataBinder that are to be recorded, including the descriptive fields listed in the Content Tracker SctServiceFilter.hda configuration file.

The SCT_LOG_EVENT service copies information out of the service DataBinder. This data is inserted into the SctAccessLog table in real time using the Content Tracker specific services sequence numbers and a type designation of “S” for service. Manual and/or scheduled reductions are only required to process the static URL access information gathered by the web server filter plugin. See Web Server Filter Plugin (page 2-5).

Note: There should be no duplication or conflicts between services logged via the service handler filter and those logged via the Content Tracker logging service. If a service is named in the Content Tracker service handler filter file then such services are automatically logged so there is no need for the Content Tracker logging service to do it. However, Content Tracker will make no attempt to prevent such duplication.

Setting Required DataBinder Fields to Call the Content Tracker Logging Service

The following table provides the SctAccessLog column names and the corresponding DataBinder fields that Content Tracker looks for when the Content Tracker logging service (SCT_LOG_EVENT) service is called. When an application calls the Content Tracker logging service, the application is responsible for setting the necessary fields in the service DataBinder for Content Tracker to find. For more detailed information about the SctAccessLog fields, see Combined Output Table (page 2-8).

SctAccessLog Column Name	Service DataBinder LocalData Field
SctDateStamp	[computed]
SctSequence	SctSequence
SctEntryType	“S”
eventDate	[computed]
SctParentSequence	SctParentSequence
c_ip	REMOTE_HOST
cs_username	HTTP_INTERNETUSER
cs_method	REQUEST_METHOD
cs_uriStem	HTTP_CGIPATHROOT
cs_uriQuery	QUERY_STRING
cs_host	SERVER_NAME
cs_userAgent	HTTP_USER_AGENT

SctAccessLog Column Name	Service DataBinder LocalData Field
cs_cookie	HTTP_COOKIE
cs_referer	HTTP_REFERER
sc_scs_dID	dID
sc_scs_dUser	dUser
sc_scs_idcService	IdcService (or SctIdcService)
sc_scs_dDocName	dDocName
sc_scs_callingProduct	sctCallingProduct
sc_scs_eventType	sctEventType
sc_scs_status	StatusCode
sc_scs_reference	sctReference (also ...)
comp_username	[computed - HTTP_INTERNETUSER or ...]
sc_scs_isPrompt	n/a
sc_scs_isAccessDenied	n/a
sc_scs_inetUser	n/a
sc_scs_authUser	n/a
sc_scs_inetPassword	n/a
sc_scs_serviceMsg	StatusMessage

Calling the Content Tracker Logging Service from an Application

You can call the SCT_LOG_EVENT service from an application. This can be done by the application developer, or by a user willing to modify the application service scripts. The application can call SCT_LOG_EVENT from Java. Or, the application can include calls to SCT_LOG_EVENT in the service script.

Calling the Content Tracker Logging Service from IdocScript

You can call the SCT_LOG_EVENT service indirectly from IdocScript, using the executeService( ) function. This is the same as calling the SCT_LOG_EVENT service from an application except that it occurs from IdocScript instead of the application Java code. Content Tracker cannot distinguish whether the SCT_LOG_EVENT service is called from Java or from IdocScript.

Appendix

A

CONFIGURING AND CUSTOMIZING CONTENT TRACKER

OVERVIEW

This section covers the following topics:

Concepts

Task

Configuration Variables

The following table lists the default values of the configuration settings used in the current version of Content Tracker. These configuration variables are contained in the Content Tracker configuration file:

