

CA Service Desk Manager

Implementation Guide

r12.6



This documentation and any related computer software help programs (hereinafter referred to as the "Documentation") are for your informational purposes only and are subject to change or withdrawal by CA at any time.

This Documentation may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA. This Documentation is confidential and proprietary information of CA and may not be used or disclosed by you except as may be permitted in a separate confidentiality agreement between you and CA.

Notwithstanding the foregoing, if you are a licensed user of the software product(s) addressed in the Documentation, you may print a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all CA copyright notices and legends are affixed to each reproduced copy.

The right to print copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO THE END USER OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is CA.

Provided with "Restricted Rights." Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Copyright © 2010 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

CA Product References

This documentation set references the following CA products:

- CA Advantage™ Data Transformer (ADT)
- CA Asset Portfolio Management (CA APM)
- CA CMDB
- CA Business Intelligence
- CA Cohesion® Application Configuration Manager (CA Cohesion ACM)
- CA Embedded Entitlements Manager (CA EEM)
- CA Enterprise Workload Automation (CA EWA)
- CA IT Process Automation Manager (CA IT PAM)
- CA Management Database (CA MDB)
- CA Management Portal
- CA Network and Systems Management (CA NSM)
- CA Portal
- CA Remote Control Manager (CA RCM)
- CA Service Desk Manager (CA SDM)
- CA Service Management
- CA Siteminder
- CA Software Delivery
- CA Spectrum® Infrastructure Manager (CA Spectrum)
- CA Wily
- CA Workflow
- Unicenter Asset Portfolio Management (UAPM)

Contact CA

Contact Technical Support

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

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

Provide Feedback

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

If you would like to provide feedback about CA product documentation, complete our short [customer survey](#), which is also available on the CA Support website, found at <http://ca.com/docs>.

Contents

Chapter 1: Introduction 15

Overview	15
Audience	15
CA Service Desk Manager Default User List	16
Primary or Secondary Server Components	17

Chapter 2: Upgrading 23

Planning Your Upgrade	23
Database Considerations	24
Retaining Your Customizations	26
LREL Migration Considerations	28
Status Transition Considerations	30
Email Upgrading	30
How the Migration Console Works	32
Migration Log File	33
Start the Migration Manually	34
How to Upgrade CA CMDB from a Previous Release	34
Support Automation Data Migration	36
How to Migrate a Support Automation Database	36
How to Convert Divisions to Tenants	37
Export CA Support Automation Data	38
Import Support Automation Data	39
How to Configure Support Automation Role Access After Migration	40
How to Migrate a Custom CA Business Intelligence Universe	40
How to Back Up a Custom Universe	40
Install a CA Business Intelligence Universe	41
Update a Universe Link	41
Post Upgrade Configuration	42
How to Upgrade CA Workflow and CA EEM	42
Clear the Webengine and Browser Cache	43
Configure Web Directory and Servers	43
LREL Post-Migration	44
Edit Access Types	47
Enable Priority Calculation	48
How to Add the Incident Priority Field to Incidents	49
Add the Urgency Field to Employee Tickets	49

How to Set Ticket Values for Self-Service Users	50
Activate Status Transitions	53
Activate Transition Types	54
Customize Functional Access Areas	54
SITEMODS.JS File	56
Adjust Access Types	56
Adjust Data Partition Settings	57
Modify Help Sets after Migrating Roles	58
Default Constraint Settings	58
Start the IIS Web Interface (CAisd)	59
How to Upgrade Knowledge Management From r11.2	60
How to Upgrade Knowledge Management From r12 or r12.1	61

Chapter 3: Planning 63

CA Service Desk Manager Default and Recommended Port List	63
CA MDB Installation Planning	66
CA MDB Considerations	67
CA Service Desk Manager Installation Planning	69
CA Service Desk Manager Considerations	72
CA EEM and CA Workflow Installation Planning	82
CA EEM Considerations	83
CA Workflow Considerations	84
CA IT PAM Integration Planning	89
Security Considerations	89
How to Set Up SSL Communications with CA IT PAM	90
CA Business Intelligence Installation Planning	92
Reporting Considerations	93
Reporting Best Practices	96
CA NSM Installation Planning	97
FAST ESP Installation Planning	97
Implementation Strategies	99
Enable Windows Authentication in Firefox	99

Chapter 4: Installing 101

How to Implement the Software	101
The CA MDB Installation	102
MDB Installations	103
Find Product Integration and Compatibility Information	103
Install on SQL Server (Windows)	104
Install on Oracle (Windows)	106
Install on Oracle (Linux\UNIX)	107

The CA Service Desk Manager Installation	108
Install on SQL Server (Windows)	109
Install on Oracle (Windows)	111
Install on Oracle (Linux\UNIX)	114
Visualizer Configuration	116
How to Configure Visualizer on a Secondary Server	116
Support Automation Component Configuration	117
The Web Screen Painter Installation	120
Install Web Screen Painter	120
The CA NSM Integration Installation	121
Install the CA NSM Integration (Windows)	122
The CA EEM Installation	123
Install CA EEM (Windows)	123
Install CA EEM (UNIX)	124
The CA Workflow Installation	126
Install CA Workflow (Windows)	127
Install CA Workflow (UNIX)	128
Install Standalone CA Workflow IDE	129
Uninstall the CA Workflow IDE Client (Windows and Linux)	130
ADT Installation	130
Install ADT	130
The CA CMDB Federation Adapters Installation	132
Create Administrator ID (SQL Server)	132
Configure ADT (SQL Server)	133
Create Administrator ID and Configure ADT (Oracle)	134
Install CA CMDB Federation Adapters	135
The FAST ESP Installation	136
Install FAST ESP (Windows)	136
Install FAST ESP (Linux)	137
The FAST ESP Installation Log	138
Install LinguisticsStudio	138
The CA Business Intelligence Installation	140
New Install of CA Business Intelligence	140
Custom Install of CA Business Intelligence (Windows)	142
Verify the Installation	145
The Install Log	145

Chapter 5: Configuring 147

Product Configuration	147
Configure the CA Service Desk Manager Components	148
Set Up the CA CMDB Audit Log	149

How to Integrate CA Cohesion ACM With CA CMDB	149
CA CMDB Visualizer Configuration on AIX	149
Modify Third-Party Scripts for CA CMDB Compatibility	150
How to Switch the Target Server for CA CMDB Reports	150
How to Configure the Web Interface	151
Enable the Web Engine on the Secondary Server (Windows)	151
Enable the Web Engine on the Secondary Server (UNIX)	152
Configure the Web Interface	153
Start the Web Interface	153
How to Configure CA Business Intelligence	154
Configure Initial CA Business Intelligence Settings	155
How to Configure Date Range Values and Join Parameters	157
Add Your CA Service Desk Manager Users to CMC	159
Add the CA Service Desk Manager Privileged User to CMC	160
Default Settings in CMC	161
Integrate CA Business Intelligence with CA Service Desk Manager	161
How to Configure Trusted Authentication with CA Service Desk Manager and BusinessObjects	163
Configure BusinessObjects LDAP Authentication	165
Connect CA Business Intelligence Server to a Different CA Service Desk Manager Server	166
Change the Maximum Size for a List of Values	168
Change the Report Record Limits	169
Change the Web Intelligence Session Time-Out	169
Replicated Database for Offline Reporting	170
Run the Automated Policies	171
Run the Knowledge Report Card	172
Knowledge Management Sample Data Import	172
How to Configure FAST ESP	172
Use the FAST Search Engine	173
Search External Repositories	175
Change the Lemmatization Strategy	177
Configure Synonyms	180
Find Similar Searches	181
pdm_k_reindex—Knowledge Re-Index Utility	182
Use pdm_k_reindex with FAST ESP	183
Increase the Search Capabilities	185
How to Back Up FAST ESP Data	185
Integrate FAST ESP on the Secondary Server	186
How to Configure CA Workflow	187
Start and Stop CA Workflow	187
Set Up Automatic Login (External Authentication)	191
Configure Worklist and Workflow Manager for Automatic Login	192
Configure Workflow Design Environment for Automatic Login	193

CA Workflow Options	194
How to Configure the Problem Management Sample Workflow	195
How to Configure the Order PC Sample Workflow	197
Change Management Process Definition Example	198
How to Configure CA IT PAM Workflow Integration	201
Verify CA IT PAM and CA Service Desk Manager Installations	201
Configure CA IT PAM Workflow Options	202
CA IT PAM User Administration	205
How to Support Single Sign-On From CA Service Desk Manager to CA IT PAM Using CA EEM	206
Set Up Automatic Login (CA MDB Authentication)	207
Configure Single Point of Entry	208
How to Implement Multi-Tenancy	209
Enable Multi-Tenancy	211
How to Export and Import Tenant Data	219
Utilities Used for Multi-Tenancy	221

Chapter 6: Customizing 231

Customization Overview	231
Notification Method Customization	232
The Notification Process	232
Notification Method Variables	232
How to Create a Customized Notification Method	237
Query and Message Customization	239
Scoreboard Queries	240
ITIL-Specific Queries	247
Activity Notification Messages Customization	247
Schema Customization	253
How to Modify the Schema Designer	254
Display the Web Screen Painter Schema Designer Tool	255
Schema Designer Tabs	255
Schema Designer Tasks	260
Web Interface Customization	266
The Web Screen Painter (WSP)	267
HTML Templates (HTMPL Form)	282
HTMPL Tags	287
Server Variables	308
Server Operations	314
Advanced Customization	322
Event Log Data Storage Customization	349
CA Business Intelligence Reports Customization	351
CA Business Intelligence Infrastructure	352

Development Environment	353
Framework	354
Schema Changes to the Infrastructure	357
Reports and Folder Structures	360
Move New Crystal Reports into CA Business Intelligence	364
How to Move Existing Access Reports into CA Business Intelligence	365
Move from Development to Production	365
Customizing Legacy Reports	366
Custom Report Design	367
How to Generate Reports	375
Report Template Reference	376
Customize Crystal Reports	390

Chapter 7: Populating CA CMDB 391

Database Population	391
How GRLoader Populates the Database	391
Federation Adapter Data Components	392
Input for Federation Adapters	392
Data Population Glossary	392
Create a Data Source Name	393
How to Import Data Using ADT	394
Open the ADT Mapper	395
Set Up the Input Table	396
Set Up the Target Data File	396
Set Up Profiles Using the Script Manager	397
Generate the Script	398
Start ADT Server and Run the Script	398
Use GRLoader to Import the Data	399
Federation Adapters	400
Prerequisites for Running a Federation Adapter	400
CA CMDB Federation Adapter Checklist	401
Family and Class Assignments	402
Source Data Mappings	402
Load CIs from an Excel Spreadsheet	402
Load Hardware Servers from an Excel Spreadsheet	404
Load Relations from an Excel Spreadsheet	404
How to Load Microsoft SMS Data	405
How to Load CA APM Data	405
How to Use the Generic Template	406
Custom Federation Adapters	407

Chapter 8: Using the MDR Launcher 409

The MDR Launcher	410
MDR Terminology	411
MDR Mapping	412
MDR Launching	412
CMDBf Viewer	412
Define an MDR to CA CMDB	413
MDR URL Definitions	417
MDR Launch URL	418
Parameters for URL Substitution	420
Federation Using GRLoader	422
Federate a CI	422
Define Multiple MDRs to a CI Using GRLoader	423
Map Between MDR CIs and CA CMDB CIs	424
How To Configure MDRs for CMDBf Viewer	425
Launching the MDR Web Browser Interface	426
CA Cohesion Integration	426

Chapter 9: Managing Web Services 427

Web Services Management	427
CA Service Desk Manager Components	428
Tips for Web Services Clients	428
Java Clients	429
Web Services Configuration	431
Redeploy the Web Services	432
Web Services Security	433
Error Handling	435
Lock Errors	436
Time Outs	436
Error Codes	436
Web Services Installation	438
How to Activate Design-Time	438
External Specifications	438
User Access Authentication	439
User Name/Password Authentication	439
Public Key Infrastructure (PKI) Authentication	440
Session and Authorization	446
Access Control and Management	447
Objects	457
System Updates and Caching	459
Categories and Properties	459

XML Object Returns	460
ITIL Methodology	463
Incident or Problem Creation	463
Query for Incidents or Problems	464
Attach an Incident to a Problem	464
Attach a Problem to a Change Order	464
Configuration Items	465
Use the Web Services	465
Logins	465
How to Perform Common Tasks	465

Chapter 10: Integrating with Other Products 469

CA Workflow Integration	469
CA Workflow Components	469
CA Workflow Access	472
CA NSM Integration	473
How to Integrate with CA NSM	474
Configure the Converter on UNIX	475
Post Integration Process	476
CA Service Desk Manager Event Converter	507
Leverage NSM to CA Service Desk Manager Integration	508
CA Portal Integration	514
Verify CA Service Desk Manager Web Interface Accessibility	515
Install and Start CA Portal	515
Configure CA Service Desk Manager to Use SSL with CA Portal	517
Mainframe Product Integration	521
Load CA Service Desk Manager Side Data	521
CA Products Currently Using CAISDI	521
CA Products Planning to Use CAISDI	522

Appendix A: Samples Directory 525

Contents of the Samples Directory	525
asset	525
How to Modify the Message Catalog	525
macro_lock	525
ntf_meth	526
pdmconf	526
call_mgt	527
sdk	528

Appendix B: Loading Supplemental Content

529

sd_content.dat	529
----------------------	-----

Chapter 1: Introduction

This section contains the following topics:

[Overview](#) (see page 15)

[Audience](#) (see page 15)

[CA Service Desk Manager Default User List](#) (see page 16)

[Primary or Secondary Server Components](#) (see page 17)

Overview

This guide provides you with the information you need for a successful CA Service Desk Manager implementation in your enterprise, including information about how to do the following:

- Plan and prepare for both a new installation and an upgrade
- Install and upgrade all of the necessary product components
- Configure the product components
- Integrate with some CA products

Note: This guide does *not* detail integrations with all CA products. For detailed information about additional integrations with CA Service Desk Manager that are not described in this guide, see the *CA Unicenter Service Desk Integrations Green Book* at <http://ca.com/support>.

Note: You can find the most current version of the Release Notes, which contain the system requirements, and optional readme file (if available) at <http://ca.com/support>.

Audience

This guide is intended for anyone who wants to understand how to install, upgrade, and configure CA Service Desk Manager. The following users may have specific tasks to complete using the information in this guide:

- *System administrators* and *administrators* use the information in this guide, and their operating system knowledge to install the product for the first time, upgrade the product from version to version, and configure the product based on your implementation requirements.

- *Integrators* use the information in this guide, along with their knowledge of CA products, to integrate CA Service Desk Manager with some CA products.

Note: This guide does *not* detail integrations with all CA products. For detailed information about additional integrations with CA Service Desk Manager that are not described in this guide, see the *CA Unicenter Service Desk Integrations Green Book* at <http://ca.com/support>.

- *Users*, when necessary, can use the information in this guide to install the product and components.

To use the information in this guide, you should have the following:

- A working knowledge of the Windows and/or UNIX operating systems, depending upon your current production environment.
- The ability to perform basic administrative tasks for your operating system.
- Depending on your working environment, you may also need to be familiar with mainframe, mobile devices, and server installations.

Note: Throughout this guide, you will see references to NX_ROOT. The references pertain to the environmental variable containing the installation path of CA Service Desk Manager. This NX_ROOT variable is set in the NX.env configuration file that is used to set environmental variables for CA Service Desk Manager.

Example NX_ROOT Definition

@NX_ROOT=C:\Program Files\CA\Service Desk Manager

CA Service Desk Manager Default User List

The following lists default user information for typical CA Service Desk Manager implementations:

OS	Product	Default Username	OS Level?	How it is Created
Windows	CA Service Desk Manager	ServiceDesk	Yes	Automatically
	CA Service Desk Manager	rhdsuser	Yes	Automatically
	FAST ESP	fastuser	Yes	Member of the Administrative Group and has special security to log on as an automatically created service Not a domain user
	CA EEM	EiamAdmin		Default Password: EiamAdmin

OS	Product	Default Username	OS Level?	How it is Created
	CA MDB SQL Server	ServiceDesk	No	Created in the MDB during configuration
	CA MDB Oracle	mdbadmin	No	Created in the MDB during configuration
UNIX	CA Service Desk Manager	srvcdesk	Yes	Manually created
	FAST ESP	fastuser	Yes	Manually created
	CA MDB Oracle	mdbadmin	No	Created in the MDB during configuration

Primary or Secondary Server Components

CA Service Desk Manager includes components that work together and run on the *primary* or *secondary* server. Before you begin your implementation, you should have a basic understanding of the following components:

Primary Server Components

The following components run on a primary server:

Daemon Manager (pdm_d_mgr)

Starts process sets as defined in the startup file, `pdm_startup.tpl`. By default, the daemon manager tries to start a failed component up to 10 times. To check the status of all CA Service Desk Manager components, use the `pdm_status` utility. The `pdm_d_refresh` utility instructs the daemon manager to start a new cycle of 10 attempts to start any process marked as previously failed.

Message Dispatcher (sslump_nxd)

Acts as a common bus or message passing system. Components that need to communicate with each other first register with the Message Dispatcher. When a component sends a message, the Message Dispatcher delivers it to those components that have registered to receive that type of message. If two components communicate so much that it would be inefficient to pass the messages through the Message Dispatcher, they create a fast channel between them. You can view a list of registered components using the `s/stat` utility.

Database Agent (platform_agent)

Performs SQL queries on the database. Database agents adhere to the logical schema of CA Service Desk Manager and translate the SQL at this level to the physical database platform SQL.

Note: The database agent detects momentary disconnection and failed queries, and attempts to reconnect and communicate with the database. This is only meant for short outages, such as for a brief network outage and momentary disconnection. It is not meant for long outages such as shutting down a database service for maintenance, and so forth. The agent will only retry the connection for a defined number of times (the default is 3 times), and only for a short time period of a few minutes. If the outage is longer than a few minutes, the agent will stop trying to connect, and CA Service Desk Manager must be recycled after the database has been made available again.

Agent Provider (platform_prov_nxd)

Starts or stops database agents. By default, a number of agents are running. If more are required to handle the number of database queries, the Agent Provider starts them. If the system no longer requires so many database agents, the Agent Provider terminates the unnecessary ones.

Virtual Database (bpvrtddb_srvr)

Enables the operation of multiple Object Managers. All Object Managers running on primary or secondary servers connect to the Virtual Database, which arbitrates their access to database agents. For example, when retrieving a new range of ticket reference numbers, the Virtual Database helps ensure that only one Object Manager at a time accesses the table containing the reference numbers. The Virtual Database also performs caching of database information for Object Managers.

Continuous Archive and Purge (arcpur_srvr)

Runs your archive and purge rules as configured by the CA Service Desk Manager administrator.

Database Monitor (dbmonitor_nxd)

Monitors changes to common tables in the CA MDB, for example *ca_contact*.

KPI Daemon (kpi_daemon)

Manages the retrieval, organization, and storage of key performance indicator (KPI) metric data. It runs continuously. When the specified refresh time of a KPI query is reached, the KPI daemon interacts with other system components to collect data, and then stores the resulting metrics in the database.

License Manager (license_nxd)

Manages CA licensing for the product.

Mail Daemon (pdm_mail_nxd)

Sends outbound email notifications.

Mail Eater (pdm_maileater_nxd)

Accepts inbound email for ticket creation and updates.

Notification Manager (bpnotify_nxd)

Manages notifications in a Windows environment.

Spell Checker (lexagent_nxd)

Performs spell checking as requested by clients.

Text API Daemon (pdm_text_nxd)

Creates and updates tickets by external interfaces, such as command line and email.

Timed Event (animator_nxd)

Runs the delay times of events. In an implementation that has many service types or contracts, there may be many active events that the Timed Event engine needs to track. In this situation, you should dedicate the primary server Object Manager entirely to the Timed Event engine. You can configure other Object Managers on the primary or secondary servers for product access as appropriate.

Time-To-Violation (ttv_nxd)

Calculates projected violation times for service types.

Primary or Secondary Server Components

The following components run on a primary or a secondary server:

Proctor Daemon (pdm_proctor_nxd)

Starts and restarts CA Service Desk Manager components, as instructed by the Daemon Manager, on primary and secondary servers. When you install a secondary server, the *pdm_proctor_nxd* process is installed as the CA Service Desk Manager Remote Daemon Proctor service. When the primary server starts, the Daemon Manager instructs the Remote Daemon Proctor to connect to the Message Dispatcher. The Daemon Manager then instructs the Remote Daemon Proctor to start components on the secondary server as defined by Process Sets in the startup file *pdm_startup.tpl*.

Object Manager (domsrvr)

Acts as the server process of CA Service Desk Manager. When you install a primary server, by default, two Object Managers are installed: one for connections to the product, and one dedicated to the Web Screen Painter. This allows you to test your modifications without affecting the production environment. When you install a secondary server, you can configure additional Object Managers.

There must always be a default Object Manager running on the primary server to which clients such as the Timed Event engine can connect.

The Object Manager also caches various records and tables for clients. If you use *pdm_userload* to manipulate these records, you can also use the *pdm_cache_refresh* utility to make the Object Manager retrieve the new data.

Method Engine (spel_srvr)

Runs SPEL code, event, macros, and so forth for an Object Manager. We recommend that every Object Manager be run with its own method engine.

Login Server (boplogin)

Performs operating system user account validation and contact record lookups using the System Login field to match a user with an access type.

If your business provides CA Service Desk Manager to other client businesses, you can place the Login server on a secondary server at a single client location. External authentication can then be enabled in access types. This avoids creating user accounts for your clients on your business systems.

LDAP Virtual Database (ldap_virtldb)

Interfaces with an LDAP directory.

Knowledge Management Daemon (bpebr_nxd)

Performs knowledge base searches. Upon CA Service Desk Manager startup, the bpebr_nxd daemon caches Knowledge Document data in its memory from the database. With a large document base, you might have memory resource issues. The bpebr_nxd daemon has the following size requirements:

Knowledge Management Search

- 100,000 documents
- Memory size = 332,000 KB

FAST ESP Search

- 2,000,000 documents
- Memory size = 6,640,000 KB

Knowledge Management/Keyword Search Indexing Daemon (bdeid_nxd)

Indexes the knowledge base.

Knowledge Management FAQ Ratings Daemon (bu_daemon)

Calculates FAQ ratings for Knowledge Management.

Knowledge Report Card Daemon (krc_daemon)

Performs calculations for the Knowledge Management Knowledge Report Card (KRC) feature. This feature enables analysts and managers to display different matrix views of their knowledge contributions and provide feedback about which documents are most effective. The information provided can be used in a variety of ways to improve the processes of creating knowledge documents and providing the best support to customers.

Knowledge Management Daemon (kt_daemon)

Manages knowledge base administration and knowledge management logic. It also manages notifications and the document approval process.

Multi-Site Support (pdm_global_nxd)

Performs replication between a region and the master region.

Repository Daemon (rep_daemon)

Manages the attachment repositories for CA Service Desk Manager and the Knowledge Management/Keyword Search Daemon.

Version Control Daemon (pdm_ver_nxd)

Synchronizes the schema files between a primary and secondary server to ensure that they are using the same schema.

Apache Tomcat Web Server (javaw)

Enables certain features to be implemented, regardless of whether Microsoft Internet Information Server (IIS) is used as the web server for access to CA Service Desk Manager. These features include CA Workflow, Graph Items, Attachments, and Web Services.

The Apache Tomcat web server can be administered with the apache Tomcat controller (*pdm_tomcat_nxd*).

Web Engine (webengine)

Connects to web browsers through a *pdmweb cgi* running on a Microsoft IIS or Apache Tomcat web server. There must be a web engine for WSP on the primary server so WSP Schema Designer can write schema files. Web engines are the true client of an Object Manager used by web browser to access the product.

Web engines cache .html web forms for connected users. You can manipulate the cache using the *pdm_webcache* utility and see connection statistics using the *pdm_webstat* utility.

Chapter 2: Upgrading

This section contains the following topics:

[Planning Your Upgrade](#) (see page 23)

[How the Migration Console Works](#) (see page 32)

[How to Upgrade CA CMDB from a Previous Release](#) (see page 34)

[Support Automation Data Migration](#) (see page 36)

[How to Migrate a Custom CA Business Intelligence Universe](#) (see page 40)

[Post Upgrade Configuration](#) (see page 42)

Planning Your Upgrade

You can upgrade directly to CA Service Desk Manager r12.5 only from r11.2, r12.0, and r12.1.

If you have an earlier version of the product, such as Unicenter Service Desk r6.0 or r11.1, you *must* upgrade to CA Service Desk Manager r11.2 before upgrading to r12.5.

Important! CA Service Desk Manager only supports ITIL. If you are upgrading from a non-ITIL system, the CA Service Desk Manager r12.5 installation updates you to an ITIL environment.

Note: For more information about upgrading from an earlier version, see the *CA Service Desk Manager Implementation Guide r11.2*. For upgrade patches and assistance, contact Technical Support at <http://ca.com/support>.

Important! If you have a combined CA Service Desk Manager r11.2 and CA CMDB r11.1 installation, you cannot upgrade directly to CA Service Desk Manager r12.5. You *must* first upgrade CA CMDB to r11.2, and then run the r12.5 upgrade. This process upgrades CA Service Desk Manager r11.2 to r12.5, and also upgrades CA CMDB r11.2 to r12.5. CA CMDB r12.0 can also be upgraded directly to CA Service Desk Manager r12.5.

Before upgrading to the current release of CA Service Desk Manager, understand the following:

- [Database Considerations](#) (see page 24)
- [Retaining Your Customizations](#) (see page 26)
- [LREL Migration Considerations](#) (see page 28)
- [Status Transition Considerations](#) (see page 30)
- [Email Considerations](#) (see page 30)
- UTF-8 locale must be installed on Linux/UNIX platforms

Important! On Linux/UNIX, CA Service Desk Manager no longer uses the `smtp_mail` script to process outgoing mail notifications. If you are an existing customer using `smtp_mail`, and you upgrade to the current release, your administrator must configure the appropriate mail options using the Default Mailbox Detail page to enable the mail notification feature of CA Service Desk Manager.

Database Considerations

Before you upgrade to CA Service Desk Manager r12.5, consider the following database information to help you upgrade:

- Back up your existing database using your typical database backup procedures.
- Archive the installation directory (`$NX_ROOT`) using your typical archive procedures. This action lowers the amount of data movement and saves disk space.
- Run the appropriate script to identify any duplicate records on your database:
 - (Oracle) Run *OracleCheckr12UniqueIndexes.sql*, located in the `\Migrate` directory on the installation media.
 - (SQL Server) Open a Command Prompt window and run *SQLCheckr12UniqueIndexes.sql* as follows:

```
cd $NX_ROOT\samples\views\SQLServer
```

Enter the command:

```
sqlcmd -E -e < SQLServer\SQLCheckr12UniqueIndexes.sql
```

Note: After you upgrade to CA Service Desk Manager r12.5, you can find these files at `$NX_ROOT/samples/views/SQLServer` or `$NX_ROOT/samples/views/Oracle` on the server.

Important! These scripts identify your duplicate records. Delete identified duplicate records before you proceed with the migration.

- Upgrade your CA Service Desk Manager r11.2 system to a supported database (SQL Server and Oracle).

Note: For more information about supported database, see the *Release Notes*.

- Upgrade from Unicenter Service Desk r11.0 to CA Service Desk Manager r11.2 before migrating your data to a supported database.
- If you enter special Windows characters, such as a long hyphen, into CA Service Desk Manager or Knowledge Management on a non-Windows system, the characters are not properly stored in the database.

- **Ingres**—If you are using an Ingres database, convert your data to Oracle or SQL Server before upgrading.

Note: For information about the conversion process, see your database documentation.

- **Oracle**—Oracle does not support case insensitive indexes for Configuration Item registration. Before you start the migration on Oracle, verify SQLPlus and Oracle DB are able to communicate using hostname. If communication fails, verify that Oracle is configured with loopback adaptor.
- **SQL Server**—If you are using SQL Server and are upgrading to the current release of CA Service Desk Manager, the default database for the configured Database Userid must be CA MDB. If the default database is not CA MDB, the migration console fails and displays the following message:

"The acctyp_v2 table does not exist on the MDB"

- **Tomcat**—If you configured Tomcat for external authentication with Unicenter Service Desk r11.0, r11.1 or CA Service Desk Manager r11.2, you *must* manually reconfigure Tomcat for external authentication after upgrading to the current product release.
- **Table Updates**—Consider the following table updates that occur during migration:
 - **Status Tables**—These tables are also updated with the appropriate status records if the same code values do not exist in your database. For example, *Cr_Status* is updated with the code *AEUR* (Awaiting End User Response).
 - **Functional Areas**—For each role, migration automatically adds a row for each *usp_functional_access* record. During migration, the access level is set to the same level for each CA Service Desk Manager r12.0 and r12.1 functional areas that are found on the *usp_role* table. New functional areas are mapped using a reference field.
- **Foreign Keys**—Consider the following information:
 - Foreign keys (SRELs) that reference tables, in which the primary key is now a UUID, are changed from integer type to UUID type (or BYTE 16).

Note: For information about setting SREL attributes with foreign key values, see the *CA Service Desk Manager Technical Reference Guide*.

- If you dropped foreign key constraints in your previous CA Service Desk Manager system to mass load data, remember to recreate the foreign key constraints before you run the upgrade. The scripts that drop the constraints are found in the following locations:

- Oracle

\$NX_ROOT/samples/views/Oracle/OracleDropConstraints.sql

- **SQL Server**

\$NX_ROOT/samples/views/SQLServer/SQLDropConstraints.sql

Note: Reapply the dropped constraints by running the appropriate script *OracleAddConstraints.sql* or *SQLServer/SQLAddConstraints.sql*. These scripts are found in the same directory as the drop constraints and contain instructions within the files mentioned.

- **MDB**—The MDB provides a consistent database schema for various IT management data. During the development of the MDB, data elements from your previous CA Service Desk Manager environment were incorporated into this schema. The size of the data elements may increase and, consequently, increase the overall database size.

Note: If you increased the size of the standard data elements beyond the column width defined for the MDB, there may be some data truncation in these elements during the upgrade process. Messages alert you to any truncation that occurs during the upgrade.

- **Distributed Setup**—We recommend that you upgrade your primary server before any secondary servers.
- **Remote Database Setup**—Consider the following information:
 - We recommend that you upgrade the database server with a new MDB before upgrading the Primary Server. If your database server is remote, run the CA MDB installation on the database server before running the upgrade.
 - If you are using a SQL Server MDB database, sqlcmd must be on the client computer before connecting to the remote MDB.

Retaining Your Customizations

Before you upgrade to CA Service Desk Manager r12.5, consider the following information if you have customized the product:

Note: If you close the Migration Console before the upgrade completes, the process continues to run in the background.

- **Custom Reports**—If you customized reports that access database tables from previous versions that have been moved to renamed tables, the column names have been changed in r12.5.

Note: For information about the tables, see the *CA Service Desk Manager Technical Reference Guide*.

- **Customized Forms**—Upgrading to the current release of CA Service Desk Manager retains the customizations on the forms from the previous version. However, you cannot see the r12.5 functionality on the upgraded customized forms.

- **Customized Admin Tree**—If you customized the Admin tree in Unicenter Service Desk r11.0, these changes are not upgraded due to modifications in the architecture to support the role-based user interface. These Admin tree customizations include the addition of new nodes, renaming of existing nodes, modifying access types, or other data alterations. If you still want to use the customizations, we recommend that you do the following:
 1. Before you upgrade, review your Unicenter Service Desk r11.0 Admin tree and note any customizations that you want to use after upgrading.
 2. After the upgrade completes, identify which roles have Admin tree customizations.
 3. Apply the customizations to the appropriate CA Service Desk Manager r12.5 role-based Admin trees.
 4. Review and test to verify that the desired functionality has been retained.

- **Customized Form Buttons**—After the upgrade completes, buttons on customized forms in `/site/mods/html` that did not have quotes around the `msgtxt(n)` part of the code result in an error message, instead of the button name.

For example, in the `detail_cr.html` form, modify `msgtxt(441)` with quotes as follows to display the correct button name:

```
ImgBtnCreate("btnchg", "msgtxt(441)", "detailSave('NEW_CHANGE')",
true, 0, msgtxt(440)); // Save and Create Change Order
```

- **Retaining Customizations**—If you need the CA Service Desk Manager r12.5 functionality and would like to preserve your customizations from a previous release, redo the customizations on a base CA Service Desk Manager r12.5 form, which has the r12.5 functions.

Note: If you customized the *acctypedtl.rpt* and *acctypesum.rpt* reports, the return data in CA Service Desk Manager r12.5 is obsolete.

- **Notification Rules**—If you removed the default activity notifications Contact, Object Contacts, and Contact Types from the previous installation of CA Service Desk Manager and want to retain this functionality, note the default contacts removed before migration. After you upgrade to the new version, remove the default Notification contacts again.
- **Role-based Functionality**—Upgrading can cause issues with the role-based functionality. Consider the following information:
 - If you customized any of the following forms, they are considered read-only by the Web Screen Painter in CA Service Desk Manager r12.5, and they include an `xxx_site.html` version where you can use custom code:
 - `ahdtop.html`
 - `menu_frames.html`
 - `reports.html`
 - `std_body.html`

- `std_footer.html`
- `std_head.html`
- `styles.html`
- `msg_cat.js`
- `menu_frames_role.html`
- **Customized HTML Files**—All customized HTML files retain their default menu bar settings after the upgrade. A pop-up window inherits its menu bar from the main page tab, as a result of the role-based user interface feature, and is not available on the previous release customized forms after the upgrade.
- Some customized HTML files from previous releases are no longer used in the current release. Do the following after you upgrade:
 - a. Run `pdm_perl $NX_ROOT/bin/migrate_to_r12_5_web_check.pl` to append the files with an *incompatible_for_r12_5* extension.
 - b. Open `$NX_ROOT/bin/migrate_to_r12_5_web_check.pl` with a text editor to see a list of incompatible forms for r12.5.
- **Foreign Keys**—If the upgrade process detects referential integrity issues when attempting to reset foreign keys, errors appear in the `migration.log` file. The associated foreign key sets to a predefined valid reference.
- **Primary/Secondary Server, Web Director Configuration**—If your previous install was configured to use secondary servers, or web directors, you *must* run `$NX_ROOT\samples\pdmconf\pdm_edit.pl` after upgrading. Follow the steps defined in `pdm_edit.pl` to complete the configuration.

Note: The secondary servers and web directors do not work until you run `pdm_edit.pl` on an upgraded system.
- **CA Support Automation Divisions**—If you want to [migrate divisions](#) (see page 37) to tenants, convert this data before enabling and configuring Support Automation in CA Service Desk Manager r12.5.

LREL Migration Considerations

A *List Relationship* (LREL) represents an association between two objects. An LREL has a left-hand side (lhs) and right-hand side (rhs) relationship. Each side of the relationship is an attribute of the majic object that contains the data relationship.

In previous releases of the product, .maj LREL statements and objects described many-to-many DBMS data relationships. Many-to-many relationships no longer use the LREL majic statement. Instead, individual tables store both sides of the relationship. Objects access the relationship with a standard BREL statement. For example, you can see the relationship between change orders and CIs by reviewing the new `usp_lrel_asset_chgnr` table and in the corresponding `lrel_asset_chgnr` object.

The LREL changes eliminate the need to store attribute names in the database. The two sides of the relationship are foreign key single relationships (SREL) that are easy to join and index. If necessary, the relationship can contain additional relational attributes.

During the upgrade, the following activities occur as LREL table data migrates to r12.5:

- The system automatically migrates tables and objects with LREL relationships to r12.5 tables and objects.
- The system names new tables using the *usp_lrel_lhsName_rhsName* format.
For example, the *usp_lrel_asset_chgnr* table has a left-hand relationship to assets and a right-hand relationship to change orders.
- The system names the corresponding objects using the *lrel_lhsName_rhsName*.
For example, the *lrel_asset_chgnr* object corresponds to the *usp_lrel_asset_chgnr* table.
- Because of a database limitation, some names are abbreviated.
- Your data migrates from the old tables to the r12.5 tables, and all CA Service Desk Manager code is modified to use these tables.
- The system no longer uses the old LREL database tables, such as *bmlrel*. However, for reference purposes, the old tables retain the data.
- A backward relation (BREL) attribute to the new object replaces the original LREL attribute in each related majic object definition.
- If you are using a supported API, such as the *CreateLrelRelationship()* web service method, the code works as expected.
- If you added any custom LREL style relationships, CA Service Desk Manager migrates them to r12.5 tables.
- Any site-defined code or reports that directly access the old LREL tables operate on old data because the system no longer uses those tables. We recommend that you update the code to use the r12.5 tables so the code and reports run properly.

Important! If your code directly accesses legacy LREL objects or tables, the code fails after migration. We recommend that you upgrade the code before migration. For example, if your code uses majic statements to establish LREL relationships, use the *createLrelRelationships()* method instead of directly populating a table.

Note: We recommend that you verify site-defined code or reports that directly access the database or address the legacy LREL majic objects such as the *lrel1* object to verify that they operate properly. You can update your code to use a supported interface, such as Web Services. You also update the necessary table names. For reports, you can also update the queries with the new DBMS table references.

Status Transition Considerations

Consider the following information if you plan to use Status Transitions after upgrading to CA Service Desk Manager r12.5:

- Status transitions are inactive when you upgrade to r12.5.
Note: All customized status code descriptions that appear on ticket forms are retained during the upgrade process.
- The *Status_Policy_Violations* option is installed and set to Warn by default after you upgrade. This setting allows undefined transitions to occur, but logs a warning.
- If you set the option to Allow, undefined transactions are not logged.
Note: For more information about the *Status_Policy_Violations* option, see the *Online Help*.

Email Upgrading

CA Service Desk Manager replaces the Options Manager, Email inbound email options with a Mailbox (*usp_mailbox* table) that supplies corresponding options. The Email outbound email options are still present under the Options Manager. When you upgrade, CA Service Desk Manager uses your existing email settings to configure a mailbox, instead of the default Mailbox settings that are supplied with CA Service Desk Manager r12.5. Each email option, except EMAIL_ATTACHMENT_DIR (which is no longer needed), is mapped to an option in the *usp_mailbox* table. Any option that is not set, is set as null in the table.

Note: For information about activating the default mailbox and about using mailbox options, see the *Administration Guide* and the *Online Help*.

The following table lists options that are removed from the Email options, provided in the *usp_mailbox* table, and indicates their labels in the Mailbox Detail page:

Email Option	usp_mailbox Option	Default Mailbox Detail Label
EMAIL_ALLOW_ANONYMOUS	allow_anonymous	Allow Anonymous
EMAIL_ATTACHMENT_DIR	N/A	N/A Note: Because EMAIL_ATTACHMENT_DIR is deprecated, you must manually select an Attachment Repository if this option was set and EMAIL_ATTACHMENT_REPOSITORY was not.
EMAIL_ATTACHMENT_REPOSITORY	attmnt_repository	Attachment Repository

Email Option	usp_mailbox Option	Default Mailbox Detail Label
EMAIL_FORCE_ATTACHMENT_SPLITOUT	split_out_attachment	Force Attachment Splitout
EMAIL_IS_ATTACHMENT	attach_email	Attach Entire Email
EMAIL_SAVE_UNKNOWN_EMAILS	save_unknown_emails	Save Unknown Emails
MAIL_EATER_IMAP_HOST_PORT	host_port	Port Override
MAILEATER_CHECK_MAIL_INTERVAL	check_interval	Check Interval
MAILEATER_HOST	host_name	Hostname
MAILEATER_LOGIN_PASSWORD	password	Password
MAILEATER_LOGIN_USERID	userid	Userid
MAILEATER_POP3_HOST_PORT	host_port	Port Override
MAILEATER_SECURITY_LEVEL	security_lvl	Security Level
MAILEATER_SERVER_TYPE	email_type	Email Type

Important! The Attachment Directory setting is deprecated in r12.5, so you *must* specify an Attachment Repository before you continue polling the mailboxes.

Maileater.cfg Considerations

Information that was previously included in the *maileater.cfg* file maps to the *usp_mailbox_rule* table in r12.5. Consider the following information about the mapping from *maileater.cfg* to *usp_mailbox_rule*:

- The ‘-i’ at the beginning of the line denotes case insensitivity and maps to the *filter_ignore_case* field.
- The search filter “Subject: *...” previously denoted a regular expression on which to filter. The “Subject:” is removed, is replaced with a “^” symbol, and the remaining value maps to the *filter_string* field. The *Filter_type* is set to “Subject Contains” type.
- The “TEXT_API xxx” denotes the object that is processed for the rule. The string “TEXT_API ” is removed, and the remainder maps to the *action_object* field. The *action_operation* field is set to Create/Update Object.
- The reply to the user typically contains “PDM_MAIL ...”. If you have “PDM_MAIL” set, set *reply_method* to 1800 or else leave it set to null.
- If the *-s* parameter is set, remove the subject field from the text and set the *reply_subject* with this value.
- The functionality maintains the order of entries. A sequence number starting at 100 and incremented by 100 is set for each valid row.

The other fields in the *usp_mailbox_rule* are set as follows:

Field	Value
mailbox	Default
action_write_to_log	0
action_log_prefix	null
delete_flag	0
description	Migrated from pdm_maileater.cfg file
reply_failure_html	<leave blank to inherit default action>
reply_failure_text	<leave blank to inherit default action>
reply_success_html	<leave blank to inherit default action>
reply_success_text	<leave blank to inherit default action>
text_api_defaults	null
text_api_ignore_incoming	null
action_subject_handling	null
last_mod_dt	null
last_mod_by	null
inclusion_list	"*"
email_address_per_hour	-1
exclusion_list	null
log_policy_violation	1

How the Migration Console Works

The Migration Console guides you through the migration and upgrade processes for CA Service Desk Manager. The console automatically detects an existing installation, such as CA Service Desk Manager r11.2. You can start the upgrade from the installation media or start it [manually](#) (see page 34).

Important! The CA Service Desk Manager Migration Console does not convert divisions to tenants. If you want to configure Support Automation in a multi-tenancy environment, you *must* [separately migrate](#) (see page 37) CA Support Automation r6.0 SR1 eFix5 divisions to r12.5 tenants before enabling Support Automation in CA Service Desk Manager.

The console performs the following tasks:

1. Verifies that your product is CA Service Desk Manager r11.2, r12.0, or r12.1.
2. Converts passwords to a FIPS 140-2 compliant format.
3. Applies MDB updates.
Important! The remote MDB version must be at least CA MDB r1.5, or the migration fails.
4. Migrates LREL data.
5. Converts customized files to UTF-8.
6. Converts access type records to the CA Service Desk Manager r12.5 role and access type records.
7. Migrates and upgrades user scoreboard queries for role-based operations.
8. Migrates and upgrades existing notifications to use notification rules and notification message templates.

Note: After you select Perform Upgrade and click Install, you cannot roll back the migration and upgrade. If you close the Migration Console before the process finishes, it continues to run in the background.

More information:

[The CA MDB Installation](#) (see page 102)

[Product Configuration](#) (see page 147)

[How to Upgrade CA CMDB from a Previous Release](#) (see page 34)

[Support Automation Data Migration](#) (see page 36)

Migration Log File

If you encounter problems during the migration and upgrade, the migration log provides a record of the entire process. You can access this log in the following location:

\$NX_ROOT/log/pdm_migrationr12_5.log

Start the Migration Manually

You can manually start the upgrade using the following command:

- **Windows**

```
$NX_ROOT\bin\migrate_to_r12_5.vbs
```

- **Linux/UNIX**

```
$NX_ROOT/bin/migrate_to_r12_5.sh
```

Note: If migration fails with a "Schema Validation" error, use the previously mentioned command to run the upgrade again.

More information:

[Database Considerations](#) (see page 24)

[Retaining Your Customizations](#) (see page 26)

[LREL Migration Considerations](#) (see page 28)

[How the Migration Console Works](#) (see page 32)

[Migration Log File](#) (see page 33)

[Start the Migration Manually](#) (see page 34)

How to Upgrade CA CMDB from a Previous Release

You can upgrade CA CMDB to CA Service Desk Manager r12.5 from a previous release, such as r11.2 and r12.0. If you cancel migration, you *must* execute the script to relaunch the migration console. The script is located in the */bin* directory of the product, such as *C:/CMDB/bin*. For example, if you cancel migration on Linux or Unix, execute the *migration_to_r12.5.sh* script.

Note: If you are upgrading from a standalone CA CMDB release, you can continue to use standalone CA CMDB functionality in CA Service Desk Manager r12.5. If you are upgrading from an environment with CA CMDB and CA Service Desk Manager, or if you are upgrading from a CA Service Desk Manager environment without CA CMDB, the full CA Service Desk Manager r12.5 installs during the upgrade.

You upgrade CA CMDB as follows:

1. Launch the CA Service Desk Manager installer from the installation media.

The installer detects your version of the product, such as CA CMDB r11.2.

Note: If you are upgrading from a combined CA Service Desk Manager and CA CMDB installation, the installer displays the detected environment as a CA CMDB release, not CA Service Desk Manager.

2. Click Next.

The installation warns you not to use CA Service Desk Manager and Knowledge Management until migration completes.

3. Click Next.

If the installation detects Visualizer, you are prompted to uninstall it manually.

Important! After you uninstall Visualizer, you *must* reboot and relaunch the CA Service Desk Manager installer.

4. Accept the terms of the License Agreement and click Install.

The installation backs up your data and shuts down services.

Note: The CA CMDB upgrade does *not* back up your database.

After the installation completes, the migration console appears with a warning to review your migration documentation.

5. Click Migrate.

The migration console loads system data, updates your MDB, and recycles services.

Note: You can actively view the migration log during the process.

6. The console verifies tables, processes data, backs up, and upgrades your product to r12.5.

The CA Service Desk Manager configuration appears.

7. (Optional) Configure CMDB Only.

If you are upgrading from a standalone CA CMDB environment, a *Configure CMDB Only* check box displays on the General Settings form.

Important! During configuration, when you migrate from CA CMDB stand alone version to CA Service Desk Manager, a *Configure CMDB Only* check box displays. When you clear the *Configure CMDB Only* check box and you click Next, you cannot configure CA CMDB again. Even if you click Back, the *Configure CMDB Only* check box is no longer available. A message warns you of this behavior in the configuration dialog. If you cancel the configuration before it completes and rerun it, the *Configure CMDB Only* check box is available.

The *Configure CMDB Only* check box controls the value of the CA Service Desk Manager environment variable NX_CMDB. The environment variable controls whether the Support Automation feature is configured. If the check box is cleared, Support Automation is configureable, otherwise it is not. The environment variable affects the behavior of some Web forms.

If you are upgrading from a standalone CA CMDB environment, and want to use standalone CA CMDB functionality in CA Service Desk Manager r12.5, you cannot configure Support Automation.

8. Complete the configuration, as appropriate to your environment.

Support Automation Data Migration

You can migrate CA Support Automation r6.0 SR1 eFix5 data to CA Service Desk Manager r12.5 from the following environments:

- Unicenter Service Desk r11.2
- CA Service Desk Manager r12.0
- CA Service Desk Manager r12.1
- CA Support Automation r6.0 SR1 eFix5 without CA Service Desk Manager.

Note: You can only migration data from CA Support Automation r6.0 SR1 eFix5. We recommend that you run a full backup of the CA Support Automation r6.0 SR1 eFix5 database before migrating.

Important! Branding customizations from CA Support Automation r6.0 SR1 eFix5 do not migrate to CA Service Desk Manager r12.5 automatically. We recommend that you review the customized branding to verify that it corresponds to the CA Service Desk Manager branding. If necessary, copy and paste the Header, Footer, and CSS URL data of each division to the corresponding tenant (or public) in CA Service Desk Manager to migrate the branding data.

How to Migrate a Support Automation Database

You can configure the migration tool to migrate data from the Support Automation database to the CA Service Desk Manager database, including Support Automation names transformation into the CA Service Desk Manager database conventions. Migrate the data from the Support Automation database to the CA Service Desk Manager database before the first usage of Support Automation.

The following process outlines how you migrate the data:

1. [Export](#) (see page 38) the CA Support Automation data using the script on the installation media.