<install_dir>/custom/ContentTracker/resources/sct.cfg

Config. Setting	Default Value	Remarks
SctAutoTruncateDataStrings	FALSE	Used by: JAVA Determines whether the reduction process will truncate data strings to fit into the corresponding table column.
SctComponentDir	*<install_dir>*/custom/ContentTracker/	Used by: JAVA Path to the directory where Content Tracker is installed.
SctDebugLogEnabled	FALSE	Used by: JAVA Set TRUE to enable Java code execution trace. Used with SctDebugLogFilePath. For information about using this variable for troubleshooting, see SctDebugLogEnabled (page B-3).
SctDebugLogFilePath	*<install_dir>*/custom/ContentTracker/log/SCT_DEBUG_TRACE.log	Used by: JAVA Directory for Java code execution trace. Used with SctDebugLogEnabled. For information about using this variable for troubleshooting, see SctDebugLogFilePath (page B-3).
SctDebugServiceBinderDumpEnabled	FALSE	Used by: JAVA Set TRUE to enable diagnostic output of Service DataBinder objects during Service logging. For information about using this variable for troubleshooting, see SctDebugServiceBinderDumpEnabled (page B-3).
SctExternalUserLogEnabled	TRUE	Used by: JAVA Set TRUE to enable replication of External user account and role information to UserSecurityAttributes table.

Config. Setting	Default Value	Remarks
SctFilterPluginLogDir	<*install_dir*>/custom/ContentTracker/data/	Used by: filter plugin Path to the directory where filter plugin will store the event logs.
SctIdcAuthExtraConfigParams	SctFilterPluginLogDir, SctLogEnabled, SctIgnoreFileTypes, SctLogSecurity, SctUseLock, SctLockPort	List of Content Tracker configuration parameters that are passed along to the filter plugin, merged programmatically into idcAuthExtraConfigParams by the Content Tracker startup filter.
SctIgnoreDirectories	/stellent/resources/; /stellent/common/	Used by: filter plugin Directs filter plugin to disregard URLs contained within the listed directory roots.
SctIgnoreFileTypes	gif,jpg,js,css	Used by: filter plugin Directs filter plugin to disregard URLs with the listed filetypes.
SctLockPort	4477	Used by: filter plugin Network port used by filter plugin to connect to Content Server Lock Provider.
SctLogDir	<*install_dir*>/custom/ContentTracker/data/	Used by: JAVA Path to the directory(s) where Content Tracker looks for the raw event logs - sctLog, etc. May be multi-valued, e.g. dir1;dir2;…;dir*n*.
SctLogEnabled	TRUE	Used by: filter plugin, JAVA If False, directs service handler filters and web server filter plugin to ignore all events and create no logs. This is the Content Tracker Master On/Off switch.

Config. Setting	Default Value	Remarks
SctLogSecurity	TRUE	Used by: filter plugin, JAVA If true, directs filter plugin to record IMMEDIATE_RESPONSE_PAGE events in the sctSecurityLog event log, and the reduction process to read the event log.
SctMaxRecentCount	5	Used by: JAVA Maximum number of days worth of reduced data kept in the "Recent" state. Overflow from Recent is moved to Archive state.
SctMaxRereadTime	3600	Used by: JAVA Maximum number of seconds that can occur between consecutive references by a particular user to a particular content item, e.g. a PDF file, and have the adjacent references be considered a single sustained access. Consecutive references which occur further apart in time count as separate accesses.
SctPostReduction Exec	[none]	Used by: JAVA Path to Post Reduction Executable (assumed to be in <cs_root>/custom/ContentTracker/bin/)
SctProxyNameMax Length	50	Used by: JAVA Maximum number of characters in the name of any Content Server proxy server in the configuration. Used to increase the size of user name fields in Content Tracker table creation.
SctReductionAvail ableDatesLookback	0	Used by: JAVA Used with SctReductionRequireEventLogs to limit Available Dates range. Unit = Days. Zero = unlimited.
SctReductionLogDir	<*install_dir*>/custom/ContentTracker/log/	Used by: JAVA Path to the directory where the Content Tracker reduction logs are stored.

Config. Setting	Default Value	Remarks
SctReductionRequireEventLogs	TRUE	Used by: JAVA Used in Detached configurations. FALSE means proceed with Reduction even if no event logs are found.
SctScheduledReductionEnable	TRUE	Used by: JAVA Used in Multi-JVM configurations to select which Content Server instance performs the reduction.
SctSnapshotEnable	FALSE	Used by: JAVA Set TRUE to enable Snapshot functions. Set from Data Engine Control Center.
SctSnapshotLastAccessEnable	FALSE	Used by: JAVA Set TRUE to enable Last Access Date Snapshot function. Set from Data Engine Control Center.
SctSnapshotLastAccessField	[none]	Used by: JAVA Metadata field name for Last Access Date, e.g. xLastAccessDate. Set from Data Engine Control Center.
SctSnapshotLongCountEnable	FALSE	Used by: JAVA Set TRUE to enable "Long" interval access count Snapshot function. Set from Data Engine Control Center.
SctSnapshotLongCountField	[none]	Used by: JAVA Metadata field name for Long Interval Count, e.g. xAccessesInLast90Days. Set from Data Engine Control Center.
SctSnapshotLongCountInterval	[none]	Used by: JAVA Number of days for "Long" Interval. Set from Data Engine Control Center.
SctSnapshotShortCountEnable	FALSE	Used by: JAVA Set TRUE to enable "Short" interval access count Snapshot function. Set from Data Engine Control Center.