The export tool converts the data into .DAT format. The tool performs the following major steps when you migrate the data:

- Import the Support Automation database schema into the CA Service Desk Manager database.

This schema creates the necessary tables that Support Automation uses.

- Migrate the data from the Support Automation database migration XML into the CA Service Desk Manager database.

The CA Service Desk Manager migration tool generates necessary UUIDs and creates necessary records which represent relationships between Support Automation ID and CA Service Desk Manager UUID.

2. Copy the CA Support Automation data export folder to the following directory on the CA Service Desk Manager server:

NX_ROOT/site/sbmigration/SA60

The data export completes.

3. [Import](#) (see page 39) the data into CA Service Desk Manager using the CA Support Automation Migration tool.

The data loads into the database and migration completes.

Historical Data Migration

If you do not want to migrate all the historical data from the CA Support Automation r6.0 SR1 eFix5 database, you can purge some of the historical data. You can configure the number of days to leave in the database in the purge script setup.

You can download the purge script from the following location:

ftp://ftp.ca.com/pub/supportbridge/6.0/patch-01/purge_history_6.0_sp1.zip

How to Convert Divisions to Tenants

You can only migrate divisions from CA Support Automation r6.0 SR1 eFix5. You convert these divisions to tenants to use multi-tenancy in a Support Automation environment. You can migrate each division separately as its own tenant. During the initial data import, all rows in tenant-optional tables are made tenanted.

Important! You migrate this data before enabling Support Automation in CA Service Desk Manager.

You can migrate divisions to tenants as follows:

1. [Export](#) (see page 38) the division data using the script on the installation media.

You can export a single division or all divisions.

2. The export tool converts the data into *.DAT* format.

The tool displays the status of the division export.

3. Copy the CA Support Automation data export folder to the following directory on the CA Service Desk Manager server:

`NX_ROOT/site/sbmigration/SA60`

The data export completes.

4. [Import](#) (see page 39) the data into CA Service Desk Manager using the CA Support Automation Migration tool.

The data loads into the database and migration completes.

Note: For more information about configuring Support Automation, see the *Online Help*.

Export CA Support Automation Data

You export CA Support Automation r6.0 SR1 eFix5 data by converting it to the `.DAT` format that CA Service Desk Manager uses. You can export divisions into separate tenants, and import the data to a public environment. The export tool logs the process and displays the output directory of the log file after the export completes. The export process records the successful output of each table and indicates any unexpected conditions or errors it encounters.

Important! You can only migrate divisions from CA Support Automation r6.0 SR1 eFix5.

To export CA Support Automation data

1. Execute the `SA60Export` script from the installation media in the following directory:

`/casd.nt/SAMigration`

Note: The file extension depends on your operating system. For example, Windows uses “bat,” UNIX uses “sh” for the bourne shell script, “csh” for the C shell, or “ksh” for the korn shell, and so on.

The CA Support Automation Migration tool appears.

2. Do the following:
 - a. Enter the CA Support Automation r6.0 SR1 eFix5 "WEB-INF" installation parent directory.
 - b. Enter a directory to export your CA Support Automation data.

Note: After the export completes, move this folder to the `NX_ROOT/site/sbmigration/SA60` directory on the CA Service Desk Manager server.
 - c. (Optional) Specify whether passwords can be exported. If you select this option, passwords are exported for users, default credentials, and automated task credentials.
 - d. (Optional) Export a single division or all the divisions. If you select this option, a drop-down list displays all active divisions.

3. Click Run.

The Process Status displays information about the export, such as the database table being migrated, and a count of the records in the table.

A message appears if the tool detects unrecoverable errors.

Note: You can stop the export by selecting Stop from the toolbar or file menu.

The data export completes.

4. Configure and implement Support Automation, as appropriate to your environment.

Note: For more information about configuring Support Automation options, see the *Online Help*.

Import Support Automation Data

You import Support Automation data after converting it to .DAT format. You can import the data into CA Service Desk Manager using the Support Automation migration script. You invoke the utility after installing and configuring CA Service Desk Manager. You can also execute *sa_migrate.pl* using the *pdm_perl* command.

You can access the script in the *NX_ROOT\bin* directory. The migration script performs tasks such as processing tables to maintain database constraints, creates corresponding CA Service Desk Manager objects, maps tenant column values, and so on.

The default location for migration-related files in the CA Service Desk Manager installation is in the *NX_ROOT/site/sbmigration* directory. For example, you can find the import configuration file in the *NX_ROOT/site/sbmigration/config* folder. The *sa_migration_config.dat* file stores the id, prop_name, value, and prop_description columns in the CA Service Desk Manager data format.

The directory stores CA Support Automation r6.0 SR1 eFix5 export data, migration utility code, and Perl scripts, and so on.

To import Support Automation data

1. Start the CA Service Desk Manager Service.

The service starts and you can verify that it is running.

2. Enter the following on the command line:

```
pdm_perl <NX_ROOT>\bin\sa_migrate.pl
```

The Support Automation data loads into the database from the export package.

How to Configure Support Automation Role Access After Migration

If you configure Support Automation before migration, the role access configures properly. If you configure Support Automation after migration, set the Support Automation access field for each role to the appropriate value. If the role access is not properly set up, you cannot access the Support Automation Analyst Interface or the End User interface.

The following process outlines how you configure the Support Automation role access after migration:

1. Install the *supportautomation_url* option.
2. Set the Support Automation access field for each role you want to access Support Automation.

Note: For more information about installing the *supportautomation_url* option or creating Support Automation access roles, see the *Online Help*.

How to Migrate a Custom CA Business Intelligence Universe

You can migrate a custom universe as follows:

1. Follow the customization instructions to link a custom universe to the CA Service Desk Manager universe.
2. Back up the *biar* file of any site-defined universes or reports.
3. Install a universe from the current release.
4. Link the custom universe to the universe from the current release.

How to Back Up a Custom Universe

If you linked a custom universe to the CA Service Desk Manager universe, do the following before you install the CA Service Desk Manager r12.5 universe and reports:

1. Use the Import Wizard to create a backup *biar* file of any site-defined universes or reports.
2. Start Universe Designer.
3. Import your custom universe.
4. Save a copy of the universe on your local drive.

The custom universe is backed up.

More information:

[Move from Development to Production](#) (see page 365)

Install a CA Business Intelligence Universe

You install the universe to replace the CA Business Intelligence r12.0 universe and reports.

To install the universe

1. From the CA Service Desk DVD #2, execute setup.exe.
2. Select the Product Installs tab.
3. Scroll down and select CA Business Intelligence Configuration.

The CA Service Desk Component Installer Wizard appears.

4. Click Next and enter your user credentials.

Note: If CA Service Desk Manager is not installed on this computer, specify the CA Service Desk Manager hostname.

5. Click Finish.

The CA Business Intelligence r12.0 universe and reports are replaced and the r12.5 reports are created in the subfolders.

Update a Universe Link

Although the universe name remains unchanged in CA Service Desk Manager r12.5, you *must* update the universe link to the universe. Otherwise, universe connection problems can occur when you run your reports and when you import the universe. The following error message can occur:

Unable to resolve core universe ID

To update the universe link

1. Start Universe Designer on the same computer that you used to back up the custom universe.
2. Select File, Open to open your custom universe.
3. Select File, Parameters.

The Universe Parameters dialog appears.

4. Click the Links tab.
5. Under the Name column, click the CA Service Desk Manager universe.
The Change Source button becomes enabled.
6. Click the Change Source button.

7. Navigate to and select the location of the CA Service Desk Manager *.unv* file.

Note: The *.unv* file is typically located in the CA Universes folder.

8. Click Open, and click OK.
The universe link is updated.
9. Export the custom universe.

Post Upgrade Configuration

You configure the product when the upgrade completes. Use the configuration wizard to verify your existing customizations.

Note: If the Configuration dialog closes without completing post-upgrade configuration, run *pdm_configure -s* from the command line.

How to Upgrade CA Workflow and CA EEM

You install CA Workflow and CA EEM separately in CA Service Desk Manager r12.5.

When migrating to r12.5, you must upgrade to at least CA EEM r8.3 and CA Workflow r1.1.5.

Note: You can upgrade from CA EEM r8.1 directly to r8.4 SP3.

To upgrade CA EEM and CA Workflow, perform the following tasks:

1. Verify that you upgraded to CA Service Desk Manager r12.5.
2. Insert the installation media into your drive.
3. Install CA EEM.

Note: You can install CA EEM from the installation media or download the installer from [CA Support Online](#).

4. Install CA Workflow.

After you install CA Workflow and CA EEM, you *must* manually set the appropriate options in the Options Manager. Review the upgraded options carefully, as the default Tomcat port for CA Workflow is no longer port 8080, and the default Process Manager URL is no longer pmService.

Note: The *cawf_pm_url* option has changed to a default of: *http://<wf_hostname>:<wf_tomcat_port>/pm/service/pmService2*, so you *must* manually change "pmService" to "pmService2" for CAWF communication to remain functional.

Important! After upgrading CA EEM, you *must* set the *eiam_hostname*, *use_eiam_artifact*, and *use_eiam_authentication* options in Options Manager, Security if you previously used eIAM CA Service Desk Manager user authentication. For more information about these options, see the *Online Help*.

Clear the Webengine and Browser Cache

After you upgrade to CA Service Desk Manager r12.5, run the *pdm_webcache* utility to clear the cache for the webengine and browser. If you had customized forms in the previous release, run *pdm_webcache* after executing the *migrate_to_r12_5_web_check.pl* script.

Note: If you did not have customized forms in the previous release, you can run the utility immediately after the CA Service Desk Manager configuration completes.

```
pdm_webcache -b -H
```

-b

Warns the user to clear their browser cache.

-H

Clears the webengine cache.

Configure Web Directory and Servers

After you upgrade, we recommend that you configure the Web Director, the primary server, and all secondary servers.

To configure the Web Director and servers

1. If the previous version was configured to use secondary servers, or web directors, execute the following script:

```
$NX_ROOT\samples\pdmconf\pdm_edit.pl
```

2. Follow the steps defined in *pdm_edit.pl* to complete the configuration.

Note: The secondary servers and the Web Director cannot operate until you run *pdm_edit.pl* on an upgraded system.

LREL Post-Migration

After migration, complete the following verification steps:

1. Query the contents of the new tables to verify that the tables contain the correct data.
2. Update each site-defined report to verify that report data originates from the new LREL tables.
3. Test site-defined reports.

Deprecated Object and Tables

The following objects and tables are deprecated for this release of CA Service Desk Manager. During migration, the system copies the data to r12.5 LREL tables. The system uses the LREL tables and objects, but for reference purposes, the old tables retain the data that was present at the time of the upgrade.

Note: For more information about LREL tables, see the *Technical Reference Guide*.

DBMS Name	Object Name
Attachment_Lrel	attmnt_lrel
Business_Management_Repository_Lrel	bmlrel
Chgcat_Group	chgcat_grp
Chgcat_Loc	chgcat_loc
Chgcat_Workshift	chgcat_workshift
Group_Loc	grp_loc
Isscat_Group	isscat_grp
Isscat_Loc	isscat_loc
Isscat_Workshift	isscat_workshift
Knowledge_Lrel_Table	kmlrel
Lrel_Table	lrel1
Pcat_Group	pcat_grp
Pcat_Loc	pcat_loc
Pcat_Workshift	pcat_workshift
Wftpl_Group	wftpl_grp

LREL Tables and Objects

The migration automatically creates the following tables and objects to manage many-to-many data relationships:

DBMS Name	Object Name
usp_lrel_asset_chgnr	lrel_asset_chgnr
usp_lrel_asset_issnr	lrel_asset_issnr
usp_lrel_att_cntlist_macro_ntf	lrel_att_cntlist_macro_ntf
usp_lrel_att_ctplist_macro_ntf	lrel_att_ctplist_macro_ntf
usp_lrel_att_ntflist_macro_ntf	lrel_att_ntflist_macro_ntf
usp_lrel_attachments_changes	lrel_attachments_changes
usp_lrel_attachments_issues	lrel_attachments_issues
usp_lrel_attachments_requests	lrel_attachments_requests
usp_lrel_aty_events	lrel_aty_events
usp_lrel_bm_reps_assets	lrel_bm_reps_assets
usp_lrel_bm_reps_bmhiers	lrel_bm_reps_bmhiers
usp_lrel_cenv_cntref	lrel_cenv_cntref
usp_lrel_dist_cntlist_mgs_ntf	lrel_dist_cntlist_mgs_ntf
usp_lrel_dist_ctplist_mgs_ntf	lrel_dist_ctplist_mgs_ntf
usp_lrel_dist_ntflist_mgs_ntf	lrel_dist_ntflist_mgs_ntf
usp_lrel_false_action_act_f	lrel_false_action_act_f
usp_lrel_false_bhv_false	lrel_false_bhv_false
usp_lrel_kwrds_crsolref	lrel_kwrds_crsolref
usp_lrel_notify_list_cntchgntf	lrel_notify_list_cntchgntf
usp_lrel_notify_list_cntissntf	lrel_notify_list_cntissntf
usp_lrel_notify_list_cntntf	lrel_notify_list_cntntf
usp_lrel_ntfr_cntlist_att_ntfrlist	lrel_ntfr_cntlist_att_ntfrlist
usp_lrel_ntfr_ctplist_att_ntfrlist	lrel_ntfr_ctplist_att_ntfrlist
usp_lrel_ntfr_macrolist_att_ntfrlist	lrel_ntfr_macrolist_att_ntfrlist
usp_lrel_ntfr_ntflist_att_ntfrlist	lrel_ntfr_ntflist_att_ntfrlist
usp_lrel_oenv_orgref	lrel_oenv_orgref
usp_lrel_status_codes_tsktypes	lrel_status_codes_tsktypes

usp_lrel_svc_grps_svc_chgcat	lrel_svc_grps_svc_chgcat
usp_lrel_svc_grps_svc_isscat	lrel_svc_grps_svc_isscat
usp_lrel_svc_grps_svc_pcat	lrel_svc_grps_svc_pcat
usp_lrel_svc_grps_svc_wftpl	lrel_svc_grps_svc_wftpl
usp_lrel_svc_locs_svc_chgcat	lrel_svc_locs_svc_chgcat
usp_lrel_svc_locs_svc_groups	lrel_svc_locs_svc_groups
usp_lrel_svc_locs_svc_isscat	lrel_svc_locs_svc_isscat
usp_lrel_svc_locs_svc_pcat	lrel_svc_locs_svc_pcat
usp_lrel_svc_schedules_chgcat_svc	lrel_svc_schedules_chgcat_svc
usp_lrel_svc_schedules_isscat_svc	lrel_svc_schedules_isscat_svc
usp_lrel_svc_schedules_pcat_svc	lrel_svc_schedules_pcat_svc
usp_lrel_true_action_act_t	lrel_true_action_act_t
usp_lrel_true_bhv_true	lrel_true_bhv_true

Verify LREL Conversion

During data migration, the system adds LREL tables to manage many-to-many relationships. You can verify the contents of the new tables and updated site-defined code and reports.

To verify data in the LREL tables

1. Query the contents of the tables to verify that they contain the correct data.
2. Update each site-defined report to verify that report data originates from the new LREL tables.
3. Update the queries with the new DBMS table references.
4. Test site-defined reports and code. Update your code to use the new LREL tables and a supported interface, such as Web Services. If necessary, update the table names in your code.

Verify Database Customizations

You can verify that your database customizations migrated correctly to the current release of the product.

To verify database customizations

1. Review each customized table using either your database management product or the Web Screen Painter.
2. Verify that your customized files appear in the following directory:

`$NX_ROOT/site/mods/`

Verify Web Form Customizations

You can verify that your web form customizations migrated correctly to the current release of the product.

To verify web form customizations

1. Verify that your customized forms appear in the `$NX_ROOT/site/mods/www/html` directory.
2. Verify that your web form opens correctly within a browser.
3. Verify that you web form opens correctly in Web Screen Painter.

Edit Access Types

When you upgrade from CA Service Desk Manager r11.2, the upgrade process automatically creates roles for all access types and correctly assigns access and permissions to the roles. If you want to take advantage of the new roles in r12.5, you can create roles for access types.

To create roles for access types

1. Log in to the web interface as a user with the ability to access the Administration tab.
2. Click the Administration tab.
3. In the tree on the left, select Security and Role Management, Access Types.

All available access types display.

4. Click an available access type.

The detail page for the access type displays.

5. Click the Roles tab.
6. Select a new role for the access type and click Update Roles.

The new role is associated with the access type.

Note: You can also create a custom role and assign it to the access type. For information about creating roles, see the *Administration Guide*.

Enable Priority Calculation

Priority calculation is a set of values that automatically set Priority, Urgency, and Impact values on problems and incidents. For new CA Service Desk Manager installations, the default priority calculation is enabled for problem and incident ticket types by default. However, if you are upgrading from a previous release, the default priority calculation is inactive.

If you create and activate a different priority calculation, the ticket values reflect the settings in the active priority calculation that is associated with an incident or a problem. When no priority calculation is active, users can manually set the Priority and other values on tickets.

Note: The customized forms on the Employee and Customer interface operate in the same manner as the previous versions. The Self-Service users can directly change the Priority regardless of the settings in Priority calculation.

To enable priority calculation after the migration, do the following:

1. On the Administration tab, navigate to Service Desk, Request/Incidents/Problems, Priority Calculation.
The Priority Calculation List appears.
2. Right-click the default priority calculation or another priority calculation and select Edit from the short-cut menu.
The Update Priority Calculation page appears.
3. Set the Status to Active.
4. Select one or more of the following ticket types:

Incidents

Enables this priority calculation to manage incident tickets. Only one active priority calculation can manage incidents.

Problems

Enables this priority calculation to manage problem tickets. Only one active priority calculation can manage problems.

5. Click Save.
The values on the default priority calculation apply to new tickets unless you activated another priority calculation. On new tickets that use a priority calculation, the Priority field is read-only.

Note: For information about defining a priority calculation for tenants and tickets, see the *Administration Guide* and *Online Help*.

How to Add the Incident Priority Field to Incidents

The *incident priority* is the sum of the Urgency and Impact values. The incident priority is only for the incident ticket type. The Incident Priority value appears on the incidents after you install the `use_incident_priority` option and add it to the incident Detail page form with Web Screen Painter.

To add the Incident Priority field to incident, do the following:

1. Install the `use_incident_priority` option from the Options Manager, Request Mgr.
2. Use Web Screen Painter to add the Incident Priority field to the Incident Detail pages.

The Incident Priority value appears on saved Incident Detail page when the `use_incident_priority` option is installed. When the `use_incident_priority` option is not installed, the Incident Priority value is zero.

Note: The `use_incident_priority` option only manages the Incident Priority value. This option is not related to priority calculation.

More information:

[Set the Urgency Range for Self-Service Users](#) (see page 50)

Add the Urgency Field to Employee Tickets

By default the Urgency field does not appear on Employee incidents or requests. However, you can add the Urgency field by using the `urgency_on_employee` option.

Note: When you uninstall the `urgency_on_employee` option and disable priority calculation, the Priority field appears on the Request and Incident Detail pages for self-service Users.

To add the Urgency field to Employee tickets, install the `urgency_on_employee` option from the Options Manager, Request Mgr. The Urgency field appears on Employee incidents or requests. Self-service users can override the value on the incident.

How to Set Ticket Values for Self-Service Users

You can control the Urgency and Priority values that appear to Self-Service users. The properties you set in the *web.cfg* file manage choices that appear to users while they create or edit tickets.

To set ticket values for Self-Service users, consider the following:

1. For each override value in the *web.cfg* parameter, specify one or more values.
2. For [Urgency values](#) (see page 51), specify one or more numbers from 0 through 4.
3. For [Priority values](#) (see page 52), specify one or more numbers from 1 through 5 or the word None.
4. Separate each value with a space.
5. Specify the first value that appears in the list as the default value that appears on tickets. If necessary, you can repeat the default value in the list to improve legibility.

More information:

[Set the Urgency Range for Self-Service Users](#) (see page 50)

[Urgency Property Values](#) (see page 51)

[Set the Priority Range for Self-Service Users](#) (see page 51)

[Priority Property Values](#) (see page 52)

Set the Urgency Range for Self-Service Users

For self-service incidents and requests, you can set default Urgency values in the *web.cfg* file. When you set a range of Urgency values, self-service users such as employees, VIP employees, or guests can set Urgency values on tickets. The choices that appear to self-service users are based on the range of values that you set in the *web.cfg*.

To set the default Urgency range for self-service users

1. Open the *web.cfg* file from the appropriate directory:
 - (Windows) %NX_ROOT%\bopcfg\www\
 - (UNIX) \$NX_ROOT/bopcfg/www/
2. For each parameter, specify one or more [urgency property values](#) (see page 51). Separate each value with a space:

ESCEmpUrg

Specifies how VIP employees can override Urgency on tickets.

EmpUrg

Specifies how employees can override Urgency on tickets.

AnonymousUrg

Specifies valid priorities for tickets created by guest users.

3. Save the *web.cfg*.

On new tickets, employees, VIP employees, or guests can set Urgency values based on the range of values in the *web.cfg*.

Example: Show Guests Only Two Urgency Values on a Request

1. Open the *web.cfg*.
2. Set the *AnonymousUrg* parameter as 0 4. For example, *AnonymousUrg 0 4*.
3. Save the *web.cfg*.

The Urgency values that appear to the self-service user are 1-When Possible and 5-Immediate. The default Urgency is 1-When Possible.

Urgency Property Values

The *web.cfg* contains settings to control how self-service users override Urgency on tickets. The following Urgency property values are available:

- **0**—Lets the user set the Urgency to 1-When Possible
- **1**—Lets the user set the Urgency to 2-Soon
- **2**—Lets the user set the Urgency to 3-Quickly
- **3**—Lets the user set the Urgency to 4-Very Quickly
- **4**—Lets the user set the Urgency to 5-Immediate

Set the Priority Range for Self-Service Users

You can set a range of valid priorities to allow self-service users to override Priority values on tickets. When you set the priority range, customers, employees, or guests can set Priority values based on the range of values in the *web.cfg*.

To set the priority range for self-service users

1. On the *web.cfg* file from the appropriate directory:
 - (Windows) %NX_ROOT%\bopcfg\www\
 - (UNIX) \$NX_ROOT/bopcfg/www/
2. For each of the parameters, specify one or more [Priority property values](#) (see page 52).

CstPrio

Specifies how customers can override Priority on tickets.

EmpPrio

Specifies how employees can override Priority on tickets.

AnonymousPrio

Specifies how employees can override Priority on tickets.

3. Save the *web.cfg*.

On new tickets, customers, employees, or guests can set Priority values based on the range of values in the *web.cfg*.

Example: Show Guests Only Two Priority Values

1. Open the *web.cfg*.
2. Set the *AnonymousPrio* parameter to None 4. For example: AnonymousPrio None 4.
3. Save the *web.cfg*.

When a guest works with tickets, the values for Urgency are None or 4. The default value is None.

Priority Property Values

The *web.cfg* contains settings to control how self-service users override ticket Priority. The following Priority property values are available:

- **None**—Lets the user set the Priority to None
- **1**—Lets the user set the Priority to 1 (highest priority)
- **2**—Lets the user set the Priority to 2
- **3**—Lets the user set the Priority to 3
- **4**—Lets the user set the Priority to 4
- **5**—Lets the user set the Priority to 5 (lowest priority)

Activate Status Transitions

After the upgrade, all predefined status transitions are inactive, so status transitions are not in effect. You can activate and modify these status transitions to accommodate the ticket status transition flow you want.

Note: All customized status code descriptions that appear on ticket forms are retained during the upgrade process.

To activate a status transition

1. On the Administration tab, expand the Service Desk node, and select one of the following ticket types:
 - Change Orders
 - Change Order Transitions
 - Issues
 - Issue Transitions
 - Requests/Incidents/Problems:
 - Incident Transitions
 - Problem Transitions
 - Request Transitions

The Transitions List appears.

2. Select Show Filter on the Transitions List page.

The top portion of the page reveals additional search fields.
3. Select Inactive in the Record Status field and click Search.

The Transitions List at the bottom of the page displays all inactive transitions.
4. Open the transition for editing.
5. Select Active in the Record Status drop-down list.
6. Click Save, Close Window.
7. Click Search.

The Transition List displays the active transition.

Note: For more information about status transitions, see the *Administration Guide* and *Online Help*.

Activate Transition Types

By default, all predefined transition types delivered with the product are inactive, so status transition buttons are not in effect. You can activate and modify these transition types to accommodate the status transition flow you want.

To activate a transition type

1. Select Show Filter on the Transition Type List page.
The top portion of the page reveals additional search fields.
2. Select Inactive in the Record Status field and click Search.
The Transition Type List displays all inactive transition types.
3. Right-click the title of the transition type and select Edit from the menu.
4. Select Active in the Record Status drop-down list.
5. Click Save, Close Window.
6. Click Search.

The Transition Type List displays the active transition type.

Customize Functional Access Areas

A *functional access area* is a group of objects that let you restrict user access. Previous versions of CA Service Desk Manager included eight fixed functional access groups to restrict access to code components.

During migration, the functional access groups migrate to new functional access areas for each role. Migration automatically handles the Majic changes, the default reference data, and role mapping to the new functional access areas.

After migration, consider the following actions:

- Review how the objects map to existing and new functional access areas and role permissions to each area. Use Web Screen Painter to verify the functional access areas.
- Use CA Service Desk Manager to remap or change permissions. Verify that users have the proper access to features and objects.

Note: For details about default permissions and how objects map to the new functional access areas, see the Product Support website. For information about how to change or add functional access areas, see the *Online Help*.

The following table maps Functional access areas to code components:

Functional access area	Code Component	New
Administration	admin	No
Incident/Problem/Request	call_mgr	No
Change Order	change_mgr	No
Inventory	inventory	No
Issue	issue_mgr	No
Knowledge Document	kd	No
Notification	notify	No
Reference	reference	No
Security	security	No
Announcement	announcement	Yes
Incident/Problem/Request Reference	call_mgr_reference	Yes
Incident/Problem/Request Template	call_mgr_template	Yes
Change Order Template	change_mgr_template	Yes
Change Order Reference	change_reference	Yes
Configuration Item	ci	Yes
Configuration Item Common	ci_common_ro	Yes
Configuration Item Reference	ci_reference	Yes
Contact	contact	Yes
Group	group	Yes
Issue Template	issue_mgr_template	Yes
Issue Reference	issue_reference	Yes
Location	location	Yes
Multisite Administration	multisite_admin	Yes
Multisite Reference	multisite_reference	Yes
Notification Reference	notification_reference	Yes
Organization	organization	Yes
Prioritization	prioritization	Yes
Service Level	service_level	Yes
Site	site	Yes

Stored Query	stored_queries	Yes
Survey	survey	Yes
Tenant Admin	tenant_admin	Yes
Timezone	timezone	Yes
Workflow Reference	workflow_reference	Yes
Workshift	workshifts	Yes

Post-Migration Access Level Changes

After migration, you can verify functional access levels for every role. Because the objects moved to another functional access areas, the user could have access to some screens in some situations that they were denied previously. They can also be denied access to forms to which they had access previously. Both situations can occur when a new functional access area manages permissions for two of the original functional access areas.

Note: For details about default permissions and how objects map to the new functional access areas, see the Product Support web site. For information about how to change or add access levels, see the *Online Help*.

SITEMODS.JS File

Lines of code added to the sitemods.js file of the previous version, called from an HTML page, must be merged into the current sitemods.js file before the code works.

Adjust Access Types

If you customized access types and data partitions in the previous release of CA Service Desk Manager, you may have a problem with the Knowledge Management data partitions settings after upgrading. These customizations can cause a problem with the permission groups settings on categories and documents. For example, a user has access to restricted information.

Note: Even if you recreated a data partition or an access type after deleting it, verify your access type and data partition settings after the upgrade.

To adjust access types

1. Click the Administration tab.
The Administration page appears.
2. Click Security and Role Management, Role Management, Role List.
The Role List appears.

3. Complete the following steps for each role:

- a. Right-click the role and select Edit.
- b. Review the Data Partition Name field under the Authorization tab.

If this field is empty, there is no data partition associated with the selected access type, so the user has no restrictions and can access any document or category in the product, even if you set up permission groups.

This action can be appropriate for administrators, but not for all roles. If there is no data partition associated with the role, you can create or modify one.

Adjust Data Partition Settings

You can adjust partition constraints after you configure the roles in your system. You adjust the partition constraints to verify that the appropriate permissions function properly after you upgrade to the current release of the product.

To adjust data partition constraints

1. On the Administration tab, browse to Security and Role Management, Data Partitions, Data Partition Constraints.

The Data Partition Constraints List page appears.

2. Verify the Majic code constraint settings for the following tables:

SKELETONS

Specifies the table used for Knowledge Documents.

O_INDEXES

Specifies the table used for Knowledge Categories.

The table constraint settings are verified.

3. Click Show Filter and enter the Data Partitions you previously used.

Note: You can also use the Table Name field in the Search area to limit your list. For example, enter *SKELETONS* or *O_INDEXES* in the Table Name field and click Search.

Modify Help Sets after Migrating Roles

After you upgrade, CA Service Desk Manager provides all migrated roles with the complete *Online Help*. You can modify help sets for any role, as appropriate to the needs of your online help system.

To modify help sets for a role

1. On the Administration tab, browse to Security and Role Management, Role List.
The Role List page appears.
2. Open the Role for modification, such as *customer*.
The Role Detail page appears.
3. Click Edit.
4. Select the Web Interface Tab.
Click Help View.
The list of available Help Sets appears for the selected role.
5. Select a Help Set, such as *customer*.
6. Save the Role.
The help View for the role changes to the selected online help set.

You can also view the topics available under an online help set by selecting the help set detail and clicking the View Help button.

Default Constraint Settings

The typical default settings for Constraints are listed as follows:

Constraint Settings for the Customer (like) and Employee (like) Data Partitions

Constraint Settings for Customer (like) and Employee (like) Data Partitions should be the following:

SKELETONS Table

View constraint as:

'SKELETONS READ_PGROUP in @root.pgroups or READ_PGROUP.[pgroup]
contained_roles.role in @root.id) and ACTIVE_STATE = 0'

Pre-update and Delete constraint:

'id = 0' (id=0 indicates no access)

O_INDEXES Table

View constraint as:

READ_PGROU in @root.pgrou or READ_PGROU.[pgrou]

contained_roles.role in @root.id

Pre-update and delete constraint:

WRITE_PGROU in @root.pgrou OR WRITE_PGROU.[pgrou]

contained_roles.role IN @root.role

Constraint Settings for CA Service Desk Manager Analyst (like), Knowledge Managers (like) and Knowledge Engineers (like)

Constraint Settings for CA Service Desk Manager Analyst (like), Knowledge Managers (like) and Knowledge Engineers (like) should be the following:

SKELETONS Table

View constraint as:

(ACTIVE_STATE >= 0) and (READ_PGROU in @root.pgrou or READ_PGROU.[pgrou] contained_roles.role in @root.id) OR (ACTIVE_STATE > 0 AND ASSIGNEE_ID = @root.id) OR (ACTIVE_STATE = 0 AND OWNER_ID = @root.id)) Active

Pre-update and delete constraint:

(ACTIVE_STATE >= 0) AND (WRITE_PGROU in @root.pgrou OR WRITE_PGROU.[pgrou] contained_roles.role IN @root.role) OR

(ACTIVE_STATE > 0 AND ASSIGNEE_ID = @root.id) OR (ACTIVE_STATE = 0 AND OWNER_ID= @root.id)) Active

O_INDEXES Table

View constraint as:

READ_PGROU in @root.pgrou or READ_PGROU.[pgrou]

contained_roles.role in @root.id

Pre-update and delete constraint:

WRITE_PGROU in @root.pgrou OR WRITE_PGROU.[pgrou]

contained_roles.role IN @root.role

Start the IIS Web Interface (CAisd)

After you upgrade a CA Service Desk Manager r11.2 Windows installation that had an IIS integration, the CA Service Desk Manager IIS web interface (CAisd) stops. If you want to continue using the IIS integration, manually start CAisd after you upgrade.

Important! If you want to use IIS 7.0, you *must* install the CGI and Metabase Compatability components.

How to Upgrade Knowledge Management From r11.2

Upgrading to CA Service Desk Manager r12.5 from r11.2 automatically upgrades your Knowledge Management environment. When the upgrade finishes, complete the following steps:

1. Map links created in a resolution of a document to the database to locate broken links.

Note: You use the default *Flag broken links* policy to locate broken links.

2. On the Administration tab, browse to Knowledge, Automated Policies, Policies, Scheduling.

The Scheduling page appears.

3. Select the Run Calculation check box in the Last Updated field.
4. Enter a date in the Schedule text box or click the Calendar icon to select a date.
5. Select the time interval to perform the calculation and run the policies. Click Save.

The policies are processed at the specified date and time.

6. Run `pdm_k_reindex` as follows:

`pdm_k_reindex -pm`

Fixes the document links and embedded images inside the resolution field.

Important! After you upgrade, you can get a critical error when running `pdm_k_reindex -pm`. If you get this error, browse to Knowledge, Approval Process Manager, Approval Process Settings, and change the *Permissions for Document Edit after Publish* option to *User with full permissions may edit documents*, and then run `pdm_k_reindex -pm`.

`pdm_k_reindex -ml`

Fixes the document links inside the resolution field and maps them to the database.

`pdm_k_reindex`

Indexes the documents so they are searchable in your knowledge environment.

The Knowledge Management environment is upgraded to r12.5.

Note: After the upgrade, printing Knowledge Documents can result in a large space inserted after the Resolution section of the document. This space is inserted due to an issue with upgrading document templates from a previous release. For more information about resolving this printing issue, see the *Release Notes*.

Important! After you upgrade, Knowledge Management notification data from previous releases of CA Service Desk Manager uses the r12.5 notification engine. For example, there are default activity notifications and notification rules for object types, such as the Knowledge Report Card. For more information about Support Automation and Knowledge Management using the CA Service Desk Manager notification engine, see the *Release Notes*.

More information:

[The FAST ESP Installation](#) (see page 136)

How to Upgrade Knowledge Management From r12 or r12.1

Upgrading to CA Service Desk Manager r12.5 from r12 or r12.1 automatically upgrades your Knowledge Management environment. When the upgrade finishes, complete the following steps:

1. Map links created in a resolution of a document to the database to locate broken links.

Note: You use the default *Flag broken links* policy to locate broken links.

2. On the Administration tab, browse to Knowledge, Automated Policies, Policies, Scheduling.

The Scheduling page appears.

3. Select the Run Calculation check box in the Last Updated field.
4. Enter a date in the Schedule text box or click the Calendar icon to select a date.
5. Select the time interval to perform the calculation and run the policies.
6. Click Save.

The policies are processed at the specified date and time.

7. (For Keyword Search implementations) Enter the following command at the command prompt:

```
pdm_k_reindex
```

8. (For FAST ESP implementations) do the following:
 - a. Upload the r12.5 index profile (datasearch-5.0-lemmatization.xml file) using the Matching Engines tab of FAST ESP Administration interface.

- b. Restart FAST ESP.
- c. Enter the following command at the command prompt:

```
pdm_k_reindex factory:all
```

The Knowledge Management environment is upgraded to r12.5.

Note: After the upgrade, printing Knowledge Documents can result in a large space inserted after the Resolution section of the document. This space is inserted due to an issue with upgrading document templates from a previous release. For more information about resolving this printing issue, see the *Release Notes*.

Important! After you upgrade, Knowledge Management notification data from previous releases of CA Service Desk Manager uses the r12.5 notification engine. For example, there are default activity notifications and notification rules for object types, such as the Knowledge Report Card. For more information about Support Automation and Knowledge Management using the CA Service Desk Manager notification engine, see the *Release Notes*.

More information:

[The FAST ESP Installation](#) (see page 136)

Chapter 3: Planning

This section contains the following topics:

[CA Service Desk Manager Default and Recommended Port List](#) (see page 63)

[CA MDB Installation Planning](#) (see page 66)

[CA Service Desk Manager Installation Planning](#) (see page 69)

[CA EEM and CA Workflow Installation Planning](#) (see page 82)

[CA IT PAM Integration Planning](#) (see page 89)

[CA Business Intelligence Installation Planning](#) (see page 92)

[CA NSM Installation Planning](#) (see page 97)

[FAST ESP Installation Planning](#) (see page 97)

[Implementation Strategies](#) (see page 99)

[Enable Windows Authentication in Firefox](#) (see page 99)

CA Service Desk Manager Default and Recommended Port List

The CA Service Desk Manager installation requires various ports and port ranges to be opened on your firewall. This port information helps your site and security administrators install and configure CA Service Desk Manager, as well as integrations with other CA solutions and third-party products.

The ports you need to open on your firewall depend on settings in the *NX.env* file. By default, CA Service Desk Manager chooses the appropriate port based on availability. The system reserves ports less than 1024, but can request a port number as high as 65335.

The following *NX.env* variables set the starting port (2100) and the incremental increase (plus 1) the system uses to find an open port for the process starting up:

- `NX_SLUMP_FIXED_SOCKETS=1`
- `NX_SLUMP_SECONDARY_SOCKET=2100`

The following list displays default and recommended ports (and port ranges) for a typical CA Service Desk Manager installation:

Database

- Oracle: 1521
- SQL Server: 1433

CA Service Desk Manager

- FTP: 21
- SMTP: 25

- HTTP: 80
- HTTPS: 8080
- HTTPS (secondary): 8081
- POP3: 110
- IMAP: 143
- LDAP: 389
- WebEx: 1270
- mstsc: 1389
- oaserver: 1706

Note: Port 1706 conflicts with FAST ESP. For more information about the port 1706 conflict, see the *Release Notes*.

- Slump Socket: 2100
- qserver: 2234
- Proctor Socket: 2300
- Communications: 2365
- Apache Tomcat: 8080
- Apache Tomcat Shutdown:
- SSL on Apache Tomcat: 8443

CA CMDB

- Visualizer: 9080
- Visualizer Apache Tomcat Shutdown: 9085
- CA Cohesion ACM: 9000
- CA Cohesion ACM Tomcat Shutdown: 9005

CA EEM

Administration Port: 5250

CA Workflow

- Apache Tomcat: 8090
- Apache Tomcat Shutdown: 8095

CA Business Intelligence

Note: For information about firewall port handling for BusinessObjects, see the BusinessObjects Enterprise XI r2 Deployment and Configuration Guide.

- Apache Tomcat: 8080
 - Recommended:** 8070
- Apache Tomcat Redirect: 8443
- Apache Tomcat Shutdown: 8075
- Secondary Apache Tomcat Shutdown: 8005
- ODBC Driver: 1706
- BusinessObjects Central Management Server (CMS): 8080
- BusinessObjects Application Server: 6400
- ODBC DSN (OpenAccess Database): 1706
- BEA WebLogic: 7001

FAST ESP

- Base Port: 13000
- Administration Interface: 16000 (Base Port plus 3000)
- License Server Port Range: 27000-27009

Note: We recommend that you do not use the port range 23000-27000, as port 27000 is used by the License Server. You cannot change the License Server port range.

Important! FAST ESP uses 4000 ports starting from the Base Port. All 4000 ports should have unrestricted access.

Portal Server

- Apache Tomcat: 8080
- Apache Tomcat Shutdown: 8085
- SSL Functionality: 8443
- Portal_Safe_List: 8444

Support Automation

- Main Server (Socket Server) Internal: 7005
- Main Server (Socket Server) External: 10443
- Socket Proxy Server (Socket Configuration Main Server) Internal: 7005
- Socket Proxy Server (Socket Configuration Main Server) External: 10444
- Message Routing Server (Socket Configuration) External: 10444
- Apache Tomcat: 8070
- Apache Tomcat Shutdown: 8075

CA MDB Installation Planning

To help you plan for a successful CA MDB installation and configuration, use the following information to research and gather information.

- **Research**—Read the *CA Management Database Overview* to become familiar with the CA MDB, determine your deployment strategy, and read about any SQL Server or Oracle issues you may need to be aware of to use the CA MDB with CA Service Desk Manager.
- **SQL Server**—To help ensure that you can configure the product and components on SQL Server, complete the following steps:
 - Enable TCP/IP on the computer on which you want to perform the installation and configuration.
 - Have the following information available:
 - The named instance of the server that is running SQL Server.
 - The SQL Server database user name and password.
 - The SQL Server database port number.
- **Oracle**—Have the following information available:
 - Whether the Oracle database is local or remote.
 - Whether you need to create tablespaces.
 - The Net service name.
 - The DBA user name and password.
 - The data and index tablespace name.

- The complete path for the tablespace.
- JDBC connection information, including the system identifier (SID) and listener port.

Note: If you are not sure what to enter for your database, see the *Server Configuration Online Help*.

More information:

[The CA MDB Installation](#) (see page 102)

CA MDB Considerations

Before you install the CA MDB, consider the following information to ensure a successful implementation:

- **AIX**—On some AIX computers, the maximum allowable space of the ARG/ENV list is too small to install the CA MDB. In this situation, issue the *lsattr -lsys0 -ancargs -Fvalue* command to find the current setting for the maximum size of the ARG/ENV list. If the value returned is less than 50, increase the size with the following command: *chdev -l sys0 -a ncargs=50*

- **Oracle**—Consider the following information:

- Perform an Oracle backup before an CA MDB patch is applied. The backup can be taken by a DBA or by the patch script. The patch script uses the Oracle Recovery Manager (RMAN) to back up the database using OS authentication. Your RMAN configuration may require that archive logging (ARCHIVELOG mode) be enabled in the database. Alternatively, if the DBA has taken a backup, you can suppress the backup using the patch script by specifying “no” for the -ORA_BACKUP parameter.

Note: Patching uses the RMAN command with operating system authentication.

- The Oracle user that runs the CA MDB installation must have the following database administrator privileges assigned:
 - The dba role (connect sys as sysdba; grant dba to installation_user;).
 - The sysdba role (connect sys as sysdba; grant sysdba to installation_user;).
 - The ability to grant privileges to the mdbadmin user to various system tables and views (connect sys as sysdba; grant all privileges on "sys".TABLE_NAME" to installation_user with grant option;). The values to be assumed by TABLE_NAME are: COL\$, DBA_CONSTRAINTS, DBA_CONS_COLUMNS, DBA_INDEXES, DBA_IND_COLUMNS, DBA_OBJECTS, DBA_OBJECT_TABLES, DBA_REGISTRY, DBA_TABLES, DBA_TABLESPACES, DBA_TAB_COLUMNS, DBA_TAB_PRIVS, DBA_VIEWS, DBMS_REGISTRY, EXPDEPACT\$, EXPDEPOBJ\$, EXPPKGACT\$, EXPPKGOBJ\$, KOPM\$, OBJ\$, TS\$, USER\$

- **Remote Clients (UNIX and Linux)**—When the CA MDB is created from a remote client, the CA MDB creation process ends with a return code 95 and an error message *Failed to create the table space* in the installation log unless the directory specified in the tablespace path exists on both the client computer and the Oracle server.
- **CA MDB Component Installation (SQL)**—The CA MDB component installer interface displays a Database Server Name field. This field identifies the local server name or cluster node name, if clustered. Use the network name that is used to connect to SQL Server.
- **Special Characters (UNIX, Linux, and Windows)**—Consider the following information:
 - (UNIX and Linux) The userid name of the database administrator used to create the CA MDB cannot include special characters. In addition, the following restrictions apply:
 - The DBA User and tablespace names support the # and _ special characters. All other special characters are not supported.
 - The DBA Password and MDBADMIN password support the ~, #, %, ^, -, +, _, {, }, [,], :, ., and ? special characters. All other special characters are not supported.
 - The Oracle Tablespace path, MDB Target Directory, MDB Source Directory, and MDB Patch Directory DIR support the ~, %, ^, _, -, +, [,], {, }, :, (including <space> if the path is double-quoted; however, consecutive spaces are not supported) special characters.
 - (Windows) The userid name of the database administrator used to create the CA MDB *cannot* include special characters. In addition, the following restrictions apply:
 - The DBA User and tablespace names support the #, \$, and _ characters. All other special characters are not supported.
 - The DBA Password and MDBADMIN password support the ~, !, #, \$, *, (), _, +, `, -, {}, [], \, :, ', ?, ., /, and @ special characters. All other special characters are not supported.
 - The Oracle Tablespace path, MDB Target Directory, MDB Source Directory, and MDB Patch Directory support the ~, !, #, \$, (), _, +, `, -, {}, and [] (including <space> if the path is double quoted) special characters.

- **Tablespaces**—Consider the following information:
 - The CA MDB uses the system temp tablespace to store temporary tables. At least 50 MB of space should be available for this purpose. If you decide to use existing tablespaces for the CA MDB, then the tablespaces require a minimum of 200 MB of available disk space. If this amount of space is not available, the creation of the CA MDB fails.
 - If the CA MDB creation process is used to create data or index tablespace and the size parameter is non-numeric, a return code of 350 occurs and the CA MDB installation log includes an error (rc=189) that the creation process failed to create the tablespace.
- **Users and Administrative Privileges**—If the CA MDB creation process occurs with a user that has no administrative privileges, the creation process does not work. The CA MDB installation log shows the error *ORA-00942: table or view does not exist*.

More information:

[The CA MDB Installation](#) (see page 102)

CA Service Desk Manager Installation Planning

To help you plan for a successful CA Service Desk Manager installation and configuration, use the following information to research and gather information.

- **Login Permissions**—Complete the following steps:
 - (Windows) Log in as an Administrator and have full Administrative permissions.
 - (UNIX) Log in as the root user and have the correct permissions to the root account.
- **Research**—Complete the following steps:
 - Read *both* the Release Notes and optional readme file (if available). Do not start your installation until you have read and understand that information.

Note: You can find the most current version of the Release Notes, which contain the system requirements, and optional readme file (if available) at <http://ca.com/support>.

- Verify that you have your installation media.

Note: If your computer does not have an appropriate drive for the installation media, copy the media content to the computer on which you want to install CA Service Desk Manager. Then, start the installation. For a remote installation over the network, you can also have to share a drive or folder on the network and then connect over the network to start the installation.

Important! Do not mount the CA Service Desk Manager install DVDs on a Windows XP operating system as some of the installation files are not found, and the install fails.

- Review the certification matrix for a list of third-party software products that have been certified for use with CA Service Desk Manager.

Note: You can find the most current version of the certification matrix at <http://ca.com/support>.

- Read about, and have a basic understanding of, the product components.
- Become familiar with the different types of implementation strategies that the product supports. Consider network availability, usable bandwidth, and responsiveness when deciding which implementation strategy is best for your organization.
- Review the implementation considerations for each component you install.

■ **Installation Home Directory**—Complete the following steps:

- Determine the home directory in which you want to install the product. The default home directory for the product is C:\Program Files\CA\Service Desk.
- Determine the home directory in which you want to install the shared components that the product uses (for example, the Java Runtime Environment (JRE) and Apache Tomcat). The default home directory for the shared components is C:\Program Files\CA\SC.

■ **Database**—Decide which database (either SQL Server or Oracle) to use with CA Service Desk Manager and install the database. Then, complete the following steps:

- **SQL Server.** Enable TCP/IP on the computer on which you want to perform the installation and configuration.

■ **Primary, Secondary Server, and User Configuration**—Complete the following steps:

- To help ensure that you can configure both the primary and optional secondary servers at the end of the product installation, have the following information available:
 - The domain name system (DNS) of the primary and secondary servers.
 - The name or alias of the server.
 - The name or aliases of the object manager to which you want to establish a connection.
 - The name or IP address of the secondary server.

- The slump socket port to be used for communication by all of CA Service Desk Manager.
- The proctor socket port to be used by the secondary server to listen for messages from the primary server.
- To help ensure that you can configure account information for product administration at the end of the product installation, have a privileged and restricted user name and password ready.
- (UNIX and Linux) Manually create your privileged and restricted users. On Windows, these users are automatically created during the product configuration, but on UNIX and Linux, you must manually create these users.
- **Web Interface Configuration**—Complete the following steps:
 - To ensure that you can configure the web interface at the end of the product installation, have the following information available:
 - The web host name and web server.
 - (Windows NT only) The web site to be used by the CA Service Desk Manager server.
 - The Tomcat port number and socket port number for shutdown requests.
- **Browsers and Antivirus Software**—Complete the following steps:
 - (Firefox) Optionally enable Windows authentication (enterprise single sign-on).
 - Temporarily disable any antivirus software, as this slows down the installation. After the installation, make sure you enable your antivirus software again.
- **Web Screen Painter**—Web Screen Painter automatically installs on the primary server when you install CA Service Desk Manager. You can install Web Screen Painter on separate computers using the installation media.
- **File Name Creation**—CA Service Desk Manager requires 8.3 File Name Creation. See your operating system's Help and Support documentation for more information about 8.3 File Name Creation and the disable8dot3 registry entry.

More information:

[Primary or Secondary Server Components](#) (see page 17)

[Implementation Strategies](#) (see page 99)

[Enable Windows Authentication in Firefox](#) (see page 99)

[The CA Service Desk Manager Installation](#) (see page 108)

CA Service Desk Manager Considerations

Before you install CA Service Desk Manager, consider the following information to ensure a successful implementation:

- **Apache Server (UNIX and Linux)**—If you are using the Apache Server on UNIX or Linux, some installations of the Apache Server disable the `OPTION FollowSymLinks` from the root directory. This option disables the default Apache integration for CA Service Desk Manager, because the configuration file uses the `/opt/CAisd` symbolic link created during the CA Service Desk Manager installation. To allow CA Service Desk Manager to work with Apache, you can either allow the `FollowSymLinks` from the root directory (this must be done in the `httpd.conf` file) or modify the `$NX_ROOT/bopcfg/www/CAisd_apache.conf` file to replace the symbolic link with the actual path. In the `CAisd_apache.conf` file, change all occurrences of `/opt/CAisd` to the actual location of CA Service Desk Manager (for example, `/opt/CA/ServiceDesk`). After making and saving this change, stop and start the Apache Server to apply the change.
- **Browsers**—Consider the following information:
 - (Firefox) Consider the following information:
 - You may receive an *Unprivileged Script* error when you use cut, copy, and paste functionality on the HTML editor page in Knowledge Categories. Click OK to view a technical note at mozilla.org, which shows you how to allow a script to access the clipboard.
 - When viewing the content of a file attachment in which the file name contains Latin-1 extended characters, a save as popup appears. You can either save to disk or click open and select an application to open the attachment.
 - (Internet Explorer 6 on Windows 2003) If the internet security level is set to high, the CA Service Desk Manager URL and *about:blank* must be added as Trusted Sites for the web pages to be displayed properly.
 - (Google Toolbar) The CA Service Desk Manager web interface may have a problem displaying the title bar text at the top of the window.
 - (Internet Explorer 6.0) You may experience periodic increased memory use when accessing the CA Service Desk Manager web interface. This is a known issue with the current release of Internet Explorer. To release the memory, periodically minimize your main CA Service Desk Manager web page.

■ **CA Workflow**—Consider the following information:

- If the privileged user credentials are incorrect during configuration, CA Service Desk Manager may run but CA Workflow fails. If these credentials are incorrect, you may experience the following:
 - CA Workflow does not install. A bad username/password error is cited in `/site/Workflow/wekinstall.log`.
 - Tomcat takes excessive CPU, and/or the logfile `/log/pdm_tomcat_CAWF.log` is filled with security manager exceptions and *AbandonedObjectPool is used* messages.

To correct this situation, verify the privileged username and password by logging in to the operating system with those credentials. Then, run the CA Service Desk Manager configuration and specify the correct privileged username/password. If the privileged user password is changed, configuration must be run to set the new password.

■ **Databases (All Supported)**—Consider the following information:

- If CA Service Desk Manager has been configured with one database, and then the configuration is run a second time and a different database type is selected, configuration does not work. For example, if you initially configure for SQL Server and then configure again to an Oracle database. The workaround is to restart the computer before the second configuration is run.
- Database connection information, if different, is not accepted in subsequent configurations. If an additional configuration is needed as a result of a change in the database connection information, delete the `$NX_ROOT\NX.env` file before proceeding.
- If you have upgraded from Argis 8.0 to CA Asset Portfolio Management r11.2 and you are sharing the CA MDB (database) with CA Service Desk Manager, you may encounter problems attempting to add or update assets in asset families using CA Service Desk Manager. To avoid errors, you must define the asset extension tables to CA Service Desk Manager using one of the following methods:
 - Use the Web Screen Painter to define the tables and the forms to view and edit the table entries.
 - Manually edit the tables and forms using the following guidelines:
 - a. The file `$NX_ROOT\bopcfg\majic\assetx.maj` contains a template that can be used to create a majic file to identify the columns in the asset extension table to CA Service Desk Manager. Copy this file and edit as appropriate, following the instructions located in the file.

- b. Create a .sch file in \$NX_ROOT\site\mods directory to define the database columns. The files \$NX_ROOT\site\assetx_schema.sch and \$NX_ROOT\site\assetx_index.sch can be used as templates which can be copied and edited as appropriate for your asset extension tables.

Note: Creating the form to display the asset extension data is accomplished using the Web Screen Painter.

■ **SQL Server**—Consider the following information:

- (SQL Server 2005) SQL Server 2005 has stricter user password credentials than in previous versions of SQL Server. Make sure that the CA Service Desk Manager database password adheres to the password policies as defined in SQL Server. If the CA Service Desk Manager database password does not meet the SQL Server 2005 password policy, the CA Service Desk Manager configuration does not work.
- (Microsoft clustered environment on SQL Server 2005) Within the cluster, you must create an alias for the SQL Virtual node name so when the cluster fails over, CA Service Desk Manager can still connect to the same server name, regardless of which clustered node is active.

■ **Oracle**—Consider the following information:

- If you install CA Service Desk Manager on a UNIX-based system with an Oracle database, the privileged user must belong to the group specified during the Oracle installation. The Oracle installation group can be found in file named *oralnst.loc*.
- When configuring to a 64-bit Oracle database on a 64-bit computer, the system library path (LD_LIBRARY_PATH on Solaris, SHLIB_PATH on HP-UX and LIBPATH on AIX) must point to the 32-bit Oracle libraries. The 32-bit Oracle libraries are found in \$ORACLE_HOME/lib32. This is for both configure and runtime.
- If you are using an Oracle database and you want to use existing tablespaces, you must create a Data tablespace that is at least 400 MB, and an Index tablespace that is at least 100 MB before configuring CA Service Desk Manager.

- **Externally Mounted Drives**—When installing CA Service Desk Manager on Linux with 20 or more externally mounted drives, you may experience a delay during the installation after the installation wizard pages and before the installation progress page appears. This is normal behavior, and the installation should resume after the delay.

- **Install Shield and InstallAnywhere**—Consider the following information:
 - If you receive an Install Shield error when trying to install CA Service Desk Manager, wait until *msiexec.exe* stops running. Then try installing the product.
 - If you start, and then stop, the CA Service Desk Manager installation on UNIX or Linux, you may see a directory at the root of the installation named *install.dir.#####*. This is a feature of InstallAnywhere. These files are not needed and you can safely delete them.
- **Installation Home Directory**—When installing CA Service Desk Manager, do not install the CA Shared Components in the same directory as the CA Service Desk Manager installation directory (NX_ROOT).
- **International**—Consider the following information:
 - You cannot use multi-byte characters for either the user you are logged in as or for the CA Service Desk Manager privileged username when installing on multi-byte operating systems such as Simplified Chinese and Japanese. Doing so causes the installation to fail.
 - Do not specify multi-byte characters in file path names during installation and configuration. Doing so causes one or both to fail.
 - CA Service Desk Manager must run on UTF-8 locale on Linux and UNIX platforms.
 - The Timespan Symbol names that are provided with the default CA Service Desk Manager installation (Administration tab, Service Desk, Application Data, Codes, Timespans) are in English. For example, TODAY, YESTERDAY, THIS MONTH, and so on. For localized versions of the product, administrators may want to define new localized Timespans as required. You should not delete or modify the default Timespans.
 - Date formats in CA Service Desk Manager do not support international specifiers such as, localized date-picture specifiers (for example, "jj/MM/AAAA" for French). The syntax is limited to generic specifiers such as "DD/MM/YYYY". However, many international short date-time patterns can be constructed from these generic specifiers (for example, "YYYY.MM.DD" would supply a common Japanese short date format).
 - For outbound plain-text email notifications, the NX_SMTP_HEADER_CHARSET and NX_SMTP_BODY_CHARSET options may need to be adjusted (directly in the NX.env file) to correctly tag the email message with the character encoding used by the international operating environment, so that it can be properly interpreted and displayed by recipient email systems. The default values for these options are set to UTF-8 on all platforms.

- For non-English Oracle installations running on UNIX or Linux, you must set the NLS_LANG environmental variable before running pdm_init. The Oracle DBA should check the Oracle documentation for the value assignment needed for NLS_LANG.

Important! If you do not set NLS_LANG, the Oracle client defaults to *American_America.US7ASCII*. For example, to test multi-byte Japanese in your environment, set NLS_LANG to *JAPANESE_JAPAN.UTF8*.

Note: The NLS_LANG variable must be set in the same shell before running pdm_init. For example, *setenv NLS_LANG GERMAN_GERMANY.WE8ISO8859P1*

- International users may want to adjust the DateFormat property in web.cfg to use the date and date-time formats best suited for their region.
- International users may want to change from the default spell check lexicon (LEX_LANG option in the Options Manager) to a lexicon matching the language used in their region.
- After the upgrade, Chinese, Japanese, and Korean operating environments must use the FAST ESP Search Engine.
- The English release of the CA Service Desk Manager server is certified to operate in nine non-English language operating environments. For Windows, these environments must be fully localized regional releases of the supported Windows Server operating systems. In particular, Windows operating environments which utilize a Language Interface Pack or MUI running on an English core are not supported.

Further, localized releases of CA Service Desk Manager are supported only on the matching localized Windows server operating environment. In all cases, the system's "Language for non-Unicode programs" (default system Windows ANSI code page) must be accurately configured in the Control Panel's Regional and Language Options window to support the target certified language.

Note: For more information about the localized releases of Windows Server operating systems, see the Localized Microsoft Operating Systems listing on Microsoft's Global Development and Computing Portal.

- For knowledge searches containing multi-byte Japanese characters for International users of CA Service Desk Manager and Knowledge Management to work properly with SQL Server only when SQL Server is installed with Windows collation. Make sure to specify the Collation option for your data during the SQL Server installation.
- **IPv4**—When installing on Linux, the /etc/hosts file must contain an IPV4 entry with the IP address, computer name, and the fully-qualified computer name. An example entry may look similar to the following:

```
127.0.0.1 localhost.localdomain localhost
141.202.211.11 usbegp11.ca.com usbegp11
```

- **Short File Names**—If you have disabled short file names on your Windows operating system, enable them before attempting to install CA Service Desk Manager. In addition to enabling short file names, you must also set both the TEMP and TMP environment variables to a short file name, for example, c:\temp, after enabling short file names before starting the installation process.

Note: For information, see the Microsoft Knowledge Base Article 121007 on the Microsoft Help and Support web site.

- **Special Characters and Spaces (Directory, Media Path, and Folder Name)**—Consider the following information:
 - (UNIX and Linux) If you copy the installation media to a local hard drive, make sure the location does not contain spaces or special characters (for example #) in the directory path. If the location contains spaces or special characters, the installation does not work.
 - (UNIX, Linux, and Windows) Do *not* specify spaces in the installation media path and folder name. If you do, the installation does not work.
- **Tomcat**—Consider the following information:
 - Tomcat is set as the default CA Service Desk Manager web server during the product installation. If you want to use IIS (on Windows) or Apache (on UNIX or Linux) as the default web server, run the installation and select IIS or Apache when prompted, or re-run the configuration and select IIS or Apache.
 - If Tomcat is configured with external authentication on the primary server, you must set up a secondary server with a webengine and repository daemon to allow users who are not authenticated to use attachments. The Tomcat installation on the secondary server cannot use external authentication.
 - The CA Service Desk Manager installation sets the Tomcat port to 8080. Other CA products, such as CA Asset Portfolio Management or the Service Delivery Suite of products, also default the Tomcat port to 8080. If you are installing multiple CA products on the same server, make sure you select a port number other than 8080 for subsequent CA product installations so they function properly together. To change the Tomcat port number to something other than 8080 for CA Service Desk Manager, install the product or if it has already been installed, re-run the product configuration and specify an available port number for Tomcat when prompted.
 - After a restart, the CA Service Desk Manager Tomcat process may not start properly. If this happens, stop and restart Tomcat using the following commands:

```
pdm_tomcat_nxd -c stop  
pdm_tomcat_nxd -c start
```

- **Users and Authentication**—Consider the following information:
 - User authentication does not work if the system is using shadow files and there is an x in the password field of the /etc/passwd file.
 - On HP, if you have configured security so that system passwords are stored in /etc/shadow (for example, an x is stored in /etc/passwd in place of passwords), CA Service Desk Manager user authentication fails, and users are not able to log in to CA Service Desk Manager.
 - The passwords specified for the privileged user and the restricted user must conform to the password policy constraints imposed by the network domain. If they do not meet the constraints imposed by the domain, CA Service Desk Manager configuration does not work.
- **Web Interface and Internet Information Services (IIS)**—To configure the web interface with IIS 7.0 on Windows 2008, you must install the CGI and Metabase Compatibility components of IIS 7.0. You can add these components using the Roles section of the Server Manager, by installing the IIS Management Compatibility modules.
- **Web Screen Painter**—Consider the following information:
 - When you install the Web Screen Painter as part of the CA Service Desk Manager installation, it must be configured for it to work properly.
 - In a Web Screen Painter preview session in test mode, search filters are ignored on new tables that are not published.

More information:

[The CA Service Desk Manager Installation](#) (see page 108)

CA CMDB Components

CA CMDB provides the following standard components:

CA CMDB Web Interface Pages

- Defines and tracks CI properties and relationships.
- Supports change impact and root cause analysis.

CMDB Visualizer

- Graphically displays CI relationships.
- Supports the planning function.

Versioning

- Displays CI change history and relationships.
- Compares snapshots and named milestones.

Common Asset Viewer

- Provides a common view of asset attributes across multiple products.
- Provides the launch point into supporting products.

CI Reconciliation

- Associates imported CI data with existing CIs in the CMDB.
- Uses existing Asset Registry API.

Repository

- Provides a data store for CIs, relationships, families, and so on.
- Exists in the MDB.

Integration

- Provides standard, predefined integration with CA Service Desk, CA NSM, CA Asset Portfolio Management, CA EEM, and others.

Reporting

Provides the following reports for CIs:

- Summary
- Detail
- Changed CIs
- CI Family
- CI Relationship
- Relationship Tree

CA CMDB provides the following optional Adapter components:

- Universal Federation Adapters supports the importing of third-party data into CA CMDB.
- SMS adapter supports the importing of SMS data into CA CMDB.

Support Automation Planning Considerations

You can use the following information to research and gather information to help you plan for a successful Support Automation configuration.

- Read the Release Notes to understand the requirements for Support Automation.

- **Server and Network**—Consider the following supported Support Automation server modes:
 - **Main Server**—Support Automation uses main application server. The server provides socket-based and HTTP-based communications.
 - **Socket Proxy Server**—Support Automation uses a socket proxy on the same tier as the web server which off-loads encryption/decryption processing from the main server for direct socket connections to support scalability.
 - **Message Routing Server (MRS)**—Support Automation separates high bandwidth and unpredictable traffic from the main application server to support server scalability and provide a network routing shortcut for geographical scalability using remote control connections.
- **Server Sizing**—Consider the following server variables:
 - **Network characteristics of end-user and analyst connections**—The server load is directly proportional to the data of the message routing component. Low bandwidth, high latency, and high packet loss contribute significantly to lowering the load on the server. When network conditions are optimal (high bandwidth, low latency, low packet loss), the speed on the server is much higher. The total number of concurrent analyst users and end-user logins per minute, including self-service user, can place a heavy load on the server.
 - **Connection type**—The number of socket connections as opposed to the number of HTTP connections affects the servers as follows:
 - When you connect predominantly through socket connections, the load on the servers is so light that, assuming powerful hardware, the application is network bound rather than CPU bound. The hardware does not limit the number of concurrent connections but rather the network can limit the connections.
 - When you connect through HTTP, the load on the web and application servers is higher and the application is CPU bound unless scaled significantly.
 - **Remote Control usage**—Remote Control uses significant network bandwidth in a sustained way whenever it is running. All traffic routed between end users and analysts flows through the server. The number of concurrent Remote Control connections has a significant role in any sizing assessments.

Note: Remote Control is the only high-bandwidth tool in the Live Assistance toolset. Chat and Automation are low bandwidth. Screenshot and File Transfer can use high bandwidth for short periods while files are transferred.

Support Automation Network and Bandwidth Considerations

The amount of bandwidth you consume on the end-user computer depends on the tools you use as follows:

- For the Chat and Automation features, the amount of bandwidth required is small. A dial-up modem of 56 kbps or less is adequate to support these functions.
- For the Remote Control feature, the amount of bandwidth required increases. However, Live Assistance Remote Control automatically adapts to low-bandwidth environments by reducing the image quality and refresh rate of the remote control session.

The amount of bandwidth also depends on the connection model you employ. Two connection models are available:

- HTTP connectivity—Use in cases where the end user is behind a restrictive firewall that lets only HTTP connections to the server.
- SSL direct socket—Use in cases where the end user connects to the server using a connection on the SSL port 443.

The following chart illustrates the necessary bandwidth depending on the tools you use.

Tools/Bandwidth	Chat/Automation	Remote Control
< 3 KBps (28.8 kbps dial-up)	Very fast and responsive	Slow
< 5 KBps (< 56 kbps dial-up)	Very fast and responsive	Adequate with image degradation
< 50 KBps (Cable/ADSL)	Very fast and responsive	Very fast and responsive
< 100 KBps (LAN)	Very fast and responsive	Very fast and responsive

CA EEM and CA Workflow Installation Planning

To help you plan for a successful CA EEM and CA Workflow installation and configuration, use the following information to research and gather information.

- **Research**—Complete the following steps:
 - Read your CA EEM and CA Workflow documentation, including both the Release Notes and optional readme file (if available).
Note: CA EEM is not a configuration option in CA Service Desk Manager. CA EEM requires a separate installation.
 - Find out if you have an existing CA EEM installation that you can use with CA Workflow. In this situation, do *not* install CA EEM again. Instead, install CA Workflow and reference your existing CA EEM installation.
 - Note your CA EEM administration password; you need the password during the installation.
- **JAVA_HOME System Variable**—Install the Java Runtime Environment (JRE) 1.5 and set JAVA_HOME as a system variable. If you do not set up this system variable, you are prompted for the path to your Java 1.5 installation. You can find the Java 1.5 installation on the installation media in the following location:
 - (Windows) \winsrvr.cd1\CA_tps.nt\JRE_1_5
 - (Linux) /lnxsrvr.cd1/ca_tps.lnx/JRE_1_5
 - (Solaris) /solsrvr.cd1/ca_tps.sol/JRE_1_5
 - (AIX) /aixsrvr.cd1/ca_tps.aix/JRE_1_5**Note:** For information about setting system variables, see your operating system documentation.
- **CA Workflow Installation**—Complete the following steps:
 - Before you try to install CA Workflow, verify that you have installed both CA Service Desk Manager and CA EEM. You cannot install CA Workflow until these products are installed.
 - The Tomcat port number for CA Service Desk Manager defaults to 8080, and 8090 for CA Workflow. If these port numbers are already in use by another product, change the port numbers for CA Service Desk Manager and CA Workflow so you do not have a conflict.

- **CA EEM and AIX**—If you are installing CA EEM on AIX, and you have configured AIX for Network Information Service (NIS), add a user named *dsa* before you install CA EEM and CA Service Desk Manager. In this situation, complete the following steps:

1. Create a user with the user ID *dsa*.
2. Set the *dsa* home directory to `/opt/CA/eTrustDirectory/dxserver`. If the directories do not exist, manually create them.

Note: This step is only required for CA EEM Release 8.3 and earlier. The home directory for CA EEM 8.4 has changed to `/opt/CA/Directory/dxserver` and is automatically created during installation.

3. Make *dsa* the owner of `/opt/CA/eTrustDirectory` and all subdirectories. To make *dsa* the owner, you can use a command similar to the following:

```
chown -R dsa /opt/CA/eTrustDirectory
```

4. Verify that the user exists, the home directory, and ownership of `/opt/CA/eTrustDirectory`.

More information:

[CA Workflow Considerations](#) (see page 84)

[The CA EEM Installation](#) (see page 123)

[The CA Workflow Installation](#) (see page 126)

[Install Standalone CA Workflow IDE](#) (see page 129)

CA EEM Considerations

Before you install CA EEM, consider the following information to help ensure a successful implementation:

- **External LDAP Data Store**—If CA EEM is configured to use an external LDAP data store, the privileged user must be created in the LDAP Directory.

Note: For more information about external LDAP server configuration, see the *Administration Guide*.

- **AIX**—If you are planning to install CA EEM 8.4 or later on AIX, you do not need JRE 1.6. You can run the installation script with an example switch as follows:

```
/aixsrvr.cd1/ca_tps.aix/EEM/EEMServer.sh -javahome none
```

More information:

[The CA EEM Installation](#) (see page 123)

CA Workflow Considerations

Before you install CA Workflow, consider the following information to help ensure a successful implementation:

- **AIX (Primary Server) and the Java Runtime Environment (JRE)**—On AIX for the primary server, CA Service Desk Manager installs the IBM version of the Java Runtime Environment (JRE) in the SC directory. The cryptography functionality in this JRE is limited, and in its current form must be updated to enable the CA EEM authentication feature and CA Workflow. You can download the full-featured cryptography features from the IBM website. Download the IBM SDK Policy Files, otherwise known as Unrestricted JCE Policy files for SDK 1.5.

On the *primary server*, complete the following steps:

1. Shut down CA Service Desk Manager.
2. Copy the following JRE 1.5 folder from the installation media to a folder you specify:

`/aixsrvr.cd1/ca_tps.aix/JRE_1_5/jre`

The CA Service Desk Manager privileged user needs execute permissions on this folder and its contents.

Note: You specify this JRE folder location in NX.env in the following Step 4.

3. Replace the policy JAR files in the lib/security directory using the policy JAR files from the IBM website.
 - a. Navigate to the IBM website at the following location:
`http://www.ibm.com/developerworks/java/jdk/security/50/`
 - b. Click the IBM SDK Policy Files link and follow the download instructions.
 - c. Install the JAR files and replace the policy JAR files in the lib/security directory.
4. Add the following entry to NX.env:

`NX_JRE_INSTALL_DIR_CAWF=<JRE 1.5 folder>`

Note: Replace `<JRE 1.5 folder>` with the location specified in the previous Step 2. For example,

`NX_JRE_INSTALL_DIR_CAWF=/opt/testDVD/aixsrvr.cd1/ca_tps.aix/JRE_1_5/jre`

5. Restart CA Service Desk Manager services.

You are now able to log in to CA Workflow. If necessary, use the CA Workflow design environment (IDE) to load/import manually the process definitions and actors shipped by CA Service Desk Manager. The process definitions and actors can be found in \$NX_ROOT/data/workflow on the primary server.

■ **CA MDB Database User or Password**—Consider the following information:

- If you change either the CA MDB database user or password after the initial CA Service Desk Manager installation, rerun the CA Workflow installation so CA Workflow applies the changes and can communicate with CA Service Desk Manager. When you rerun the CA Workflow installation, CA Service Desk Manager automatically sets the new user name or password for CA Workflow and you are *not* able to change the values manually during the CA Workflow installation.

Note: Oracle environmental variables must be set before installing CA Workflow on UNIX and Linux. If the ORACLE_HOME variable is not set, you receive an error message when clicking the install link.

- If you are migrating to a new database, run the [Workflow Migration Utility](#) (see page 86).

■ **IPv6**—If your Linux server is configured for IPv6 support, you might be unable to log in to the CA Workflow IDE. To correct this situation, modify the ide.sh script file located in \$NX_ROOT\site\Workflow\Client by changing the second line which reads: "VMARGS=" to read: "VMARGS=-Djava.net.preferIPv4Stack=true". Then, save the script file and start the CA Workflow IDE again.

■ **Tomcat**—Consider the following information:

- If you have any difficulties with Tomcat being started, and as a result, the CA Workflow and web interface are not available, increase the Tomcat memory.

Note: For information about how to address this situation, search for the Knowledge Base Article TEC418959 on <http://ca.com/support>.

- During the installation of CA Workflow, CA Service Desk Manager and CA EEM share a separate port number for Tomcat. If you have any difficulties with Tomcat when installing or using CA Workflow, [stop and restart](#) (see page 188) the service.
- If you have installed and changed the installation defaults for the Tomcat port (8080) or the default privileged userid/password (ServiceDesk/ServiceDesk), make the following changes to the default CA Workflow definitions/actors so they work properly.

1. Log on to Workflow IDE.
2. Complete the following steps to modify the default Service Desk Web Service Actor:
 - a. Click the Actors tab.
 - b. Expand the Web Service tree.
 - c. Right-click the Unicenter_Service_Desk_Web_Service actor and select Properties.
 - d. Select and copy the content of the WSDL URL entry.

- e. Modify the Name of the Actor and click OK.
- f. Right-click the Web Service Tree and select Add Actor.
- g. In the Name field, enter Unicenter_Service_Desk_Web_Service.
- h. In the WSDL URL entry, paste the previously copied content.
- i. Modify the port number in the URL to match the Port Number selected during the Tomcat configuration.
- j. Click OK.

More information:

[The CA Workflow Installation](#) (see page 126)

Workflow Migration Utility

The Workflow Data Migration Tool moves data from one CA Workflow database to another. This utility can move data from any of the supported databases to any of the other supported databases, and can move data to the latest CA Workflow schema.

The Workflow Data Migration Tool is provided as both a GUI and as a bean.

- **GUI**—migration.jar
- **Bean**—datamover.jar

Note: To run these Java commands, two JAR files (datamover.jar and migration.jar) must be manually unzipped and extracted into the Java path. The Migration.zip file can be found on the installation media in the \winsrvr.cd1\CA_tps.nt\CAFLOW\Resources directory.

Run the Migration Utility from the Command Line

The migration tool is provided as a bean in datamover.jar. You can run the migration tool from the command line.

Run the migration utility

```
java -cp <driver jar file(s)> com.ca.workflow.util.DataMover <source db driver name> <source db url> <source db username> <source db password> <target db driver name> <target db url> <target db username> <target db password> <number of entries in one transaction> [product keys (comma separated)]-fetchsize=15 [options]
```

Example

```
java -cp ijjdbc.jar;datamover.jar com.ca.workflow.util.DataMover ca.ingres.jdbc.IngresDriver
jdbc:ingres://155.35.26.105:EI7/mdb;cursor=readonly;auto=multi;group=workflow_admin_group root iloveca
ca.ingres.jdbc.IngresDriver
jdbc:ingres://tadra01-w2k3-v:EI7/mdb;cursor=readonly;auto=multi;group=workflow_admin_group tadra01
ca#33333 50 wekProductKey1 --clean-target --fetchsize=15
```

driver jar file(s)

Identifies the source and target driver file(s). Separate multiple driver files using semicolons (;).

source db driver name

Identifies the source database driver.

source db URL

Identifies the source database URL.

source db username

Identifies the source database user.

source db password

Identifies the source database password.

target db driver name

Identifies the target database driver.

target db URL

Identifies the target database URL.

target db username

Identifies the target database user.

target db password

Identifies the target database password.

number of entries in one transaction

Specifies the number of entries to move in one transaction. If this number is too large, then there is a chance to overrun the Ingres transaction log. 50 is the recommended default.

clean-target

Removes all data related to the specified product key(s) from target database before proceeding to transfer data. If no product key is specified, it removes all data from the target database before proceeding to transfer data. This option can be used to clean up a target database if a migration has previously failed midway through running.

fetchsize

If the -fetchsize command line option is present, it specifies the number of rows to be fetched at a time in a scrollable result set. If it is not specified, the utility uses the default fetchsize of the source database. Ingres does not have any default value for the number of rows to be fetched at a time in a scrollable result set; in that case, the default of 10 rows is used if it is not specified as a command line option.

product keys

Specifies product keys for which to migrate data. This is an optional parameter. If specified, it can be either a single product key or a comma separated list of multiple product keys. If the product keys parameter is present, then the migration utility only migrates data from the source database that is constrained by the specified product keys. If this parameter is omitted, all workflow data is migrated regardless of the product key.

Log File

A log file is created when you transfer data from a command line.

Driver File

All driver files should be placed in the migration tool directory, or you must specify the location of the database driver JAR file.

Run the Migration Utility from the GUI

To run the migration from GUI

1. At the command line, enter the following command:

```
java -cp datamover.jar;migration.jar -Dwfllog="c:\wf.log" -Dbxlimit="0" com.ca.workflow.util.WFMigration
```

The Workflow Data Migration Tool dialog appears.

2. For both the Source and Target Database sections, select the database type from the Database Type drop-down menu.

The driver names and URLs update automatically. You can modify these default values, if necessary.

- For Ingres, use -iijdbc.jar
- For Oracle 10g, use ojdbc14.jar
- For Oracle 11g, use ojdbc5.jar
- For MS SQL 2000, use msutil.jar, msbase.jar, mssqlserver.jar
- For MS SQL 2005, use sqljdbc.jar
- For Sybase Adaptive Server Enterprise 12.5.4, use jconn3.jar
- For Sybase Adaptive Server Enterprise 15.0, use jconn3.jar

Important! For a list of supported databases for CA Service Desk Manager, see the *Release Notes*.

3. Complete the appropriate fields.
4. Click Start.

The data migration starts. During the migration, the status updates at the bottom of the dialog.

CA IT PAM Integration Planning

CA IT PAM is a stand-alone CA product with features for automating and tracking hardware and software administration tasks in enterprise IT environments. CA IT PAM automates tasks and manages user interactions, such as approvals and notifications for compliance and accuracy within production environments.

To plan for CA IT PAM integration, consider the following:

- Allow extra time to install and configure the CA IT PAM product.
- Assess whether CA IT PAM and CA Service Desk Manager can coexist on a single server when the server architecture supports both products. When CA IT PAM or CA Service Desk Manager components cannot coexist on the same server, consider installing each product on separate servers.
- Allow for time to verify that both products are installed and working independently.

Note: For information about CA Service Desk Manager and CA IT PAM product requirements, see the *CA Service Desk Manager Release Notes*. For information about the CA IT PAM supported platforms, required steps, and options for installation, see the CA IT PAM installation and configuration documentation.

Security Considerations

In addition to the stated minimum requirements from the Release Notes, consider the following recommendations for the CA IT PAM installation:

- Configure CA IT PAM to use CA EEM as an authentication server. CA EEM eliminates plain text user names and passwords from being passed for authentication purposes.
- If you are using multi-tenancy for CA IT PAM, CA EEM installation is required.

Note: For information about implementing multi-tenancy with CA IT PAM, see the CA IT PAM installation and configuration documentation.

- If you are using multi-tenancy for CA Service Desk Manager CA EEM installation is required.

Note: If you are not using multi-tenancy, CA EEM configuration for CA Service Desk Manager is optional.

- Configure CA IT PAM to communicate using secure communications over HTTPS. HTTPS URLs use SSL/TLS to eliminate plain text exchanges, protecting proprietary and other sensitive data from accidental or malicious disclosure.

Note: For information about configuring CA IT PAM to use HTTPS, see the CA IT PAM installation and configuration documentation.

How to Set Up SSL Communications with CA IT PAM

For security reasons, CA IT PAM implementers may have chosen to install or reconfigure CA IT PAM to require SSL communications. If CA IT PAM is configured to require SSL communications, integrated applications such as CA Service Desk Manager require a certificate from the CA IT PAM keystore for communication.

To set up SSL communications with CA IT PAM, do the following:

1. Configure CA Service Desk Manager options to use the CA IT PAM HTTPS address.
2. Export the CA IT PAM keystore certificate to a file and copy the file to CA Service Desk Manager.
3. Load the certificate file into CA Service Desk Manager using the CA Service Desk Manager `pdm_keystore_mgr` utility.
4. If applicable to your CA Service Desk Manager architecture, update the version control files to deliver the CA Service Desk Manager keystore to all secondary servers.
5. Restart CA Service Desk Manager.

How to Enable Communications When CA IT PAM is SSL Enabled

When CA IT PAM communicates with SSL, you must configure the primary and secondary CA Service Desk Manager servers to communicate with CA IT PAM.

To enable communications when CA IT PAM is SSL enabled, do the following:

1. Verify that you can use CA IT PAM in a browser, without launching CA Service Desk Manager. Record the CA IT PAM URL and use it for reference when you configure the CA IT PAM Workflow options in Options Manager.
2. Log in to CA Service Desk Manager and install or modify the CA IT PAM Workflow options in Options Manager. For each of the following options, use the syntax `https://server:8443` instead of `http://server:8080` for reaching the SSL enabled CA IT PAM application. However, if the CA IT PAM installation uses another port instead of the 8443 SSL port, specify the appropriate port number.

- `caextwf_endpoint`
- `caextwf_processdisplay_url`
- `caextwf_worklist_url`

Note: If the values do not match the actual CA IT PAM installation values, CA Service Desk Manager cannot communicate with CA IT PAM and a runtime error occurs. Verify that the values match the actual CA IT PAM installation values because the CA IT PAM installer might have selected a different port instead of port 8443.

3. On the CA IT PAM server, locate the KEYSTOREID entry in the following file:
C:\Progra~1\ITPAM\server\c2o\config\OasisConfig.properties
4. Copy the KEYSTOREID. Be prepared to paste the KEYSTOREID value as the password after you issue the keytool command.
5. On the CA IT PAM server, issue the following keytool command as one line on the command line:
C:\Progra~1\ca\sc\jre\1.6.0_00\bin\keytool.exe -keystore C:\Progra~1\ITPAM\server\c2o\config\c2o\keystore -export -alias c2o-j -file itpam.cer
The keytool utility prompts you for a password.
6. Paste or type the KEYSTOREID value as the password.
The keytool utility uses the final parameter (-file itpam.cer) to create a file named *itpam.cer*. The *itpam.cer* file contains the necessary certificate information for communications with CA Service Desk Manager.
7. Move the *itpam.cer* file to one of the following locations on the CA Service Desk Manager server:
 - (Windows) %NX_ROOT%\bin
 - (UNIX) \$NX_ROOT/bin
8. Import the CA IT PAM certificate information into CA Service Desk Manager by entering the following command:
(Windows) pdm_perl %NX_ROOT%\bin\pdm_keystore_mgr.pl -import %NX_ROOT%\bin\itpam.cer
(UNIX) pdm_perl \$NX_ROOT/bin/pdm_keystore_mgr.pl -import \$NX_ROOT/bin itpam.cer
The *pdm_keystore_mgr.pl* script generates the keystore file in the following locations:
 - (Windows) %NX_ROOT%\pdmconf\nx.keystore
 - (UNIX) \$NX_ROOT/pdmconf/nx.keystore
9. If your CA Service Desk Manager architecture includes secondary servers, the *nx.keystore* must be delivered to all CA Service Desk Manager secondary servers. Open the *server_secondary.ver* file from one of the following locations:
 - (Windows) %NX_ROOT%\site\server_secondary.ver
 - (UNIX) \$NX_ROOT/site/server_secondary.ver
10. Modify the *server_secondary.ver* for version control by add the following information:
[SSL_Keystore]
filename = "nx.keystore"
directory = "\$NX_ROOT/pdmconf"
component_type = "file"
O_mode = "RW"
g_mode = "RW"
w_mode = "RW"
file_ctl

Note: For information about managing version control, see the *Administration Guide*.

11. Restart CA Service Desk Manager.

The CA Service Desk Manager server can communicate with the SSL enabled CA IT PAM application.

Note: For information about configuring CA IT PAM Workflow options, see the *Online Help*.

CA Business Intelligence Installation Planning

To help you plan for a successful CA Business Intelligence installation and configuration, use the following information to research and gather information.

- **Research**—Complete the following steps:
 - Read the Release Notes to understand the requirements for CA Business Intelligence.
 - Read your CA Business Intelligence documentation, including both the Release Notes and optional readme file (if available).

Note: For a detailed list of supported environments and hardware requirements, see the various Supported Platforms documents on the CA Business Intelligence DVD. These documents include specific version and patch-level requirements for web application servers, web browsers, and operating systems.

- **Installation Home Directory**—Determine the home directory in which you want to install CA Business Intelligence. The default locations are:
 - For windows 32bit computers: C:\Program Files\CA\SC\CommonReporting3
 - For x64 computers: C:\Program Files X(86)\CA\SC\CommonReporting3If this directory does not exist, create it before starting the installation.
- **Antivirus Software**—Temporarily disable any antivirus software scanning on the computer on which you install CA Business Intelligence.
- **Application Server Support**—The CA Service Desk Manager integration with CA Business Intelligence only supports Apache Tomcat as the CA Business Intelligence application server.

Important CA Business Intelligence installation is limited to Windows platforms only for CA Service Desk Manager r12.5. You can, however, integrate CA Business Intelligence with CA Service Desk Manager on all supported operating systems.

More information:

[Reporting Considerations](#) (see page 93)

[The CA Business Intelligence Installation](#) (see page 140)

[How to Configure CA Business Intelligence](#) (see page 154)

[Integrate CA Business Intelligence with CA Service Desk Manager](#) (see page 161)

Reporting Considerations

Before you install CA Business Intelligence, consider the following information to help ensure a successful implementation.

- An installation of CA Business Intelligence r3.0 (BusinessObjects Enterprise Release 3) is required to view the r12.5 reports included in this release.
- CA Business Intelligence r2.1 (BusinessObjects Enterprise Release 2) is not supported.
- You can install CA Business Intelligence r3.0 during the upgrade process from r12.1 to r12.5.
- The installation differs based on the installation type that you select (custom opposed to new).
- The credentials for the BusinessObjects Administrator Account must be defined before running the installer for both new and custom installations.
- The Configuration Management Server (CMS) must be installed on port 6400 (default); otherwise the CA Business Intelligence configuration fails during setup.
- CA Service Desk Manager users must be added to the Administrator's list in CMS before using the reports.
- Some additional configuration steps must be completed if you want to install Apache Tomcat 5.5.25 for use as the application server for Business Objects.
- SAP Business Objects users with an existing installation of Business Objects can install and configure CA Business Intelligence (recommended) or they can use their existing Business Objects installation.
- If you plan to access InfoView with Firefox, the supported version of Firefox must be used.
- It may be necessary to modify the version of the JRE provided with Business Objects to a version that better fits your specific environment.
- Crystal Reports Explorer and Desktop Intelligence are *not* supported even though they can be invoked after the CA Business Intelligence installation.

New Opposed to Custom Installation (Windows)

Several differences exist between a new installation and custom installation. The installation flow differs based on the installation type that you select.

New

Installs all components on one computer. Select this installation type to quickly set up a complete deployment, with all server and client components on a single computer. A new installation provides:

- MySQL as the database to store CMS information.
- Apache Tomcat as the application server.

Note: BusinessObjects Enterprise requires a database to store information about users and groups, security levels, BusinessObjects Enterprise content, and servers. The primary database, which the CMS maintains, is known as the CMS database. During the installation of CA Business Intelligence, you specify the CMS you want to use and enter the required parameters for authentication. For more information about CMS database requirements and preparation, see the *CA Business Intelligence Implementation Guide*.

Custom

Installs the components that you select on the computer. Select this installation type to specify which components to install when performing a distributed deployment, or when adding servers to an existing deployment.

Important! Perform a custom installation if you are installing the BusinessObjects CMS on a Microsoft SQL or Oracle database.

A custom installation provides the options to do the following:

Install MySQL or use a pre-existing CMS, including:

- Oracle
- Microsoft SQL
- Apache Tomcat—You can control the installation of the Apache Tomcat instance.

BusinessObjects Administrator Credentials

The BusinessObjects administrator password must be identified before running the installer for both new and custom installations.

This password must be mixed-case, at least six characters long, and cannot contain the word administrator in any form. It should also contain at least two of the following character types:

- Uppercase
- Lowercase

- Numeric
- Punctuation

BusinessObjects Enterprise XI Application Server

BusinessObjects Enterprise requires an application server to process the server-side scripts that make up web applications.

During the installation of CA Business Intelligence, you specify the application server to use and enter the required configuration parameters. CA Business Intelligence provides the option to install Apache Tomcat 5.0.27, or you can use your existing application server.

If you want to install Apache Tomcat 5.5.25 for use as the application server for BusinessObjects, do the following:

- Install the Java 2 Standard Edition JDK 1.5.
- Set the environment variable \$JAVA_HOME to the JDK 1.5 home directory.
- Install Apache Tomcat 5.5.25 and set the Java Virtual Machine path to the directory in which the Java 2 Standard Edition JRE 1.6 is installed on your system.
- Complete a Custom CA Business Intelligence Install.
- On the Web Server Configuration screen, select "No" when prompted to Install a new copy of Tomcat for BusinessObjects XI.
- Select Tomcat from the list of application servers.
- Enter the location of the Apache Tomcat 5.5.25 web server in the Install Directory field and complete the install.
- After completing the CA Business Intelligence Install, run the CA Business Intelligence Configuration.

Note: If you have already installed CA Business Intelligence with JDK 1.5 (using an external nondefault application server) and you want to employ JRE 1.6 on a client environment accessing the InfoView Web Intelligence tool, do not install the 1.4.2 JRE when prompted. Instead, install JRE 1.6 on the client environment. If applicable, restart your browser after installing JRE 1.6. You can download JRE 1.6 from http://java.sun.com/javase/downloads/index_jdk5.jsp (download Java Runtime Environment).

Existing Installation of BusinessObjects (SAP)

For SAP BusinessObjects users, if you have an existing installation of BusinessObjects, which was not installed through CA Business Intelligence, we recommend for the greatest level of compatibility and supportability that you install and configure CA Business Intelligence. If you prefer to use your existing BusinessObjects installation, however, you can skip the CA Business Intelligence installation steps and proceed to this section: [How to Configure CA Business Intelligence](#) (see page 154).

Note: You must be using BusinessObjects Enterprise Release 3 and have deployed Tomcat as an application server. You also must be licensed for Web Intelligence. Regarding Crystal Reports, you must have a runtime license or greater.

Access InfoView with Firefox

If you plan to access BusinessObjects InfoView with Firefox, you must use Firefox 2 with Java Virtual Machine (Java Runtime) 1.5.0_xx and 1.6.0_02+.

Note: For more information about supported platforms and release levels, see the *BusinessObjects Enterprise XI for Windows Guide*.

Reporting Best Practices

Use the following best practices when installing, maintaining, and using CA Business Intelligence:

- Install and maintain one universe for each CA product. If required, you can build linked universes.
- Before applying service packs, patches, and other updates to your customized universe, back up all your customizations.
- Do not modify the default universe. Instead, copy it and modify the copy. Otherwise, your customizations may be erased when you apply service packs, patches, and other updates.
- Build your own folders.
- If reports stop running, verify that the Central Management Server (CMS) is running.
- Do not overwrite pre-defined reports.
- Always use a pre-defined report as a base to build a custom report. Doing so helps ensure consistent formatting in all reports.

Note: For details about completing these tasks, see your BusinessObjects Enterprise documentation.

CA NSM Installation Planning

To help you plan for a successful CA NSM installation and configuration, use the following information to research and gather information.

- **Research**—Complete the following steps:
 - Read your CA NSM documentation, including both the Release Notes and optional readme file (if available).
 - Ensure that CA NSM is properly installed and configured.

More information:

[The CA NSM Integration Installation](#) (see page 121)

FAST ESP Installation Planning

To help you plan for a successful FAST ESP installation and configuration, use the following information to research and gather information.

- **Research**—Complete the following steps:
 - Read the *Release Notes* and the *FAST ESP Installation Guide* to understand the installation requirements, such as supported hardware, operating systems, and their combinations, as well as the default search capabilities for the FAST ESP search engine.

Important! There is a known issue with installing FAST ESP on a server with Daylight Savings Time enabled. For more information, see the *Release Notes*.
 - Read your FAST ESP documentation to become familiar with the FAST ESP search engine, learn how to configure and deploy it for optimal query performance, and how to properly secure the engine. The FAST ESP installation guide is provided on the CA Service Desk Manager installation media.
- **Installation Home Directory**—Determine the home directory in which you want to install FAST ESP. By default, the home directory for the FAST ESP installation is C:\FastESP (Windows) and /opt/FastESP (Linux).
- **After the installation**—Configure Knowledge Management to use the FAST ESP search engine.

Note: For more information, see the *Administration Guide*.

Important! FAST ESP does not install correctly if there are any traces of a previous or current FAST ESP installation on your server. Be sure to close all FAST ESP processes and uninstall the product before attempting to install FAST ESP.

■ **Server and User Configuration**—Complete the following steps:

- Have the fully-qualified domain name or IP address (recommended) of the dedicated server on which you install FAST ESP.

Important! The server must have a static IP address. If the server does not have a static IP address, do not continue with the installation. Reconfigure the server with a static IP address and then continue with the installation.

- Determine a user name and password for FAST ESP security administration. This user is added to the Administrative group and has special security to log on as a service.

Important! When you log into FAST ESP for the first time, you must use the user name "admin" with no password. After this initial login, you can create users, groups and change the default user password. For more information about the initial FAST ESP login, see the *FAST ESP Home Guide*.

- (Linux) Create your FAST ESP user. On Windows, this user is automatically created during the FAST ESP installation, but on Linux, you must manually create this user.

Important! This FAST user is a local user only; it is *not* a domain user.

- External repository searches only work when the collection is called *site*. If you already have a collection in place, it cannot be used unless its called *site*. This allows documents to be properly indexed.

Note: For information about creating and configuring collections, see the *FAST ESP Configuration Guide*.

- When setting the limit of pure text size of attachments per document, set the max-index-size parameter for the filetext and attstext fields of the index profile between 0 and 2GB. The default value is 8196KB.

Note: For more information about max-index-size, see the *FAST ESP Configuration Guide*.

- **Antivirus Software**—Temporarily disable any antivirus software scanning on the computer on which you install FAST ESP.

More information:

[The FAST ESP Installation](#) (see page 136)

Implementation Strategies

When planning your CA Service Desk implementation strategy, consider the following information, in addition to network availability, usable bandwidth, and responsiveness.

- **Centralized**—Installs and configures all product components on one primary server. This is the default installation. You can implement multiple Object Managers and web engines for load balancing and failover, but your business may outgrow this implementation.
- **Distributed**—Installs and configures product components on servers that are closer to the clients receiving the service. For example, a branch location of a business that has a number of subnets may have many analysts using the Web Client. Placing a secondary server at this branch location reduces the network traffic and response times. Network traffic between the branch location and the primary server location is also reduced because the secondary server performs caching. This type of implementation supports the implementation of multiple Object Managers and web engines for load balancing and failover.
- **Global**—Consists of two or more centralized or distributed implementations known as regions. The primary server of a region replicates minimal information to and from a master region. This allows a single region to have all necessary information about all other regions. This enables an analyst to be aware of tickets from all regions, but only connect to a region when required. This type of implementation is useful when network bandwidth is too limited for a distributed implementation. For example, you may have business locations in different countries with a slow link between them.

Enable Windows Authentication in Firefox

NT LAN Manager (NTLM) authentication allows the login credentials of a Windows user, who is logged in to a Windows domain, to be automatically passed to an IIS web server in the same domain. By default, Windows authentication is not enabled in Firefox. To prepare for your implementation, you should enable Windows authentication in Firefox.

To enable Windows authentication in Firefox

1. Start Firefox.
2. Navigate to the following URL:
about:config
3. Confirm that you want to change the advanced settings.
The Preference Name list appears.
4. Locate and double-click the *network.automatic-ntlm-auth.trusted-uris* preference name:
The Enter string value dialog appears.

5. Specify the list of trusted sites by entering the full URL (for example, `http://web.example.com`) or the server name (for example, `web.example.com`)

Note: Do not enter a trailing slash. If you want to specify multiple servers, separate them with a comma. You can match all the servers in a particular domain, but typical wildcards do not work. For example, you must specify *.example.com* rather than **.example.com*.