Config. Setting	Default Value	Remarks
SctSnapshotShortCountField	[none]	Used by: JAVA Metadata field name for Short Interval Count, e.g. xAccessesInLast10Days. Set from Data Engine Control Center.
SctSnapshotShortCountInterval	[none]	Used by: JAVA Number of days for "Short" Interval. Set from Data Engine Control Center.
SctTrackerInfoFile	<*install_dir*>/custom/ContentTracker/bin/trackerinfo.txt	Used by: JAVA Path to special configuration file used to maintain Content Tracker Scheduler parameters.
SctUrlMaxLength	3000	Used by: JAVA Maximum expected length (characters) for URL fields. Used to determine column widths when creating tables. There may be several such columns in a given table.
SctUseGMT	FALSE	Used by: filter plugin, JAVA Set TRUE for logged event times to be converted to Universal Coordinated Time. FALSE uses local time.
SctUseLock	TRUE	Used by: filter plugin If true, directs filter plugin to obtain lock from Content Server Lock Provider before writing to raw log files (sctLog-yyyymmdd.txt, etc.). Otherwise, no locks enforced and sequence numbers assigned internally.

Manually Setting Content Tracker Configuration Variables

To set or edit any of the Content Tracker configuration variables:

1. In a text editor, open the *sct.cfg* file:

   <*install_dir*>/custom/ContentTracker/resources/sct.cfg

2. Locate the configuration variable to be edited.
3. Enter the applicable value.
4. Save and close the *sct.cfg* file.
5. Restart Content Server to apply the changes.

Note: Optionally, you can add or edit the configuration variables for the activity metrics metadata fields with the user interface included in the Data Engine Control Center. These include the following variables:

- SctSnapshotEnable (page A-5)
- SctSnapshotLast AccessEnable (page A-5)
- SctSnapshotLast AccessField (page A-5)
- SctSnapshotLong CountEnable (page A-5)
- SctSnapshotLong CountField (page A-5)
- SctSnapshotLong CountInterval (page A-5)
- SctSnapshotShort CountEnable (page A-5)
- SctSnapshotShort CountField (page A-6)
- SctSnapshotShort CountInterval (page A-6)

For more information about the user interface and the activity metrics functions, see the Data Engine Control Center (page 3-2) and the Snapshot Tab (page 3-11).

Activity Metrics SQL Queries

The snapshot feature enables you to log and track search relevance custom metadata fields. Content Tracker fills these fields with content item usage and access information that reflects the popularity of particular content items. The information includes the date of the most recent access and the number of accesses in two distinct time intervals. For more information about the snapshot feature, see Snapshot Tab (page 3-11).

If the snapshot feature and activity metrics are enabled, the values in the custom metadata fields are updated following the reduction processing phase. When users access content items, the values of the applicable search relevance metadata fields change accordingly. Subsequently, Content Tracker runs three SQL queries as a post-reduction processing step to determine which content items were accessed during the reporting period. For more information about the post-processing reduction step, see Data Reduction Process with Activity Metrics (page 2-3).

This section covers the following topics:

- Customizing the Activity Metrics SQL Queries (page A-8)
- Customizing the Autoload Option SQL Query (page A-9)

Customizing the Activity Metrics SQL Queries

The SQL queries are available as a resource and can be customized to fulfill your specific needs. You may want to filter out certain information from the final tracking data. For example, you might want to exclude accesses by certain users in the tabulated results. The SQL queries are included in the sctQuery.htm file:

<*install_dir*>/custom/ContentTracker/resources/SctQuery.htm

Note: In general, you should feel free to modify the WHERE clause in any of the SQL queries. However, it is recommended that you leave everything else as is.

The following SQL queries are used for the search relevance custom metadata fields:

- qSctLastAccessDate (page A-9)
- qSctAccessCountShort and qSctAccessCountLong (page A-9)

qSctLastAccessDate

For the last access function, the qSctLastAccessDate SQL query uses the SctAccessLog table. It checks for all content item accesses on the reduction date and collects the latest timestamp for each dID. The parameter for the query is the reduction date. In this case, dates may be reduced in random order because the comparison test for the last access date will only signal a change if the existing DocMeta value is older than the proposed new value.

For more information about the last access field on the Snapshot tab, see Enable Last Access updates check box and corresponding Field metadata field (page 3-13).

qSctAccessCountShort and qSctAccessCountLong

For the short and long access count functions, the qSctAccessCountShort and qSctAccessCountLong SQL queries are identical except for the "column name" for the count. They use the SctAccessLog table to calculate totals for all accesses for each dID across the time intervals specified (in days) for each. The parameters are the beginning and ending dates for the applicable rollups.

For more information about the short and long access count fields on the Snapshot tab, see Enable Short / Long Access Count updates check boxes and corresponding Fields / Intervals (page 3-15).

Customizing the Autoload Option SQL Query

The Autoload option on the Snapshot tab of the Data Engine Control Center enables you to backfill the Last Access field for all existing content. When invoked, Autoload runs the qSctLastAccessDateAutoload query which fills the empty (NULL) Last Access fields in Content Server’s DocMeta database table with the current date and time.

However, the qSctLastAccessDateAutoload query is available as a resource and can be customized to fulfill your specific needs. For example, you may want to set the Last Access field to dCreateDate, dReleaseDate, or any other time that meets the requirements of your application. The qSctLastAccessDateAutoload query is included in the sctQuery.htm file:

<*install_dir*>/custom/ContentTracker/resources/SctQuery.htm

For more information about the last access field and the Autoload option on the Snapshot tab, see Enable Last Access updates check box and corresponding Field metadata field (page 3-13) and Autoload check box (page 3-14).

External Users and Content Item Tracking

You have the option to control whether Content Tracker includes data about external user accesses in the applicable reports. These authenticated users are qualified based on their user roles and accounts. By default, the configuration parameter SctExternalUserLog Enabled (page A-2) is set to true (enabled). This allows Content Tracker to monitor external user logons and automatically propagate their role and account information to the UserSecurityAttributes table.

Regardless of whether the SctExternalUserLogEnabled configuration variable is enabled or disabled, all of the content item access information for external users is tracked and recorded. But when it is enabled, this variable ensures that this data is included in reports that explicitly correlate externally authenticated user names with their associated user roles and accounts. Specifically, the Top Content Items by User Role (page 4-12) report and the Users by User Role (page 4-13) report will include all of the content item access activity by external users.

Note: Optionally, you can manually disable the SctExternalUserLogEnabled configuration variable. If you choose to do so, however, content item accesses by externally authenticated users will be included in the more general reports, such at Top Content Items. This data is omitted from reports that use document access counts qualified by user role and account information.

To manually disable the SctExternalUserLogEnabled configuration variable, see Manually Setting Content Tracker Configuration Variables (page A-7).

Appendix

B

TROUBLESHOOTING

OVERVIEW

This section covers the following topics:

Concepts

Tasks

ABOUT CONTENT TRACKER TROUBLESHOOTING

Content Tracker 10gR3 has two execution trace mechanisms: the web server filter plugin and the Java code. These are intended for diagnosing problems at customer installations and are not to be used in production.

Web Server Filter Plugin Debugging Support

The web server filter plugin honors PLUGIN_DEBUG. Set this in the Content Server Filter Administration page and the Content Tracker web server filter plugin will issue execution trace information. The trace is only meaningful to someone with access to the source. Customers with a problem are expected to enable PLUGIN_DEBUG, run the test scenario, and then send the log segments to Customer Service for evaluation. Otherwise, PLUGIN_DEBUG should be left turned off.