6. Click OK.
7. Restart Firefox.

Chapter 4: Installing

This section contains the following topics:

[How to Implement the Software](#) (see page 101)

[The CA MDB Installation](#) (see page 102)

[The CA Service Desk Manager Installation](#) (see page 108)

[The Web Screen Painter Installation](#) (see page 120)

[The CA NSM Integration Installation](#) (see page 121)

[The CA EEM Installation](#) (see page 123)

[The CA Workflow Installation](#) (see page 126)

[ADT Installation](#) (see page 130)

[The CA CMDB Federation Adapters Installation](#) (see page 132)

[The FAST ESP Installation](#) (see page 136)

[The CA Business Intelligence Installation](#) (see page 140)

[Verify the Installation](#) (see page 145)

[The Install Log](#) (see page 145)

How to Implement the Software

You implement CA Service Desk Manager based on a number of factors, such as whether you are upgrading from a previous release, you are installing the product for the first-time, your operating system, your database, the products you want to integrate, and so on. In general, follow these steps to implement the software:

1. Read the implementation considerations that are documented in the *Release Notes*.
2. If you are *upgrading* from a previous version, follow the steps to upgrade the database, console, and CA Workflow.
3. Install the CA MDB.

If your database (SQL Server or Oracle) is installed on a server other than the CA Service Desk Manager primary server, install the CA MDB on the remote database server (that is, the computer on which SQL Server or Oracle is installed). If the CA MDB and primary server are on the same computer, CA MDB automatically installs.

4. Install CA Service Desk Manager on either the primary server or optional secondary server.

You can customize web forms and the schema after installation because Web Screen Painter automatically installs on the server.

5. (Optional) Enable CA CMDB Visualizer during the CA Service Desk Manager configuration.
6. (Optional) Enable Support Automation during the CA Service Desk Manager configuration.
Important! If you want to migrate CA Support Automation r6.0 SR1 eFix5 divisions to tenants, convert this data before enabling and configuring Support Automation in CA Service Desk Manager r12.5.
7. (Optional) Install CA EEM for authentication.
8. (Optional) Install CA Workflow to manage your business processes.
9. (Optional) Install CA Business Intelligence for managing reports with BusinessObjects technology.
10. (Optional) Install the CA NSM integration.
11. (Optional) Install the FAST ESP search engine for use with Knowledge Management.

More information:

[The CA MDB Installation](#) (see page 102)
[The CA Service Desk Manager Installation](#) (see page 108)
[The CA EEM Installation](#) (see page 123)
[The CA Workflow Installation](#) (see page 126)
[The CA Business Intelligence Installation](#) (see page 140)
[The CA NSM Integration Installation](#) (see page 121)
[The FAST ESP Installation](#) (see page 136)
[Verify the Installation](#) (see page 145)
[The Install Log](#) (see page 145)
[Start the Web Interface](#) (see page 153)

The CA MDB Installation

Note: Before you install this component, read the information about how to plan for a successful installation.

Important! Mapping the DVD image using a UNC path is not supported for the CA MDB installer.

If the database (either SQL Server or Oracle) you want to use with CA Service Desk Manager is installed on a server other than the CA Service Desk Manager primary server, manually install the CA MDB on the remote database server (that is, the computer on which SQL Server or Oracle is installed). If the CA MDB and primary server are on the same computer, you do not have to install the CA MDB manually. CA Service Desk Manager automatically installs the CA MDB during the CA Service Desk Manager installation.

Note: If you want to install the Management Database on the CA Service Desk Manager Primary server, you *must* start and run the CA MDB installation on the remote computer hosting the database server.

More information:

[CA MDB Installation Planning](#) (see page 66)

[Install on SQL Server \(Windows\)](#) (see page 104)

[Install on Oracle \(Windows\)](#) (see page 106)

[Install on Oracle \(Linux\UNIX\)](#) (see page 107)

MDB Installations

For MDB installations, the following requirements apply:

- When you perform an MDB Installation to install the SQL database on a different server, the SQL Native Client, Client and Management Tools must be installed on the CA CMDB client.
- For any Oracle database setup, the Oracle Client software also must be installed. You can download Oracle Client Software from the Oracle website.

Find Product Integration and Compatibility Information

You can use information that CA Support Online provides to understand CA MDB compatibility with CA Service Desk Manager and other products.

- *CA Service Accounting and CA Service Catalog Integration Guide*—Describes CA MDB versions and how to determine if you must install a CA MDB compatibility patch so that other products can integrate with CA MDB.
- *CA Management Database Mixed Version Installation*—Describes CA MDB compatibility.

To find integration with CA MDB information

1. Open a browser and go to <http://support.ca.com>.
The CA Support Online page appears.
2. Log in to CA Support Online.
3. Click the Documentation link in the left pane.
The Documentation page appears.

4. Select CA Service Catalog, r12, and US English from the drop-down lists. Click Go.
The CA Service Catalog documentation list appears.
5. Click the PDF link for the Integration Guide J02775-1E.
The *CA Service Accounting and CA Service Catalog Integration Guide* appears.
6. Click the Enabling Integration with CA Products that Use an Earlier Version of the MDB bookmark.
You can use the information to help you integrate products with CA MDB.

To find compatibility information

1. Open a browser and go to <http://support.ca.com>.
The CA Support Online page appears.
2. Log in to CA Support Online.
3. Click Knowledge Base Search in the Support pane.
The Knowledge Base Search page appears.
4. Enter **MDB104 Compatibility** in the search field. Click Search.
Knowledge Base Search Results appear.
5. Click the CA Management Database Mixed Version Installation link.
The *CA Management Database Mixed Version Installation* document appears.

Install on SQL Server (Windows)

If the CA MDB you are using with CA Service Desk Manager is on a different computer than the CA Service Desk Manager server, you *must* install the CA MDB on the *remote database server* (that is, the computer on which SQL Server is installed). If the CA MDB and primary server on the same computer, you do not have to complete these steps. During the CA Service Desk Manager installation, the CA MDB automatically installs.

Note: Do not complete these steps on the CA Service Desk Manager primary server. You *must* start and run the CA MDB installation on the remote computer having the database server.

To install the CA MDB on the remote database server

1. Insert the installation media into your drive.
Important! If your computer does not have an appropriate drive for the installation media, copy the media content to the computer on which you want to install CA Service Desk Manager. Then, start the installation. For a remote installation over the network, you can share a drive or folder on the network, and then connect over the network to start the installation.

2. Select your language from the list, click Select Language.

The Installation Menu appears.

Note: If the Installation Menu does not automatically appear, start the installation by double-clicking the setup.exe file, located at the root of the installation media.

3. Click the Product Installs tab.
4. Click CA MDB.
5. To continue with the installation, follow the on-screen instructions.
6. When prompted, select SQL Server as the database type and enter the appropriate information for SQL Server.

Note: If you are not sure what to enter for SQL Server, see the *Server Configuration Online Help*.

7. Continue following the on-screen instructions to complete the CA MDB installation.
The CA MDB is installed on SQL Server.

More information:

[The Install Log](#) (see page 145)

Install on Oracle (Windows)

If the CA MDB you are using with CA Service Desk Manager is on a different computer than the CA Service Desk Manager server, you *must* install the CA MDB on the *remote database server* (that is, the computer on which Oracle is installed). If the CA MDB and primary server on the same computer, you do not have to complete these steps. During the CA Service Desk Manager installation, the CA MDB automatically installs.

Note: Do not complete these steps on the CA Service Desk Manager primary server. You *must* start and run the CA MDB installation on the remote computer hosting the database server.

To install the CA MDB on the remote database server

1. Insert the installation media into your drive.

Important! If your computer does not have an appropriate drive for the installation media, copy the media content to the computer on which you want to install CA Service Desk Manager. Then, start the installation. For a remote installation over the network, you can share a drive or folder on the network, and then connect over the network to start the installation.

The Installation Menu appears.

Note: If the Installation Menu does not automatically appear, start the installation by double-clicking the setup.exe file, located at the root of the installation media.

2. Click the Product Installs tab.
3. Click CA MDB.
4. To continue with the installation, follow the on-screen instructions.
5. When prompted, select Oracle as the database type and enter the appropriate information for Oracle.

Note: If you are not sure what to enter for Oracle, see the *Server Configuration Online Help*.

6. Continue following the on-screen instructions to complete the CA MDB installation.

The CA MDB is installed on Oracle.

More information:

[The Install Log](#) (see page 145)

Install on Oracle (Linux\UNIX)

If the CA MDB you are using with CA Service Desk Manager is on a different computer than the CA Service Desk Manager server, you *must* install the CA MDB on the *remote database server* (that is, the computer on which Oracle is installed). If the CA MDB and primary server on the same computer, you do not have to complete these steps. During the CA Service Desk Manager installation, the CA MDB automatically installs.

Important! Do not complete these steps on the CA Service Desk Manager primary server. You *must* start and run the CA MDB installation on the remote computer hosting the database server.

Note: Verify that your Oracle environment is correctly set up and accessible.

To install the CA MDB on the remote database server

1. Mount the installation media on your drive.

Important! If your computer does not have an appropriate drive for the installation media, copy the media content to the computer on which you want to install CA Service Desk Manager. Then, start the installation. For a remote installation over the network, you can share a drive or folder on the network, and then connect over the network to start the installation.

2. Navigate to the root directory of the installation media and start the installation by running the following command:

```
sh ./setup.sh
```

3. Select your language from the list, click Select Language.

The Installation Menu appears.

4. Click the Product Installs tab.

5. Click CA MDB.

6. To continue with the installation, follow the on-screen instructions.

7. When prompted, select Oracle as the database type and enter the appropriate information for Oracle.

Note: If you are not sure what to enter for Oracle, see the *Server Configuration Online Help*.

8. Continue following the on-screen instructions to complete the CA MDB installation.

The CA MDB is installed on Oracle.

More information:

[The Install Log](#) (see page 145)

The CA Service Desk Manager Installation

Note: Before you install this component, be sure to read the information about how to plan for a successful installation. In addition, when you install on Linux and UNIX, some pop-up messages cannot view clearly if your color properties are white on white.

When you start and install CA Service Desk Manager on either the primary or secondary server, the following files, components, and features are installed to use the product:

- Primary and secondary server functionality, based on how you configure the product after installation.
- The SQL Server or Oracle Interface
- The ODBC Interface

Important! Installation of the ODBC interface is intended solely for use to access the ODBC driver for Business Objects reporting in CA Service Desk Manager with CA Business Intelligence. Use of the ODBC driver by other applications is not directly supported, certified, or warranted by CA and you use it at your own risk.

- The Web Interface
- CA CMDB

Important! During configuration, when you migrate from CA CMDB stand alone version to CA Service Desk Manager, a Configure CMDB Only check box displays. When you clear the Configure CMDB Only check box and you click Next, you cannot configure CA CMDB again. Even if you click Back, the Configure CMDB Only check box is no longer available. A message warns you of this behavior in the configuration dialog. If you cancel the configuration before it completes and rerun it, the Configure CMDB Only check box is available.

The Configure CMDB Only check box controls the value of the CA Service Desk Manager environment variable NX_CMDB. The environment variable controls whether the Support Automation feature is configured. If the check box is cleared, Support Automation is configureable, otherwise it is not. The environment variable affects the behavior of some Web forms.

- Visualizer
- Support Automation

You install the *supportautomation_url* option after configuring Support Automation during CA Service Desk Manager configuration. For more information about this option, see the *Online Help*.

Important! When you uninstall CA Service Desk Manager, CA Workflow is also uninstalled.

More information:

[CA Service Desk Manager Installation Planning](#) (see page 69)

[Install on SQL Server \(Windows\)](#) (see page 109)

[Install on Oracle \(Windows\)](#) (see page 111)

[Install on Oracle \(Linux\UNIX\)](#) (see page 114)

Install on SQL Server (Windows)

When you start and install CA Service Desk Manager on either the primary or secondary server, the following components and features are installed:

- Primary and secondary server functionality, based on how you configure the product after installation.
- The SQL Server Interface
- The Web Interface
- The ODBC Interface

Important! Installation of the ODBC interface is intended solely for use to access the ODBC driver for Business Objects reporting in CA Service Desk Manager with CA Business Intelligence. Use of the ODBC driver by other applications is not directly supported, certified, or warranted by CA and you use it at your own risk.

To install CA Service Desk Manager on SQL Server

1. Insert the installation media into your drive.

Important! If your computer does not have an appropriate drive for the installation media, copy the media content to the computer on which you want to install CA Service Desk Manager. Then, start the installation. For a remote installation over the network, you can share a drive or folder on the network, and then connect over the network to start the installation.

2. Select your language from the list, click Select Language.

The Installation Menu appears.

Note: If the Installation Menu does not automatically appear, start the installation by double-clicking the setup.exe file, located at the root of the installation media.

3. Click the Product Installs tab.

4. Click CA Service Desk Manager.
5. To continue the installation, follow the on-screen instructions.

The Configuration Wizard appears.

Note: If Internet Protocol Version 4 (IPv4) and Internet Protocol Version 6 (IPv6) hosts coexist on the network, verify the appropriate transition strategies, tools, and mechanisms to support these technologies are in place before you start the product configuration. For information about configuring servers, see the *Administration Guide*.

6. Enter and select the information to configure the product.

Note: For information about the fields that appear in the wizard, see the *Online Help*.

7. Continue following the on-screen instructions to complete the installation.

CA Service Desk Manager is installed on SQL Server.

8. After you install CA Service Desk Manager (or migration to CA Service Desk Manager from a CA CMDB standalone system), run the `cmdb_update_ambiguity` utility. Use the `-h` command to view the mandatory options. For more information about calculating the ambiguity index, see the *Administration Guide*.

Note: If configuration fails during the Validate Extension Tables step, database connectivity can be an issue. Run the configuration again, and verify that you provided the correct database connectivity information.

More information:

[The Install Log](#) (see page 145)

[Verify the Installation](#) (see page 145)

[Start the Web Interface](#) (see page 153)

Install on Oracle (Windows)

When you start and install CA Service Desk Manager on either the primary or secondary server, the following components and features are installed:

- Primary and secondary server functionality, based on how you configure the product after installation.
- The Oracle Interface
- The Web Interface
- The ODBC Interface

You must be the administrator to install CA MDB or CA Service Desk Manager.

Important! Installation of the ODBC interface is intended solely for use to access the ODBC driver for Business Objects reporting in CA Service Desk Manager with CA Business Intelligence. Use of the ODBC driver by other applications is not directly supported, certified, or warranted by CA and you use it at your own risk.

To install CA Service Desk Manager on Oracle

1. Insert the installation media into your drive.

Important! If your computer does not have an appropriate drive for the installation media, copy the media content to the computer on which you want to install CA Service Desk Manager. Then, start the installation. For a remote installation over the network, you can share a drive or folder on the network, and then connect over the network to start the installation.

2. Select your language from the list, click Select Language.

The Installation Menu appears.

Note: If the Installation Menu does not automatically appear, start the installation by double-clicking the setup.exe file, located at the root of the installation media.

3. Click the Product Installs tab.
4. Click CA Service Desk Manager.
5. To continue the installation, follow the on-screen instructions.

The Configuration Wizard appears.

Note: If Internet Protocol Version 4 (IPv4) and Internet Protocol Version 6 (IPv6) hosts coexist on the network, verify the appropriate transition strategies, tools, and mechanisms to support these technologies are in place before you start the product configuration. For information about configuring servers, see the *Administration Guide*.

6. Enter and select the information to configure the Oracle Database.

CA Service Desk Manager requires a Net Service Name specifying the Oracle database where the MDB resides. CA Service Desk Manager also requires a system identifier (SID) for the database. These names may be different, although they specify the same MDB database. Two name values are required for CA Service Desk Manager because it accesses the database with both Oracle client technology and JDBC technology.

Note: For information about Service Names and SIDs, see your Oracle documentation.

Load default data

Specifies whether to load the default CA Service Desk Manager data into the Oracle database.

If this check box is selected, the system loads the CA Service Desk Manager default system data into the Oracle database. If you modified any system default values, this option replaces the values. Select this check box for first-time installations. This option replaces existing data when it is not a first-time installation. If this check box is not selected, no default data loads.

Remote Database

Indicates an Oracle database is installed on a server other than the CA Service Desk Manager primary server. You manually install the CA MDB on the remote database server (that is, the computer on which Oracle is installed) before you run the CA Service Desk Manager Install.

If the Oracle server is local (not selected), CA Service Desk Manager creates the MDB if it is not already created.

Create Tablespaces

Specifies whether to create tablespaces for the MDB database or use tablespaces already created.

- Clear this check box if you already created tablespaces manually. You provide the names of existing tablespaces. The tablespace path field in the product is disabled and the Data Tablespace Name and Index Tablespace Name fields are enabled.
- Select this check box to create tablespaces with the default names MDB_DATA and MDB_INDEX. Provide a tablespace path. The Data and Index tablespace name fields are disabled.

Note: If you are using Oracle Automated Storage Manager (ASM), manually create an Oracle tablespace before you run the CA Service Desk Manager Install. When Oracle is installed and configured for ASM, we are unable to create an Oracle tablespace during our MDB install. ASM uses virtual storage, and we are looking for a physical directory, so we cannot successfully create the tablespace with this configuration.

Net Service Name

Identifies the Net Service Name of the Oracle database where the MDB resides. If the database is remote, use the Net Service Name defined within the Oracle client on the local computer. CA Service Desk Manager accesses the database using a local installation of the Oracle client, which may specify a Net Service Name that is different than the service name on the Oracle server.

mdbadmin User Password

Specifies the mdbadmin user password. Provide the existing password, or specify a new password if CA Service Desk Manager creates the user.

DBA User Name

Specifies the name of an Oracle user with DBA access (usually SYS). This field is only used if the Oracle server is on the local computer.

DBA Password

Identifies the password for the DBA user. This field is only used if the Oracle server is on the local computer.

Data Tablespace Name

Creates the data tablespace name specified when the Create Tablespace check box is selected. This field is only used if the Oracle server is on the local computer.

Index Tablespace Name

Creates the index tablespace name specified when the Create Tablespace check box is selected. This field is only used if the Oracle server is on the local computer.

Tablespace Path

Specifies the directory path to the physical tablespace location created if the tablespaces previously mentioned do not exist. This field is only used if the Oracle server is on the local computer.

Oracle Home Path

Specifies the directory path to the Oracle home path.

JDBC Connectivity

Specifies whether to use JDBC Connectivity. Several components of CA Service Desk Manager use JDBC technology to access the database and require specific information about the Oracle server.

Identify the Database Host Name, SID, and Listener Port that you configured previously.

7. Continue following the on-screen instructions to complete the installation.
CA Service Desk Manager is installed on Oracle.
8. After you install CA Service Desk Manager (or migration to CA Service Desk Manager from a CA CMDB standalone system), run the *cmdb_update_ambiguity* utility. Use the -h command to view the mandatory options. For more information about calculating the ambiguity index, see the *Administration Guide*.

More information:

[The Install Log](#) (see page 145)

[Verify the Installation](#) (see page 145)

[Start the Web Interface](#) (see page 153)

Install on Oracle (Linux\UNIX)

When you start and install CA Service Desk Manager on either the primary or secondary server, the following components and features are installed:

- Primary and secondary server functionality, based on how you configure the product after installation.
- The Oracle Interface
- The Web Interface
- The ODBC Interface

Important! Installation of the ODBC interface is intended solely for use to access the ODBC driver for Business Objects reporting in CA Service Desk Manager with CA Business Intelligence. Use of the ODBC driver by other applications is not directly supported, certified, or warranted by CA and you use it at your own risk.

Note: Verify that your Oracle environment is correctly set up and accessible.

To install CA Service Desk Manager on Oracle

1. Mount the installation media on your drive.

Important! If your computer does not have an appropriate drive for the installation media, copy the media content to the computer on which you want to install CA Service Desk Manager. Then, start the installation. For a remote installation over the network, you can share a drive or folder on the network, and then connect over the network to start the installation.

2. Navigate to the root directory of the installation media and start the installation by running the following command:

```
sh ./setup.sh
```

3. Select your language from the list, click Select Language.

The Installation Menu appears.

Note: When installing on Linux and UNIX, you may not be able to view some pop-up messages clearly, if your color properties are white on white.

4. Click the Product Installs tab.
5. Click CA Service Desk Manager.
6. To continue the installation, follow the on-screen instructions.

The Configuration Wizard appears.

Note: If Internet Protocol Version 4 (IPv4) and Internet Protocol Version 6 (IPv6) hosts coexist on the network, verify the appropriate transition strategies, tools, and mechanisms to support these technologies are in place before you start the product configuration. For information about configuring servers, see the Administration Guide.

7. Enter and select the information to configure the product.

Note: For information about the fields that appear in the wizard, see the *Online Help*.

8. Continue following the on-screen instructions to complete the installation.

CA Service Desk Manager is installed on Oracle.

9. After you install CA Service Desk Manager (or migration to CA Service Desk Manager from a CA CMDB standalone system), run the `cmdb_update_ambiguity` utility. Use the `-h` command to view the mandatory options. The `-d` option is not required in a Linux/Unix environment as it defaults to Oracle.

Note: For more information about calculating the ambiguity index, see the *Administration Guide*.

More information:

[The Install Log](#) (see page 145)

[Verify the Installation](#) (see page 145)

[Start the Web Interface](#) (see page 153)

Visualizer Configuration

You can configure Visualizer during the CA Service Desk Manager installation. When you select the Configure Visualizer check box, the panels for Visualizer configuration display.

When you select the wizard Next button, you validate the configuration information. After a successful validation, the Visualizer configuration task uses the validated Visualizer configuration information.

How to Configure Visualizer on a Secondary Server

You can configure Visualizer on a secondary CA Service Desk Manager server to use Web Services. You start and configure Visualizer on a secondary server during CA Service Desk Manager configuration and by using the *pdm_edit* utility.

Note: When configuring to use secondary web services on a secondary installation of CA CMDB Visualizer, select the server name from the drop-down menu. The default selection is the primary.

To configure Visualizer on a secondary server

1. Select Configure Visualizer in the Configuration Wizard Visualizer page.
2. Select secondary CA Service Desk Manager server as the web host and complete the web host port.

If the configuration is running on a secondary CA Service Desk Manager server, you can use the drop-down menu to select available web service hosts `general.primary_server_node` and the `general.local_host` property values.

3. Enter the CA Service Desk Manager port number for the secondary Visualizer server.

4. Continue following the on-screen instructions to complete the installation.
Visualizer is installed and configured on a secondary CA Service Desk Manager server.
5. Open a command prompt and Navigate to `$NX_ROOT\samples\pdmconf` folder.
Execute the following command:
`pdm_perl pdm_edit.pl`
A menu appears.
6. Select Z to edit the Visualizer Tomcat Servers. Enter A to add a new Visualizer Tomcat Server.
You are prompted to enter a host name.
7. Complete the secondary host name.
The secondary host name is saved and a list of templates appear.
8. Follow the instructions provided by the `pdm_edit.pl` utility.
The secondary visualizer tomcat configuration is complete.

Support Automation Component Configuration

CA Service Desk Manager installs and configures the following Support Automation components:

- End User Client
- Support Automation Analyst Interface
- Server

You install and configure the following components separately:

- End User Agent
- Automated Tasks Editor IDE

Important! You *must* install the `supportautomation_url` option after installing CA Service Desk Manager, in addition to configuring Support Automation during CA Service Desk Manager configuration. For more information about the `supportautomation_url` option, see the *Online Help*.

How to Configure Support Automation Server Modes

You can configure the following Support Automation server modes during the installation:

- **Main Server**—If you select the Main Server Configuration Type, the Host Name or IP field defaults to the local Host Name. All parameters must be provided for the Main Server except the Internal Port section and the Bind to IP in Socket Server section, which are optional.

Important! When you set the *supportautomation_url* option, this URL must use the URL of the Support Automation main server. It should not reference the proxy server or load balancer server.

Note: If you select the Main Server option, and are also planning to configure one or more socket proxy servers, you *must* set the Socket Server host name and external port to the socket proxy host and external port. For multiple socket proxies, you set the Socket Server to the host and external port of the load balancer server.

- **Socket Proxy Server**—If you configured a CA Service Desk Manager Secondary server, you can select the Socket Proxy Server Configuration Type. The default values for fields are displayed. All parameters must be provided for Socket Proxy Server except the Bind to IP field in the Socket Configuration section, which is optional.
- **Message Routing Server**—If you configured a CA Service Desk Manager Secondary server, you can select the Message Routing Server Configuration Type. The default values for fields are displayed. All parameters must be provided for Message Routing Server except the Bind to IP field in the Socket Configuration section, which is optional.

How to Configure Support Automation on a Secondary Server

You can configure Support Automation on a secondary CA Service Desk Manager server.

Note: When configuring Support Automation on a secondary CA Service Desk Manager server, be sure to change the value for the *supportautomation_url* option, if it is currently configured to point to a Support Automation instance on the primary server.

To configure Support Automation on a secondary server

1. Select Support Automation in the Configuration Wizard.
2. Enter the appropriate information for your configuration types on secondary server, such as the host name and port on a Socket Server.

Important! After you define the Support Automation server, its role as primary or secondary server cannot be changed.

3. Continue following the on-screen instructions to complete the installation.
Support Automation is installed and configured on a secondary CA Service Desk Manager server.
4. Open a command prompt and navigate to `$NX_ROOT\samples\pdmconf` folder.
5. Execute the following command:
`pdm_perl pdm_edit.pl`
A menu appears.
6. Enter M to edit the Support Automation Tomcat Servers. Enter A to add a Support Automation Tomcat Server.
You are prompted to enter a host name and server type.
7. Complete the secondary host name and server type.
The secondary host name is saved and a list of templates appear.
8. Follow the instructions provided by the `pdm_edit.pl` utility.
The secondary Support Automation Tomcat configuration is complete.

How to Configure Automated Tasks

You install and configure the Automated Tasks Editor to manage automated tasks that Support Automation analysts use to provide support for end users. The end user can launch an automated task from a knowledge document and the self-service interface, or an analyst executes an automated task during an assistance session. Automated tasks provide analysts with detailed information about an end-user computer. You create self-service automated tasks that interact with the end user and process their input. These tasks can change the file system, registry, download install software, and so on. You configure automated tasks as follows:

1. Install the Automated Tasks Editor.
You launch the installer from the following location on the installation media:
`casd.nt\SAScriptWriter`
Note: You can also copy the installer and deploy it to the appropriate users in your support environment.
The Automated Task Editor is installed.
2. Open the Automated Tasks Editor.
The Automated Tasks Editor installation creates a shortcut on your desktop.

3. Set the following connection parameters:
 - a. Click Tools, Server.
The Server Configuration dialog appears.
 - b. Enter your hostname and port.
Default Port: 8070
 - c. Enter the user name and password of a user with read/write access to the Automated Task Editor, such as a Support Automation Analyst.
 - d. Click Test.
 - e. Click OK.

4. Create automated tasks and upload them to your server.

You can upload public tasks or assign them to specific tenants and subtenants.

Important! Only roles from the Service Provider tenant with the Update Public flag enabled can upload tasks and libraries to the server. All task library content and static content are stored as public data.

The Web Screen Painter Installation

Note: Before you install this component, verify and read the information about how to plan for a successful installation.

Web Screen Painter lets you customize the CA Service Desk Manager schema and web interface. Web Screen Painter is installed on the primary server by default, but you can also install Web Screen Painter on another computer.

Install Web Screen Painter

If you plan to use Web Screen Painter for managing your schema and web interface customizations in CA Service Desk Manager, start and run the installation on a CA Service Desk Manager primary or secondary server.

To install Web Screen Painter

1. Insert the installation media into your drive.

Important! If your computer does not have an appropriate drive for the installation media, copy the media content to the computer on which you want to install CA Service Desk Manager. Then, start the installation. For a remote installation over the network, you can share a drive or folder on the network, and then connect over the network to start the installation.

2. Select your language from the list, click Select Language.

The Installation Menu appears.

Note: If the Installation Menu does not automatically appear, start the installation by double-clicking the setup.exe file, located at the root of the installation media.

3. Click the Product Installs tab.
4. Click Web Screen Painter.
5. Select your language, click OK.
6. Continue following the on-screen instructions to complete the Web Screen Painter installation.

Web Screen Painter is installed and you can customize schema and the web interface.

The CA NSM Integration Installation

Note: Before you install this component, verify and read the information about how to plan for a successful installation.

You can install CA NSM Integration in the following locations:

- CA Service Desk Manager Primary Server
- CA Service Desk Manager Secondary Server
- Standalone server (not CA Service Desk Manager)

CA NSM Integration components are automatically installed on Primary and Secondary servers during the CA Service Desk Manager Server installation. On a standalone server, run the CA NSM Integration component from the CA Service Desk Manager installation menu.

Note: This component is only run on a Windows server that is not a CA Service Desk Manager Primary or Secondary server. For a Primary or Secondary server, CA NSM Integration is installed automatically.

More information:

[CA NSM Installation Planning](#) (see page 97)

[Install the CA NSM Integration \(Windows\)](#) (see page 122)

Install the CA NSM Integration (Windows)

If you are integrating CA NSM and CA Service Desk Manager to control network management issues and coordinate critical management events automatically, start and run the CA NSM integration installation on a CA NSM server that has no other CA Service Desk Manager components on it.

To install the CA NSM integration

1. Insert the installation media into your drive.

Important! If your computer does not have an appropriate drive for the installation media, copy the media content to the computer on which you want to install CA Service Desk Manager. Then, start the installation. For a remote installation over the network, you can share a drive or folder on the network, and then connect over the network to start the installation.

2. Select your language from the list, click Select Language.

The Installation Menu appears.

Note: If the Installation Menu does not automatically appear, start the installation by double-clicking the setup.exe file, located at the root of the installation media.

3. Click the Product Installs tab.
4. Click CA NSM Integration.
5. Follow the on-screen instructions, and when prompted complete the following information:

- Slump Host Name
- Slump Host IP Address
- Slump Port ID
- NSM Repository

The installer creates the NX.env file and a Windows service for starting and stopping the Event Converter.

6. Update the topology.cfg file on the Primary or Secondary CA Service Desk Manager server to reflect the remote IP address of the CA NSM server.

The CA NSM integration is installed and can integrate the two products.

More information:

[How to Integrate with CA NSM](#) (see page 474)

The CA EEM Installation

Note: Before you install this component, verify and read the information about how to plan for a successful installation.

CA EEM authenticates and authorizes users of CA products such as CA Workflow and CA Service Management. Authentication means that a user ID and password, or other information, verify that the user is a valid product user. Authorization means validating that the logged-in user can access particular functionality in the product.

Each time a user tries to log in, CA EEM must authenticate their information; when authenticated, access is either granted or denied. For example, a user must have a CA EEM user record to access the CA Workflow IDE or Worklist application. If you use CA Workflow in CA Service Desk Manager to manage your business processes, first start and run the CA EEM installation on a supported operating environment. Then, immediately install CA Workflow.

Important! If you have an existing CA EEM installation that you can use to authenticate CA products, do not install CA EEM again. Instead, install the CA products and reference your existing CA EEM installation. However, you need CA EEM r8.3 at a minimum before integrating with CA Service Desk Manager r12.5.

More information:

[CA EEM and CA Workflow Installation Planning](#) (see page 82)

[Install CA EEM \(Windows\)](#) (see page 123)

[Install CA EEM \(UNIX\)](#) (see page 124)

Install CA EEM (Windows)

You can start and run the CA EEM installation on a supported operating environment.

Important! If you have an existing CA EEM installation that you can use to authenticate CA products, do not install CA EEM again. Instead, install the CA products and reference your existing CA EEM installation. However, you need CA EEM r8.3 at a minimum before integrating with CA Service Desk Manager r12.5.

To install CA EEM

1. (Optional) Verify that you have installed the Java Runtime Environment (JRE) 1.5 and have set the JAVA_HOME variable.

Note: You can skip this step in CA EEM 8.4 or later, but a warning appears saying that you cannot integrate CA SiteMinder and SAML.

2. Insert the installation media into your drive.

Important! If your computer does not have an appropriate drive for the installation media, copy the media content to the computer on which you want to install CA Service Desk Manager. Then, start the installation. For a remote installation over the network, you can share a drive or folder on the network, and then connect over the network to start the installation.

3. Click Select Language.

The Installation Menu appears.

Note: If the Installation Menu does not automatically appear, start the installation by double-clicking the setup.exe file, located at the root of the installation media.

4. Click the Product Installs tab.
5. Click CA Embedded Entitlements Manager.
6. Continue following the on-screen instructions to complete the CA EEM installation.

CA EEM is installed and you can install other CA products.

More information:

[CA EEM and CA Workflow Installation Planning](#) (see page 82)

Install CA EEM (UNIX)

CA EEM for Linux and UNIX uses a self-extracting shell script that guides you through the installation process. During the installation process, the script displays the license information and prompts for installation parameters. After the installation parameters are entered, the installation begins.

Important! If you have an existing CA EEM installation that you can use to authenticate CA products, do not install CA EEM again. Instead, install the CA products and reference your existing CA EEM installation. However, you need CA EEM r8.3 at a minimum before integrating with CA Service Desk Manager r12.5.

To install CA EEM

1. Run the appropriate installation script on the target computer:

- (Linux) /lnxsrvr.cd1/ca_tps.lnx/EEM/EEMServer.sh
- (AIX) /aixsrvr.cd1/ca_tps.aix/EEM/EEMServer.sh
- (Sun) /solsrvr.cd1/ca_tps.sol/EEM/EEMServer.sh

The file decompresses and the installation begins.

2. Enter Y to accept the Terms and Conditions of the license agreement (or N to decline and abort the installation).

The script prompts for the installation parameters.

3. Enter the installation parameters.

Example:

- a. Enter the installation path for the CA EEM (or accept the default).
- b. If you are installing CA EEM r8.4 or higher, use the following command to avoid setting the JAVA_HOME variable:

```
./EEMServer.sh --javahome none
```

Note: If you are using a previous version of CA EEM, enter the value for the \$JAVA_HOME variable at the prompt. The iGateway installer uses JAVA_HOME to locate the Java Virtual Machine (JVM), which is required for proper operation of CA EEM Server. The installer script prompts you for this variable only if it is not already set in the environment.

A confirmation screen appears with the installation parameter values you entered.

4. If the information about the confirmation screen is correct, enter Y to continue the installation. If you Enter N, the installer exits.
5. Enter the EiamAdmin password.

Note: The default administrator username is EiamAdmin.

The installation procedure depends on the command-line parameters and the type of CA EEM package being installed.

The installer script completes the installation of CA EEM on your computer.

CA EEM Installation Script Parameters

The CA EEM installer accepts the following command-line parameters:

eiampath

Specifies the path where you want to install CA EEM. The default is C:\Program Files\CA\SC\Embedded IAM.

etdirpath [path]

Specifies the path where you want to install CA Directory. The default is C:\Program Files\CA\eTrust Directory.

igpath [path]

Specifies the path where you want to install iGateway. The default is C:\Program Files\CA\SC\iTechnology.

ingpath [directory]

Specifies the path where you want to install Ingres. The default is C:\Program Files\CA\Ingres [EI].

db [database schema]

Specifies CA EEM to use iTechPoz database schema with CA EEM MDB Server for storing policy information.

Example

```
EEMServer_8.3_[builddate]_win32.exe -s -a /z"db=iTechpoz; "
```

The CA Workflow Installation

Note: Before you install this component, verify and read the information about how to plan for a successful installation.

Each time a user tries to log in to CA Workflow, CA EEM authenticates their information, and when authenticated, access is either granted or denied. A user must have a CA EEM user record to access the CA Workflow IDE or Worklist.

After you have installed CA EEM, start and run the CA Workflow installation on a CA Service Desk Manager primary or secondary server to manage your business processes.

Important! If you install CA Workflow, and then uninstall CA Service Desk Manager, CA Workflow is also uninstalled.

CA Workflow logs (*pdm_install_wf.log* and *pdm_tomcat_CAWF.log*) are located in *\$NX_ROOT/log*. If the CA Workflow install is successful, *\$NX_ROOT/site/Workflow/log* contains *wf_admin.log*, *wf_process.log*, *wf_security.log*, and *wl_debug.log*.

More information:

[CA EEM and CA Workflow Installation Planning](#) (see page 82)

[Install CA Workflow \(Windows\)](#) (see page 127)

[Install CA Workflow \(UNIX\)](#) (see page 128)

[Start and Stop CA Workflow](#) (see page 187)

Install CA Workflow (Windows)

If you want to use CA Workflow to manage your business processes in CA Service Desk Manager, start and run the CA Workflow installation on a CA Service Desk Manager primary or secondary server.

To install CA Workflow

1. Insert the installation media into your drive.

Important! If your computer does not have an appropriate drive for the installation media, copy the media content to the computer on which you want to install CA Service Desk Manager. Then, start the installation. For a remote installation over the network, you can share a drive or folder on the network, and then connect over the network to start the installation.

2. Select your language from the list, click Select Language.

The Installation Menu appears.

Note: If the Installation Menu does not automatically appear, start the installation by double-clicking the setup.exe file, located at the root of the installation media.

3. Click the Product Installs tab.
4. Click CA Workflow.
5. Continue following the on-screen instructions to complete the CA Workflow installation.

CA Workflow is installed and you can configure it for use with CA Service Desk Manager.

More information:

[How to Configure the Problem Management Sample Workflow](#) (see page 195)

[CA Workflow Integration](#) (see page 469)

Install CA Workflow (UNIX)

If you want to use CA Workflow to manage your business processes in CA Service Desk Manager, start and run the CA Workflow installation on a CA Service Desk Manager primary or secondary server.

To install CA Workflow

1. Mount the installation media on your drive.

Important! If your computer does not have an appropriate drive for the installation media, copy the media content to the computer on which you want to install CA Service Desk Manager. Then, start the installation. For a remote installation over the network, you can share a drive or folder on the network, and then connect over the network to start the installation.

2. Navigate to the root directory of the installation media and start the installation by running the following command:

```
sh ./setup.sh
```

3. Select your language from the list, click Select Language.

The Installation Menu appears.

4. Click the Product Installs tab.
5. Click CA Workflow.
6. Continue following the on-screen instructions to complete the CA Workflow installation.

CA Workflow is installed and you can configure it for use with CA Service Desk Manager.

More information:

[How to Configure the Problem Management Sample Workflow](#) (see page 195)

[CA Workflow Integration](#) (see page 469)

Install Standalone CA Workflow IDE

You can install the CA Workflow IDE standalone client from the CA Service Desk Manager installation media on Windows and Linux. For UNIX installations, this client allows connection to the UNIX CA Workflow Server. You can point to any CA Workflow server install regardless of operating environment by changing the URL in its login GUI.

CA Service Desk Manager is not required to install CA Workflow IDE. The location of the installation log is different on systems with and without CA Service Desk Manager.

- If CA Service Desk Manager is installed on Windows and Linux, the log is located in the following location:
`$NX_ROOT/logs`
- If CA Service Desk Manager is not installed, the log is located in the following temporary folders:

Windows

`%temp%`

Linux

`/tmp`

To install CA Workflow IDE on Windows or Linux

1. Open the installation media and select CA Workflow IDE.
The CA Service Desk Manager Component Installer Wizard appears.
2. Specify the Workflow Client Install Location
3. Click Next.
The installation completes.

Verify the CA Workflow IDE Installation

You *must* verify that the CA Workflow IDE installation completed successfully.

To verify the CA Workflow IDE installation

1. Navigate to the directory specified in the Workflow Client Install Location field in the CA Service Desk Manager Component Installer Wizard.
2. Verify that the directory contains the following:
 - Folders named "Client" and "uninstall"
 - Files named *version.txt* and *wekinstall.log*

Note: If there are errors in the installation or if the folders and files are in the incorrect locations, view `pdm_install_wf_client.log`.

On Windows, you can launch CA Workflow IDE from the Start Menu by selecting Programs, CA, Service Desk, CA Workflow IDE.

Uninstall the CA Workflow IDE Client (Windows and Linux)

To uninstall the CA Workflow IDE client

1. Run the following command:

```
<install_location>\jre\bin\java" -cp "<install_location>\uninstall\uninstall.jar" run -silent
```

install_location

Specifies the root folder where the CA Workflow IDE client was installed.

■ **Windows Example**

If "c:\test" was the root folder where the CA Workflow IDE client was installed:

```
"c:\test\jre\bin\java" -cp "c:\test\uninstall\uninstall.jar" run -silent
```

■ **Linux Example**

If "/test" was the root folder where the CA Workflow IDE client was installed:

```
"/test/jre/bin/java" -cp "/test/uninstall/uninstall.jar" run -silent
```

2. Delete the root folder where CA Workflow IDE client was installed.
3. (Windows) Delete the associated start menu shortcut.

ADT Installation

Before you install the ADT component, verify and read the information about how to plan for a successful installation.

More information:

[Install ADT](#) (see page 130)

Install ADT

You can configure ADT for your environment.

Note: ADT is not certified on Windows 2000.

To install ADT

1. Log on as administrator to the computer where you want to install ADT and insert the installation media into your drive.

Important! If your computer does not have an appropriate drive for the installation media, copy the media content to the computer on which you want to install CA Service Desk Manager. Then, start the installation. For a remote installation over the network, you can share a drive or folder on the network, and then connect over the network to start the installation.

2. Select your language from the list, click Select Language.

The Installation Menu appears.

Note: If the Installation Menu does not automatically appear, start the installation by double-clicking the setup.exe file, located at the root of the installation media.

3. Click the Product Installs tab.
4. Click Advantage Data Transformer.
5. To continue the installation, follow the on-screen instructions.

The Configuration Wizard appears.

6. Enter and select the information to configure the product.
7. Continue following the on-screen instructions to complete the installation.

ADT is installed.

8. Click Yes to restart your computer and proceed with the Federation Adapters installation.

ADT Patches: Non-UTF-8 Characters

Symptom:

By default, ADT uses the UTF-8 (Unicode) character set when generating the XML input document that GRLoader reads, which results in the creation of an invalid XML document.

The document is invalid because it specifies that it contains UTF-8 characters, but it can contain other characters. When GRLoader attempts to read this invalid document, it generates a UTFDataFormatException in the GRLoader.log and fails to import the data into CMDB.

Solution:

Instead of the UTF-8 character set, some other character set can be more appropriate. To use a non-UTF-8 character set, do the following:

1. Research ISO-8859 to help determine a more appropriate character set.
2. Determine an alternate character set.

ADT Published Solution QO87072 (available from support.ca.com) lets you change the default character set from UTF-8 to another character set.

3. Install the fix and change the Windows registry according to the instructions.

GRLoader can load the data, including the non-UTF-8 characters that you need.

The CA CMDB Federation Adapters Installation

Note: Before you install this component, verify and read the information about how to plan for a successful installation.

The CA CMDB Federation Adapters let you import third-party data into CA CMDB. Before you install the CA CMDB Federation Adapters on your operating environment (SQL Server or Oracle), create an ADT User ID and configure it on your database.

More information:

[Create Administrator ID \(SQL Server\)](#) (see page 132)

[Configure ADT \(SQL Server\)](#) (see page 133)

[Create Administrator ID and Configure ADT \(Oracle\)](#) (see page 134)

[Install CA CMDB Federation Adapters](#) (see page 135)

Create Administrator ID (SQL Server)

When using a SQL Server database, the database administrator ID must have sufficient privileges to create a user. In SQL Server, the user must be assigned either **sysadmin** or **securityadmin** roles. The CMDBAdmin user and the administrator id that is specified during a remote MDB installation do not have sufficient privileges to create the ADT ID (infopump) userid. The default system administrator **sa** can create SQL Server logon IDs.

To create the administrator ID and configure ADT

1. Log on as administrator to the computer where you want to install the CA CMDB Federation Adapters and insert the installation media into your drive.

Important! If your computer does not have an appropriate drive for the installation media, copy the media content to the computer on which you want to install CA Service Desk Manager. Then, start the installation. For a remote installation over the network, you can share a drive or folder on the network, and then connect over the network to start the installation.

2. Select your language from the list, click Select Language.

The Installation Menu appears.

Note: If the Installation Menu does not automatically appear, start the installation by double-clicking the setup.exe file, located at the root of the installation media.

3. Click the Product Installs tab.
4. Click the Install CA CMDB Federation Adapters link.

The initial page appears.

5. To continue the installation, follow the on-screen instructions.

The Configuration Wizard appears.

Note: The ADT administrator id field automatically populates and cannot be changed.

A message informs you that the ADT ID infopump was created.

6. Enter IDB in the Name field. In the Server field, use the drop-down list to select your database server.

Note: If your database server does not appear in the drop-down list, you can manually enter a *hostname* or *hostname\instancename* (for an SQL Server named instance) in the Server field.

7. Continue following the on-screen instructions to complete the installation.
8. Click Finish.

A verification page appears.

9. Click Test Data Source.

The ODBC Data Source that you created is tested. When the test is completed successfully, a success message appears.

Configure ADT (SQL Server)

You must configure ADT before you can install the CA CMDB Federation Adapters. The ADT configuration begins immediately after ADT creates the Administrator ID.

To configure ADT

1. Click OK at the Success screen.
The SQL Server ODBC Data Source is created and the ODBC Login page appears.
2. Enter the user ID and password to log in to the database and click Next.
The ADT Server page appears.
3. Follow the on-screen instructions.
4. Click OK and Close in response to the messages and pages that display.
The Script Manager configuration begins and the ADT Configuration page appears.
5. Click Next to accept the selected script manager.
The Summary page appears.
6. Click Finish.
You can now install the Federation Adapters.

Create Administrator ID and Configure ADT (Oracle)

When using an Oracle database, create an administrator ID and configure ADT.

To create the administrator ID and configure ADT

1. Log on as administrator to the computer where you want to install the CA CMDB Federation Adapters and insert the installation media into your drive.
Important! If your computer does not have an appropriate drive for the installation media, copy the media content to the computer on which you want to install CA Service Desk Manager. Then, start the installation. For a remote installation over the network, you can share a drive or folder on the network, and then connect over the network to start the installation.
2. Select your language from the list, click Select Language.
The Installation Menu appears.
Note: If the Installation Menu does not automatically appear, start the installation by double-clicking the setup.exe file, located at the root of the installation media.
3. Click the Product Installs tab.
4. Click the Install CA CMDB Federation Adapters link.
The Unable to determine database type page appears.
5. Enter Oracle as database type and click OK.
The Create ADT page appears.

6. Continue following the on-screen instructions to complete the installation.
7. Click Finish.

The operation completes.

Install CA CMDB Federation Adapters

The CA CMDB Federation Adapter installation begins immediately after the CA Advantage Data Transformer configuration completes, or you can install it from the installation media.

To install the Federation Adapters on the database

1. Log on as administrator to the computer where you want to install CA Advantage Data Transformer and insert the installation media into your drive.

Important! If your computer does not have an appropriate drive for the installation media, copy the media content to the computer on which you want to install CA Service Desk Manager. Then, start the installation. For a remote installation over the network, you can share a drive or folder on the network, and then connect over the network to start the installation.

2. Select your language from the list, click Select Language.

The Installation Menu appears.

Note: If the Installation Menu does not automatically appear, start the installation by double-clicking the setup.exe file, located at the root of the installation media.

3. Click the Product Installs tab.
4. Click CA Advantage Data Transformer on the Product Install page.

The Welcome page appears.

5. Enter the password for the CA Advantage Data Transformer Administrator ID (infopump).
6. Verify that IDB is selected as the ODBC Datasource and click OK.

The Federation Adapter installation status page shows progress and completion of the installation.

The FAST ESP Installation

Before you install this component, verify and read the information about how to plan for a successful installation.

If you want to integrate FAST ESP with the Knowledge Management functionality of CA Service Desk Manager, install the FAST ESP search engine. Start and run the installation of the FAST ESP search engine on a dedicated stand-alone server. If you install CA Service Desk Manager and FAST ESP on the same computer, always start CA Service Desk Manager before FAST ESP to avoid port conflicts.

Note: For more information, see the *Release Notes*.

Important! For complete details on the installation prerequisites, see the *FAST ESP Installation Guide*, located in \CA_tps.nt\FastESP\Doc\en-US. The FAST ESP documentation covers several installation scenarios, but CA Service Desk Manager only supports FAST ESP integration when it is installed from the CA Service Desk Manager installation media.

More information:

[FAST ESP Installation Planning](#) (see page 97)

[Install FAST ESP \(Windows\)](#) (see page 136)

[Install FAST ESP \(Linux\)](#) (see page 137)

[Install LinguisticsStudio](#) (see page 138)

Install FAST ESP (Windows)

If you want to integrate FAST ESP with the Knowledge Management functionality of CA Service Desk Manager to improve your knowledge searches, start and run the installation of the FAST ESP search engine on a dedicated stand-alone server.

To install the FAST ESP search engine

1. Insert the installation media into your drive.

Important! If your computer does not have an appropriate drive for the installation media, copy the media content to the computer on which you want to install CA Service Desk Manager. Then, start the installation. For a remote installation over the network, you can share a drive or folder on the network, and then connect over the network to start the installation.

2. Select your language from the list, click Select Language.

The Installation Menu appears.

Note: If the Installation Menu does not automatically appear, start the installation by double-clicking the setup.exe file, located at the root of the installation media.

3. Click the Product Installs tab.

4. Click FAST ESP.
5. Continue following the on-screen instructions to complete the FAST ESP installation.

The FAST ESP search engine is installed and you can configure it for use with CA Service Desk Manager in Options Manager.

Note: For more information, see the *Administration Guide*.

More information:

[The FAST ESP Installation Log](#) (see page 138)

[How to Configure FAST ESP](#) (see page 172)

[How to Back Up FAST ESP Data](#) (see page 185)

Install FAST ESP (Linux)

If you want to integrate FAST ESP with the Knowledge Management functionality of CA Service Desk Manager to improve your knowledge searches, start and run the installation of the FAST ESP search engine on a dedicated stand-alone server.

To install the FAST ESP search engine

1. Mount the installation media on your drive.

Important! If your computer does not have an appropriate drive for the installation media, copy the media content to the computer on which you want to install CA Service Desk Manager. Then, start the installation. For a remote installation over the network, you can share a drive or folder on the network, and then connect over the network to start the installation.

2. Navigate to the root directory of the installation media and start the installation by running the following command:

```
sh ./setup.sh
```

3. Select your language from the list, click Select Language.

The Installation Menu appears.

4. Click the Product Installs tab.
5. Click FAST ESP.
6. Continue following the on-screen instructions to complete the FAST ESP installation.

The FAST ESP search engine is installed and you can configure it for use with CA Service Desk Manager in Options Manager.

Note: For more information, see the *Administration Guide*.

More information:

[The FAST ESP Installation Log](#) (see page 138)

[How to Configure FAST ESP](#) (see page 172)

[How to Back Up FAST ESP Data](#) (see page 185)

The FAST ESP Installation Log

When you install the FAST ESP search engine, an installation log file is created to document and list the actions, events, and system changes that occurred during the installation. If the search engine does not install correctly, you can view the errors in the log file to help fix problems so the search engine installs correctly.

You can find the FAST ESP installation log in the `\KTInstallLog` folder of the FAST ESP installation directory. By default, the home directory for the FAST ESP installation is `C:\FastESP` (Windows) and `/opt/FastESP` (Linux). Open the file with a text editor such as Notepad or the vi editor.

Important! A known issue occurs with installing FAST ESP on a server with Daylight Savings Time enabled. For more information about this issue, see the *Release Notes*.

Install LinguisticsStudio

You can use LinguisticsStudio (a product component of FAST ESP) to configure lemmatization for both document indexing and query processing. In addition, you can use LinguisticsStudio to create and edit synonym dictionaries.

Install LinguisticsStudio to configure synonym functionality with FAST ESP. Complete all the following steps on the FAST ESP computer on which you want to install LinguisticsStudio.

To install LinguisticsStudio

1. Install the Java development kit (JDK) 6 (`jdk-6u7-windows-i586-p.exe`).
2. Set the `JAVA_HOME` environment variable to reference JDK 6. For example, `JAVA_HOME=c:\jdk6`.
3. Add the location of JDK 6 to the `PATH` environment variable. For example, `c:\jdk6\bin`.
4. Extract `$FASTROOT/LinguisticsStudio/linguisticsstudio-1.23-win32.win32.x86.zip` into a folder `$ls`.

5. Download the following necessary files for the installation based on `$ls/installer/lingstudio_install.xml`:
 - `eclipse-SDK-3.2.1-win32.zip`
 - `hibernate-3.0.5.zip`
 - `JacORB_2_2_1-compact.zip`
 - `xxl_1_0.zip`
6. After you download all the previous files, copy the files into the `www` folder in your ESP directory on which you want to install LinguisticsStudio. The following are sample folders:
 - `$FASTROOT\www\xxl_1_0.zip`
 - `$FASTROOT\www\JacORB_2_2_1-compact.zip`
 - `$FASTROOT\www\hibernate-3.0.5.zip`
 - `$FASTROOT\www\eclipse-SDK-3.2.1-win32.zip`
7. Update the `lingstudio_install.xml` file to reflect the new URL location of the files (if you are installing on the same computer that is your admin node). The following are sample URL entries from `lingstudio_install.xml`:
 - `<mirror>http://localhost:16000/xxl_1_0.zip</mirror>`
 - `<mirror>http://localhost:16000/JacORB_2_2_1-compact.zip</mirror>`
 - `<mirror>http://localhost:16000/hibernate-3.0.5.zip</mirror>`
 - `<mirror>http://localhost:16000/eclipse-SDK-3.2.1-win32.zip</mirror>`
8. Run the `$ls/setup.cmd` command to deploy the required files based on your updated `lingstudio_install.xml` file.
9. Run the `$ls/linguisticsstudio/linguisticsstudio.exe` command.

More information:

[The FAST ESP Installation Log](#) (see page 138)

[Configure Synonyms](#) (see page 180)

[How to Configure FAST ESP](#) (see page 172)

[How to Back Up FAST ESP Data](#) (see page 185)

The CA Business Intelligence Installation

Note: Before you install this component, verify and read the information about how to plan for a successful installation.

If you want to use CA Business Intelligence with CA Service Desk Manager to customize existing reports or design your own reports, install CA Business Intelligence. You typically start and run the CA Business Intelligence installation on a dedicated stand-alone server.

During the installation, you create a password for the BusinessObjects administrator. This password *must* be mixed-case, at least six characters long, and cannot contain the word *administrator* in any form. We recommend that the password also contains at least two of the following character types:

- Uppercase
- Lowercase
- Numeric
- Punctuation

Important! Apache Tomcat is automatically selected for an Express installation. If you select Custom, you *must* select Apache Tomcat if you intend to use the CA Service Desk Manager Reports tab.

Note: For more information about CA Business Intelligence, see the *CA Business Intelligence Installation Guide*.

More information:

[CA Business Intelligence Installation Planning](#) (see page 92)
[New Install of CA Business Intelligence](#) (see page 140)

New Install of CA Business Intelligence

If you want to use CA Business Intelligence with CA Service Desk Manager to customize existing reports or design your own reports, start and run the CA Business Intelligence installation on a dedicated stand-alone server.

Important! Do not install CA Business Intelligence on the same computer as FAST ESP.

To install CA Business Intelligence

1. Insert the installation media into your drive.

Important! If your computer does not have an appropriate drive for the installation media, copy the media content to the computer on which you want to install CA Business Intelligence. Then, start the installation. For a remote installation over the network, you may also have to share a drive or folder on the network and then connect over the network to start the installation.

The Installation Menu appears.

Note: If the Installation Menu does not automatically appear, start the installation by double-clicking the setup.exe file, located at the root of the installation media.

2. Select your language from the list, click Select Language.

The Installation Menu appears.

3. Click the Product Installs tab.
4. Click CA Business Intelligence Installation.

Important! The CA Business Intelligence Configuration link performs some post-installation and configuration steps, which you must complete only after you install CA Business Intelligence.

5. Continue following the on-screen instructions to complete the CA Business Intelligence installation.

Note: For further installation instructions, see the *CA Business Intelligence Implementation Guide*.

CA Business Intelligence is installed and you can configure it for use with CA Service Desk Manager.

Note: Restart your computer if requested to do so at the end of installation.

More information:

[How to Configure CA Business Intelligence](#) (see page 154)

[Configure Initial CA Business Intelligence Settings](#) (see page 155)

Custom Install of CA Business Intelligence (Windows)

If you use a database other than MySQL, you *must* create the CMS database before running a custom CA Business Intelligence installation.

If Microsoft SQL is your database of choice, do the following:

1. Create a CMS database.

Note: BusinessObjects Enterprise requires a database to store information about users and groups, security levels, BusinessObjects Enterprise content, and servers. The primary database, which the CMS maintains, is known as the CMS database. During the installation of CA Business Intelligence, you specify the CMS you want to use and enter the required parameters for authentication. For more information about CMS database requirements and preparation, see the *CA Business Intelligence Implementation Guide*.

2. If you want to use a separate database for auditing, create a CMS Audit database.
3. Create a user account for the CMS database with *db_accessadmin*, *db_owner*, and *db_securityadmin* as owned schemas and role members.
4. If you created a separate CMS Audit database, create a user account for the CMS Audit database with *db_accessadmin*, *db_owner*, and *db_securityadmin* as owned schemas and role members.
5. Create a DSN for the CMS database and the CMS Audit database (if used).
6. Run a custom installation, and when you are prompted to select a database type, select Use Existing DBMS and Microsoft SQL Server.

Note: A 32-bit ODBC connection is required for CA Business Intelligence 3.0. If you have a 64-bit system, you can easily create a 32-bit ODBC connection by running the 32-bit version of "create dsn." To create this connection, invoke odbcad32.exe from the C:\Windows\SysWow64 directory.

If Oracle is your database of choice, do the following:

1. Install the Oracle database server, and configure the server to use Unicode encoding for the character data types.
2. Verify that the Oracle environment variables are set up correctly.
3. Create a database and user for the CMS database.

Note: You *must* use the same *tns* name, user name, and password specified during the Business Objects installation when you create the CMS and CMS Audit databases.

4. Create a database and user for the CMS Audit database (if you want to use a separate database for auditing).
5. Run a custom installation, and when you are prompted to select a database type, select Use Existing DBMS and Oracle.

Perform a Custom Install

A custom installation lets you select and configure the application features to be installed.

To perform a custom installation of CA Business Intelligence

1. Enter and confirm the BusinessObjects XI Administrator password.
2. Select the CMS database type.

To administer the CMS database, set up a database account for CA Business Intelligence as follows:

Microsoft SQL or Oracle:

- Create or select a user account that provides BusinessObjects Enterprise with the appropriate permissions to access your database server.
- Verify that you can log on to your database and perform administrative tasks using the account defined for use by the CMS.

For MySQL

- If you want to install MySQL when you install CA Business Intelligence, you are prompted during installation to set up this account.
3. Specify the database connection properties.

Microsoft SQL Server:

You set up this database before installation. Then, create a Data Source using ODBC Data Source Administrator on the System DSN tab for this connection.

- Click Microsoft SQL Server, and then click Next.
- Enter the DSN Name.
- Enter the Database Name.
- Enter the User Name.
- Enter and confirm the Password.

Repeat these steps for the auditing database, if you plan on using one and then click Next; otherwise, click Next. Selecting "Use the same setting for Audit database" applies the CMS database connection setting to the auditing database.

For Oracle

- Click Oracle.
- Enter the Oracle *tnsname* in the TNS Name field.
- Enter the credentials for the server in the User Name and Password fields.

- Repeat these steps for the auditing database if you plan on using one and then click Next; otherwise, click Next. Selecting “Use the same setting for Audit database” applies the CMS database connection setting to the auditing database.

For MySQL

- Click MySQL.
- Enter the database name in the Database field for the CMS database.
- Enter the host name in MySQL hostname field.
- Enter the port that MySQL uses in the Port field.
- Enter the credentials for the server in the User Name and Password fields.
- Repeat these steps for the auditing database, if you plan on using one and then click Next; otherwise, click Next. Selecting “Use the same setting for Audit database” applies the CMS database connection setting to the auditing database.

4. Specify the web server configuration. CA Business Intelligence provides the option to install its own version of Apache Tomcat, or you can use your existing application server.

For Apache Tomcat

- Click Tomcat.
- Enter the location of the existing Apache Tomcat installation.

Note: When Tomcat is already installed, the installer only prompts for this information.

- Enter the Tomcat Connection Port.

Note: We recommend that you change the default *connection* port number to avoid interference with other applications using this common port number. The recommended connection port for a typical installation is 8070.

- Enter the Tomcat Redirect Port.
- Enter the Tomcat Shutdown Port.

Note: We recommend that you change the default *shutdown* port number to avoid interference with other applications using this common port number. The recommended shutdown port for a typical installation is 8075.

5. Specify whether to install sample database and templates.
6. Review the settings and install CA Business Intelligence.

Proceed to the chapter in the *CA Business Intelligence Implementation Guide* related to the installation type you selected.

Verify the Installation

After you install CA Service Desk Manager, use the following information to verify that the installation was successful.

1. Verify that a system environment variable for the path is set for the product to the installation directory you specified. The default home directory is C:\Program Files\CA\Service Desk Manager.
2. Verify the following:
 - a. In the Control Panel (Add or Remove Programs), verify that an entry appears for the product.
 - b. From the Start menu, verify that an option appears to:
 - View the documentation.
 - Start the Configuration Wizard.
 - Start the Web Interface.
 - Contact Technical Support.
 - Start the Web Screen Painter.
 - Uninstall CA Service Desk Manager.
3. Verify that you can start the Web Interface.

Important! If you cannot verify this information, the product has not been installed correctly. In this case, start the CA Service Desk Manager installation again to modify the installation.

More information:

[Start the Web Interface](#) (see page 153)

The Install Log

When you install CA Service Desk Manager, an installation log file is created to document and list the actions, events, and system changes that occurred during the installation. If the product does not install correctly, you can view the errors in the log file to help fix problems so the product installs correctly.

You can find the *ServiceDesk_r12_5_Install.log* file in the \log folder of the installation directory. For Remote Components installations, the log is located in the %TEMP% directory. Open the file with a text editor such as Notepad or the vi editor.

Note: If you cancel the installation before it is finished, the installation log is created on your desktop (Windows) or on the root directory (UNIX and Linux).

Chapter 5: Configuring

This section contains the following topics:

[Product Configuration](#) (see page 147)
[Configure the CA Service Desk Manager Components](#) (see page 148)
[How to Configure the Web Interface](#) (see page 151)
[How to Configure CA Business Intelligence](#) (see page 154)
[Run the Automated Policies](#) (see page 171)
[Run the Knowledge Report Card](#) (see page 172)
[Knowledge Management Sample Data Import](#) (see page 172)
[How to Configure FAST ESP](#) (see page 172)
[How to Configure CA Workflow](#) (see page 187)
[How to Configure CA IT PAM Workflow Integration](#) (see page 201)
[Set Up Automatic Login \(CA MDB Authentication\)](#) (see page 207)
[Configure Single Point of Entry](#) (see page 208)
[How to Implement Multi-Tenancy](#) (see page 209)

Product Configuration

After you install CA Service Desk Manager and any additional products you select, there are configuration steps that you must complete so that the products work together correctly. To configure the product, complete the following steps:

- Configure the CA Service Desk Manager components (primary and secondary servers, the database, the web interface).
- Configure the web interface when the web server and primary server are on different computers.
- Configure Support Automation.
- Implement Knowledge Document Life Cycle Reports for Automated Policies.
- Configure CA Business Intelligence.
- Configure CA Workflow.
- Configure FAST ESP.
- Implement multi-tenancy.

More information:

[Configure the CA Service Desk Manager Components](#) (see page 148)

[How to Configure the Web Interface](#) (see page 151)

[Run the Automated Policies](#) (see page 171)

[How to Configure CA Business Intelligence](#) (see page 154)

[How to Configure the Problem Management Sample Workflow](#) (see page 195)

[How to Configure FAST ESP](#) (see page 172)

[How to Implement Multi-Tenancy](#) (see page 209)

Configure the CA Service Desk Manager Components

If you do not configure CA Service Desk Manager during the installation, or if you manually configure the product after installation, you can use the product to configure the primary and secondary servers, the database, the web interface, and additional configuration options.

To verify that you can successfully configure the product and components on SQL Server, enable TCP/IP on the computer on which you want to perform the installation and configuration.

Note: If Internet Protocol Version 4 (IPv4) and Internet Protocol Version 6 (IPv6) hosts coexist on the network, verify the appropriate transition strategies, tools, and mechanisms to support these technologies are in place before you start the server configuration. For information about configuring servers, see the *Administration Guide*.

To configure the product components

1. Select Start, Programs, CA, Service Desk, Configuration.

The Configuration Wizard appears.

2. Enter and select the information to configure the product.

Note: For information about the fields that appear in the wizard, see the *Server Configuration Online Help*.

3. Continue following the on-screen prompts to complete the product configuration.

A default value that works for the TCP service number on most installations is displayed the first-time you run the configuration. To determine the TCP service number at your installation, open a telnet session from your Windows workstation to the server:

- If your network is using NIS, enter the following:

```
ypcat services | grep slump.
```

The output includes a line similar to the following:

```
slump nnnn/tcp #This is required for slump to work!
```

- If your network is not using NIS, enter the following:
`grep slump /etc/services.`

The output includes a line similar to the following:

```
#slump nnnn/tcp
```

Enter the number *nnnn* in the TCP Service Number field.

Note: If configuration fails during the Validate Extension Tables step, database connectivity can be an issue. Run the configuration again, and verify that you provided the correct database connectivity information.

Set Up the CA CMDB Audit Log

The object and trigger definitions, attributes, and html forms that CA CMDB uses for the audit log have changed in the product.

To set up the CA CMDB r12.5 audit log

1. Remove the `cmdb_write_audit_log_site` trigger if you have created `site/mods/extension.mod` (*extension* specifies the extension name).
In this release of the product, auditing is automatically created and enabled.
2. Add 'UI_INFO("AUDIT_LOG")' to each attribute that you want to log.
3. Use the new templates provided by the product to migrate your existing HTML forms.

Note: For more information about attributes and HTML forms, see the *Administration Guide*.

How to Integrate CA Cohesion ACM With CA CMDB

You can integrate CA Cohesion ACM with CA CMDB using a patch. Contact CA Support to obtain the patch specific for your release of CA CMDB and CA Cohesion ACM.

CA CMDB Visualizer Configuration on AIX

Valid on IBM AIX

CA Service Desk Manager installs CMDB Visualizer by default on all operating environments. You can configure CA CMDB Visualizer if necessary. IBM AIX requires additional security policy files.

To configure CMDB Visualizer on IBM AIX

1. Verify that CMDB Visualizer is configured.
2. Download unrestricted policy files (version 1.4.2 or later) from the Unrestricted JCE policy files page at the IBM website

Note: Register on the IBM website to download the policy files.

3. Replace the local_policy.jar and US_export_policy.jar files in your Shared Components JRE directory (default location: /opt/CA/SC/JRE/1.6.0/lib/security) with the policy files that you downloaded from the IBM website.

4. Stop and start Visualizer using the following commands:

```
pdm_tomcat_nxd -c STOP -t VIZ  
pdm_tomcat_nxd -c START -t VIZ
```

CMDB Visualizer is configured on IBM AIX.

Modify Third-Party Scripts for CA CMDB Compatibility

For scripts in the current product release, the ext_asset attribute is renamed to ID. For CA CMDB compatibility with other CA products that use the ID attribute, you can modify third-party scripts that use CA Service Desk Manager web services to update CA CMDB extension tables.

To modify third-party scripts for CA CMDB

1. Open the third-party script that you want to modify.
2. Replace all SQL references of ext_asset with ID.

The script is compatible with the current product release.

How to Switch the Target Server for CA CMDB Reports

Typically, a single CA Cohesion ACM system exports CI data to a single CA CMDB server to create CA CMDB reports. You can switch the target CA CMDB server for the export of CI data by doing the following:

1. Use a CA CMDB server as the target for exporting CI data, and run CA CMDB Reports.
2. Restart the CA Cohesion ACM Server service.
3. Switch to a different target CA CMDB server for exporting CI data, and run CA CMDB Reports.
4. (Optional) Repeat Steps 2 and 3.

How to Configure the Web Interface

When you install CA Service Desk Manager, the *web interface* (commonly referred to as the *browser interface*) is automatically installed and you can configure the web interface as part of the installation. When the web server and the primary server are on the same computer, no additional action is necessary. However, if you are using a Windows web server and the primary server is installed on a different computer, install and configure two servers, a primary server and a secondary server.

The secondary server is the Windows computer on which the web server resides and where you plan to install and configure the web interface. Install this server *after* you install the primary server.

Important! By default, Tomcat is the default web server. If you want to use IIS as your web server, manually configure the product and select IIS. For information, see the *Server Configuration Online Help*.

To configure the web interface, complete the following steps:

1. (Required) Enable the web engine on the secondary server.
2. (Required) Configure the web interface.

You can then start the web interface.

More information:

[Enable the Web Engine on the Secondary Server \(Windows\)](#) (see page 151)

[Enable the Web Engine on the Secondary Server \(UNIX\)](#) (see page 152)

[Configure the Web Interface](#) (see page 153)

[Start the Web Interface](#) (see page 153)

Enable the Web Engine on the Secondary Server (Windows)

After the CA Service Desk Manager primary server is installed and configured, you must enable the web engine on your secondary server.

To enable the web engine on the secondary server

1. On the computer on which you installed the primary server, navigate to the *installation-directory\samples\pdmconf* directory.
2. Use the following command to start the pdm_edit utility:

```
pdm_perl pdm_edit.pl
```
3. Follow the on-screen instructions based on your requirements.
4. On the main menu, enter W to select the Edit Web Engines option.

5. Enter A to add, then enter the IP address or DNS name of the secondary server when prompted for a host name.
Important! This value is case-sensitive, so be precise if you enter a DNS name. If you are unsure, check the `NX_LOCAL_HOST` entry in the `NX.env` file on the secondary server.
6. When prompted for an object manager and a configuration file, you can typically accept the default values. If necessary, change the values.
7. Press the Enter key to return to the main menu.
8. Enter X to create a file named `pdm_startup.rmt` that stores your new configuration values and exit.
9. On the computer on which you installed the primary server, create a backup of the `pdm_startup.tpl` file in the `installation-directory\pdmconf` directory.
10. Replace this file with the `pdm_startup.rmt` file you previously created.
11. Use the Configuration Wizard on the primary server without making any changes, but configure the secondary server. For information, see the *Server Configuration Online Help*.

Note: Your new configuration settings take effect the next time you start the CA Service Desk Manager server.

Enable the Web Engine on the Secondary Server (UNIX)

After the CA Service Desk Manager primary server is installed and configured, you must enable the web engine on your secondary server.

To enable the web engine on the secondary server

1. On the computer on which you installed the primary server, navigate to the `$NX_ROOT/samples/pdmconf` directory.
2. Use the following command to start the `pdm_edit` utility:

```
pdm_perl pdm_edit.pl
```
3. Follow the on-screen instructions based on your requirements.
4. On the main menu, enter W to select the Edit Web Engines option.
5. Enter A to add, then enter the IP address or DNS name of the secondary server when prompted for a host name.
Important! This value is case-sensitive, so be precise if you enter a DNS name. If you are unsure, check the `NX_LOCAL_HOST` entry in the `NX.env` file on the secondary server.
6. When prompted for an object manager and a configuration file, you can typically accept the default values. If necessary, change the values.

7. Press the Enter key to return to the main menu.
8. Enter X to create a file named *pdm_startup.rmt* that stores your new configuration values and exit.
9. On the computer on which you installed the primary server, create a backup of the *pdm_startup.tpl* file in the `$NX_ROOT/pdmconf` directory.
10. Replace this file with the *pdm_startup.rmt* file you previously created.
11. Use the Configuration Wizard on the primary server without making any changes, but configure the secondary server. For information, see the *Server Configuration Online Help*.

Note: Your new configuration settings take effect the next time you start the CA Service Desk Manager server.

Configure the Web Interface

If the default configuration specified for the web interface during the CA Service Desk Manager installation does not meet your requirements, modify the *web.cfg* file, located in the *installation-directory\bopcfg\www* directory. Edit the file using a text editor that does not add formatting or control characters, such as Notepad or WordPad.

Each entry in the file consists of a single line containing a property name, optionally followed by a value. Lines beginning with a pound sign (#) are treated as comments and are ignored.

Note: For information about the entries in the *web.cfg* file, see the *Administration Guide*.

Start the Web Interface

Before you can start the web interface and use CA Service Desk Manager, you must ensure that the Daemon Server services and the database server are started. If you have configured a secondary server (for example, if you have the web interface installed on a web server that resides on a different computer than the primary server), the Remote Daemon Proctor service must be running before you start the primary server service.

- (Windows) To start the services, start the Control Panel (Administrative Tools, Services). Then, right-click CA Service Desk Manager Remote Daemon Proctor and select Start.
- (Linux) To start the services, open a command prompt and execute the *pdm_client* command.

After starting the services, you can start the web interface. How you start the web interface depends on whether the computer from which you start the web interface is a primary or secondary server, and whether you are using IIS. You can also start the web interface from an internal web site.

- To start the web interface, select Start, Programs, CA, Service Desk, Service Desk Web Client.
- To start the web interface from a computer that is not the primary server or a secondary server, open a web browser and enter the following URL:

`http://servername:8080/CAisd/pdmweb.exe`

In this URL, *servername* is the name of the computer that is hosting the CA Service Desk Manager web server.

- To start the web interface from a computer that is not the primary server or a secondary server, and you are using IIS as your web server, open a web browser and enter the following URL:

`http://servername/CAisd/pdmweb.exe`

- To start the web interface from an internal website, add */pdmweb.exe* to the URL for your web pages. Use the following sample HTML code as a guide:

A HREF=`http://<server-name>:<port-no>/CAisd/pdmweb.exe`

In this URL, *server-name* identifies your computer and *port-no* is the port on which your web server is listening.

Note: If your Internet Explorer browser's security is set to high, a content warning message appears when you start the web interface. To avoid this message, add the website to your trusted sites, or lower your security settings.

How to Configure CA Business Intelligence

After you install CA Business Intelligence, you configure CA Business Intelligence so it works correctly with CA Service Desk Manager.

1. Configure initial CA Business Intelligence settings. This step loads the CA Service Desk Manager universe and reports, creates groups, and optionally creates one user for each group and establishes group authorizations.
2. Add your CA Service Desk Manager users and groups to the BusinessObjects Central Management Console (CMC). This step lets you control end user access to InfoView and other BusinessObjects applications.

Note: For more information about adding users and groups and configuring data partitions security, see the *Administration Guide*.

3. Add the default CA Service Desk Manager Privileged User Account to the CMC.
4. Modify default security and Web Intelligence settings in CMC.
5. Integrate CA Business Intelligence with CA Service Desk Manager. This step lets you specify Web Reporting options in Options Manager.
6. Configure Trusted Authentication for BusinessObjects and CA Service Desk Manager. The authentication process allows users to log in without providing passwords more than once during a session.
7. (Optional) Configure BusinessObjects LDAP Authentication.
8. (Optional) Connect the CA Business Intelligence server to a different CA Service Desk Manager server.
9. (Optional) Change the maximum size for a list of values.
10. (Optional) Change the report record limits.
11. (Recommended) Change the Web Intelligence time-out value.

Configure Initial CA Business Intelligence Settings

After you install CA Business Intelligence, there is a post-installation and configuration step that must be completed, so CA Business Intelligence works correctly with CA Service Desk Manager. This step loads the CA Service Desk Manager universe and reports, creates groups, and optionally creates one user for each group and establishes group authorizations.

To configure initial CA Business Intelligence settings

1. Complete the following step:
 - Insert the installation media into your drive. If the Installation Menu does not automatically appear, start the installation by double-clicking the setup.exe file, located at the root of the installation media.

Important! If your computer does not have an appropriate drive for the installation media, copy the media content to the computer on which you want to install CA Business Intelligence. Then, start the configuration. For a remote configuration over the network, you can share a drive or folder on the network and then connect over the network to start the configuration.
2. Select your language from the list, click Select Language.

The Installation Menu appears.
3. Click the Product Installs tab.

4. Click CA Business Intelligence Configuration.
5. Complete the following fields on the CA Business Intelligence configuration:

BI Admin User Name

Specifies the user name of a CA Business Intelligence user who belongs to the Administrators group. The user name is Administrator for a newly installed CA Business Intelligence installation.

BI Admin Password

Specifies the password for the CA Business Intelligence administrative user.

Service Desk Admin User Name

Specifies the user name for the CA Service Desk Manager Privileged user.

Service Desk Admin Password

Specifies the password for the CA Service Desk Manager Privileged user.

Create Default Users

Adds a set of CA Service Desk Manager groups to CA Business Intelligence that matches the CA Service Desk Manager roles. Select this check box if you want to use a sample user for each of these groups.

If you installed CA Business Intelligence on a different computer than CA Service Desk Manager, the following fields appear on the CA Business Intelligence configuration:

Service Desk Primary Host

Provide the host name of the CA Service Desk Manager primary server.

ODBC Port

Specifies the port number of the CA Service Desk Manager ODBC driver (1706 is recommended).

ODBC Install Location

Specifies the location where the CA Service Desk Manager ODBC driver is installed.

6. [Verify](#) (see page 157) the CA Business Intelligence configuration.
7. (Optional) [Connect](#) (see page 166) CA Business Intelligence server to a different CA Service Desk Manager server.

Verify the CA Business Intelligence Configuration

After you configure initial settings, verify the configuration of CA Business Intelligence.

To verify successful configuration of CA Business Intelligence

1. Start InfoView.
2. Log in with the following information:
 - **System**—Specify the hostname of the server where CA Business Intelligence was installed.
 - **User Name**—Specify the name of the CA Business Intelligence administrative user (typically Administrator).
 - **Password**—Specify the password of the CA Business Intelligence administrative user.
 - **Authentication**—Specify Enterprise in the list.
3. Click the Document List, and do the following:
 - Expand Public Folders.
 - Expand CA Reports.
 - Expand CA Service Desk Manager.
4. Select the Asset folder in the left pane.
5. Select the Asset List report.

The report returns with zero or more Results Found.

Important! If the report did not run successfully, or you do not see the previously mentioned folder structure in InfoView, review the CA Business Intelligence configuration log. This file is located in your user temp directory in a subfolder named BIconfig. Review the biconfig.log file in this directory for any error messages. You can run the CA Business Intelligence configuration again to resolve these errors.

How to Configure Date Range Values and Join Parameters

After you install CA Business Intelligence, do the following:

- Configure the date range values so that the date range filters in CA Business Intelligence work correctly.
- Configure the join parameters so that universe outer joins are supported.

To configure date range values and join parameters, do the following:

1. On the computer on which CA Business Intelligence has been installed, navigate to the following location:

```
C:\Program Files\CA\SC\CommonReporting3\BusinessObjects Enterprise  
12.0\win32_x86\dataAccess\connectionServer\odbc\
```

2. Using a text editor, open the `odbc.prm` file, navigate to the `<Configuration>` section, and locate the following line:

```
<Parameter Name="USER_INPUT_DATE_FORMAT">{d 'yyyy-mm-dd'}</Parameter>
```

3. Locate the following line to configure date range values:

```
<Parameter Name="USER_INPUT_DATE_FORMAT">{d 'yyyy-mm-dd'}</Parameter>
```

4. Modify the line to include "HH:mm:ss am/pm" as shown in the following:

```
<Parameter Name="USER_INPUT_DATE_FORMAT">{d 'yyyy-mm-dd HH:mm:ss am/pm'}</Parameter>
```

5. Locate the following line to configure join parameters:

```
<Parameter Name="EXT_JOIN">NO </Parameter>
```

6. Modify the line to replace NO with YES as shown in the following:

```
<Parameter Name="EXT_JOIN">YES</Parameter>
```

7. Locate the following line:

```
<Parameter Name="OUTERJOINS_GENERATION">NO</Parameter>
```

8. Modify the line to replace NO with FULL_ODBC as shown in the following:

```
<Parameter Name="OUTERJOINS_GENERATION">FULL_ODBC</Parameter>
```

9. Add the following three lines after the `OUTERJOINS_GENERATION` parameter:

```
<Parameter Name="LEFT_OUTER"></Parameter> <Parameter Name="RIGHT_OUTER"></Parameter>  
<Parameter Name="OUTERJOINS_COMPLEX">Y</Parameter>
```

10. Save the `odbc.prm` file.

11. Restart the Business Objects Enterprise services.

Date range values and join parameters are configured. Date range filters work with CA Business Intelligence and universe outer joins are supported.

Add Your CA Service Desk Manager Users to CMC

The CMC is an administrative utility that lets you control users' access to InfoView and other BusinessObjects applications. With CMC, you can assign security and user access permissions to folders and documents.

Note: During the configuration phase, an optional check box indicates whether sample users are added to the CMC. If you selected this option, your CMC contains several sample users. You can use these samples as models when defining user permissions and authentication options for your reporting environment. For more information about sample users, see the *Administration Guide*.

To add your CA Service Desk Manager users to CMC

1. From the Start menu on the CA Business Intelligence server, select BusinessObjects XI Release Release 3.1, BusinessObjects Enterprise, BusinessObjects Enterprise Central Management Console.

The CMC Management Console appears.

2. Type the privileged user name and password.
3. Select Enterprise in the Authentication Type list.
4. Click Log On.

The CMC Home page appears.

5. Click Users and Groups in the Organize section of the CMC home page.
6. Click Manage, New, New User.

The New User dialog appears.

7. Select Enterprise from the Authentication Type list.
8. Under Account Name, specify the CA Service Desk Manager User ID.
9. On the Properties tab, specify your password information and settings as follows:

Password

Enter the password and confirm. This password should match the CA Service Desk Manager user's password. The maximum password length is 64 characters.

This password must be mixed-case, at least six characters long, and cannot contain the word administrator in any form. It should also contain at least two of the following character types:

- Uppercase
- Lowercase
- Numeric
- Punctuation

Password never expires

Select the check box.

User must change password at next logon

This check box is selected by default. If you do not want to force users to change the password the first time they log on, clear the check box.

10. To restrict data access for the reports with data partition or tenancy constraints, select the Enable Data Source Credentials for Business Objects Universes check box. In the fields that display, specify the user's CA Service Desk Manager account name and password, and then confirm the password.
11. Click the Actions, Members Of to specify the groups the user should belong to.
12. Click the Join Group to view the available groups. By default the user is a member of the Everyone group.
13. In the Available groups area, select one or more additional groups.
14. Click the > arrow to add the group(s).
15. Click OK.

The Members Of dialog appears and lists the groups in which the user is a member.

Note: For more information about adding users and groups and configuring data partitions security, see the *Administration Guide*.

Add the CA Service Desk Manager Privileged User to CMC

The CA Service Desk Manager Universe connection is configured by default to use the CA Service Desk Manager User and Password when accessing data. This user account is added to the CMC as a new CA Business Intelligence user. You need this user if you plan to set up data partition security for reporting and to test reports from the Reports tab. The Reports tab requires a user who is defined to both CA Service Desk Manager and CA Business Intelligence.

To add the CA Service Desk Manager Privileged User to CMC

1. Click Users and Groups management area of the CMC.
2. Under Account Name, select the CA Service Desk Manager Privileged User account.

3. On the Properties tab, specify your password information.
4. Select the Enable Database Credentials for Business Objects Universes check box. In the fields that display, specify the privileged user's account name and password, and then confirm the password.
5. Click the Members Of tab to specify the group the privileged user belongs to.
6. Click the Join Group button to view the available groups.
7. In the Available groups area, select Administrators and CA Universe Developer.
8. Click the > arrow to add these groups.
9. Click OK.

The Member Of tab appears and lists the groups in which the user is a member.

Default Settings in CMC

Most of the reporting configuration is performed silently during the CA Business Intelligence installation. Reporting configuration involves:

- Setting up security
- Deploying reports
- Deploying universes
- Deploying program objects
- Configuring Web Intelligence settings

The administrator can log on to the BusinessObjects CMC and modify the default settings at any time. Users are authorized access based on the CA Service Desk Manager group to which they belong.

Note: For more information about the BusinessObjects CMC, see the *CA Business Intelligence Implementation Guide*.

Integrate CA Business Intelligence with CA Service Desk Manager

After you install CA Business Intelligence, update the Web Reporting options so CA Service Desk Manager is properly integrated with CA Business Intelligence.

To integrate CA Business Intelligence with CA Service Desk Manager

1. On the Administration tab, select Options Manager, Web Report.
The Option List appears.

- Set the correct values for the following Web Report options:

bo_server_auth

Specify which type of authentication you want to use for reporting. You can specify the following types of authentication:

- **secEnterprise.** (Default) Specify Enterprise Authentication as your authentication type if you prefer to create distinct accounts and groups in BusinessObjects for use with CA Business Intelligence, or if a user hierarchy has not been set up in a Windows NT user database, an LDAP server or a Windows AD server.

Note: Before you use the secEnterprise option, add your CA Service Desk Manager report users to the BusinessObjects Central Management Console (CMC). In the CMC, enter the same user names and passwords configured in CA Service Desk Manager. For detailed instructions, see [Add CA Service Desk Manager Users to CMC](#) (see page 159).

- **secLDAP.** Specify LDAP Authentication as your authentication type if you have already set up an LDAP directory server and want to use your LDAP user accounts and groups in BusinessObjects for use with CA Business Intelligence.

When you map LDAP accounts to BusinessObjects, users can access CA Business Intelligence with their LDAP user name and password. This eliminates the need to recreate individual user and group accounts within BusinessObjects.

- **secWinAD.** Specify Windows AD Authentication as your authentication type if you are working in a Windows 2000 environment and want to use your existing Active Directory user accounts and groups in BusinessObjects for use with CA Business Intelligence.
- **secExternal.** Specify External Authentication as your authentication type if you integrate the BusinessObjects authentication solution with a third-party authentication solution (for example, using JCIFS with Tomcat). This authentication type requires setting up Trusted Authentication in BusinessObjects to allow users to log on without providing their passwords.

Note: For information about alternative security options, see the *CA Business Intelligence Implementation Guide*.

bo_server_cms

Specify the name of the Central Management Server (CMS) that is responsible for maintaining a database of information about your BusinessObjects that you use with CA Business Intelligence.

For the *bo_hostname*, use the hostname of the computer where CA Business Intelligence is installed. The default *bo_cms_port* is 6400. For detailed information about advanced server configuration options, see the *CA Business Intelligence Installation Guide*.

bo_server_location

Specify `bo_hostname` by using the hostname of the computer where CA Business Intelligence is installed. CA Service Desk Manager uses this URL to put together report URLs for requesting reports from the BusinessObjects server. The CMS location is specified by hostname and port.

Note: For detailed information about each option, see the *CA Business Intelligence Implementation Guide*.

2. Click Save, Refresh.

The Options Detail page is updated with your selection.

3. Click Close Window.

4. Stop and start the service named CA Service Desk Manager Server.

The Web Reporting settings are correctly configured to work with CA Service Desk Manager. You can now set up web-based reports.

Note: For information about setting up web-based reports, see Managing CA Business Intelligence in the *Administration Guide*.

How to Configure Trusted Authentication with CA Service Desk Manager and BusinessObjects

Trusted Authentication lets you use a simple form of Single Sign On when integrating CA Service Desk Manager and CA Business Intelligence. With Single Sign On, users can log on to the system without providing a password more than once in a session. To configure Trusted Authentication between CA Service Desk Manager and Business Objects, do the following:

1. Install and configure CA Service Desk Manager.
2. Install and configure CA Business Intelligence.
3. Login to the CMC as Administrator.
4. Access the Central Management Console (CMC) to set up Trusted Authentication.
5. Create CA Service Desk Manager contacts and BusinessObject users.
6. Install the CA Service Desk Manager web report options and set the `bo_server_auth` option to Enterprise.

Note: For more information about `bo_server_auth`, see the *Online Help*.

7. Cycle the BusinessObjects Apache Tomcat.
8. Cycle the CA Service Desk Manager server in Windows Services.

Configure Trusted Authentication in CA Business Intelligence

Configuring Trusted Authentication for CA Business Intelligence first requires editing the web.xml file.

To configure trusted authentication in CA Business Intelligence

1. Login to CMC as Administrator.
2. Go to the Authentication management area.
The Enterprise page appears.
3. At the bottom of the page, select the "Trusted Authentication is enabled" option, and specify a text string in the Shared secret field.

Note: The shared secret is used to create a trusted authentication password.

4. Enter a time-out value for your trusted authentication requests.

Note: The time-out value determines how long the CMS waits for the IEnterpriseSession.logon() call from the client application.

5. Click Update.
6. Edit the web.xml file from the CA Business Intelligence installation directory as follows:

Note: You can locate this file in the Tomcat55\webapps\OpenDocument\WEB-INF subdirectory of the CA Business Intelligence installation directory (the default installation directory is C:\Program Files\CA\SC\CommonReporting3).

<param-name>	Default Value	New Value
param-name	Default value	New value
Opendoc.cms.default	Hostname: port of your CMS	Provide the hostname and port number of your BusinessObjects CMS server
Opendoc.siteminder.enabled	true	false
Opendoc.sso.enabled	false	true
Opendoc.trusted.auth.user.retrieve	(blank)	REMOTE_USER (as specified following this table)
Opendoc.trusted.auth.user.param	(blank)	(blank)
Opendoc.trusted.auth.shared.secret	(blank)	(blank)

Enter the parameter *Opendoc.trusted.auth.user.retrieval* as follows:

```
<context-param>
  <param-name>Opendoc.trusted.auth.user.retrieval</param-name>
  <param-value>REMOTE_USER</param-value>
</context-param>
```

Note: Do not copy and paste this sample as it may corrupt the web.xml file.

Configure Trusted Authentication in CA Service Desk Manager

Configuring Trusted Authentication in CA Service Desk Manager requires editing the TrustedPrincipal.conf file.

To configure trusted authentication in CA Service Desk Manager

1. Open the TrustedPrincipal.conf file located in the following directory:
NX_ROOT\bopcfg\www\CATALINA_BASE\webapps\CAisd
2. Edit this line
SharedSecret=<Shared Secret as entered in CMC>

to reflect the following:
SharedSecret=xxxx
(where xxxx is the shared secret entered into the CMC in step 3)
3. Click Save.
4. Recycle the CA Business Intelligence Tomcat server.

Trusted authentication in CA Service Desk Manager is configured.

Configure BusinessObjects LDAP Authentication

When configuring LDAP authentication, the LDAP Server Administration Credentials “cn” value must be the fully qualified name (First Name, Last Name). Configuring LDAP authentication allows you to re-map LDAP attributes to use the user's logon name.

Important! The BusinessObjects user's Account Name must match the CA Service Desk Manager contact's User ID when configuring LDAP authentication.

To configure BusinessObjects LDAP Authentication

1. Access the Authentication management area of the Central Management Console (CMC).
2. Double-click LDAP.

3. Enter the name and port number of your LDAP hosts in the Add LDAP host (hostname:port) field. For example, "myserver:123".
4. Click Add and then OK.
5. Select Custom for the server type from the LDAP Server Type list.
6. Follow the prompts in the CMS configuration wizard to complete the configuration.

Note: For more information about configuring LDAP Authentication, see the *CMS Online Help* and the *BusinessObjects Enterprise Administrator's Guide*.

Connect CA Business Intelligence Server to a Different CA Service Desk Manager Server

After you configure CA Business Intelligence, do not run this configuration process again. The configuration is only used once after the initial installation of CA Business Intelligence. If you want to connect this CA Business Intelligence server to a different CA Service Desk Manager server, do the following:

1. [Create an ODBC DSN for the CA Service Desk Manager server](#) (see page 166).
2. [Connect the CA Service Desk Manager universe to this CA Service Desk Manager server](#) (see page 167).

Create an ODBC DSN for the CA Service Desk Manager Server

Use the ODBC Data Source Administrator to create an ODBC DSN.

To create an ODBC DSN for the CA Service Desk Manager server

1. Start the Windows ODBC Data Source Administrator (Data Sources (ODBC)).
2. On the ODBC Data Source Administrator form, select the System DSN tab, and select Add.
3. On the Create New Data Source form, select the DataDirect OpenAccess driver, and select Finish.
4. On the DataDirect OpenAccess ODBC 32 Setup form, assign an ODBC Name and select Advanced.

The naming convention to use is casd_hostname. For example, if the hostname of the CA Service Desk Manager server is MyServer, you would use casd_MyServer.

5. On the OpenAccess Database Configuration form, select Add.
6. On the OpenAccess Database Setup form, enter the following information:
 - **Name**—Specify casd_hostname.
 - **IP Address**—Specify the IP address of the CA Service Desk Manager server.
 - **Port**—Specify 1706.
 - **Type**—Select SQL.
7. Click OK.
8. On the Open Access Database Configuration form, select OK.
9. On the DataDirect OpenAccess ODBC 32 Setup form, select casd_hostname from the Database dropdown and select OK.

The ODBC DSN is created.

Connect the CA Service Desk Manager Universe to the Server

Use the Designer in BusinessObjects Enterprise to establish a connection.

To connect the CA Service Desk Manager universe to this CA Service Desk Manager server

1. From the Start menu, browse to BusinessObjects XI Release 2, BusinessObjects Enterprise, Designer.
2. Log in to the Designer with the following credentials.
 - **System**—Specify the hostname of the server where CA Business Intelligence was installed.
 - **User name**—Specify the name of the CA Business Intelligence administrative user (typically Administrator).
 - **Password**—Specify the password of the CA Business Intelligence administrative user.
 - **Authentication**—Select Enterprise.

The Designer window appears.

3. Click File, Import.

The Import Universe dialog appears.
4. Select the CA Universes folder from the drop-down list and then select the CA Service Desk Manager universe and click OK.

Note: If this is the first time you are using Designer, you may first need to select Browse to select the CA Universes folder.

5. Click OK to the "Universe successfully imported" message.
The universe window appears.
6. Select File, Parameters.
The Universe Parameters dialog appears.
7. On the Definition tab, click Edit.
The Login Parameters dialog appears.
8. Select Edit.
9. Select the ODBC DSN you created (casd_hostname) from the Data source name drop-down list and specify the CA Service Desk Manager Privileged User and Password for User name and Password.
10. Click Next, Test Connection, and step through the universe connection dialogs that appear.
11. Click OK to finish.
12. Select File, Export
The Export Universe dialog appears.
13. Select /CA Universes from the Domain drop-down list.
14. Select Everyone from the Groups list.
15. Click OK.
The universe is exported and the connection to the server is established.

Change the Maximum Size for a List of Values

When you install CA Business Intelligence, the maximum number of values that can be returned in a batch for a list of values in Crystal reports is automatically set to 5,000 records from the database. For performance reasons, you can change the size so that the list of values returned to the user is in several batches of this size or less.

Note: For information about improving the performance of the Web Intelligence Report Server, see your BusinessObjects documentation.

To change the maximum size for a list of values

1. Start the Registry Editor.
2. In the tree, expand HKEY_CURRENT_USER, Software, Business Objects, Suite 11.0, Crystal Reports, Database.
3. Create a registry key named QPMaxLOVSize.
4. Select New, DWORD Value.

5. In the Name field, enter DWORD.
6. In the Edit DWORD Value dialog, enter 1000 in the Value data field and click OK.
7. Select File, Exit to close the Registry Editor.
8. Using BusinessObjects Enterprise, log in to the Central Management Console.
9. Navigate to the property settings for the Web Intelligence Report Server.
10. Set the *List of Values Batch Size* and *Maximum Size of List of Values for Custom Sorting* options to 1000, or an appropriate setting based on your requirements.
11. Save the setting.

The maximum size is changed and is used when returning a list of values.

Change the Report Record Limits

When you install CA Business Intelligence, the number of records that the server retrieves from the database when a user runs a query or report in Crystal reports is automatically set to 20,000 records. You can change the setting so users running reports receive the record sets they expect.

Note: For complete details about administrative tasks you can complete for the Crystal Reports Page Server, see your BusinessObjects documentation.