Setting the Debug Plugin

To set PLUGIN_DEBUG:

1. In Content Server, click the **Admin Applets** link in the Administration tray.

   The Administration page is displayed.

2. Click the **Filter Administration** icon or link.

   The Configure Web Server Filter page is displayed.

3. Select the PLUGIN_DEBUG option check box.

4. Click Update.

Java Code Debugging Support

Each of the configuration variables available for Java code debugging must be set in the *sct.cfg* file located in the following directory:

> <*install_dir*>/custom/ContentTracker/resources/sct.cfg

For more information about the Content Tracker configuration variables, see Appendix A (*Configuring and Customizing Content Tracker*). For more information about manually setting the variable values in the sct.cfg file, see Manually Setting Content Tracker Configuration Variables (page A-7).

The Java code now supports the following debugging configuration variables:

- SctDebugLogEnabled (page B-3)
- SctDebugLogFilePath (page B-3)
- SctDebugServiceBinderDumpEnabled (page B-3)

SctDebugLogEnabled

The values for this configuration variable include:

- `SctDebugLogEnabled=True` configures Content Tracker ensure that the Java code writes an execution trace to a datestamped log file. The amount of information written to this file is immense and should be used specifically for debugging.
- `SctDebugLogEnabled=False` inhibits the Java code from writing an execution trace to a datestamped log file. This is the default value.

For more information about this configuration variable, see SctDebugLogEnabled (page A-2).

SctDebugLogFilePath

If SctDebugLogEnabled=True, then Content Tracker is configured to use SctDebugLogFilePath to determine which directory contains the trace log. The default value for SctDebugLogFilePath is:

> <*install_dir*>/custom/ContentTracker/log/SCT_DEBUG_TRACE.log.

SctDebugLogFilePath is only meaningful if SctDebugLogEnabled=True. For more information about this configuration variable, see SctDebugLogFilePath (page A-2).

SctDebugServiceBinderDumpEnabled

This configuration variable controls whether the Content Tracker Service Handler Filter (page 2-5), used in logging service events, will write out the service DataBinder for events being logged. Additional information about the SctDebugServiceBinderDumpEnabled is available as follows:

- For more information about this configuration variable, see DataBinder Dump Facility (page B-4).
- For more information about viewing the contents of a DataBinder dump file, see Accessing the DataBinder Object Dump File (page B-5).
- For more information about this configuration variable, see SctDebugService BinderDumpEnabled (page A-2).
- For more information about field maps related to logging customized field data for services, see the Field Map Screen (page 3-21) and Extended Service Call Tracking Function (page 5-3).

DataBinder Dump Facility

This section covers the following topics:

- Values for the DataBinder Dump Facility (page B-4)
- About DataBinder Object Dump Files (page B-4)
- Location of the DataBinder Object Dump Files (page B-5)
- Names of the DataBinder Object Dump Files (page B-5)

Values for the DataBinder Dump Facility

The values for this configuration variable include:

- **`SctDebugServiceBinderDumpEnabled=False`** prevents the Content Tracker service handler filter from writing out the DataBinder objects into dump files. This is the default value.
- **`SctDebugServiceBinderDumpEnabled=True`** configures the Content Tracker service handler filter to write out the DataBinder objects into dump files. Consequently, you can use a dump file as a diagnostic aid when you are developing field maps for extended service logging. If you are creating field maps for services, the dump files enable you to see what data is available at the time the service events are recorded.

About DataBinder Object Dump Files

As soon as Content Tracker records a specific service in the log file, the contents of that service's DataBinder object are written to a serialized dump file. The contents of these files are useful for debugging when you are creating field maps to use the extended service call tracking function. These dump files allow you to see the available LocalData fields for the recorded service.

Note: The Content Tracker service handler filter only creates dump files for DataBinder objects if the associated services are defined in the SctServiceFilter.hda file. For more information about this file, see About the Service Call Configuration File (page 5-2).

Caution: The dump files for DataBinder objects will continue to accumulate until you manually delete them. Therefore, it is recommended that you are careful to use the SctDebugServiceBinderDumpEnabled configuration variable only as necessary.