To change the report record limits

1. Using BusinessObjects Enterprise, log in to the Central Management Console.
2. Navigate to the page displaying the servers.
3. Click Crystal Reports Page Server.
4. On the Properties tab, change the setting for the *Database Records To Read When Previewing or Refreshing a Report* field to either unlimited records, or specify a specific record limit.
5. Click Apply.
6. Restart the Crystal Report Page Server.

The report record limit changes and is used when running reports.

Change the Web Intelligence Session Time-Out

Users have Full Control access to the Web Intelligence application by default.

The Web Intelligence application has a session time-out of 20 minutes by default. Unsaved reports are lost when the session times out and the user must log on again to use the application.

Administrators can modify the connection session time-out value using the Central Management Console (CMC).

To change the Web Intelligence session time-out

1. Select Servers from the CMC Home page.
The Servers window opens.
2. Select Web_IntelligenceReportServer in the Server Name column.
3. Type the appropriate time-out value (number of minutes) in the Connection Time Out field.
4. Click Apply.
Your changes take effect after the server is restarted.
5. Click OK
The session time-out value is set.

Replicated Database for Offline Reporting

To manage potential performance issues that may affect the reporting components installed with CA Service Desk Manager, you can create a replicated database for offline reporting purposes.

Note: For more information about creating a replicated database for offline reporting, see the sample documentation and scripts delivered in the NX_ROOT\samples\reporting directory.

Run the Automated Policies

In Knowledge Management, the Automated Policies feature provides a set of default automated policies that allow you to manage knowledge for your organization more efficiently and effectively. An automated policy describes the condition by which documents are flagged for correction and promoted to publication or retirement throughout the various stages of the document lifecycle process. For example, you can specify the "fix broken links" default policy that matches documents found in the knowledge base with broken links. The task of fixing the problem can be assigned to an analyst.

The Automated Policies List page contains the details of the policies you can manage. To display this page, select the Administration tab, Knowledge, Automated Policies.

Each policy contains a stored query that executes when documents are matched during processing. After processing, a Lifecycle Policy report appears on the CA Service Desk Manager Scoreboard. To view a report, select Knowledge Documents, Automated Policies. From the scoreboard, the analyst can manage their own documents, and by default, the administrator can manage all documents for each role.

To implement the reports, you must run a batch process with the Automated Policies Scheduler. The scheduler runs on the server and displays the data required to view the reports. When you are finished, run the [Knowledge Report Card](#) (see page 172).

To run the Automated Policies Scheduler

1. Select the Administration tab, browse to Knowledge, Automated Policies, Scheduling.

The Automated Policies Scheduler appears.

2. Complete the following fields:

Last Updated

Select the Run Calculation check box.

Schedule

Specify a date and time from which CA Service Desk Manager performs the calculation and runs the policies.

3. Click Save.

Run the Knowledge Report Card

To run the Knowledge Report Card

1. Click the Administration tab.
2. Browse to Knowledge, Knowledge Report Card.

The Knowledge Report Card appears.

3. Complete the following fields:

Last Updated

Select the Run Calculation check box.

Schedule

Specify a date and time from which CA Service Desk Manager performs the calculation and runs the Report Card.

4. Click Save.

Note: For information about using Automated Policies and the Knowledge Report Card, see the *Knowledge Management Administration Guide*.

Knowledge Management Sample Data Import

Sample Knowledge data from Knowledge Broker and Knowledge Accelerators is provided for your use. If you decide you want to use the sample Knowledge data you must import the data into the Knowledge Management database. Follow the instructions outlined below to do so.

Windows installation

1. Go to `$NX_ROOT\samples\data` and unzip `SampleData.zip` into the same directory.
2. From the command window, go to `$NX_ROOT/bin` and run `ImportSampleData.bat`.

UNIX installation

1. Run the command `tar -xvf SampleData.tar` from `$NX_ROOT/samples/data`.
2. From the command window, go to `$NX_ROOT/bin` and run `ImportSampleData.sh`.

How to Configure FAST ESP

After you install FAST ESP, do the following steps to configure FAST ESP so it works with CA Service Desk Manager correctly:

1. (Required) Specify that you want to use the FAST ESP search engine.
2. Set up search and indexing for external repositories (websites and local directories).

3. Change the lemmatization strategy for a language.
4. Configure synonyms.
5. Increase the FAST ESP search capabilities.
6. Back up FAST ESP data.
7. Integrate FAST ESP on a secondary server.

Important! If you stop and restart FAST ESP services for any reason, you *must* also stop and restart the indexing daemon (bpeid_nxd). FAST ESP *must* be running for the daemon to start, or you get errors in *stdlog* and when you create or search for knowledge documents.

Note: For detailed configuration information, see the *FAST ESP Configuration Guide*.

More information:

[Use the FAST Search Engine](#) (see page 173)
[Search External Repositories](#) (see page 175)
[Change the Lemmatization Strategy](#) (see page 177)
[Configure Synonyms](#) (see page 180)
[Find Similar Searches](#) (see page 181)
[pdm_k_reindex—Knowledge Re-Index Utility](#) (see page 182)
[Use pdm_k_reindex with FAST ESP](#) (see page 183)
[Increase the Search Capabilities](#) (see page 185)
[How to Back Up FAST ESP Data](#) (see page 185)
[Integrate FAST ESP on the Secondary Server](#) (see page 186)

Use the FAST Search Engine

By default, CA Service Desk Manager is configured to use the Knowledge Management search engine. If you have installed the FAST ESP search engine, you can select to use that search engine as your default for Knowledge Management searches. Complete these steps on the CA Service Desk Manager primary or secondary server.

To use the FAST search engine

1. Click the Administration tab.
The Administration console appears.
2. In the tree on the left, click Options Manager, Search Engine.
The Option List appears.

3. Specify the following settings:

ebr_version

Specify FAST [Search Engine].

ebr_search_engine_baseport

Specify the base port of the dedicated server on which the FAST ESP search engine is installed. Consider the following when specifying the base port:

- Make sure that the port number you enter here matches the base port specified during the FAST ESP installation (default port is 13000).
- The base port is used by FAST ESP to calculate static ports.
- On each host, there should be a span of 4000 ports between the FAST ESP base port and the FAST ESP installation because FAST ESP offsets its static ports from the base port (that is, the administration interface port number if 16000).
- Make sure this port range does not conflict with other products or services the server uses.
- You cannot use ports below 1024.
- We recommend that you *do not* use a base port in the range of 23000 through 27000, because the license server always uses port number 27000.

Important! All hosts must have unrestricted network access in the full port range (for example, ports 13000 through 16999).

ebr_search_engine_host

Specify the fully-qualified domain name or IP address (recommended) of the dedicated server on which the FAST ESP search engine is installed.

4. Click Save, Refresh.

The Options Detail page is updated with your selection.

5. Click Close Window.
6. Stop and start the service named CA Service Desk Manager Server.
7. Run the `pdm_k_reindex index factory:all` command to index knowledge documents and tickets in FAST ESP.

Note: For information about running the `pdm_k_reindex` command, see the *Administration Guide*.

Search External Repositories

From the Knowledge Search page, or from the Knowledge tab in a service desk ticket, analysts and knowledge managers can retrieve unstructured knowledge from external repositories (websites and file systems) by selecting the External Repositories data source on the Knowledge Type list. Users can retrieve external repository content collected from local directories, the Internet or an Intranet.

Important After the FAST ESP and CA Service Desk Manager installation, there are no additional steps required to configure search and indexing for knowledge documents, forums, knowledge files, and service desk tickets. By default, this content is retrieved, processed, made searchable, and then grouped into the default *kd* collection in the FAST ESP administrative interface.

In FAST ESP, you can use either or both of the following methods to configure search and indexing for websites and local directories:

- [Add External Repositories for Website Searches](#) (see page 175)
- [Add External Repositories for Local Directories](#) (see page 177)

Important! The default license agreement for FAST ESP search specifies the following limitations: two searches a second; two million indexed objects. Because of these limitations, we recommend that you do not add large websites such as ca.com. In addition, you should not change the Request Rate and Refresh Interval parameters without a specific reason.

Add External Repositories for Website Searches

You can use the FAST ESP administrative interface to add a set of URIs from which to start web crawling. For example, <http://www.my-site-to-index.com>. The Enterprise Crawler connects the collection of URIs that you define to a web crawler for content retrieval.

Note: Some websites block the crawl ability for applications. For more information about troubleshooting the Enterprise Crawler, see the FAST ESP Enterprise Crawler Guide, located in `\CA_tps.nt\FastESP\Doc\en-US`.

Complete these steps on the computer on which the FAST ESP search engine is installed.

To add external repositories for website searches

1. Launch the FAST ESP administrator interface from your web browser.
 - Assuming that you kept the default installation setting (13000) for the FAST base port range, you can enter the administrator interface at address `http://machinename:16000`, where port number is base port + 3000.

- If you change the default base port range, you can log on to the FAST ESP administrator interface at address *http://machinename:default start port range + 3000/*.

Note: From the Administration tab, in Options Manager, the base port range is managed through the `ebr_search_engine_baseport` option.

2. To log in to FAST Home for the first time as the administrator, use Admin as the user name, and leave the password field blank. Then navigate to FAST Home, User Administration, Create Users & Groups to enter a new password for the admin account.

3. From the home page, click the ESP Admin GUI link located on the right side.

Note: The ESP Admin GUI link launches the FAST ESP administrative interface. You can ignore the Clarity link that launches the FAST Monitoring tool. For more information about these links, see the *FAST Operations Guide*.

The Collection Overview page appears.

4. From the *site* collection, select the Edit icon that appears in the Docs column.

The Collection Details page appears.

5. From the Control Panel, click the Edit Data Sources icon.

The Edit Collection page appears.

6. From the Available Data Sources list, select Enterprise Crawler.

The Enterprise Crawler connects the collection to a web crawler for content retrieval.

7. Click the *add selected* button.

The Edit Collection, New Data Source setup page appears.

Note: Request rate and Refresh interval defines how often the crawler visits the web servers of the indicated web domain.

8. Enter a start URI in the Start URIs box as shown. The Start URI field allows you to add a set of URIs from which to start crawling (for example, `http://www.my-site-to-index.com`). At least one URI (or a file containing a list of URIs) must be defined in order for the system to start crawling. Make sure the URI ends with a slash (/); you cannot add the same URI twice.
 - a. Click the add arrow icon and the URI is added to the set of URIs in the text box on the right.

To remove a start URI from the list, highlight the URI and click the remove arrow icon.

At the same time, an *exact hostname include filter* is added by default to the list of allowed hosts in the Hostname include filters field. All servers in the host `www.mysitename.com` are crawled.

- b. Click Submit.

Wait a few minutes, and review the indexed documents count to verify that the indexing is in progress.

9. Click OK.

Search is now available from the External Repositories data source on the Knowledge Search page, or from the Knowledge tab in a service desk ticket.

Add External Repositories for Local Directories

You can use the ESP File Traverser to retrieve files from directories on file servers and submit them to a specified collection for further processing.

Note: For complete details about *filetraverser* commands, see the ESP File Traverser Guide, located in `\CA_tps.nt\FastESP\Doc\en-US`.

Complete these steps on the computer on which the FAST ESP search engine is installed.

To add external repositories for local directories

1. Open a command prompt window, and use the *filetraverser* command to index a directory that contains HTML files to index.

The command you use to run the File Traverser might look similar to the following:

```
filetraverser -r C:\HTML -s html -c site -p http://test03.ca.com/test
```

Note: In this example, HTML files are indexed from the `C:\test` directory into the collection site when the `http://test.my-files.com/test` prefix is assigned to all pages.

2. Wait a few minutes, and review the indexed documents count on the Collection Overview page in the ESP Administrator interface to verify that indexing is complete.

Search is now available from the External Repositories data source on the Knowledge Search page, or from the Knowledge tab in a service desk ticket.

Change the Lemmatization Strategy

In CA Service Desk Manager, you specify the query language by setting the `NX_EBR_QUERY_LANGUAGE` variable in the `$NX_ROOT\NX.env` file. By default, no variable is required. You must define this variable for the Chinese, Japanese, and Korean languages.

Consider the following information when changing your lemmatization strategy and specifying the query language:

Important! For complete details about changing your lemmatization strategy, see the FAST ESP documentation available on the installation media, including the *FAST ESP Advanced Linguistics Guide*, *FAST ESP Troubleshooting Guide*, and *FAST ESP Query Language Parameters Guide*. In addition, setting up a new language or changing the lemmatization strategy requires that you re-process all indexed documents.

- CA Service Desk Manager supports query-side linguistic support within FAST ESP for the following languages:

Group	Supported Languages
1	<ul style="list-style-type: none">■ English (en)■ French (fr)■ German (de)■ Japanese (ja)
2	<ul style="list-style-type: none">■ Italian (it)■ Simplified Chinese (zh-simplified)■ Spanish (es)■ Brazilian Portuguese (pt)
3	<ul style="list-style-type: none">■ Korean (ko)■ Traditional Chinese (zh-traditional)■ Thai (th)

- When you install FAST ESP as part of the CA Service Desk Manager installation, you cannot specify multiple-language support. To use multiple-language support, you must deploy the *LemmatizationConfig.xml* file (found in the \$FASTSEARCH/etc directory) that defines the lemmatization strategy for all supported languages.

This LemmatizationConfig.xml file will include a list of all languages having advanced linguistic support in InStream 5.1.3, including the following:

■ Dutch	■ German	■ Korean
■ Arabic	■ Estonian	■ Hungarian
■ Polish	■ Slovak	■ Ukrainian
■ English	■ Italian	■ Norwegian
■ Czech	■ Finnish	■ Hindi
■ Romanian	■ Swedish	■ Portuguese
■ French	■ Japanese	■ Latvian
■ Danish	■ Hebrew	■ Spanish
■ Russian	■ Turkish	■ Lithuanian

Note: All other languages will not have advanced linguistic support.

- Certain languages require specialized linguistic processing so that the content can be indexed and searched. The languages include Chinese, Japanese, Korean and Thai. In FAST ESP, both the documents to be indexed and queries undergo linguistic processing, where much of the processing is language-specific.
- FAST ESP can identify the language of each document automatically when it is sent for indexing. However, because many languages share the same or similar scripts, search queries are typically too short for accurate language identification to be guaranteed. Therefore, it is important that the language of a query is set by default, or specified when each query is submitted. If you do not, the query may be incorrectly processed.

Example: Identify short documents by their language and make them searchable

Note: This example explains how to identify short Japanese documents and make them searchable in FAST ESP. You can also identify short Chinese and Korean documents.

The language can be forced in the CA Service Desk Manager (webcluster) pipeline as long as the proper criteria is defined.

If all Japanese Documents are pushed within the FAST ESP collection, then the language can be forced using an AttributeAssigner type of stage, or configured using the FallbackLanguage parameter of the LanguageAndEncodingDetector stage.

If the FAST ESP collection has a mix of Japanese and other language documents, then the criteria have to be defined on how to determine the language (such as id-based criteria). Therefore, a custom stage must be created to implement those criteria and assign the correct language.

You can also assign a language from where documents are coming from such as the database, Crawler or other extraction mechanism. If the language is already known at the extraction phase, it can be passed through the pipeline and be maintained as it is by disabling the language detection.

Configure Synonyms

When you use *synonyms* with FAST ESP, you can search and find relevant content, even when the content does not actually contain any of the terms or phrases you are looking for. For example, if you define the synonym *music* for the terms mp3 and mp4, users will find documents containing the word mp3 or mp4 when they search for music. You can also use synonyms to rewrite a term to show results you think are more appropriate in your enterprise. For example, you can rewrite the term *laptop* to return results for a specific laptop model only.

Note: For complete details and steps to manage synonyms, see the *FAST ESP Search Business Center Guide*.

To configure synonyms

1. Log in to the FAST ESP Linguistics Studio using the instructions provided in your FAST ESP documentation.
2. Use LinguisticsStudio to complete the following steps:
 - a. Create a linguistics project.
 - b. Create a dictionary and name it *casynonyms*.
 - c. Add new synonyms to the dictionary. For example, music for the terms mp3 and mp4.

The following new dictionary file is created:

```
$FASTROOT\resources\dictionaries\synonyms\qt\casynonyms.aut
```

- d. Add the following entry to the new dictionary to the `$FASTROOT\etc\config_data\QRServer\webcluster\etc\qrserver\qtf-config.xml` file:

```
<instance name="synonym" type="external" resource="qt_synonym">
  <parameter-list name="qt.synonym">
    <parameter name="enable" value="1"/>
    <parameter name="synonymdict1" value="resources/dictionaries/synonyms/qt/short_spellvars.aut"/>
    <parameter name="synonymdict2" value="resources/dictionaries/synonyms/qt/short_wordnet.aut"/>
    <parameter name="synonymdict3" value="resources/dictionaries/synonyms/qt/casynonyms.aut"/>
  </parameter-list>
</instance>
```

3. Stop the *qrserver* service.
4. Deploy the new dictionary file to FAST ESP.
5. Start the *qrserver* service.
6. Run the *setupenv.cmd* command in the `$FASTROOT\bin` directory.
7. Run the *view-admin.cmd -a -m refresh* command in the `$FASTROOT\bin` directory.
8. Update `NX.env` with `@NX_EBR_QUERY_WITH_SYNONYMS=Yes`.
9. Restart CA Service Desk Manager.

More information:

[Install LinguisticsStudio](#) (see page 138)

More information:

[pdm_k_reindex—Knowledge Re-Index Utility](#) (see page 182)

[Use pdm_k_reindex with FAST ESP](#) (see page 183)

Find Similar Searches

You can search for similar tickets from the Knowledge tab of tickets, including Issues, Requests, Incidents, Problems, and Change Orders. Find Similar is only enabled when you install FAST ESP. This functionality lets you search for similar tickets using the ticket summary and description as a search query. Searching for similar tickets helps you avoid creating redundant tickets in your support environment.

Important! If you want to use Find Similar, you must run [pdm_k_reindex](#) (see page 183) to sync the documents. You must also reindex each of the CA Service Desk Manager objects, such as Call Requests, Change Orders, and so on.

pdm_k_reindex—Knowledge Re-Index Utility

The Knowledge Re-Index utility, `pdm_k_reindex.exe`, is located under the Knowledge Management installation directory. You can also use this utility in a [FAST ESP integration](#) (see page 183).

Note: Re-indexing the documents in the knowledge base can be a time-consuming operation, depending upon the size of your database. We recommend that you run the Knowledge Re-Index utility after all the changes have been added.

To run Knowledge Re-Index, enter the following command at the command prompt:

```
pdm_k_reindex
```

Following are the options available with this command.

Interface:

-D

Defines the debug mode, such as printing to the command window.

-v

Defines the verbose mode, such as printing to the stdlog file.

-i

Does not create table indexes in the re-index table after re-indexing.

Note: Parameters with dash as a prefix, such as “-D”, should precede other parameters that do not have this prefix.

file:reindex.txt

Documents are re-indexed to the appropriate file.

+i

Creates indexes of the re-indexed table only, which is the search table after re-indexing. Old indexes are dropped before re-indexing.

+t

Switches names of search and re-index tables only.

Note: A “+” prefix denotes only this parameter applies.

sdtout

Defines the frequency of statistic appearing in the command window. By default the Knowledge Re-index utility provides statistics into the command window for every 1000 documents processed. However, sometimes statistics are required to be provided more often. Use the following parameter:

```
pdm_k_reindex -i sdtout:10
```

In this case, statistics display in the command window for every ten documents.

Important! On UNIX, the LIBPATH must be set before running several CA Service Desk Manager utilities. Use *pdm_task* to set the LIBPATH before running a utility. For example, input "pdm_task pdm_clean_attachments ...".

Use pdm_k_reindex with FAST ESP

The pdm_k_reindex utility is used to selectively re-index, de-index and synchronize (re-index and de-index) document factories such as KD, cr, iss, and so on. The utility is invoked as follows:

```
pdm_k_reindex [operation] [factory] [mode]
```

operation

The following operations are valid:

-h

Displays help on the utility.

index

(Default) Re-indexes the documents.

deindex

De-indexes the documents.

sync

Allows you to re-index and/or de-index documents to synchronize search engine after purge or restore being executed. The following modes are valid for the sync operation:

- **purge**

Synchronizes the search engine after purge being executed.

- **restore**

Synchronizes the search engine after restore being executed.

status

Displays a count of remaining documents to process.

factory

The following factories are valid:

all

Sets all factories.

KD

(Default) Sets the KD factory.

cr

Sets the cr factory.

iss

Sets the iss factory.

mode

The following modes are valid:

purge

Synchronizes search engine after purge being executed.

restore

Synchronizes search engine after restore being executed.

Note: If no operation or factory arguments are selected, the KD factory is indexed.

Example index parameters

```
pdm_k_reindex [index] [factory:KD|cr|iss|all]
```

Example de-index parameters

```
pdm_k_reindex deindex factory:KD|cr|iss|all [clearrange]
```

clearrange

(Optional) De-indexes documents that are out of the predefined range of the factory.

Note: A factory argument is required with the deindex parameter.

Example sync parameters

```
pdm_k_reindex sync[:purge|restore] factory:KD|cr|iss|all
```

Note: If you omit purge or restore modes, both modes are invoked. A factory argument is also required with the sync parameter.

Miscellaneous parameters

`pdm_k_reindex -pm`

Fixes the document links and embedded images inside the resolution field.

`pdm_k_reindex -ml`

Fixes the document links inside the resolution field and maps them to the database.

Note: For more information about the `pdm_k_reindex` utility, see the *Implementation Guide*.

Increase the Search Capabilities

When you install the FAST ESP search engine in CA Service Desk Manager to improve your knowledge searches, the default license agreement for the search engine specifies the following search limitations:

- Two searches a second
- Two million indexed objects

If you need to increase the default search capabilities specified by your license agreement, contact FAST and purchase additional licenses at CAKnowledgeT@fastsearch.com.

How to Back Up FAST ESP Data

After you install, configure, and have been using the FAST ESP search engine, there are several maintenance steps that you should perform on a regular basis to ensure that you properly back up your FAST ESP data. The frequency on which you back up your data depends on your business needs. If your configuration is constantly changing, or if new content is being added on a regular basis, you will probably want to run your backups on a more frequent basis.

To back up your FAST ESP data, complete the following steps:

1. Back up the following directories:
 - `$FASTSEARCH/data/data_fixml`
 - `$FASTSEARCH/data/data_index`
 - `$FASTSEARCH/etc`

2. Back up the index profiles in the following location:

- \$FASTSEARCH/index-profiles

Note: For complete information about backing up and restoring your FAST ESP data, see the *FAST ESP Operations Guide* that discusses these operations.

Integrate FAST ESP on the Secondary Server

You can use the product to integrate FAST ESP on the secondary server.

To integrate FAST ESP on the secondary server

1. Click the Administration tab.

The Administration console appears.

2. In the tree on the left, click Options Manager, Search Engine.

The Options List appears.

3. Specify the following setting on the secondary server:

ebr_version

Specify FAST [Search Engine].

ebr_search_engine_baseport

Specify the base port of the dedicated server on which the FAST ESP search engine is installed. Consider the following when specifying the base port:

- Make sure that the port number you enter here matches the base port specified during the FAST ESP installation (default port is 13000).
- The base port is used by FAST ESP to calculate static ports.
- On each host, there should be a span of 4000 ports between the FAST ESP base port and the FAST ESP installation because FAST ESP offsets its static ports from the base port (that is, the administration interface port number if 16000).
- Make sure this port range does not conflict with other products or services the server uses.
- You cannot use ports below 1024.
- We recommend that you *do not* use a base port in the range of 23000 through 27000, because the license server always uses port number 27000.

Important! All hosts must have unrestricted network access in the full port range (for example, ports 13000 through 16999).

ebr_search_engine_host

Specify the fully-qualified domain name or IP address (recommended) of the dedicated server on which the FAST ESP search engine is installed.

4. Click Save, Refresh.

The Options Detail page is updated with your selection.

5. Click Close Window.
6. Stop and start the service named CA Service Desk Manager Server.

How to Configure CA Workflow

To configure CA Workflow for use with your CA Service Desk Manager installation, perform the tasks described in this section.

More information:

[Start and Stop CA Workflow](#) (see page 187)

[Set Up Automatic Login \(External Authentication\)](#) (see page 191)

[Configure Worklist and Workflow Manager for Automatic Login](#) (see page 192)

[Configure Workflow Design Environment for Automatic Login](#) (see page 193)

[CA Workflow Options](#) (see page 194)

[How to Configure the Problem Management Sample Workflow](#) (see page 195)

[How to Configure the Order PC Sample Workflow](#) (see page 197)

[Change Management Process Definition Example](#) (see page 198)

Start and Stop CA Workflow

You must first install and configure CA Service Desk Manager before installing CA Workflow. The default CA Service Desk Manager Tomcat is automatically configured, started and stopped by the CA Service Desk Manager daemon manager. The `pdm_tomcat_nxd` command line utility can be used to start and stop the Tomcat instance as well.

To start and stop CA Workflow

1. Install and configure the CA Service Desk Manager primary server.
2. Install and configure the CA Service Desk Manager secondary server, if you intend to install CA Workflow on a secondary server.

3. Run `pdm_edit` (on the primary server) to configure specific processes to be started on the secondary server.

4. Recycle CA Service Desk Manager services.

This will version control specific environment variables to a secondary server that are required to install and run CA Workflow.

5. Install and configure CA Workflow.

6. Run `pdm_tomcat_nxd -d start -t CAWF`

This manually starts CA Workflow.

Note: Run `pdm_tomcat_nxd -d stop -t CAWF` to stop CA Workflow.

7. Install CAWF options on the primary server.

This allows CA Workflow to start and stop automatically, and will allow communications between CA Service Desk Manager and CA Workflow.

More information:

[pdm_tomcat_nxd—Start or Stop a Tomcat Instance](#) (see page 188)

pdm_tomcat_nxd—Start or Stop a Tomcat Instance

The `pdm_tomcat_nxd` utility runs as a daemon process and as a command line utility. The utility can do the following:

- Handles requests from the command line utility to START and STOP a specific Tomcat instance
- Returns the Tomcat STATUS.
- EXIT the daemon and stops the Tomcat instance.

The daemon also handles requests from a Tomcat "listener" servlet to update the STATUS of the Tomcat instance when it is stopped or started.

The `pdm_tomcat_nxd` command line utility directs START, STOP, STATUS and EXIT requests to a Tomcat daemon or can START or STOP a specific Tomcat instance directly without using a Tomcat daemon.

`Pdm_tomcat_nxd` provides multiple daemons that are started on a single CA Service Desk Manager server with each daemon maintaining a specific Tomcat instance. Each daemon is initialized with a specific Tomcat and Server name which are used to compose the daemon's slump address, its "listener's" slump address and the Catalina Base directory where the Tomcat instance is defined. This change is supported by a new Tomcat name parameter.

Use the following command to start a `pdm_tomcat_nxd` daemon. When the daemon starts, it automatically starts a Tomcat instance with a specified Tomcat name.

Note: In the following examples, the default Tomcat name is empty (implies "ServiceDesk").

```
pdm_tomcat_nxd -s [ -t tomcat ]
```

Example: Send a request to a specific Tomcat daemon on a specific Server

If the request is STATUS, a message is sent first to the Tomcat "listener". If there is no response, the request is then sent to the Tomcat daemon. The default Server name is `NX_LOCAL_HOST`. The default Tomcat name is empty; (which implies "ServiceDesk").

```
pdm_tomcat_nxd -c <request> [-t <tomcat_name>] [-S <server_name>]
```

request

STOP | SHUTDOWN | START | STATUS | EXIT

Example: Start or stop a specific Tomcat on this Server (without using the Tomcat daemon)

```
pdm_tomcat_nxd -d <request> [ -t <tomcat_name> ]
```

request

STOP | START

Example: Display the Tomcat command that will be used to start and stop a Tomcat instance

```
pdm_tomcat_nxd -T [-t <tomcat_name>]
```

When the daemon process begins, it logs in to the Slump server using a process name comprised of the server_name and the tomcat name (if it is non-empty). The command line utility uses this address to send messages to the daemon. To send STATUS requests to the Tomcat "listener" the command line utility uses the listener process name also comprised of the server_name and the tomcat_name (if it is non-empty). The daemon also opens a log file for the Tomcat instance using a log filename comprised of the tomcat name (if it is non-empty). The following example displays the daemon_name, listener_name, and log filename variables:

```
daemon_name
```

```
"pdm_tomcat -" +  
server_name  
[ + "-" + tomcat_name ] if tomcat_name is non empty
```

```
listener_name = server_name + "-" +  
"container-pdmContextListener"  
[ + "-" + tomcat_name ] if tomcat_name is non empty
```

```
tomcat_log = pdm_tomcat +  
"-" + tomcat_name + ] if tomcat_name is non empty  
".log"
```

When the command line utility or the daemon process starts or stops a Tomcat instance, a command (catalina_cmd) is generated using various NX environment variables. All of the following environment variables are required except for NX_JAVA_OPTIONS and NX_TOMCAT_HOTSPOT ,which is only used on an HP system.

The environment variables (for the default ServiceDesk Tomcat instance) do not incorporate a Tomcat name in their name. Whereas the environment variables for all other Tomcat instances may optionally use a Tomcat name as part of their variable name except for NX_XXX_CATALINA_BASE where the tomcat name (XXX) must be specified. If a Tomcat specific environment variable does not exist then the default environment variable name is used.

```
CATALINA_BASE | NX_XXX_CATALINA_BASE  
NX_TOMCAT_INSTALL_DIR + [ _XXX ]  
NX_TOMCAT_INSTALL_DIR + [ _XXX ]  
NX_JRE_INSTALL_DIR + [ _XXX ]  
NX_JDBC_DRIVER + [ _XXX ]  
NX_JAVA_OPTIONS + [ _XXX ]  
NX_JDBC_DRIVER_CLASSPATH + [ _XXX ]  
NX_TOMCAT_HOTSPOT + [ _XXX ]
```

Set Up Automatic Login (External Authentication)

To automate the login process for CA Workflow Worklist tasks when working with change orders, requests, incidents, problems, and issues, you can set up automatic login. When you use automatic login, you will not have to enter a userID and password when you click a Worklist link. Automatic login uses your Windows operating system domain credentials to log in to the Worklist.

Note: For automatic login to work correctly, the user does *not* have to be included in the Workflow Administrators, Workflow Process Initiators, or Workflow SuperUser groups in CA EEM. These groups provide different access levels to users logged in to the Workflow Design Environment.

To set up automatic login for Worklist tasks (Windows only)

1. [Install and configure CA Workflow](#) (see page 126).
2. [Configure Worklist and Workflow Manager for automatic login](#) (see page 192).
3. Configure CA EEM to reference an external directory (for example, Microsoft Active Directory) to authenticate users. For information, see your CA EEM documentation.

Note: A user must have a CA EEM user record, as well as a matching valid CA Service Desk Manager contact record for the same user account, to access the Worklist and for Automatic Login to work correctly. For an item to appear in the Worklist, and to take advantage of the Workflow Administrators, Workflow Process Initiators, and Workflow SuperUser groups in CA EEM, a valid CA EEM user record is needed.

4. Log in to CA Service Desk Manager.
5. Click the Administration tab.
6. In the tree on the left, click Options Manager, CA Workflow.
The Options List appears.
7. Verify that the user name in cawf_username and password in cawf_password is a valid user on the domain set up in CA EEM. If not, use CA EEM to add the user to the domain.

Important! In the Options Manager, the values in cawf_username and cawf_password are set based on the values specified for the privileged user name and privileged user password when configuring CA Service Desk Manager after the installation. If the privileged user specified during the CA Service Desk Manager configuration is not available when CA EEM is configured to reference the external directory, automatic log will *not* work. In this situation, use CA EEM to add the user to the domain, or change the information in cawf_username and cawf_password to include a valid domain user set up in CA EEM.

8. Log in to CA Service Desk Manager and create a change order, request, incident, problem, or issue with the appropriate category.

Note: Make sure the category is configured to use CA Workflow and has a CA Workflow definition attached to it before creating the change order, request, incident, problem, or issue.

You will now be able to click a link in the Worklist and log in without being asked for a userID and password.

More information:

[Configure Workflow Design Environment for Automatic Login](#) (see page 193)

Configure Worklist and Workflow Manager for Automatic Login

You can enable automatic login for Worklist and Workflow Manager by changing the value for variable *automaticLoginEnabled* to true in the Worklist and Workflow Manager web.xml files (wl.xml and pm.xml). The entries are the same in both web.xml files.

```
<env-entry>
  <env-entry-name>automaticLoginEnabled</env-entry-name>
  <env-entry-value>true</env-entry-value>
  <env-entry-type>java.lang.Boolean</env-entry-type>
</env-entry>
```

The initial value for Worklist and Workflow Manager is defined during the installation based on the setting in the response file.

Enable or disable Automatic Login (also known as Single Sign On or SSO)

-P serverAction.automaticLoginEnabled="true"

To configure Worklist and Workflow Manager for automatic login

1. Navigate to the pm.xml and wl.xml files in the following directory:
\$NX_ROOT\bopcfg\www\CATALINA_BASE_WF\conf\Catalina\localhost
2. Edit pm.xml and wl.xml and change the value of the following line:
 - (Before editing) <Environment name="automaticLoginEnabled" override="true" type="java.lang.Boolean" value="false"/>
 - (After editing) <Environment name="automaticLoginEnabled" override="true" type="java.lang.Boolean" value="true"/>

3. Stop the Tomcat server using the following command:

```
pdm_tomcat_nxd -d STOP -t CAWF
```
4. Start the Tomcat server to re-create these two folders using the following command:

```
pdm_tomcat_nxd -d START -t CAWF
```
5. Access a ticket with a Workflow task and select one of the tasks.
6. The Worklist appears and the automatic login is successful.

Note: If you reconfigure CA Workflow and reference a different CA EEM server, you must shut down the CAWF Tomcat, delete both the *pm* and *wl* folders found under *\$NX_ROOT\bopcfg\www\CATALINA_BASE_WF\webapps*, and then restart the CAWF Tomcat to have these folders re-created with the new CA EEM server information.

Configure Workflow Design Environment for Automatic Login

Automatic login for the Workflow Design Environment is enabled using the command-line parameter *-auto*.

CA Workflow Options

The following options control the CA Workflow functionality.

Important! After installing CA Workflow and CA EEM, you must manually set options in the Options Manager, such as the port number. For example, a separate Tomcat web application is used for CA Workflow, so it can no longer use the default port 8080. To avoid port conflict, the default value 8090 is used. For more information, see the *Implementation Guide*.

cawf_hostname

Specifies the host name of the server where CA Workflow is installed. Set this variable only if CA Workflow is installed from the CA Service Desk Manager media; otherwise, do not install this option.

cawf_password

Specifies the password for the CA Workflow IDE user. During CA Workflow installation, the CA Service Desk Manager Component Installer Wizard uses the Workflow User Password value to set the cawf_password.

cawf_pm_location

Specifies the location of the CA Workflow Process Manager application. The URL must take the following format:

`http://<wf_hostname>:8090/pm`

cawf_pm_url

Specifies the CA Workflow Process Manager Web Service URL. The URL must take the following format:

`http://<wf_hostname>:8090/pm/services/pmService2`

cawf_username

Specifies the CA Workflow username for the CA Workflow IDE user. During CA Workflow installation, the CA Service Desk Manager Component Installer Wizard uses the Workflow User Name value to set the cawf_username.

cawf_wl_location

Specifies the location of the CA Workflow Worklist Manager application. The URL must take the following format:

`http://<wf_hostname>:8090/wl`

cawf_wl_url

Specifies the CA Workflow Worklist Web Service URL. The URL must take the following format:

`http://<wf_hostname>:8090/wl/services/wlService`

Note: Installation of these options requires restarting the CA Service Desk Manager server.

How to Configure the Problem Management Sample Workflow

The Problem Management sample workflow guides the process of deciding whether to create a change order for a particular problem ticket. After a problem assignee has researched a problem, they complete a problem survey to develop a recommendation. The survey gives the approver an idea of the impact of the problem and indicates whether it is necessary to resolve it through change management. The underlying concept of this workflow (problem pain/value analysis) is derived from ITIL v3.

Note: For details about performing the CA Service Desk Manager tasks in the following process, see the *Online Help*.

To configure the Problem Management sample workflow:

1. Create a CA Service Desk Manager contact for an analyst (for example, Jane Analyst) using the following values:

Contact Type

Select Analyst.

Data Partition

Select Service Desk Analyst.

Access Type

Select Service Desk Staff.

Email Address

Enter an e-mail address.

Note: For testing purposes, you may want to enter your own e-mail address.

Notifications Method

Select Email for Low, Normal, High, and Emergency notifications.

2. Create a CA Service Desk Manager contact for a supervisor (for example, Joe Supervisor) using the following values:

Contact Type

Select Manager.

Data Partition

Select Service Desk Administrator.

Access Type

Select Service Desk Management.

Email Address

Enter an e-mail address.

Note: For testing purposes, you may want to enter your own e-mail address.

Notifications Method

Select Notification for Low, Normal, High, and Emergency notifications.

3. Create an associated EEM user record for the Jane Analyst and Joe Supervisor CA Service Desk Manager contacts.

Note: The userids of the EEM records must match those of the contact records. For information about creating EEM user records, see the *EEM Online Help*.

4. Assign Joe Supervisor as the Supervisor on Jane Analyst's contact record.
5. Create a CI (for example, Exchange Server).
6. Create a Root Cause code (for example, RAM Upgrade Required).
7. Create a Problem Area (for example, ProblemMgmt.SampleWF).
8. Associate the ProblemMgmt.SampleWF Problem Area with the Problem Management sample workflow.
9. Create a problem ticket with the following attributes:

Problem Area

Select ProblemMgmt.SampleWF.

Configuration Item

Select Exchange Server.

Asignee

Select Analyst, Jane.

10. Log in to CA Service Desk Manager as Jane Analyst and test the workflow configuration.

Note: You must complete the problem research before you complete the first form in the workflow. For information about performing the workflow tasks, see the *CA Workflow Online Help*.

More information:

[Set Up Automatic Login \(CA MDB Authentication\)](#) (see page 207)

[Set Up Automatic Login \(External Authentication\)](#) (see page 191)

[Start and Stop CA Workflow](#) (see page 187)

How to Configure the Order PC Sample Workflow

The Order PC sample workflow guides the change management process of ordering a new PC.

Note: For details about performing each of the CA Service Desk Manager tasks in the following process, see the *Online Help*.

To set up and test the Order PC sample workflow:

1. Create a CA Service Desk Manager contact for an employee (for example, William Employee) using the following values:

Contact Type

Select Employee.

Data Partition

Select Employee.

Access Type

Select Employee.

Email Address

Enter an e-mail address.

Note: For testing purposes, you may want to enter your own e-mail address.

Notifications Method

Select Email for Low, Normal, High, and Emergency notifications.

2. Create a CA Service Desk Manager contact for an analyst (for example, Jane Analyst) using the following values:

Contact Type

Select Analyst.

Data Partition

Select Service Desk Analyst.

Access Type

Select Service Desk Staff.

Email Address

Enter an e-mail address.

Note: For testing purposes, you may want to enter your own e-mail address.

Notifications Method

Select Email for Low, Normal, High, and Emergency notifications.

3. Create an associated EEM user record for the William Employee and Jane Analyst CA Service Desk Manager contacts.

Note: The userids of the EEM records must match those of the contact records. For information about creating EEM user records, see the *EEM Online Help*.

4. Create a Change Category (for example, PC.Order).
5. Assign Jane Analyst as the assignee of the Change Category.
6. Create a change order ticket with the following attributes:

Change Category

Select PC.Order.

Assignee

Select Analyst, Jane.

7. Log in to CA Service Desk Manager as Jane Analyst and test the workflow configuration.

Note: For information about performing the workflow tasks, see the *CA Workflow Online Help*.

Change Management Process Definition Example

This example demonstrates how the Change Management Process Definition manages change orders using the following ITIL v3 guidelines:

- Risk assessment
- Conflict and impact analysis
- Approvals by both the Change Manager and the CAB
- Implementation assessment and review

Note: This example assumes that CA Service Desk Manager is using CA Workflow and that the Change Management Process Definition is set up. For information about setting up the Change Management Process Definition, see the *Administration Guide*.

How to Prepare a Change Order for Approval

This example configures a change category and prepares a change order for approval. As an administrator, you configure the change category to use the Change Management Process Definition and set up groups and contacts. As a requester, you complete a Risk Assessment Survey, perform conflict and impact analysis, and perform change analysis.

Note: For information about configuring the change category and working with the Change Management Process Definition, see the *Online Help* and the *Administration Guide*.

To prepare a change order for approval, do the following:

1. Log in as administrator and create or edit a change category with the following options and click Use CA Workflow:
 - **CAB**—Select the CAB group.
 - **Group**—Specify the Implementation group.
 - **Risk Survey**—Select General.
 - **CA Workflow Definition Name**—Select Change Mgmt - Service Desk r12.1 (from the Workflow tab).
2. Create the following CA Service Desk Manager user IDs and contacts and assign them to the respective groups in CA Service Desk Manager and CA EEM:
 - **Don Requester**—The contact who creates the Change Order.
 - **John Approver**—The manager of the Implementation group who acts as the Change Manager to approve a change order. A manager of the CAB group who acts as a CAB approver.
 - **Sue Implementer**—The member of the Implementation Group who also completes the change order work items.
3. Log in as Don Requester and create a change order with the following values:
 - **Requester**—Specify Don Requester.
 - **Category**—Specify the change category with the Change Management Process Definition.
 - **Type**—Select Normal.
 - **Order Summary/Order Description**—Specify the reason for the change order.
 - **Schedule Start Date**—Specify a start date.
 - **Schedule Duration**—Specify the duration.
 - **CIs**—Specify the affected CIs (from Config Items tab).

Don Requester receives an email notification to complete the Risk Assessment Survey. The Change Order Detail page shows Status-RFC.
4. Click Risk Survey and answer the Risk Assessment Survey questions so the change order has a High Risk. Click Submit and Confirm.

The system generates a risk value for the change order and the requester receives an email notification to start conflict and impact analysis.

Note: You can only move to the next task after you click Confirm on the Perform Task page.
5. On the Workflow Tasks tab, follow the links to navigate to the change order Conflicts tab.
6. Click Conflict Analysis to review and resolve all scheduling conflicts for CIs.

7. On the Config Items tab, click Impact Analysis to review information about each CI.
8. On the Config Items tab, click Impact Explorer and navigate to the CMDB Relationships tab to consider the change order impact on related CIs.
9. Navigate to the Perform Task page and click Confirm.
Don Requester receives an email notification to perform change analysis.
10. On the Workflow Tasks tab, follow the links to navigate to the change order Chg Analysis tab. Answer the questions to confirm the change order and click Submit.
John Approver receives an email notification to approve the change order. The Change Order Detail page shows the following:
 - Status-Approval In progress
 - CAB Approval-Yes

How to Approve and Implement the Change Order

This example shows the approver and implementation tasks to perform while using the Change Management Process Definition to manage change orders.

As an approver, you review the change analysis information and approve the change order. As a member of the implementation group, you complete the assigned work on the change order and a post implementation review that describes the outcome of the change order.

Note: For information about working with the Change Management Process Definition, see the *Administration Guide*.

To approve and implement the change order, do the following:

1. Log in as John Approver and open the change order.
2. On the Workflow Tasks tab, follow the links to navigate to the Chg Mgr Approval tab and click Approve.
Because the change order is High Risk, the status is Approval in progress. Because this example also uses John Approver as a member of the CAB board, John Approver receives an email notification to review the change order as a CAB member.
3. On the CAB Approval tab, click Approve.
The Change Order Detail page shows the status of Approved. Sue Implementer receives an email notification.
4. Log in as Sue Implementer and open the change order.
5. On the Workflow Tasks tab, navigate to the Perform Tasks page and click Confirm.
The Change Order Detail page shows Status-Implementation in progress.
6. On the Workflow Tasks tab, follow the links to navigate to the Impl Complete tab.

7. Answer the questions to describe how the change order progressed and click Complete.

The change order closes. The Change Order Detail page shows Status-Implemented and Closure Code-Successful. Sue Implementer receives an email notification to perform the PIR.

8. On the Workflow Tasks tab, follow the links to navigate to the PIR tab.
9. On the PIR tab, answer the questions to describe the resolution and click Submit.

The change order closes. The Change Order Detail page shows Status-Closed. The Workflow Tasks tab shows the Close Change Order link with additional information. The Change Order Activities tab shows the final status of the Change Management Process Definition.

How to Configure CA IT PAM Workflow Integration

To configure the CA IT PAM Workflow Integration, do the following:

1. Verify that CA IT PAM and CA Service Desk Manager operate as stand-alone entities.
2. Configure the CA IT PAM Workflow Options.
3. Create the necessary groups and user IDs to grant CA Service Desk Manager users the appropriate access to CA IT PAM.

More information:

[Verify CA IT PAM and CA Service Desk Manager Installations](#) (see page 201)

[Configure CA IT PAM Workflow Options](#) (see page 202)

[CA IT PAM User Administration](#) (see page 205)

[How to Support Single Sign-On From CA Service Desk Manager to CA IT PAM Using CA EEM](#) (see page 206)

Verify CA IT PAM and CA Service Desk Manager Installations

You can integrate CA IT PAM and CA Service Desk Manager to coexist on a single server when the server architecture supports both products. When CA IT PAM or CA Service Desk Manager components cannot integrate on the same server, consider installing each product on separate servers.

Before you configure CA IT PAM and CA Service Desk Manager, you can confirm that both products are installed and working independently.

Note: For information about CA Service Desk Manager and CA IT PAM product requirements, see the *CA Service Desk Manager Release Notes*.

To verify CA IT PAM and CA Service Desk Manager installations

1. Open a browser on the server that hosts CA Service Desk Manager and verify that a CA IT PAM user can log in to CA IT PAM. Change the place holders to match the target CA IT PAM installation.

`http(s)://<server>:CA Portal/itpam`

2. Enter the following URL. Change the place holders to match the target CA IT PAM installation.

`http(s)://<server>:CA Portal/itpam/JNLPRRequestProcessor?processType=startUI`

The CA IT PAM product is accessible from the CA Service Desk Manager host.

Configure CA IT PAM Workflow Options

When you configure CA IT PAM Workflow options, you specify connectivity between CA Service Desk Manager and CA IT PAM. If you are using CA EEM for authentication, you also specify the CA EEM host name.

To configure CA IT PAM Workflow options

1. On the Administration Tab, select Options Manager, CA IT PAM Workflow.

The Option List appears.

2. Right-click the name of each option and select Edit from the short-cut menu. Configure the following options:

caextwf_eem_hostname

Specifies the name of the CA EEM server. For example, `http://pam.host.com` identifies the authentication host. You install `caextwf_eem_hostname` only if you configured CA IT PAM to use CA EEM as an authentication server. CA Service Desk Manager uses this value to transform a user name and password into an CA EEM token, so that a user name and passwords do not pass in plain text over HTTP.

Note: If the CA IT PAM installation is not using CA EEM, do not place a value in the `caextwf_eem_hostname` option, and do not install the `caextwf_eem_hostname`. Placing a false value or installing `caextwf_eem_hostname` when it is not necessary causes the integration to fail.

caextwf_endpoint

Specifies the URL that points to the CA IT PAM web services by including the CA IT PAM host name, port, and the mandatory `/itpam/soap` path. For example, `http://pam.host.com:CA Portal/itpam/soap` identifies the endpoint. Installing the `caextwf_eem_hostname` option is required for the integration between CA IT PAM and CA Service Desk Manager to operate properly.

caextwf_log_categories

Specifies a comma separated list of CA IT PAM process instance log category names to appear on the CA Service Desk Manager Request, Change Order, and Issue Workflow Tasks tab. For example, *Operator,Response,MyOwnCategory* supplies three log categories.

You install `caextwf_log_categories` based on business decisions from the CA Service Desk Manager and CA IT PAM process design personnel. This option adjusts the default data that appears on the Workflow Tasks tab for requests, change orders, and issues.

When you install the `caextwf_log_categories` option, all CA IT PAM process instance log messages from the Process category and the categories that you specify appear on the Workflow Tasks tab. When you do not install `caextwf_log_categories`, only the CA IT PAM process instance log messages from the Process category appear on the Workflow Tasks tab.

Note: For information about the CA IT PAM predefined log message categories, and defining custom message categories, see the CA IT PAM reference documentation.

caextwf_processdisplay_url

Specifies how to launch a graphical snapshot of a CA IT PAM process instance by supplying the host name and the mandatory `/itpam/JNLPrequestProcessor?processType=startUI&roid` path. For example, `http://pam.host.com:CA Portal/itpam/JNLPrequestProcessor?processType=startUI&roid=` launches a snapshot of a process instance. On the Workflow Tasks tab of a request, change order or issue, the user selects View Process to see the snapshot.

Installing the `caextwf_processdisplay_url` option is required for the integration between CA IT PAM and CA Service Desk Manager to operate appropriately.

caextwf_worklist_url

Specifies the process instance path by supplying the host name and the mandatory `/itpam?webPage=mytaskfilter&view=tasklist` path. For example, `http://pam.host.com:CA Portal/itpam?webPage=mytaskfilter&view=tasklist` enables CA Service Desk Manager users to see a list of CA IT PAM process instances that require attention. The list appears in CA IT PAM when the CA Service Desk Manager user selects a link associated with any listed task in the request, change order, or issue Workflow Tasks tab.

Installing the `caextwf_worklist_url` option is required for the integration between CA IT PAM and CA Service Desk Manager to operate properly.

caextwf_ws_password

Specifies the administrative password associated with the CA IT PAM user name from the `caextwf_ws_user` option. CA Service Desk Manager uses the user name and password to access the CA IT PAM web service functions to perform integration activities such as selecting start request forms, process definition information, and process instance information.

Installing the `caextwf_ws_password` option is required for the integration between CA IT PAM and CA Service Desk Manager. The password and user name that you specify requires the appropriate access to CA IT PAM. However, it is not necessary the CA IT PAM user name and password to exist within the CA Service Desk Manager contact records.

caextwf_ws_user

Specifies the CA IT PAM administrative user name associated with the CA IT PAM user name from the `caextwf_ws_password` option. CA Service Desk Manager uses the user name and password to access the CA IT PAM web service functions to perform integration activities such as selecting start request forms, selecting process definition information, selecting process instance information, or launching process instances.

Installing the `caextwf_ws_user` option is required for the integration between CA IT PAM and CA Service Desk Manager to operate. The user name and password that you specify requires the appropriate access to CA IT PAM. However, it is not necessary the CA IT PAM user name and password to exist within the CA Service Desk Manager contact records.

3. Click Install.
4. Restart the CA Service Desk Manager service.

The CA Service Desk Manager and CA IT PAM products can communicate even though there is no process instance data. CA Service Desk Manager and CA IT PAM are ready for you to create CA IT PAM process definitions and CA IT PAM start request forms.

Note: For more information about creating CA IT PAM process definitions and CA IT PAM start request forms, see the *Administration Guide* and the CA IT PAM user documentation.

CA IT PAM User Administration

Both CA IT PAM and CA Service Desk Manager, as stand-alone products, have individual requirements for authentication and authorization. To support a unified Service Oriented Architecture (SOA) strategy, you can configure both products to use CA EEM for authentication.

When you install CA IT PAM with CA EEM as the authentication server, the installer creates several policies and four essential entities by default:

- Two application users: itpamadmin, itpamuser
- Two application groups: ITPAMAdmins, ITPAMUsers

CA Service Desk Manager users who also use CA IT PAM can be divided between ITPAMAdmins and ITPAMUsers as follows:

- CA Service Desk Manager analysts must be members of ITPAMUsers when their duties entail:
 - Approving, rejecting, or otherwise responding to CA IT PAM Interaction Request Forms.
 - Listing CA IT PAM process instances assigned to the user.
 - Viewing the graphical display by clicking the View Process button of CA IT PAM's process status screen. The CA IT PAM ITPAMUsers group requires an additional CA IT PAM policy to grant access the graphic.
- CA Service Desk Manager analysts are members of ITPAMAdmins when their duties entail:
 - Creating and checking in CA IT PAM process definitions and/or start request forms.
 - Terminating process instances directly within CA IT PAM. Terminating process instances are an administrative exception to expected integration procedures.
 - Delegating CA IT PAM process instance tasks.
 - If the user is the user name defined in CA Service Desk Manager Options Manager.
- CA Service Desk Manager users require no access to CA IT PAM when their duties entail:
 - Creating requests, change orders, and issues that launch CA IT PAM instances.
 - Reviewing the Workflow tab which shows CA IT PAM process instance status and task information.

- Changing the status of a request, change order, or issue which causes the termination of a CA IT PAM process (such as canceling a change order.
- Selecting a CA IT PAM process definition on a CA Service Desk Manager request area, change category, issue category.

Note: For information about CA EEM configuration, see the CA IT PAM documentation.

How to Support Single Sign-On From CA Service Desk Manager to CA IT PAM Using CA EEM

When you have CA Service Desk Manager and CA IT PAM integrated, you can set up single sign-on to work from CA Service Desk Manager to CA IT PAM as follows:

1. Verify that the following requirements have been met:
 - CA Service Desk Manager and CA IT PAM are configured to use the same CA EEM installation.
 - The user that logs in to CA Service Desk Manager is also a user in CA IT PAM.
 - When CA EEM uses the internal database as a user store, the users must have either global permissions or belong to the same folder. Otherwise, if CA EEM references an external user store like an external directory or CA Siteminder, the users must be of the same store to access single sign-on.
2. Install CA EEM from the CA Service Desk Manager DVD install media or use any existing CA EEM install (for example, CA EEM for CA IT PAM).
3. On the CA Service Desk Manager Administration tab, install the following options from under the Options Manager, Security folder:
 - `eiam_hostname`
 - `use_eiam_authentication`

Note: You do not need to install the option `caextwf_eem_hostname` under Options Manager, CA IT PAM folder. But if you do install it, the value must be the same as `eiam_hostname` option.

4. Restart CA Service Desk Manager.
5. To create a user in CA IT PAM, do the following:
 - a. Log in to CA EEM using the CA IT PAM application context using the EiamAdmin userid or any other administration user.
 - b. Select the Manage Identities tab and click the icon next to the Users folder.

- c. On the New User page, the Name field at the top is the userid that must match the userid in the CA Service Desk Manager contact table.
- d. Click the Add Application User Details button and do the following:
 - Add any or both groups that are listed. Add at least one of these two groups in order to be able to log in to CA IT PAM.
 - Complete the New User fields, such as First Name, Last Name, Display, and Password.
6. Create a user in the CA Service Desk Manager contact table with the same userid. Verify that the Access Type Validation Type field for the user is set to CA EEM.

You can log in to CA EEM and CA Service Desk Manager with this user and the password specified in CA EEM.

Set Up Automatic Login (CA MDB Authentication)

To automate the login process for CA Workflow Worklist tasks when working with change orders, requests, incidents, problems, and issues, you can set up automatic login. When you use automatic login, you will not have to enter a userID and password when you click a Worklist link. Automatic login uses your Windows operating system domain credentials to log in to the Worklist.

Note: For automatic login to work correctly, the user does *not* have to be included in the Workflow Administrators, Workflow Process Initiators, or Workflow SuperUser groups in CA EEM. These groups provide different access levels to users logged in to the Workflow Design Environment.

To set up automatic login for Worklist tasks (Windows only)

1. [Install and configure CA Workflow](#) (see page 126).
2. [Configure Worklist and Workflow Manager for automatic login](#) (see page 192).
3. Configure CA EEM to use the CA MDB to authenticate users. When installing and configuring CA Workflow, your *Workflow User Name* and *Workflow User Password* are automatically added to CA EEM. For information, see your CA EEM documentation.

Note: A user must have a CA EEM user record to access the Worklist.

4. Log in to CA Service Desk Manager and create a change order, request, incident, problem, or issue with the appropriate category.

Note: Make sure the category is configured to use CA Workflow and has a CA Workflow definition attached to it before creating the change order, request, incident, problem, or issue.

You will now be able to click a link in the Worklist and log in without being asked for a userID and password.

More information:

[Configure Workflow Design Environment for Automatic Login](#) (see page 193)

Configure Single Point of Entry

When CA Service Desk Manager and CA Service Management are integrated, Single Point of Entry can permit single sign-on to CA Service Catalog.

To configure Single Point of Entry

1. Install the catalog_server General option.
Note: For more information, see the Options Manager online help.
2. Restart CA Service Desk Manager.
3. Navigate to the CA Service Desk Manager Employee Self-Service page.
4. Click Browse Catalog Services.

The CA Service Management logon page appears.

When CA EEM is configured for both CA Service Desk Manager and CA Service Management, Single Point of Entry can permit single sign-on to CA Service Catalog. When single sign-on is configured, the CA Service Management logon page is not displayed.

Note: Single sign-on is not available if you enter CA Service Desk Manager Employee Self-Service as a guest.

To configure single sign-on to CA Service Management

1. Verify that CA EEM Security options eiam_hostname and use_eiam_authentication are installed.
2. Set up your users in CA EEM and verify that these users are also CA Service Management users.
3. Click the Administration tab.
4. Open the Security and Role management folder.
5. Click Access Types
6. Select the Employee role.
7. Verify that the validation type under the Web Authentication tab is set to CA EEM-Use CA Embedded Entitlements Manager.
8. Log in as the defined CA EEM user and navigate to the Employee Self-Service page.
9. Click Browse Catalog Services.

The CA Service Management main page appears.

How to Implement Multi-Tenancy

The *service provider* is the primary tenant of a CA Service Desk Manager multi-tenancy installation.

Note: For more information about the service provider and multi-tenancy, see the *Administration Guide*.

You must define tenants and populate the tenant attribute in as many tenant-required and tenant-optional objects as you require before enabling multi-tenancy in other than setup mode. However, you can select multi-tenancy enforcement before assigning tenants to all objects that need them. Objects with a null tenant in a tenant-required table are treated as public objects visible to all users, and the user interface detects and supports updates to these objects. Depending on the extent of multi-tenancy enforcement, CA Service Desk Manager may or may not require a user to update a tenant-required object to supply a tenant.

Note: You must create a tenant before an instance of a tenant-required object can be updated. In addition, the `pdm_settenant` and `pdm_buildtenant` utilities can be used after activation of multi-tenancy to complete the setting of the attribute. If tenant required tables incorrectly contain untenanted data in a multi-tenancy system, a public data drop-down appears in tenant required tables.

Important! Before you run `pdm_buildtenant`, you *must* configure the service provider.

To implement multi-tenancy, complete the following steps:

1. [Set the Multi-Tenancy option to "setup."](#) (see page 211) and recycle CA Service Desk Manager. You must cycle services each time you change the tenant state (setup, on, on(allow)).

Important! When multi-tenancy is in setup mode, web interface changes are active for service provider administrators, allowing tenancy-related objects and data to be viewed and edited on the web interface. However, tenancy restrictions are not enforced, and users other than service provider administrators do not see any product interface changes. Therefore, you can continue to use the product in untenanted mode while being prepared for multi-tenancy.

2. [Map any existing tenant implementation to the tenant object.](#) (see page 213)
3. [Populate the tenant columns](#) (see page 214).

Note: We recommend that you populate the tenant column in contacts first, then use the tenanted contacts to populate the tenant column in the other objects.

4. Verify that the tenant is set in all tenant-required tables.

5. [Create tenants.](#) (see page 215)

Important! The first tenant that you create must be the service provider tenant. After you create the service provider tenant, log out of CA Service Desk and log in again as a member of the service provider. An easy way to do this login is to log in as the privileged user (for example, *ServiceDesk*), because this user automatically belongs to the service provider tenant. If you want to log in as a different contact, use `pdm_settenant` first to assign the contact to the service provider.

6. [Create tenant groups.](#) (see page 217)
7. [Set the Multi-Tenancy option to "on."](#) (see page 211)

After you select this option, restart services and complete the following steps:

- a. Log in using the privileged username (typically *ServiceDesk*).
- b. Verify that the privileged user is assigned to the service provider tenant.
- c. Verify that your multi-tenancy restrictions are enforced.

Important! If untenanted data remains in the database, you can set multi-tenancy enforcement to *warn* or *allow*. This allows updates to tenant-required tables with a null tenant, preventing the loss of data when a service level agreement (SLA) or attached event executes against a ticket that has not yet been updated to contain a tenant.

Note: You can run `pdm_settenant` even after multi-tenancy is active both to monitor the completeness of the implementation (with the `-r` option) and to mass-update the tenant as required.

8. Back up data partition constraints and roles.

Multi-tenancy reduces both the number and complexity of data partition constraints required, allowing them to be simplified. Before you make changes, back up both the `Domain_Constraint` and the `usp_role` tables.

9. Remove unneeded data partition constraints.

Many domain constraints that were defined for single tenancy are not needed after multi-tenancy is activated and can be removed. Removing these additional constraints improves query performance.

10. Review and audit.

Run `pdm_settenant -r` to review the need for additional tenant population in tenant-required tables. Review the need for adding tenancy to tenant-optional tables, such as Category and Activity Notification.

11. (Optional) Disable multi-tenancy if problems occur. If there are issues after implementation, you can optionally complete the following steps:

- a. Restore the Domain_Constraint and usp_role tables.
- b. Set the Multi-Tenancy option back to "setup".
- c. Recycle the system.

The site can resume previous operations while you continue to populate tenant columns and correct whatever issues required the reversion.

Note: If tenant required tables incorrectly contain untenanted data in a multi-tenancy system, a public data dropdown appears in tenant required tables and you get the following message: "AHD05358 There were *nn* untenanted active xxx objects at Service Desk startup."

Enable Multi-Tenancy

Using the product, you can enable multi-tenancy.

To enable multi-tenancy

1. Click the Administration tab.
2. In the tree on the left, click Options Manager, Multi-Tenancy.
The Option List page appears.
3. Click multi_tenancy.
The multi_tenancy Options Detail page appears.
4. Click Edit.
The Update Options page appears.
5. From the Option Value drop-down list, select *one* of the following options:

off

Disables the multi-tenancy feature.

setup

Enables multi-tenancy in setup mode. In this mode, CA Service Desk Manager displays editable tenant-related objects and attributes for service provider administrators, but does not enforce tenancy requirements. Other users can continue working as if multi-tenancy was not installed.

on

Enables the multi-tenancy feature as fully operational. You can select from the following values:

- **strict**—(Default) CA Service Desk Manager fails a checkin to a tenant-required table when tenant is null and the product cannot default it from an SREL to a tenanted table.
- **warn**—CA Service Desk Manager writes an error to the log but allows the checkin to proceed when a tenant-required object with a null tenant is created or updated.
- **allow**—CA Service Desk Manager writes a warning to the log but allows the checkin to proceed when a tenant-required object with a null tenant is created or updated.

Note: If you have not populated the `ca_tenant` table, you must select `setup` from the Option Value drop-down list. For new implementations of multi-tenancy, select `setup`.

6. Click Install.

The `multi_tenancy` option is installed.

7. Click Refresh.

The page displays your changes.

8. Close the window.

The Option List page reappears.

9. Restart services.

How to Initialize a New Tenant

As the service provider, you may want to build a standard set of data for a new tenant, such as categories, data partitions, ticket templates, and so on. This task can be done by using `pdm_extract` or `pdm_tenant_extract` to build a `pdm_userload` input file containing the data you want.

If necessary, you can edit this file with any text editor. It can then be loaded into the database using `pdm_userload` with the `-t` argument to set the tenant column to the new tenant. For information on `pdm_userload` arguments other than `-t`, see the *Administration Guide*.

The following process describes how to initialize a new tenant:

1. Create the tenant in the `ca_tenant` table.
Use the online [Create Tenant](#) (see page 215) page.
2. Load the standard data as previously described.
Use `pdm_userload -t` to set the tenant.
3. Create contact records for the new tenant.
Load outside data or use `pdm_userload -t`.

How to Convert an Existing Tenant Implementation to the Tenant Object

You may have used data partitions and another CA Service Desk Manager object to achieve some of the functionality now provided by multi-tenancy. If you want to convert an implementation to multi-tenancy, the first step is to map the data in the previously-used object to the new tenant object. The previously-used object is called the *pre-tenant* object. For most sites with these requirements, the `org` (organization) object is the pre-tenant object, but the following approach can be used for any pre-tenant object.

1. If the pre-tenant object is not `org`, verify that its Majic object definition specifies `TENANT_REQUIRED`.
2. Verify the attribute mappings from the pre-tenant object to the new tenant object in the `buildtenant.xml` file in the following location:

`$NX_ROOT/samples/multi_tenancy`

Note: You must copy `buildtenant.xml` to the `$NX_ROOT/site/cfg` directory. In addition, `buildtenant.xsd` must be in the same directory as `buildtenant.xml`, or you will receive an error. When you install the product, `buildtenant.xsd` is located in `$NX_ROOT/site/cfg`, so you do not have to copy this file.

The default settings are based on `org`. If the pre-tenant object is not `org`, you must edit the file.

3. [Run `pdm buildtenant -f`](#) (see page 222).
A new tenant is created for each pre-tenant object, and sets the tenant attribute in the pre-tenant object to reference the new tenant.
4. Log in to CA Service Desk Manager, and review both the tenant object and the pre-tenant object.

Note: In some situations, you want to map multiple pre-tenant objects to a single tenant object. To do this, manually update the pre-tenant objects affected, and then delete or inactivate the unused tenants.

How to Populate the Tenant Attributes in Your Tables

To populate the tenant attribute in all or a subset of a table, use the *pdm_settenant* utility. This utility uses a configuration file to select the objects to be tenanted and to specify where to obtain the tenant for the objects. You can specify an explicit tenant, or specify that the tenant should be derived from an SREL reference in the object to be tenanted.

To populate the tenant attributes in your tables using *pdm_settenant*, complete the following steps:

1. Create or edit a configuration file.

The configuration file selects the rows that will have their tenant attribute set and specifies a source for the tenant attribute's value. The product provides a sample *settenant.xml* file in the following location:

`$NX_ROOT/samples/multi_tenancy`

Note: You can modify the sample *settenant.xml* file, or create a file and copy it to the `$NX_ROOT/site/cfg` directory. In addition, *settenant.xsd* must be in the same directory as *settenant.xml*, or you will receive an error. When you install the product, *settenant.xsd* is located in `$NX_ROOT/site/cfg`, so you do not have to copy this file.

2. [Run *pdm_settenant* -f \[configuration file\] -r](#) (see page 223)

The *pdm_settenant* utility reads its configuration file and processes each rule it defines in sequence.

We recommend that you use this utility first to populate the tenant attribute in the *cnt* (contact) object, and then use the *cnt* object as a source for populating tenant into other objects.

After the *cnt* object is properly tenanted, it can be used as a base for setting tenant in other tables by completing the following steps:

- a. Specify a *TenantRule* with `type="SREL"` in the configuration file for an attribute referencing the *cnt* object to set tenant in other tables.
 - b. (Optional) Specify a *TenantRule* with `type="Name"` < tenantname > to set tenant explicitly in some of the tables.
3. Run *pdm_settenant* with a new configuration file.
 4. Rerun *pdm_settenant* as required.

After you have populated the tenant column in an object, you can use SRELs to that object as the basis of an SREL *TenantRule* for setting tenant in other objects.

Example: SREL Type Syntax

SREL type syntax checks for cnt objects that do not have a tenant value specified and uses the tenant value from the linked organization object:

```
<Object name="cnt">  
<TenantRule type="SREL">organization</TenantRule>  
<Where>tenant is null</Where>  
</Object>
```

Example: Name Type Syntax

Name type syntax checks for org objects that do not have a tenant value specified and sets their tenant field to the name of an actual Tenant object:

```
<Object name="org">  
<TenantRule type="Name">Tenant A</TenantRule>  
<Where>tenant is null</Where>  
</Object>
```

Create a Tenant

You can use the product to create a tenant.

To create a tenant

1. Select Security and Role Management, Tenants on the Administration tab.

The Tenant List page appears.

Note: The Security and Role Management, Tenants option is available only when multi-tenancy is installed (either on or setup).

2. Click Create New.

The Create New Tenant page appears.

3. Complete the editable fields if necessary:

Name

Displays the tenant name.

Service Provider

Identifies whether a tenant is the service provider. The first created tenant is always the Service Provider.

Tenant Number

(Information Only) Displays the tenant number. This field is not used by CA Service Desk Manager.

Record Status

Sets the tenant to Active or Inactive.

Parent Tenant

Specifies another tenant above this tenant, making this tenant a *subtenant* in a tenant hierarchy.

Subtenants Allowed

Allows this tenant to have subtenants. The tenant cannot modify the setting.

Tenant Depth

(Information Only) Indicates the tenant depth of this tenant.

Supertenant Group

(Information Only) Identifies the system-maintained tenant group that contains this tenant and all tenants above it in the tenant hierarchy.

Subtenant Group

(Information Only) Identifies the system-maintained tenant group that contains this tenant and all tenants below it in the tenant hierarchy.

Foreign Key Group

(Information Only) Identifies the system-maintained tenant group that contains tenants that can be referenced from an SREL in data that belongs to this tenant. The foreign key group is the same as the supertenant group.

Related Tenant Group

(Information Only) Identifies the system-maintained tenant group consisting of both the supertenant and subtenant groups for this tenant.

Terms of Usage

Specifies the Terms of Usage statement for the tenant.

Logo

Specifies the URL for the tenant logo file, which can be any web image type.

Location

Displays the Location lookup page.

Contact

Displays the Contact lookup page.

Note: If no contact is associated with the respective tenant, the Email Address and Pager Email Address fields are inactive.

4. Click Save.

The tenant is created.

5. Close the window.

6. Right-click the Tenant list and click Refresh.

The Tenant List is updated and displays the created tenant.

7. (Optional) To assign this tenant to user-defined tenant groups, click Update Tenant Groups on the Tenant Groups tab.

Create a Tenant Group

You can use the product to create a tenant group.

To create a tenant group

1. On the Administration tab, select Security and Role Management.

2. Click Tenant Groups.

The Tenant Group List appears.

Note: The Security and Role Management, Tenant Groups option is available only when multi-tenancy is installed (either on or setup).

3. Click Create New.

The Create New Tenant Group page appears.

4. Complete the following fields:

Tenant Group Name

Displays the name of the tenant group.

Record Status

Sets the tenant group as active or inactive.

Description

Displays a description of the tenant group.

5. Click Save.

The tenant group is created.

6. Close the window.

The Tenant Group List appears.

7. Right-click the Tenant List and select Refresh.

The Tenant Group List is updated.

8. Click Update Tenants on the Tenant Group Detail page to add tenant members to the group.

Tenant Hierarchies

A *tenant hierarchy* is a structured tenant group that is system-created or modified when you assign a *parent tenant* to a tenant. The tenant becomes a *subtenant* of the parent and higher tenants (if any) in that hierarchy.

Note: The service provider can create multiple unrelated hierarchies, or none. Even in a system with tenant hierarchies, you can define standalone tenants.

A subtenant typically represents a subdivision within its *supertenants*. A subtenant can have its own business rules and data, and supertenant data is "pushed" to the subtenant automatically on a read-only basis.

CA Service Desk Manager supports a tenant hierarchy of unlimited depth. However, the *service provider* can specify a limit on the total number of tenants and the depth of tenant hierarchies (default is four levels). The service provider also determines whether individual tenants can have subtenants.

Note: The service provider can participate in tenant hierarchies, but this is not required. The service provider cannot have a parent tenant.

Create a Subtenant

Subtenancy allows you to build and modify tenant hierarchies for organizational and data-sharing purposes. To place a tenant into a tenant hierarchy, you assign it a parent tenant.

To create a subtenant

1. On the Administration tab, select Security and Role Management, Tenants.

The Tenant List appears.

Note: The Security and Role Management, Tenants option is available only when multi-tenancy is enabled.

2. Click an existing tenant to Edit, or click Create New.

The Tenant Detail page appears. Enter any required data or changes.

3. Select a Parent Tenant.

Note: The Parent Tenant drop-down only displays tenants that are allowed to have subtenants.

4. Click Save.

The tenant is a subtenant of the parent tenant.

Note: When a tenant is a subtenant, it belongs to the Subtenant group of the parent tenant, as do the subtenants (if any) of that subtenant, and so on. The parent tenant joins the Supertenant group of the subtenant, as do the supertenants (if any) of that supertenant, and so on. Each joins the Related Tenants group of the other.

System-Maintained Tenant Groups

CA Service Desk Manager generates and maintains three tenant groups automatically for each tenant in a tenant hierarchy (*tenant* is the tenant name):

- *tenant_subtenants* (tenant, its child tenants, and their lower subtenants)
- *tenant_supertenants* (tenant, its parent tenant and its higher supertenants)
- *tenant_relatedtenants* (entire single hierarchy)

System-maintained tenant groups can be used like user-defined tenant groups. However, only their names and descriptions can be modified.

How to Export and Import Tenant Data

The service provider can extract tenant data from an existing multi-tenancy implementation and import it into a new system.

Note: Depending on the volume of your data, the extraction process can take several hours. You may have to perform the extract and import in multiple phases, as follows:

Initial

Extracts a base line and creates a control file used in subsequent phases .

Update

Uses the control file to extract only data that has changed since the previous run.

Final

Performs the same steps as Update, except that it also extracts animations. Animations are omitted from both the Initial and Update phases.

To extract data from one database and import it into another, complete the following steps:

1. Run an initial phase of [pdm_tenant_extract](#) (see page 227) to extract base-line data. This builds the control file used by subsequent phases.
2. Prepare a new, clean MDB for the extracted data.

Important! The output from the initial phase *must* be loaded into a database that has never been used for the product or for any other product. Each table loaded from initial phase data is truncated prior to the load, which could cause loss of data if the database is already in use.

3. To avoid duplicate privileged contacts appearing on the new system, you must inactivate the privileged contacts. Log onto CA Service Desk Manager and change the Status of these contacts to "inactive" before loading the extracted data.
4. To avoid referential problems during the data load, run the appropriate drop constraints script:
 - (Oracle) Run `$NX_ROOT/samples/views/Oracle/OracleDropConstraints.sql`
 - (SQL Server) Run `$NX_ROOT/samples/views/SQLServer/SQLDropConstraints.sql`
5. Use [pdm_userload](#) (see page 228) to load the data from the initial phase into the clean MDB prepared in steps 2 and 3.
6. Run an update or final phase of `pdm_tenant_extract` to extract additional data created or modified since the previous phase. `pdm_tenant_extract` uses the control file created in step 1 to determine the data already processed by the previous phase.
7. Use `pdm_userload` to load the data extracted in step 5 into the same MDB containing data loaded from the previous phases.

Note: For more information about this utility, see [pdm_userload](#) (see page 228).
8. Repeat steps 5 and 6 as required until all data has been imported into the new database. The last run should be the final phase.
9. To protect the integrity of the new database, restore the constraints dropped in step 3 by running the appropriate add constraints script:
 - (Oracle) Run `$NX_ROOT/samples/views/Oracle/OracleAddConstraints.sql`
 - (SQL Server) Run `$NX_ROOT/samples/views/SQLServer/SQLAddConstraints.sql`
10. Use [pdm_tenant_delete](#) (see page 226) to delete the extracted data from the original database.
11. Ensure that all repositories associated with extracted tenants are copied to the target settings.

How to Handle Attachments and Repositories

Attachments are stored in repositories. You must copy all repositories that are associated with extracted tenants to the target system, including public repositories. This process is primarily a manual operation, with the following steps:

1. Redefine location-specific information for all repositories, after completion of the initial load of the data into the target system. This task includes changing the following values:
 - Server Name
 - Upload Path
 - Servlet Path
 - Archive Path
2. Manually create all required directories and folders.
3. Copy all attachment files from the previous location to the new repository location after (or during) the load of the data from the Final phase.

After you complete these steps, all references for attachments in the target system should be successful. However, copies of the attachments remain on the source system. [Use the `pdm clean_attachments.pl` utility](#) (see page 223) to clean these redundant attachments.

Utilities Used for Multi-Tenancy

This section describes utilities that are used to manage a multi-tenancy environment.

Note: Required parameters are enclosed within "{ }", while optional parameters are in "[]".

More information:

[pdm buildtenant—Creating Tenants from Another Object](#) (see page 222)

[pdm clean_attachments—Delete Redundent Attachments After Importing Tenant Data](#) (see page 223)

[pdm settenant—Assigning Tenants to Objects](#) (see page 223)

[pdm tenant_delete—Deleting Tenant Data from a Database](#) (see page 226)

[pdm tenant_extract—Extracting Tenant Data](#) (see page 227)

[pdm userload—Load Tenant Data](#) (see page 228)

pdm_buildtenant—Creating Tenants from Another Object

The *pdm_buildtenant* utility is used to create tenants from another object. You may have used data partitions and another CA Service Desk Manager object to achieve some of the functionality now provided by multi-tenancy. If you want to convert an implementation to multi-tenancy, the first step is to use *pdm_buildtenant* to map the data in the previously-used object to the new tenant object.

Important! Before you run *pdm_buildtenant*, you *must* configure the service provider.

In this section, the object used to hold tenant-like information is named the pre-tenant object. For most sites with these requirements, the org (organization) object is the pre-tenant object, but the following approach can be used for any pre-tenant object.

The *pdm_buildtenant* utility builds the tenant objects from pre-tenant objects. This application creates a tenant for each pre-tenant object, and sets the tenant attribute in the pre-tenant object to reference the new tenant. This utility has the following syntax:

```
pdm_buildtenant [-h] | [-f [configuration_file]]
```

-f configuration_file

(Optional) Specifies the location of a configuration file specifying the rules for creating tenants from the pre-tenant object. If this argument is not included, *pdm_buildtenant* uses the configuration file from the `$NX_ROOT/site/cfg` directory. This file assumes the pre-tenant object is org; if this is not the case, you *must* edit the configuration file before using *pdm_buildtenant*.

Note: You *must* copy *buildtenant.xml* to the `$NX_ROOT/site/cfg` directory. In addition, *buildtenant.xsd* must be in the same directory as *buildtenant.xml*, or you receive an error. When you install the product, *buildtenant.xsd* is located in `$NX_ROOT/site/cfg`, so you do not have to copy this file.

-h

Displays usage information for *pdm_buildtenant*.

The following is the format of the configuration file:

```
<?xml version="1.0" encoding="utf-8" ?>
<BuildTenant>
  <Object from="MajicObjectName">
    <Attribute from="sourceAttribute1" to="tenantAttribute1" />
    <Attribute from="sourceAttribute2" to="tenantAttribute2" />
  </Object>
</BuildTenant>
```

The *from* attribute of the Object tag identifies the pre-tenant object. Each Attribute tag identifies an attribute to be copied from the pre-tenant object to an attribute of the new tenant.

Important! For UNIX implementations of multi-tenancy, you *must* run *pdm_task* to export LIBPATH before executing the *pdm_settenant* and *pdm_buildtenant* utilities. If you do not run *pdm_task* before executing these utilities, you receive system errors. Use *../pdm_task* to run the command.

pdm_clean_attachments—Delete Redundent Attachments After Importing Tenant Data

After importing tenant data, you should delete redundant attachments. This utility has the following syntax:

```
pdm_perl pdm_clean_attachments.pl [-h] | [-n repository_name] | [-S|-K]
```

-h

Specifies to display command line help.

-n repository_name

Specifies the name of the repository to process. If not specified, all repositories are processed.

-S

Specifies that only CA Service Desk Manager repositories are processed.

-K

Specifies that only Knowledge Management and Embedded Images repositories are processed.

Note: Running the *pdm_clean_attachments.pl* command without any arguments processes all repositories.

Important! On UNIX, the LIBPATH must be set before running several CA Service Desk Manager utilities. Use *pdm_task* to set the LIBPATH before running a utility. For example, input "*pdm_task pdm_clean_attachments ...*".

pdm_settenant—Assigning Tenants to Objects

After you define tenants, you must use the *pdm_settenant* utility to set the tenant column in other objects. This utility has the following syntax:

```
pdm_settenant [-h] | {-f [configuration_file] | -r} [-d domsrvr]
```

-d domsrvr

(Optional) Specifies a domsrvr to use. If this argument is not specified, *pdm_settenant* uses the default domsrvr.

-f configuration_file

(Optional) Specifies the location of a configuration file specifying the data that will be updated and the rules for updating the file. If this argument is not specified, pdm_settenant uses the configuration file from the \$NX_ROOT/site/cfg directory (after the configuration file is copied to the \$NX_ROOT/site/cfg folder).

Note: You can modify the sample settenant.xml file, or create a file and copy it to the \$NX_ROOT/site/cfg directory. In addition, settenant.xsd must be in the same directory as settenant.xml, or you will receive an error. When you install the product, settenant.xsd is located in \$NX_ROOT/site/cfg, so you do not have to copy this file.

The following sample XML code describes the format of this file:

```
<?xml version="1.0" encoding="utf-8" ?>
<SetTenant>
  <Object name="MajicObjectName">
    <TenantRule type="SREL">MajicColumnName</TenantRule>
    <Where>tenant is null</Where>
  </Object>
  <Object name="MajicObjectName">
    <TenantRule type="Name">TenantName</TenantRule>
    <Where>tenant is null</Where>
  </Object>
</SetTenant>
```

Each Object tag specifies a CA Service Desk Manager object to be tenanted. The TenantRule tag specifies how pdm_settenant should determine the tenant, and the Where tag selects the objects to be tenanted. There are two types of TenantRule tags:

- type="Name"
Specifies an explicit tenant by name.
- type="SREL"
Specifies an SREL attribute in the object. Pdm_settenant copies the tenant of the object referenced by the SREL.

-h

Displays usage information for pdm_settenant.

-r

Outputs a report displaying the total number of rows in each tenant-required table, and how many have a null tenant column.

Note: If both the -f and -r arguments are specified, pdm_settenant outputs a report after completing its update. If you only specify the -r argument, pdm_settenant outputs a report, but does not update any data.

Running `pdm_settenant` without any arguments displays usage information. To run `pdm_settenant` using the default configuration file, specify the `-f` option without the `configuration_file` argument. The `pdm_settenant` utility reads its configuration file and processes each rule it defines in sequence. It writes output to the `pdm_settenant.log` file in the `$NX_ROOT/log` directory.

You can run `pdm_settenant` as many times as needed. The first pass may take a significant time (possibly several hours at a large site). Subsequent passes run faster, as they only need to process rows that have not been updated. This prepares the database prior to installing the multi-tenancy option.

Important! On UNIX implementations of multi-tenancy, you must run `pdm_task` to export `LIBPATH` before executing the `pdm_settenant` and `pdm_buildtenant` utilities. If you do not run `pdm_task` before executing these utilities, you will receive system errors. Use `../pdm_task` to run the command.

Assign Tenants to Objects Considerations

After you define tenants, you can use the `pdm_settenant` (assign tenants to objects) utility to set the tenant column in other objects. When you change the tenant for an object, you must consider whether to change the tenancy on related tenanted objects in order to maintain data integrity. Failure to keep these objects synchronized can cause data to appear missing from CIs, relationships, MDRs, versioning, and so on. The following CA CMDB objects are tenanted:

- `nr`—CI definitions
- `nr_com`—Log entries associated with a CI
- `bmhier`—Relationships associated with CIs
- `mdr_idmap`—MDR provider definitions
- `ci_mdr_idmap`—CI/MDR federated mappings

For each CI, do the following to synchronize data when you use `pdm_settenant` to change tenancy:

- Specify `nr` for the CI object name.
- Change the log entries associated with the CI in `nr_com` so that you can view the log entries for the new tenant.

Note: For information about the `pdm_settenant` command, see the *Administration Guide*.

Example: XML to Change the Tenant and Log

The following XML changes the tenant for a CI named CITest to T2 and also changes the corresponding log entries in nr_com:

```
<TenantRule type="Name">T2</TenantRule>
<Where>name = 'CITest'</Where>
</Object>
<Object name="nr_com">
<TenantRule type="Name">T2</TenantRule>
<Where>asset_id.name = 'CITest'</Where>
\</Object>
```

pdm_tenant_delete—Deleting Tenant Data from a Database

The *pdm_tenant_delete* utility removes all data for a specified tenant from the database.

Important! The referential constraints on the *ca_* tables must be dropped before running *pdm_tenant_delete* and restored afterwards.

This utility has the following syntax:

```
pdm_tenant_delete -h|-t tenant_name [-C|-R] [-Q]
```

-h

Displays the usage information for *pdm_tenant_delete*.

-t *tenant_name*

Specifies the name of the tenant of the data to be deleted.

Note: The tenant must be marked inactive before you can use this utility to delete the data.

-C

Specifies that all contacts for a tenant will be marked inactive. Since contacts can be shared between products, default logic should not mass delete or mass inactivate contacts unless explicitly requested.

Note: This option is ignored if the -R option is specified.

-R

Specifies that all rows in all tenanted tables marked CA_COMMON in *ddict.sch* will be deleted, including the tenant object itself.

Important! These tables are shared between multiple products, so use this option with caution.

-Q

Specifies quick query processing to execute database queries as fast as possible. If this argument is not specified, the utility uses background query processing so that queries run only when the system is otherwise idle. This argument improves running time at the expense of a greater impact on an active system.

Important! On UNIX, the LIBPATH must be set before running several CA Service Desk Manager utilities. Use *pdm_task* to set the LIBPATH before running a utility. For example, input "pdm_task pdm_clean_attachments ...".

pdm_tenant_extract—Extracting Tenant Data

The *pdm_tenant_extract* utility extracts all data for a specified tenant from the database. It extracts the data in *pdm_userload* format so that it can easily be loaded into another database. This utility has the following syntax:

```
pdm_tenant_extract -h | -c control_file [-d domsrvr] [-g yes|no] [-o output_file] -p phase [[-t tenant_name]...] [-Q]
[table1 [table2...]]
```

-h

Displays the usage information for *pdm_tenant_extract*.

-c control_file

Specifies the location of the control file for this tenant extract. For the Initial phase, the file is created in the specified location (and must not already exist). The file must exist for the Update and Final phases.

-d domsrvr

(Optional) Specifies a domsrvr to use.

-g yes|no

(Optional) Specifies whether or not public data is included in the output file. If this argument is not specified, public data from all tables is included.

-o output_file

(Optional) Specifies the location of the output file. If this argument is not specified, output is directed to stdout.

-p phase

Specifies the phase of the extract. Use one of the following values:

I—Initial

U—Update

F—Final

-t *tenant_name*

Specifies the name of a tenant to be extracted. This argument is required for the Initial phase and can be repeated for multiple tenants. It is not valid on the Update and Final phase.

-Q

Specifies quick query processing to execute database queries as fast as possible. If this argument is not specified, the utility uses background query processing so that queries run only when the system is otherwise idle

table1 [table2...]

(Optional) Specifies the tables to extract. If omitted, all tables are extracted.

Important! The output from the initial phase must be loaded into a database that has never been used for CA Service Desk Manager or for any other product. Each table loaded from initial phase data is truncated prior to the load, which could cause loss of data if the database is already in use.

Note: To avoid referential problems during the data load, run the appropriate drop constraints script (\$NX_ROOT/samples/views/Oracle/OracleDropConstraints.sql or \$NX_ROOT/samples/views/SQLServer/SQLDropConstraints.sql). After the loads complete, re-apply the constraints with the appropriate xxxAddConstraints.sql script found in the same directory.

Important! On UNIX, the LIBPATH must be set before running several CA Service Desk Manager utilities. Use *pdm_task* to set the LIBPATH before running a utility. For example, input "pdm_task pdm_clean_attachments ...".

More information:

[How to Export and Import Tenant Data](#) (see page 219)

pdm_userload—Load Tenant Data

The *pdm_userload* utility is used to load data into a CA Service Desk Manager database. This utility is available even if multi-tenancy is not installed. Multi-tenancy adds support for one additional argument (-t) that specifies the name of a tenant whose id must be inserted into the tenant column of all rows inserted or updated into a tenanted table. The specified tenant must already be in the database.

When extracting data, complete the following steps to avoid errors in stdlog:

1. Before you start loading data, shut down CA Service Desk Manager and restart the product in DBADMIN mode as follows:

Windows

Run `pdm_d_mgr -s DBADMIN`

UNIX

Run `pdm_init -s DBADMIN`

2. After the data loads, shut down CA Service Desk Manager using the `pdm_halt` command.
3. Restart CA Service Desk Manager in normal mode.

Note: For more information about the `pdm_userload` utility, see the *Administration Guide*.

Chapter 6: Customizing

This section contains the following topics:

[Customization Overview](#) (see page 231)

[Notification Method Customization](#) (see page 232)

[Query and Message Customization](#) (see page 239)

[Schema Customization](#) (see page 253)

[Web Interface Customization](#) (see page 266)

[Event Log Data Storage Customization](#) (see page 349)

[CA Business Intelligence Reports Customization](#) (see page 351)

[Customizing Legacy Reports](#) (see page 366)

Customization Overview

CA Service Desk Manager is an exceptionally flexible product designed to fulfill various IT Service Management functions. The product provides a broad feature set, and various best-practice content to help ensure that your service management needs are met as rapidly, and fully, as possible.

While we believe that the default implementation of CA Service Desk Manager closely matches the processes and terminology used in most IT organizations, we recognize the need to extend the product to work with the unique aspects of your organization. To that end, the product includes a broad spectrum of approaches to customize the product to meet your unique needs, including the following:

- End-user personalization
- Systemwide configuration
- Tool-based adaptation
- Code-level customization

Different types of approaches exist that are available to customize the product.

Notification Method Customization

The CA Service Desk Manager automatic notification methods notify personnel at key points in the service desk management process. The standard notification methods are shown as follows:

- Email
- Notification (Log)
- Pager_Email

You can define customized notification methods to specify a new method of transmission, such as voice mail, display boards, or a specific printer. You can also access data from another application and include it in the notification message.

The Notification Process

Ticket notifications (applicable to issues, change orders, and requests) are processed when the ticket is saved, as described by the following:

- If the notification method is other than Notification, such as Email, the notification processor executes the notification method for each contact in the list. This method is typically an executable or shell script, which is launched in a new process. Details about the notification are stored in environment variables for easy access by the executable/script.
- For each notification requested, the notification processor sets the NX_NTF_MESSAGE and NX_NTF_SUMMARY environment variables using the Notification Message Title and Notification Message Body information provided on the Message Template notebook page of the Activity Notifications Detail window. If the recipient is a valid contact, additional environment variables are created using information in their Contact Detail record.
- If the Write To File option is selected for the notification, a text file is created with additional information that the notification method can use to obtain more detailed information.
- A list of contacts to receive the notification is built from the information provided on the Objects, Contacts, Types, and Survey notebook pages of the Activity Notifications Detail window. For those having a notification method matching the Notify Level and the log_all_notify Options Manager option installed, a notification is generated first to the notification log.

Notification Method Variables

Two sets of variables are created and made available to the notification method.

Basic Environment Variables

The first set of variables is created for every notification sent, independent of whether you select the Write To File option for the notification. They are written to the environment as environment variables that can be accessed by the notification method in the standard way. If you select the Write To File option for the notification method, these variables are also written to the notification file in the notification section.

The following environment variables give you basic information about the notification. They are always defined, even if the corresponding value is empty:

Environment Variable	Description
NX_NTF_MESSAGE	The completed message template text, including full expansion of all variables
NX_NTF_SUMMARY	The completed message template header, including full expansion of all variables
NX_NTF_URGENCY	The notification urgency (1 is low, 4 is emergency)

The following environment variables are created only if the recipient is a valid CA Service Desk Manager contact, in which case they are set using values from the recipient's Contact Detail record as shown in the following table:

Variable	Contact Detail Window Fields
NX_NTF_BEEPER_PHONE	Pager Number
NX_NTF_COMBO_NAME	Last Name, First Name, Middle Name
NX_NTF_CONTACT	Contact ID
NX_NTF_EMAIL_ADDRESS	Email or Pager Email Address (depending on notification type)
NX_NTF_FAX_PHONE	Fax Number
NX_NTF_PUBLIC_PHONE	Phone Number
NX_NTF_USERID	User ID
NX_NTF_VOICE_PHONE	Alt. Phone #

Note: These variables are not created if the corresponding values are empty (with the noted exception of NX_NTF_CONTACT, which cannot be empty).

Attribute Variables

The second set of variables, called attribute variables, is available only if you select the Write To File option when you define the notification method. They are written to the notification file only—not to the environment. They are of the form:

```
NX_NTF_attribute[.secondary_attribute]=value
```

where:

attribute

The name of the attribute whose value you want to obtain. This is the attribute name as defined for the object. For a complete list of all attribute names for any object, see the CA Service Desk Manager Technical Reference Guide. The most common objects associated with notifications are the ticket, which has an object name dependent of the type of ticket (for example, cr for requests), and the contact identifying the recipient, which has an object name of cnt. For example, the environment variable for the description attribute of a ticket might look as follows in the notification file:

```
NX_NTF_DESCRIPTION=This is a sample description.
```

secondary_attribute

If the first *attribute* is an internal identifier for another object, a secondary attribute is often attached to give more meaningful information using the dot notation. In database terms, attribute is a foreign key that points to a row in another table, rather than a simple data value. Using this raw key value would probably have little meaning. To save you the effort, many of these types of fields are resolved or de-referenced for you. When this is the case, *secondary_attribute* will be the value in the referenced table. For example, instead of writing the value for the assignee attribute, which is actually stored as the unique ID of the contact record for the assignee, the assignee's combined name is written by referring to the *combo_name* attribute for the contact object, as shown in the following example:

```
NX_NTF_ASSIGNEE.COMBO_NAME=Armstrong, Beth
```

If an attribute does not have a value, the corresponding value is usually (NULL) or blank. For example:

```
NX_NTF_CALL_BACK_DATE=(NULL)
```

```
NX_NTF_GROUP.COMBO_NAME=
```

Note: An attribute variable that exists for both the ticket and the recipient is `NX_NTF_ID` (the id attribute), which is the unique database ID for the object.

The Notification File

If you select the Write To File option when you define a notification method, all the basic environment and attribute variables are written to a text file, which is closed before executing the notification method script or program. This notification file is written every time the notification method is invoked for a contact, and is a handy mechanism for passing relevant information to the notification script that is not otherwise available in the environment.

The full path of the notification file is set in the `NX_NTF_FILENAME` environment variable, which is available to the notification method's process. The file name is also added to the end of value you enter in the Notification Method field when defining the notification method. For example, if the Notification Method is `'pdm_perl -w mymethod.pl'`, then the actual process executes `'pdm_perl -w mymethod.pl unique_notification_file_name'`.

Important! The administrator can clean up the notification files. This clean-up is especially important for a site using a high volume of notifications, which can result in thousands of notification files a day. The files are located in the standard temporary directory (TEMP on Windows and TMP on UNIX). One suggestion is to delete the file at the end of the notification method script/program.

The notification file is a standard text file that is divided into sections. Each line contains either an attribute/value pair or a section marker. Each notification file has three sections, as described by the following. All sections begin with "-----" followed by a new line.

SECTION=obj, where obj identifies the object type of the ticket

Iss

Provides information about the issue.

Chg

Provides information about the change order.

Cr

Provides information about the request.

SECTION=cnt

Provides information about the recipient.

SECTION=notification

Provides the same information that is available from the basic environment variables.

Note: The section names for the ticket and recipient are actually the object names for the attributes in that section. For a complete list of all attribute names for any object, see the *CA Service Desk Manager Technical Reference Guide*.

Several lines of attribute/value pairs, each of which represents an attribute of the corresponding object, are contained in each section. The Attribute Variables in this section provide the detailed information about how these lines are formatted and what they mean.

Line breaks in an attribute value are reproduced as new lines in the notification file. Your notification method process can only use the attribute or value lines that begin with NX_NTF, and section markers. Generate a sample file and look at its contents before you work with a notification file in your notification method process.

More information:

[Basic Environment Variables](#) (see page 233)

[Attribute Variables](#) (see page 234)

Using Perl Scripts

Most notification methods invoke an executable or shell script to read the environment variables and send the message. This action works well on most UNIX servers, but difficulties arise reading the environment variables on a Windows server.

You can use a Perl script to overcome environment problems on Windows. CA Service Desk Manager includes a ready-to-use installation of the Perl interpreter named `pdm_perl`. Any Perl script invoked with `pdm_perl` as a notification method can reliably obtain the environment variables. The Perl script can read and format the environment variable values and carry on with the rest of the notification, such as invoking a pager or sending an email.