Location of the DataBinder Object Dump Files

The serialized DataBinder objects are written to:

<*install_dir*>/custom/ContentTracker/DEBUG_BINDERDUMP/<*dump_file_name*>

Names of the DataBinder Object Dump Files

The dump file of DataBinder Objects are text files and their names consist of three parts as follows:

<*service_name*>_<*filter_function*>_<*serial_number*>.hda

Where:

- *service_name* is the name of the logged service (such as, GET_FORM_FILE).
- *filter_function* is one of the following:

End	Filter Event "onEndServiceRequestActions" - Normal end-of-service event.
EndSub	Filter Event "onEndScriptSubServiceActions" - Normal end-of-service for service called as Sub Service.
Error	Filter Event "onServiceRequestError" - End of service where an error occurred. May happen in addition to End.

- *serial_number* is the unique identification number assigned to the file. This enables Content Tracker to create more than one DataBinder object dump file for a given service.

Example:

GET_SEARCH_RESULTS_End_1845170235.hda

Accessing the DataBinder Object Dump File

To access the DataBinder object dump file for a specific logged service:

1. In a text editor, open the specific data binder file in the following directory:

   <*install_dir*>/custom/ContentTracker/DEBUG_BINDERDUMP/

2. Review the contents.

   The dump files for DataBinder objects will continue to accumulate indefinitely. Therefore, it is recommended that you manually delete them when you are finished.

Setting the Debugging Configuration Variables

See Manually Setting Content Tracker Configuration Variables (page A-7) for more detailed information about setting any of the Content Tracker configuration variables.

Appendix

C

THIRD PARTY LICENSES

OVERVIEW

This appendix includes a description of the Third Party Licenses for all the third party products included with this product.

APACHE SOFTWARE LICENSE

```
* Copyright 1999-2004 The Apache Software Foundation.
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*      http://www.apache.org/licenses/LICENSE-2.0
*
```

```
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
```

W3C® SOFTWARE NOTICE AND LICENSE

```
* Copyright © 1994-2000 World Wide Web Consortium,
* (Massachusetts Institute of Technology, Institut National de
* Recherche en Informatique et en Automatique, Keio University).
* All Rights Reserved.  http://www.w3.org/Consortium/Legal/
*
* This W3C work (including software, documents, or other related items) is
* being provided by the copyright holders under the following license. By
* obtaining, using and/or copying this work, you (the licensee) agree that
* you have read, understood, and will comply with the following terms and
* conditions:
*
* Permission to use, copy, modify, and distribute this software and its
* documentation, with or without modification, for any purpose and without
* fee or royalty is hereby granted, provided that you include the following
* on ALL copies of the software and documentation or portions thereof,
* including modifications, that you make:
*
*   1. The full text of this NOTICE in a location viewable to users of the
*      redistributed or derivative work.
*
*   2. Any pre-existing intellectual property disclaimers, notices, or terms
*      and conditions. If none exist, a short notice of the following form
*      (hypertext is preferred, text is permitted) should be used within the
*      body of any redistributed or derivative code: "Copyright ©
*      [$date-of-software] World Wide Web Consortium, (Massachusetts
*      Institute of Technology, Institut National de Recherche en
*      Informatique et en Automatique, Keio University). All Rights
*      Reserved. http://www.w3.org/Consortium/Legal/"
*
*   3. Notice of any changes or modifications to the W3C files, including the
*      date changes were made. (We recommend you provide URIs to the location
*      from which the code is derived.)
*
* THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS
* MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT
* NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR
* PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE
* ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.
*
* COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR
* CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR
* DOCUMENTATION.
*
* The name and trademarks of copyright holders may NOT be used in advertising
* or publicity pertaining to the software without specific, written prior
* permission. Title to copyright in this software and any associated
* documentation will at all times remain with copyright holders.
*
```

ZLIB LICENSE

* zlib.h -- interface of the 'zlib' general purpose compression library
version 1.2.3, July 18th, 2005

Copyright (C) 1995-2005 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied
warranty. In no event will the authors be held liable for any damages

arising from the use of this software.

Permission is granted to anyone to use this software for any purpose,
including commercial applications, and to alter it and redistribute it
freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly jloup@gzip.org
Mark Adler madler@alumni.caltech.edu

GENERAL BSD LICENSE

Copyright (c) 1998, Regents of the University of California

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

"Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

"Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

"Neither the name of the <ORGANIZATION> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

General MIT License

Copyright (c) 1998, Regents of the Massachusetts Institute of Technology

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Unicode License

UNICODE, INC. LICENSE AGREEMENT - DATA FILES AND SOFTWARE

Unicode Data Files include all data files under the directories http://www.unicode.org/Public/, http://www.unicode.org/reports/, and http://www.unicode.org/cldr/data/ . Unicode Software includes any source code published in the Unicode Standard or under the directories http://www.unicode.org/Public/, http://www.unicode.org/reports/, and http://www.unicode.org/cldr/data/.

NOTICE TO USER: Carefully read the following legal agreement. BY DOWNLOADING, INSTALLING, COPYING OR OTHERWISE USING UNICODE INC.'S DATA FILES ("DATA FILES"), AND/OR SOFTWARE ("SOFTWARE"), YOU UNEQUIVOCALLY ACCEPT, AND AGREE TO BE BOUND BY, ALL OF THE TERMS AND CONDITIONS OF THIS AGREEMENT. IF YOU DO NOT AGREE, DO NOT DOWNLOAD, INSTALL, COPY, DISTRIBUTE OR USE THE DATA FILES OR SOFTWARE.

COPYRIGHT AND PERMISSION NOTICE

Copyright © 1991-2006 Unicode, Inc. All rights reserved. Distributed under the Terms of Use in http://www.unicode.org/copyright.html.

Permission is hereby granted, free of charge, to any person obtaining a copy of the Unicode data files and any associated documentation (the "Data Files") or Unicode software and any associated documentation (the "Software") to deal in the Data Files or Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Data Files or Software, and to permit persons to whom the Data Files or Software are furnished to do so, provided that (a) the above copyright notice(s) and this permission notice appear with all copies of the Data Files or Software, (b) both the above copyright notice(s) and this permission notice appear in associated documentation, and (c) there is clear notice in each modified Data File or in the Software as well as in the documentation associated with the Data File(s) or Software that the data or software has been modified.

THE DATA FILES AND SOFTWARE ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THE DATA FILES OR SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in these Data Files or Software without prior written authorization of the copyright holder.

____Unicode and the Unicode logo are trademarks of Unicode, Inc., and may be registered in some jurisdictions. All other trademarks and registered trademarks mentioned herein are the property of their respective owners

MISCELLANEOUS ATTRIBUTIONS

Adobe, Acrobat, and the Acrobat Logo are registered trademarks of Adobe Systems Incorporated.

FAST Instream is a trademark of Fast Search and Transfer ASA.

HP-UX is a registered trademark of Hewlett-Packard Company.

IBM, Informix, and DB2 are registered trademarks of IBM Corporation.

Jaws PDF Library is a registered trademark of Global Graphics Software Ltd.

Kofax is a registered trademark, and Ascent and Ascent Capture are trademarks of Kofax Image Products.

Linux is a registered trademark of Linus Torvalds.

Mac is a registered trademark, and Safari is a trademark of Apple Computer, Inc.

Microsoft, Windows, and Internet Explorer are registered trademarks of Microsoft Corporation.

MrSID is property of LizardTech, Inc. It is protected by U.S. Patent No. 5,710,835. Foreign Patents Pending.

Oracle is a registered trademark of Oracle Corporation.

Portions Copyright © 1994-1997 LEAD Technologies, Inc. All rights reserved.

Portions Copyright © 1990-1998 Handmade Software, Inc. All rights reserved.

Portions Copyright © 1988, 1997 Aladdin Enterprises. All rights reserved.

Portions Copyright © 1997 Soft Horizons. All rights reserved.

Portions Copyright © 1995-1999 LizardTech, Inc. All rights reserved.

Red Hat is a registered trademark of Red Hat, Inc.

Sun is a registered trademark, and Sun ONE, Solaris, iPlanet and Java are trademarks of Sun Microsystems, Inc.

Sybase is a registered trademark of Sybase, Inc.

UNIX is a registered trademark of The Open Group.

Verity is a registered trademark of Autonomy Corporation plc

Index

I