For Windows-based servers, consider using the `launchit` utility. One of the functions of this utility is to invoke your scripts or programs in a shell environment similar to the Command Prompt with the proper environment variables set.

For example, if you write a Perl script named `read_env.pl` to read several of the environment variables described here, you can invoke it for a notification by entering the following in the Notification Method field on the Notification Method Detail window:

```
pdm_perl script_path/read_env.pl
```

This notification method starts the Perl interpreter and executes the instructions in `read_env.pl` script.

How to Create a Customized Notification Method

To create a customized notification step, complete the following steps:

1. Create a script to process the message template and transmit it to the recipient. The script can be any executable, depending on the platform. Third-party or public domain interpreters can also be used. Typically, Bourne shell scripts are used on UNIX and .bat files are used on Windows. If your script requires a special template, you must create it.
2. Add the new notification method to your site using the web interface.

More information:

[Create a Script](#) (see page 237)

[Add the Notification Method](#) (see page 237)

Create a Script

Use the following steps to create a notification method script:

1. Determine how you want the notification to be delivered (for example, printed on a particular printer).
2. Determine the contents of the notification message.
3. Specify what information from the message template to include in the notification.
4. Set up a script to transmit the notification.
5. Place the script in an executable file in the path of the CA Service Desk Manager server.

Add the Notification Method

After you create a script, you must define the new notification method to CA Service Desk Manager. There are two ways to add a notification method:

- Using the web interface
- Using a UNIX shell script

More information:

[Add a Notification Method Using the Web Interface](#) (see page 238)

[Add a Notification Method Using a UNIX Shell Script](#) (see page 239)

Add a Notification Method Using the Web Interface

To add a notification method using the web interface

1. Select Notification Methods from Notifications on the Administration Interface.
The Notification Method List appears.
2. Click the Create New button.
A Create New Notification Method window appears.
3. Enter data in the following fields:

Symbol

(Required) Identify the notification method.

Write to File

Select this check box to write the context information of the notification method to a file.

Description

Describe the notification method.

Notification Method

Specify the full path of the executable script for the notification method. If the script or program can be resolved using the system path, you do not have to specify the full path. For a Windows Server, consider using the launchit.exe utility to invoke your script or program.

Note: For more information about the launchit utility, see the *Online Help*.

Note: Because the notification method runs from the CA Service Desk Manager server, you must put the notification method script in a directory that can be accessed from the path on the server or specify the full path to the script. On UNIX, depending on the shell you are running, you can check this by executing this command:

which pathname_to_script

If there appears to be a problem with the notification methods, examine the logs in the \$NX_ROOT/log directory on UNIX or \$NX_ROOT\log on Windows.

Add a Notification Method Using a UNIX Shell Script

The following steps create a notification method shell script that sends the notification message to the service desk printer, SDPR2. In this example, the notification message will consist of the message header and the message text from the message template:

1. Set up the shell script to assemble the notification text and transmit it, as follows:

```
#!/bin/sh  
  
echo "  
  
TO:    $NX_NTF_USERID  
  
SUBJECT: $NX_NTF_SUMMARY  
  
MESSAGE:  
  
$NX_NTF_MESSAGE" |lp -dSDPR2
```

2. Name the executable file `sd_print`, and place it in any directory used for common scripts at your site, such as `/usr/local/netbin`.
3. Make the shell script an executable file using `chmod`.
4. Select Notification Methods from Notifications on the Administration Interface.
5. Select New from the File menu.
6. Enter data in these fields:

Symbol

SDPR2

Description

Send backup notification to service desk printer SDPR2

Notification Method

`/usr/local/netbin/sd_print`

7. Click the Save button to save the new record. Then click Close Window to close the detail window.

Query and Message Customization

CA Service Desk Manager provides a number of features that let you narrow the focus of information so you can concentrate on requests, change orders, and issues that apply to your immediate situation. One of these functions stores queries that you can use to see relevant information on the scoreboard of the administrative or web interface. Another lets you customize the messages that notify key personnel of ticket activities.

Stored queries can provide a focus on tickets related to the logged-in user and customize the counter fields in the scoreboard area of the administrative and web interfaces. You can customize activity notification messages to include attributes from the activity log object and information on specific tickets.

Scoreboard Queries

One of the tables in the database, Cr_Stored_Queries, defines stored queries. These stored queries, which are similar to SQL queries, can be used to customize the counter fields on nodes in the scoreboard area of the administrative and web interfaces. The counter fields tell how many records match the query. For example, they can tell how many of various types of requests have been assigned to the logged-in user.

Each user can customize the counter fields that appear on his or her scoreboard (this is explained in the online help.) However, the system administrator must first define the various types of requests that can be counted in these counter fields as stored queries. For information about scoreboard queries, see the *Administration Guide*.

Note: Scoreboard counts will be incorrect if database query values are equal to NULL. For example, if your Scoreboard query specifies that assignee.organization = xyz, and an assignee field is blank (NULL) for a record, then that record will not be part of the Scoreboard count.

Stored Queries for Logged in User

Two of the fields that must be defined on the Stored Query Detail window are Where Clause and Label. Both of these fields can contain expressions that are customized to the logged-in user. Stored queries refer to objects and attributes, rather than to table names and columns. A stored query that is customized to the logged-in user consists of two parts, as follows:

The object (such as cr for a request)

This is usually specified on the left of the equal (=) sign. The syntax for this part of the stored query is:

`att_name[.att_name...].SREL_att_name`

A stored query always has a Type, which is an object name that the query is executed against and provides context for the query. In the syntax above, the first att_name must be an attribute name of the context object.

The logged-in user (the instance of the cnt object for this user)

This must be specified on the right of the equal (=) sign if the tickets are to be selected based on an attribute of the logged-in user. The syntax for this part of the stored query is:

`@att_name[.att_name...].SREL_att_name`

Note: For more information about objects and attributes, see the *CA Service Desk Manager Technical Reference Guide*.

Syntax for cr Object

Use this syntax if the reference is to the request (cr) object:

`att_name[.att_name...].SREL_att_name`

This example identifies the location of the person assigned to handle a ticket. In this example, the object name is omitted, as the type of the Stored Query implies the cr object:

`assignee.location=@cnt.location AND active=1`

assignee

The attribute in the request object that maps to the assignee field in the corresponding table. For example, the assignee attribute is defined in the cr object with SREL agt, which means it refers to the agt factory. The agt factory is part of the cnt object definition.

location

The attribute in the cnt object that maps to the c_l_id field in the Contact table. The location attribute is defined in the cnt object with SREL loc, which means it refers to the loc object.

WHERE Clause

The following example demonstrates a value you can code in a WHERE clause:

`assignee.location=@cnt.location AND active=1`

Given the Stored Queries type is a Request, this query selects all active requests where the assignee's location is the same as the location of the logged-in user.

Label

Attributes in the cnt object can be included in labels the same way they are included in WHERE clauses. Here is an example of the use of an attribute in the cnt object in a label:

```
@cnt.location.name Calls
```

This label will include the name of a location, for example, Phoenix, where Phoenix is substituted for @cnt.location.name when the label is displayed on a window. The label will be displayed as Phoenix Calls.

The IN Keyword

The IN keyword allows a stored query to reference two (or more) tables without creating a join. This can result in significant efficiencies in executing the query. It is coded as follows:

```
SREL_att_name IN ( value1 [, value2 [...]] )
```

For example, a request query could be coded as:

```
category.sym IN ('\Soft%', '\Email')
```

This results in the following SQL WHERE clause:

```
category IN (SELECT persid FROM prob_ctg WHERE sym LIKE 'Soft%' OR sym = 'Email')
```

One use of IN is to avoid Cartesian products. For example, the following query results in a Cartesian product and is very inefficient:

```
assignee.last_name LIKE 'MIS%' OR group.last_name LIKE 'MIS%'
```

By using IN, the query does not create a Cartesian product; in fact, it creates no joins at all, as illustrated by the following example:

```
assignee.last_name IN 'MIS%' OR group.last_name IN 'MIS%'
```

Note: The parentheses that normally enclose the list of values on the right side of IN can be omitted if there is only one value in the list. Similarly, you should avoid joins in data partitions by converting a data partition, illustrated as follows:

```
assignee.last_name LIKE 'Smith'
```

to:

```
assignee = U'374683AA82ACE34AB999A042F3A0BA2E'
```

where:

U

indicates that the value is a uuid.

'374683AA82ACE34AB999A042F3A0BA2E'

The 32 characters in single quotes indicates the string representation of an actual uuid.

This avoids the join with some loss in clarity. Using IN, the same partition can be written as illustrated in the next example, with the clarity of the first version and almost the same efficiency as the second version:

```
assignee.last_name IN 'Smith'
```

CA Service Desk Manager supports the IN clause applied to QREL or BREL lists. For example, if you want to find all the Requests with Assets that are parents of another specific Asset (with id 374683AA82ACE34AB999A042F3A0BA2E), the appropriate where clause is as follows:

```
affected_resource.[parent]child_hier.child IN  
(U'374683AA82ACE34AB999A042F3A0BA2E')
```

The first part of the clause, *affected_resource*, is an SREL (foreign key) of the cr (Request) object, pointing to the Network_Resource table. The *child_hier* portion is a list of hier objects pointing to the hierarchical relationships. The last part, *child*, forms the first part of the where clause for the IN sub query. The *374683AA82ACE34AB999A042F3A0BA2E* portion is the foreign key value to match on *child*. *[parent]* specifies the sub query return. Since the id value is a string representation of a UUID it must be indicated as such and written as *U'374683AA82ACE34AB999A042F3A0BA2E'*

The following is an example of the actual SQL generated, which provides all the Requests where the Asset is a parent of a specific Asset:

```
SELECT Call_Req.id FROM Call_Req WHERE Call_Req.affected_rc IN (SELECT hier_parent FROM Asset_Assignment WHERE hier_child = U'374683AA82ACE34AB999A042F3A0BA2E')
```

To query on multiple parents, you can provide a comma-separated list in the () portion of the SQL, as shown by the following example:

```
affected_resource.[parent]child_hier.child IN (U'374683AA82ACE34AB999A042F3A0BA2E', U'374683AA82ACE34AB999A042F3A0BA2E')
```

The attribute name in brackets ([]) is used to form the SELECT portion of the sub-clause. Bracket notation is not used for the group Stored Queries shipped with Unicenter Service Desk Version 6.0, as illustrated in this example:

```
(assignee = @cnt.id OR group.group_list.member IN (@cnt.id)) AND active = 1
```

Note: If bracket notation is not used, the SQL subsystem assumes that it is the attribute name of the first symbol in the dot-notation portion. It works in this case, more out of luck, that the group_list object has an attribute named 'group' in it. If it were named anything else, the where clause would fail to parse! The equivalent clause with brackets illustrated as follows:

```
(assignee = @cnt.id OR group.[group]group_list.member IN (@cnt.id)) AND active = 1
```

Note: You cannot extend the dot notation. For example, the following does not work:

```
affected_resource.[parent]child_hier.child.name IN ('chicago1')
```

Query Based on Priority

In the database, the Priority table has two columns named sym and enum. The value the users see are the sym values. But the application sees the sym based on the enum values. At present, the default sym values 1 to 5 are reversed in their enum value.

Example

Sym	Enum
1	5
2	4
3	3
4	2
5	1

Therefore, when writing the stored query, when you reference a value of 5, you are actually looking for priority 1 unless you use a .sym to specify which attribute to look at.

Important! Do not change the default enum values the product assigns. Instead, when adding new sym values, just continue from the highest enum value and so on.

Time-Based Queries

Time spans can be used to create time-based stored queries. A time span specifies a period of time, which can be relative to the current date. For example, a time span could refer to today, yesterday, last week, or last month. A time span has a name, such as TODAY or YESTERDAY. You refer to a time span in a stored query by using either of two built-in functions, as follows:

StartAtTime (timespan-name)

This refers to the beginning of the period described by the time span.

EndAtTime (timespan-name)

This refers to the end of the period described by the time span.

The syntax rules for stored queries require that the time span name be enclosed in single quotes, with each single quote preceded by a backslash. For example, to refer to the beginning of last week, you would specify:

```
StartAtTime('\PAST_WEEK')
```

The passage of time makes it necessary to periodically refresh a stored query containing a reference to a time span. For example, the interval described by “yesterday” changes at midnight. You specify the Start Time, End Time, and Trigger Time for refreshes in the Timespan Detail window.

Start Time

Start Time specifies the beginning of the time span in absolute or relative terms. The following table describes the fields within the Start Time section of the Timespan Detail window:

Year

An explicit year, such as 2000, or a relative year, such as +1 (next year) or –1 (last year)

Month

An explicit month from 1 (January) to 12 (December), or a relative month, such as +1 (next month) or –1 (last month)

Day

An explicit day from 1 to 31, or a relative day, such as +1 (tomorrow) or –1 (yesterday)

Hour

An explicit hour from 0 to 24, or a relative hour, such as +1 (next hour) or –1 (last hour)

Minute

An explicit minute from 0 to 59, or a relative minute, such as +1 or –1

End Time

End Time specifies the end of the time span in absolute or relative terms. The End Time fields of the Timespan Detail window are the same as the Start Time fields of the Timespan Detail window.

Trigger Time

The Trigger Time field specifies when the WHERE clause of a stored query containing a reference to the time span is recreated and the stored query refreshed. Trigger Time must be relative to the current time as described in the following table:

Year

Must be a relative year from –1 (last year) to +36 (36 years from now).

Month

Must be a relative month from –1 (last month) to +11 (11 months from now).

Day

Must be a relative day from –1 (yesterday) to +31 (31 days from now).

Hour

Must be a relative hour from –1 (last hour) to +23 (23 hours from now).

Minute

Must be a relative minutes from +9 (9 minutes from now) to +59 (59 minutes from now).

ITIL-Specific Queries

Problems and Incidents are requests with one of two values in the *type* attribute: “I” for Incidents or “P” for Problems.

The following stored query obtains all Incidents in which the Assignee's Organization or the Group's Organization equals the logged-in Analysts Organization:

```
assignee.organization IN @cnt.organization OR group.organization IN @cnt.organization) AND active = 1 AND  
type = 'I'
```

For Problems, the query is identical except for type = 'P'

Activity Notification Messages Customization

Notification messages can be sent automatically when request activities occur.

Note: For information about notification messages and instructions for defining activity notifications, see the *Administration Guide*.

Two of the fields that must be defined on the Activity Notifications Detail window are Notification Message Title and Notification Message Body. Both of these fields can contain attributes from the activity log object (alg for Requests/Incidents/Problems, chgalg for Change Orders and issalg for Issues. These three activity log objects are almost identical) and can identify the specific request related to the activity.

Formatting Attributes for Activity Notifications

Optional formatting and escaping of individual attributes can be achieved using the properties listed below. This can be useful especially if formatting HTML notification where the data in the attribute may need to be escaped to conform to HTML standards.

To include formatting, use the following syntax:

```
@{property=value property=value:attribute_name}
```

Property values pairs are separated by at least one space and are not case sensitive. A colon separates the formatting properties from the attribute name. If no properties are listed, no formatting or escaping will be done on attribute.

The following table the available formatting properties:

Property	Description
DATE_FMT	Specifies the date format for attribute. Valid values are: MM/DD/YYYY MM-DD-YYYY DD/MM/YYYY DD-MM-YYYY YYYY/MM/DD YYYY-MM-DD Valid only for Date attributes. Dates embedded in strings are not affected.
ESC_STYLE=NONE HTML URL	Specifies the escape type of the formatted text. Valid values are: NONE Default setting. Specifies that no special treatment be given to any character in the content body. HTML Give special treatment to the following characters, which are meaningful in HTML text: <ul style="list-style-type: none">■ & becomes &amp;■ " becomes &quot;■ < becomes &lt;■ > becomes &gt; URL Translate all characters other than letters, digits, and '@*-_.#' to '%xx', where xx is the hexadecimal coding of the translated character.

Property	Description
JUSTIFY=LEFT CENTER RIGHT TRUNCATE WRAP LINE	<p>Specifies the justification of the formatted text. Valid values include:</p> <p>TRUNCATE (default if formatting) Truncates text to WIDTH property value if a positive integer. If ESC_STYLE=HTML, eliminates HTML formatting by replacing '<' and '>' with &lt; and &gt; (see KEEPLINKS and KEEPTAGS).</p> <p>LEFT CENTER RIGHT Produces exactly WIDTH characters, truncated or padded with spaces as necessary, with any embedded new lines replaced by a single space. If ESC_STYLE=HTML, the output text is delimited by [set the pre variable for your book] and </pre> tags. The WIDTH argument must be specified as a positive integer.</p> <p>WRAP Same as LEFT, except that text wrapping honors word boundaries (line breaks are not placed within words).</p> <p>LINE Same as TRUNCATE, except that it also replaces all embedded line breaks with
 tags if ESC_STYLE=HTML.</p>
KEEPLINKS=YES NO	If KEEPLINKS=YES is specified, the action of JUSTIFY=LINE or JUSTIFY=TRUNCATE is modified to preserve HTML anchor tags (Action:) while converting all other '<' and '>' characters. Mutually exclusive with KEEPTAGS. Only valid if ESC_STYLE=HTML.
KEEPNL=YES NO	The normal action of PDM_FMT is to convert all embedded new lines and any following spaces to a single space. If KEEPNL=YES is specified, embedded new lines are preserved. This argument is ignored for JUSTIFY=LINE.
KEEPTAGS=YES NO	If KEEPTAGS=YES is specified, the action of JUSTIFY=LINE or JUSTIFY=TRUNCATE is modified to preserve all HTML tags. Mutually exclusive with KEEPLINKS. Only valid if ESC_STYLE=HTML.
PAD=YES NO	If PAD=NO is specified, PDM_FMT does not convert empty strings to a single space. This is the normal action when WIDTH is non-zero, or JUSTIFY is TRUNCATE or WRAP.
WIDTH=nn	When non-zero, specifies that the text should be formatted to exactly WIDTH characters.

For example, to format the Request description for an HTML notification by escaping HTML specific characters, adding
 tags for line breaks and keeping any HTML Links as links, enter the following:

```
@{ESC_STYLE=HTML JUSTIFY=LINE KEEPLINKS=YES:call_req_id.description}
```

To format the open_date of a Request to European format, enter the following:

```
@{DATE_FMT=DD-MM-YYYY:call_req_id.open_date}
```

Attributes from the Activity Log Object

To include an attribute from the activity log object, include this in the Notification Message Title or Notification Message Body field:

`@{att_name}`

The name of the object, alg or chgalg or issalg, is the default and need not be specified. For example, to include the type of activity in the message title, enter this in the Notification Message Title field (along with the rest of what you want in the title):

`@{type}`

To include the description of the activity in the message body, enter this in the Notification Message Body field (along with the rest of what you want in the body):

`@{description}`

Information on Specific Change Orders

For messages to provide information on the specific change order that triggered the notification, the Notification Message Title or Notification Message Body fields must contain an attribute in the activity log object that references the change order object. Enter the reference in this format:

`@{change_id.chg_att_name}`

In this reference, the following information applies:

@

Indicates to replace this expression.

change_id

The attribute in the activity log object that links it to a specific instantiation of the change order object (chg).

chg_att_name

Any attribute in the chg object.

For example, to include the priority of the change order in the message title, enter the following in the Notification Message Title field, along with the rest of what you want in the title:

`@{change_id.priority.sym}`

To identify who reported the change order (Affected End User) in the message body, enter the following in the Notification Message Body field, along with the rest of what you want in the body:

`@{change_id.requestor.combo_name}`

If you want to reopen a specific change order by number, and want the message to appear as follows, use the following syntax:

Reopen Change Order @{change_id.chg_ref_num}

Note: For messages to provide information about an issue that triggered a notification, the Notification Message Title or Notification Message Body fields must contain an attribute in the activity log object that references the issue object, iss. Using the information for requests and change orders presented in this section, along with the information about objects and attributes in the *CA Service Desk Manager Technical Reference Guide*, you can see how to accomplish this.

For example, to include the priority of the issue in the message title, enter the following in the Notification Message Title field, along with the additional information you want in the title:

@{issue_id.priority.sym}

Information on Specific Requests

For messages to provide information on the specific request that triggered the notification, the Notification Message Title or Notification Message Body fields must contain an attribute in the activity log object that references the request object. Enter this reference in this format:

`@{call_req_id.cr_att_name}`

@

Indicates to replace this expression.

call_req_id

The attribute in the activity log object that links it to a specific instantiation of the request object (cr).

cr_att_name

Any attribute in the cr object.

For example, to include the impact of the request in the message title, enter this in the Notification Message Title field (along with the rest of what you want in the title):

`@{call_req_id.impact.sym}`

To identify the affected resource in the message body, enter this in the Notification Message Body field (along with the rest of what you want in the body):

`@{call_req_id.affected_resource.name}`

If you want to reopen a specific request by number, and want the message to appear as follows, use the following syntax:

Reopen Request `@{call_req_id.ref_num}`

There are several other mechanisms by which messages can be sent which are in the context of the request itself (or change order or issue). When the context is the request itself, you do not need (and cannot use) the "call_req_id" part of the reference. So, in these cases, you need to use:

`"@{ref_num}"` rather than `"@{call_req_id.ref_num}"`

Schema Customization

You can use the Web Screen Painter Schema Designer to modify the flexible database schema of CA Service Desk Manager to meet your needs. The Schema Designer provides an easy-to-use graphical user interface to review and modify the CA Service Desk Manager schema. The Web Screen Painter also allows you to test your schema changes on your own web forms before updating the physical DBMS schema or affecting other users.

Here are the kinds of schema changes you can make and use in your own forms and reports:

- Add new tables to the database
- Add new columns to existing tables
- Make a column required
- Change a table or column display name or function group

Consider the following before using Web Screen Painter:

- You cannot use the Web Screen Painter to change the length of an existing column, and we strongly recommend that you *do not* use other tools to do so. Changes to the length of an existing column are not supported, and may cause other applications accessing the CA Service Desk Manager database to fail.

Important! Do not shorten a field or delete an existing field, because these actions could cause CA Service Desk Manager to fail.

- Be careful when adding columns to an existing table, because you can inadvertently exceed the record length capacity of the underlying database. Check the specifications for the database that you are using with CA Service Desk Manager and make modifications within the limits of that database.
- Publishing changes to the database schema could require limited or considerable downtime, depending on the changes you make and the capabilities of your underlying database.
- If you are a new user of CA Service Desk Manager, it is easier to make all of your changes during testing instead of waiting until you are in production.
- Review general procedures you must complete before and after changing the database schema.
- Use specific procedures to customize your schema. Most of these procedures are followed by an example of a change you might want to make to the standard database schema.

Important! The Web Screen Painter verifies that the first letter of a new table or column name is “z”, and inserts a “z” if necessary. This verification helps ensure that your user-defined field names do not conflict with field names used by CA Service Desk Manager, now or in future releases.

How to Modify the Schema Designer

To modify the CA Service Desk Manager schema, complete the following steps:

1. Make your changes with the Web Screen Painter Schema Designer. Changes can include modifying existing tables and columns and defining new ones.
2. Put your changes into test mode. Changes in test mode are defined to the Object Engine associated with the Web Screen Painter, but are not defined to the physical database. The Web Screen Painter users can access the modified schema, but typical CA Service Desk Manager users are not exposed to them.
3. Update or create web forms using the modified schema. You can examine data in your web forms, and even create or update records in site-defined tables without affecting the CA Service Desk Manager database. All updates affect only the Object Engine associated with the Web Screen Painter.
4. Repeat Steps 1 through 3 until you are satisfied with your schema changes and with the web forms that use them.
5. Publish your schema changes. Publishing requires shutting down CA Service Desk Manager.

Display the Web Screen Painter Schema Designer Tool

You can make changes to CA Service Desk Manager using the Schema Designer.

To display this tool

1. Start the Web Screen Painter.
 - (Windows) From the Windows Start menu, select Program Files, CA, CA Service Desk Manager, Web Screen Painter.
 - (UNIX) Enter the command `pdm_wsp` with `$NX_ROOT/bin` in your path.

A Web Screen Painter login window appears.

2. Enter your User Name and Password.
3. Select Schema Designer from the Tools menu.

The Schema Designer window appears.

The left side of the Schema Designer window shows the CA Service Desk Manager database in a tree format. The initial display lists tables, each preceded by a plus sign and a yellow folder icon. To see the columns in a table, either double-click the table name, or click the plus sign. The plus sign will change to a minus sign, and the Web Screen Painter will display the columns in the table in a tree format.

The Web Screen Painter shows both tables and columns in sequence by their Object Name. In addition, if the Display Name differs from the Object Name of the table or column, the Web Screen Painter shows the Display Name in parentheses after the Object Name.

The right side of the window displays the properties of the table or the column that is selected.

Schema Designer Tabs

Table Info Tab

When you click a table from the Tables tree, the Web Screen Painter populates the Table Info tab with the information about the table. You can see the following information in the Table Info tab:

Name

The object name of the table. For example, the object name of the `cr` table is `"cr"`. This is a read-only field.

Display Name

The user-friendly name of the table. For example, the Display Name of the `cr` table is `"Request"`. You can change the Display Name of a table by entering a new name in this field.

Schema Name

The name used to refer to the table in CA Service Desk Manager utilities, such as pdm_userload. This field is read-only for standard tables. For site-defined tables, Schema Name defaults to the Object Name. You can change the Schema Name by entering a new value in this field.

DBMS Name

The name used to refer to the table in the physical DBMS. This field is read-only for all tables. For site-defined tables, it is always the same as Schema Name.

Description

A brief description of the table.

Default Display Field (common name)

The column displayed on the UI for a field that references this table. For example, the assignee field of a request is a reference to the Contact table. Since the common name of the Contact table is combo_name (last, first middle), the combo name of the referenced contact is shown on the UI as the value of assignee. You cannot change the value of common name.

Foreign Key Field (rel attr)

The column stored in the database for a field that references this table. For example, the assignee field of a request is a reference to the Contact table. Since the rel attr of the Contact table is id, the assignee column in a Request contains the id of the referenced contact. You cannot change the value of rel attr.

Function Group

The name of the group that controls the level of access that users have to records in this table. Each contact's access type specifies whether or not they have read, modify, or no access to data in tables in each function group. You can change the value of rel attr by selecting a new value from the drop-down list.

Column Info Tab

In the Tables tree, when you click a column within a table, the Web Screen Painter populates the Column Info tab with information about the selected column. You can see the following information in the Column Info tab:

Name

(Display-only) The object name of the column. For example, the object name of the Contact alt_phone column is "alt_phone".

Display Name

The user-friendly name of the column. You can change the Display Name of a column by entering a new Display Name in this field. For example, the display name of the Contact alt_phone column is "alternative phone".

Schema Name

The name used to refer to the column in CA Service Desk Manager utilities, such as pdm_userload. This field is read-only for standard tables. For site-defined tables, Schema Name defaults to the Object Name. You can change the Schema Name by entering a new value in this field.

DBMS Name

The name used to refer to the table in the physical DBMS. This field is read-only for all tables. For site-defined tables, the DBMS Name is always the same as Schema Name.

Description

A brief description of the column.

Field Type

The data type of the column. This field is read-only for all standard columns in standard tables, and for site-defined columns that have been saved. You can specify or change the field type of new site-defined columns by selecting a value from the drop-down. Field types available are:

INTEGER

Indicates a numeric value.

STRING

Indicates a text string. The number of characters in a string is shown or entered in the String Length field.

DATE

Indicates a date and time. The value stored in the database is an integer containing the number of seconds since midnight on January 1, 1970.

DURATION

Indicates a period of time. The value stored in the database is an integer containing a number of seconds.

DOUBLE

Indicates a real (floating point) number.

SREL

Indicates a foreign key reference to another table. The table referenced is specified in the SRel Table field. The value stored in the database is the rel attr of the referenced table, which can be either an integer or a string. The value displayed in the product is the common name of the referenced table row. For information on setting SREL attributes with foreign key values, see the CA Service Desk Manager Technical Reference Guide.

BREL

Indicates a virtual column representing the set of all objects with an SREL to this table. It exists only in the Object Engine and is not physically stored in the database. This field type should be selected only on direction from a CA employee.

QREL

Indicates a virtual column representing a set of objects selected by the where clause on the Advanced tab. It exists only in the Object Engine and is not physically stored in the database. This field type should be selected only on direction from a CA employee.

DERIVED

Indicates a virtual column constructed by the Object Engine from the values of other columns, under the direction of a formula specified on the Advanced tab. It exists only in the Object Engine and is not physically stored in the database. This field type should be selected only on direction from a CA employee.

String Length

The length of a string column. This field is blank for non-string columns. It is read-only for all standard columns, and for site-defined columns that have been saved. You can specify or change the length of new site-defined STRING columns by entering an integer between 1 and 32767 in this field.

SRel Table

The table referenced by an SREL column. This field is blank for non-SREL columns. It is read-only for all standard columns, and for site-defined columns that have been saved. You can specify the table referenced by a new site-defined SREL by selecting it from the drop-down list.

On New Default

The default value assigned to this column when a new row of the table is defined. It should be a value appropriate to the field type. Some keyword values are available for particular field types:

NOW

Specifies the current date and time for a DATE column.

USER

Specifies the active user for an SREL to the Contact table.

On Save Set

The value assigned to this column when a row of the table is updated. It should be a value appropriate to the field type. Some keyword values are available for particular field types:

NOW

Specifies the current date and time for a DATE column.

USER

Specifies the active user for an SREL to the Contact table.

Required

When checked, this option indicates that a value must be supplied for the column before a row of the table containing it can be saved. You can set this option for both standard and site-defined columns, and you can disable an option that you have set. However, you cannot turn off the option of a standard column unless it was set by your site.

Updatable only for new record

When checked, this option indicates that a value for this column can be provided only when a row of its table is initially created, and cannot thereafter be changed. You can set this option for both standard and site-defined columns, and you can disable an option that you have set. However, you cannot turn off the option of a standard column unless it was set by your site.

Key for pdm_userload

When checked, this option indicates that this column is one of the columns tested by pdm_userload to determine whether or not its input is an update to an existing row. This option is available only for STRING columns. It is read only for all columns in standard tables.

DBMS Index Options

These options specify characteristics of a column that is an index of the physical DBMS. They are available only for columns in site-defined tables.

Unique

Specifies that the column is unique within the table and that no two rows have the same value for the column.

Ascending

Specifies that the DBMS index is listed in ascending sequence by this column. Mutually exclusive with Descending.

Descending

Specifies that the DBMS index is listed in descending sequence by this column. Mutually exclusive with Ascending

Advanced Tabs

The Schema Designer includes an Advanced tab for both tables and columns. Information on this tab is intended for CA support and field representatives. You will not need to work with this tab for most uses of the Schema Designer, and it will not be discussed further in this document.

Schema Designer Tasks

Table or Column Modification

To modify information about a table or column, select the table or column on the Schema Designer by clicking it, and enter the new information in the appropriate fields. The information you can modify depends on the status of the table or column:

- **Standard Tables**—You can modify the Display Name, Description, and Function Group fields.
- **Standard Columns**—You can modify Display Name, Description fields, the On New Default Value, and the On Save Set value. In addition, if the check boxes for Required or Updatable only for new record are not selected, you can select them. You cannot remove these options if they are set by default. However, you can reverse your own changes.
- **Site-Defined Table**—If the table is not published, you can modify all fields, except Name, which cannot be changed after the new table has been saved. After a site-defined table has been published, you can modify only the Display Name, Description, and Function Group fields.
- **Site-Defined Column**—If the column is published, you can modify all fields, except Name, which cannot be changed after the new column has been saved. After a site-defined column has been published, you can modify only the Display Name and Description fields, the On New Default Value, the On Save Set value, and the check box for Required, Updateable only for new record, Key for pdm_userload, and the DBMS index options.

Add a New Table

To add a new table to the database

1. Select Add Table from the Edit menu, or click the Add Table button

The Add New Table dialog appears.

2. Enter the table name in the New Table Name field and click OK. The name of a site-defined table must begin with the letter “z” to prevent conflict with possible future standard tables.

The Web Screen Painter verifies this, and adds a “z” to the beginning of the table name if necessary.

3. Complete the fields in the Table Info tab as appropriate.

Add a New Column

To add a new column to a table

1. Select the table for which you want to add a column (or select any of its existing columns) and select Add Column from the Edit menu or click the Add Column button.

The Add New Column dialog appears.

2. Enter the column name in the New Column Name field and click OK. The names of a site-defined column added to a standard table must begin with the letter “z” to prevent conflict with possible future standard columns.

The Web Screen Painter verifies this, and adds a “z” to the beginning of the column name if necessary.

3. Complete the fields in the Column Info tab as appropriate.

Save Changes

To save your changes to the database while you are still modifying tables or columns, select Save from the File menu or click Save. The Web Screen Painter stores your new or updated schema modifications in the database in either the wsptbl table (for table modifications) or the wspcol table (for column modifications).

Test Schema Modifications

To put schema changes in Test mode, select Save and set to Test Mode from the File menu on the Schema Designer. This selection saves your changes in the database, and creates a file on the server defining your changes to the Object Engine. This file is called wstest.mods, and is stored in the site/mods/majic subdirectory of your CA Service Desk Manager installation directory.

After creating the `wsptest.mods` file, the Web Screen Painter causes its Object Engine to recycle so that it will use the new changes. This may take from a few seconds to a couple of minutes, depending on the complexity of your schema. The Web Screen Painter displays a dialog while waiting for the Object Engine to recycle, and updates it after the recycle has completed and it is synchronizing its internal storage with the updated Object Engine. When this is complete, the Web Screen Painter displays a message indicating that the Schema has been placed into test mode. When you click OK on this message box, you can begin to use the new schema, including creating and modifying web forms that use it.

The `wsptest.mods` file affects only the Object Engine designated by the `wsp_domsrvr` option. Other Object Engines on the same server do not process this file, and the file is not distributed to other servers. In addition, new tables and columns in Test mode are defined to the Object Engine as local objects. This means that the Object Engine knows about them and you can use them on web forms. However, they do not exist in the database, and do not affect other users. Typical CA Service Desk Manager users do not use the Web Screen Painter Object Engine, so they are unaffected by the schema modifications you are testing.

Revert Schema Modifications

If you change your mind about your schema modifications after putting them in test mode, you can revert back to the published, version of the schema. Because reverting schema modifications has the potential of impacting other users, this option is available only if your installation has installed both the `wsp_domsrvr` and `wsp_webengine` options to dedicate an Object Engine and a Web Engine to the Web Screen Painter.

To revert schema changes from Test mode, select **Revert Test Mode** from the File menu. The Web Screen Painter deletes the `wsptest.mods`, causing the Web Screen Painter Object Engine to revert its schema back to the published version.

After deleting the `wsptest.mods` file, the Web Screen Painter causes its Object Engine to recycle so that it can rebuild its internal schema. This may take from a few seconds to a couple of minutes, depending on the complexity of your schema.

After the Object Engine has completed recycling, the active schema is back to its published version. Web forms modified to work with the new schema are not themselves automatically reverted, and may not work correctly when used with the published schema.

Publish Schema Modifications

After you are satisfied with your schema modifications, you can make them available to all users by publishing them. Publishing modified schema is a two step process:

1. Create or update files describing the modified schema to the Object Engine and to CA Service Desk Manager utility programs. The Web Screen Painter creates the following files on the web engine designated by the `wsp_webengine` option (which defaults to `web:local`):

wsp.mods

Describes all Web Screen Painter-maintained schema changes to the Object Engine.

wsp_schema.sch

Describes all Web Screen Painter-maintained tables and columns.

wsp_index.sch

Describes DBMS indexes for Web Screen Painter-maintained tables.

wsp.altercol

Names new columns created by the Web Screen Painter but not yet defined to the DBMS.

wsp.altertbl

Names new tables created by the Web Screen Painter but not yet defined to the DBMS. In addition, the Web Screen Painter distributes the `wsp.mods` file to all CA Service Desk Manager servers with an Object Engine.

2. Modify the physical DBMS to contain information about the new schema. This step requires bringing down CA Service Desk Manager services and running the `pdm_publish` script on the primary server.

Important! Step 2 has a significant impact on other users, so you should carefully plan publishing schema changes. We recommend you use CA Service Desk Manager Change Orders to schedule and obtain approval for your planned schema publication.

To begin schema publication, select **Save**, and **Publish** from the **File** menu. This creates the necessary files on CA Service Desk Manager servers, but does not recycle any of them. Thus, the new files have no immediate impact. However, after the files are created, they will be used the next time CA Service Desk Manager services are recycled. Therefore, you should shut down services and run the `pdm_publish` script on the primary server at your earliest convenience after publishing schema modifications.

After you have completed schema publication with the Web Screen Painter, you cannot make any further changes with the Schema Designer until you have run the `pdm_publish` script. To run `pdm_publish`, shut down CA Service Desk Manager services and enter the `pdm_publish` command at a command prompt.

The `pdm_publish` command does the following:

- Verifies that there are Web Screen Painter-produced schema modifications to publish by checking for the existence of the required files in the site mods directory.
- Verifies that you have shut down CA Service Desk Manager services.
- Merges all schema files - Web Screen Painter maintained and non-Web Screen Painter maintained - into a single master schema file called `ddict.sch`.
- Sends the appropriate SQL commands to the DBMS to define the new tables and columns.
- Writes a line to a log file, `wsp_schema.log`, after each successful DBMS definition of a table or column. In addition to documenting your schema modifications, the log file also serves as a directory to the `pdm_publish` command itself to allow it to determine which Web Screen Painter-created tables and columns have already been defined to the DBMS. Therefore, you must not move or modify this file.
- Builds the CA Service Desk Manager data dictionary.

These steps normally take only about a minute. After they are complete, you can restart CA Service Desk Manager services and begin using your modified schema. If you have created or modified web forms to use the new schema, you should start the Web Screen Painter and publish your new web forms.

Test to Production Migration

One of the design goals for the Web Screen Painter was to make it safe to develop and test schema modifications on a production database. Such features as test mode and dedicated Web Screen Painter server processes support this goal. However, many users prefer to develop their schema modifications in an independent test system and then migrate them to a separate production system after they are complete as follows:

1. Copy the contents of the `wsptbl` and `wspcol` tables from the test database to the production database. We recommend you use the CA Service Desk Manager `pdm_extract` and `pdm_userload` utilities to do this.
2. Use the Web Screen Painter on the production system to publish the schema. Then, run the `pdm_publish` script.

Note: Using the Web Screen Painter for publishing ensures that all required updates are distributed to all production servers.

More information:

[Publish Schema Modifications](#) (see page 263)

Change or Delete Site-Defined Columns after Publishing

After site-defined schema modifications are published, the Web Screen Painter treats them similarly to standard schema and restricts further changes. Sometimes, it is desirable to delete a site-defined column, or change the length of a site-defined string column. You can accomplish these tasks by manually updating the DBMS and schema outside of the Web Screen Painter, and then running the `pdm_wspupd` script to update the database `wspcol` table to synchronize the Web Screen Painter with the external changes. The following procedure can be used to do this:

1. Find the `site/mods` (UNIX) or `site\mods` (Windows) subdirectory in your CA Service Desk Manager installation directory.
2. Using any standard text editor, edit file `wsp_schema.sch` to delete unwanted site-defined columns or change the length of site-defined `STRING` columns. These are the only changes supported by this procedure.

Important! If any of the index options (such as, `UNIQUE`) were specified for a column to be deleted, use any standard text editor to edit file `wsp_index.sch` to remove references to the column. If the column was the only indexed column in the table, remove all references to the table from `wsp_index.sch`.

3. Using any standard text editor, edit file `majic/wsp.mods` (UNIX) or `majic\wsp.mods` (Windows) with the same changes made to `wsp_schema.sch` – that is, delete unwanted site-defined columns, or change the length of site-defined `STRING` columns.
4. Bring up a command window and issue the command:
`pdm_wspupd`

The `pdm_wspupd` script reads `wsp_schema.sch` and compares it with the `wspcol` table in the database, writing a line to the console for any differences. The output is similar to:

```
PDM_WSPUPD - Update wspcol table from wsp_schema.sch
Reading wsp_schema.sch to for current DBMS information...
Reading wspcol table for WSP schema information...
String column zSalesOrg.description length changed from 350 to 400
Column zSalesOrg.sym not found in wsp_schema.sch - deleting wspcol row
pdm_wspupd found 1 WSP-maintained column(s) to update and 1 to delete. Please verify that your DBMS
has been manually updated to correspond to wsp_schema.sch, then reply Y to update wspcol or anything
else to cancel.
```

Verify that the changes found by `pdm_wspupd` correspond exactly to the changes you made to `wsp_schema.sch`. If they do, type “Y” to confirm the changes. After you confirm the update, the script uses standard CA Service Desk Manager utilities to update the `wspcol` table. This causes the Web Screen Painter Schema Designer to show your changes.

5. Stop CA Service Desk Manager services.
6. Using the appropriate utility for your DBMS, alter the DBMS definition of the columns you changed. You should delete from the database any columns you deleted from `wsp_schema.sch`, and change the database length of any string columns you changed in `wsp_schema.sch`. Take care to ensure that the changes you make to the DBMS correspond exactly to the changes you made to `wsp_schema.sch`.
7. Run `pdm_publish` as described in Publish Schema Modifications above.
8. Start CA Service Desk Manager services.

Web Interface Customization

The CA Service Desk Manager web interface (also referred to as the browser interface) provides you with CA Service Desk Manager functionality through the Internet, including the ability to open, update, or close tickets, display and post announcements, and access supporting data tables. It enables independent browsing of the knowledge base, thus reducing the number of calls to the service desk and speeding resolution times. The web interface can be fully customized and can be used with many web browsers.

If you have installed and configured the web interface, you can integrate it into your existing web interface and otherwise customize it to suit your needs. For customization, be familiar with HTML and the web browser in use at your site.

Note: The Web Screen Painter Design view works for CA Service Desk Manager controls (PDM_MACROS). When working on forms that do not contain CA Service Desk Manager controls, you can only work on the Source tab. The Employee and Customer web forms do not contain CA Service Desk Manager controls and therefore appear on the Source tab rather than the Design tab. Some Analyst forms do not contain CA Service Desk Manager controls, and therefore would appear on the Source tab too.

Important! Technical support cannot provide assistance with design or debugging of customizations (including documentation, such as online help systems). We provide general information for customizing the CA Service Desk Manager web interface. When doing so, be aware that you are solely responsible for your own customizations. CA Service Desk Manager technical support can assist you in interpreting and understanding customization.

Support for the customization techniques here extends to helping ensure that the techniques and facilities perform as documented. Be careful not to exploit undocumented features or to extend documented features beyond their documented capabilities. Such exploitation is not supported and can result in system problems or instability that may appear unrelated to the customization. For this reason, support may ask you to remove customizations to reproduce the problems. Sites should prepare for this eventuality by carefully following the guidelines on placing all modifications in the site mods directory tree and maintaining change logs. Sites that make frequent, complex, or extensive changes should consider approaching CA Service Desk Manager customization as a software engineering project with disciplined source control, testing, and controlled releases to production.

Migrating customizations between releases can present unique challenges, and we have developed the product in ways to preserve the efforts put into customization. However, we always assume that the product has been customized only as documented in this guide, particularly with regard to placement of all customizations in the site mods tree. In addition, if Level Two support supplies a patch to a system, the patch is written with these same assumptions. Patching or upgrading a system with undisciplined customizations is a risky undertaking that often results in costly system down time. Avoid it by following this guide and practicing sound software engineering principles.

Note: For information about how to secure and configure the web interface, see the *Administration Guide*.

The Web Screen Painter (WSP)

The primary customization tool for CA Service Desk Manager is the Web Screen Painter. You can install this tool on any CA Service Desk Manager server. It provides a simple and easy-to-use user interface that allows you to customize web forms and schema to the requirements of your site without programming. You can use the Web Screen Painter for many tasks including the following:

- Changing field labels.
- Moving fields in a form or changing the appearance of a list.
- Adding fields to a form, or columns to a list.
- Adding a notebook to a form, or changing the tabs in an existing notebook.
- Creating forms and forms groups.
- Customizing Cascading Style Sheet (CSS) files.

- Previewing your changes in a browser window with your own data before publishing them to other users.
- Adding new tables or columns to the database, or changing the characteristics of existing columns.
- Previewing forms that use customized schema before changing the database.

You can accomplish all these tasks with simple drag-and-drop or point over desired control from the Control Palette and double-click, without any programming, and without even looking at form source code. However, if you want to review and modify source code, the Web Screen Painter also provides a source code editor with keyword highlighting, and seamlessly integrates your source level changes with your design view changes.

However, some of the Knowledge Management forms cannot be customized in the design view of the Web Screen Painter. For these forms, there are alternate approaches to providing customization such as.

- Document View—The document template that is used when creating the document determines the contents of this page. These templates can be modified on the Administration tab under Documents, Document Templates.
- Knowledge Categories document list—You can modify this page by using Web Screen Painter, but it is also manage by user preferences. The “Preferences” screen provides personalization per user for defining which document properties displays in the document list and how many documents display per page.

Note: Modifying the schema by adding new tables and columns requires administrator authorization.

More information:

[Schema Customization](#) (see page 253)

Start the Web Screen Painter (Windows)

You can start the Web Screen Painter at any time to customize, without programming, web forms and the schema based on your requirements. To start the Web Screen Painter, select Start, Programs, CA, Service Desk, Web Screen Painter. The Web Screen Painter displays a standard CA Service Desk Manager login form in a browser. After you log in, the Web Screen Painter displays the main form.

Start the Web Screen Painter (UNIX)

You can start the Web Screen Painter at any time to customize, without programming, web forms and the schema based on your requirements. To start the Web Screen Painter, enter the command *pdm_wsp* with *\$NX_ROOT/bin* in your path. After you log in, the Web Screen Painter displays the main form.

Note: When you use UNIX, you must have Firefox installed to use the Web Screen Painter.

Open a Form for Editing

You can open a form in the Web Screen Painter to modify the content and appearance of the information on the form.

To open a form for editing

1. Select File, Open.

The Open Form dialog appears.

2. Select the Interface (Analyst, Customer, Employee, Default) or File Type (CSS Stylesheet, JavaScript, or HTML) and the forms group that contains the form you want to edit.
3. Select either the form you want from the scrolling list, or enter its name in the textbox.

When you enter a name in the textbox, the Web Screen Painter automatically scrolls the list to the first name matching the characters entered.

You can use the Files of Status drop-down list to restrict the list of files displayed:

Site Modified with Unpublished Changes (+)

Restricts the list to files that have been modified with the Web Screen Painter, but not yet published. These files are identified with a plus sign (+) after the file name.

Site Modified (*)

Restricts the list to forms modified at your site, both published and unpublished. Unpublished files are identified with a plus sign (+) after the file name; published site modifications are identified with an asterisk (*) after the file name.

All

Displays the list with no restrictions. Unpublished files are identified with a plus sign (+) after the file name; published site modifications are identified with an asterisk (*) after the file name.

Important! When you create or edit a detail or list form, make sure to use the "list_" and "detail_" prefixes" to name the HTML file. For example, use "list_test.html" and "detail_test.html." Adding this prefix lets you correctly preview a form. When you save a detail template with a custom name, you must also manually edit the <PDM_WSP> tag. For example, <PDM_WSP mode=edit preview="test.html+OP=CREATE_NEW" factory=cr>.

Create a Form

You can use the Web Screen Painter to create a form.

To create a form

1. Select File, New.

The New Form dialog appears.

2. Select an Interface and Forms Group for the new form, and whether the form is to be a detail form, list form, or menu bar.
3. Select a factory (or table) for the new form.

Note: Only one detail or list form can exist for each table in a forms group, so edit an existing form (rather than create one) for tables that already have an existing form. If you want to have multiple versions of a form, create one or more form groups to hold the additional versions.

Important! When you create or edit a detail or list form, use the "list_" and "detail_" prefixes" to name the HTML file. For example, use "list_test.html" and "detail_test.html." Adding this prefix lets you correctly preview a form. When you save a detail template with a custom name, also manually edit the <PDM_WSP> tag. For example, <PDM_WSP mode=edit preview="test.html+OP=CREATE_NEW" factory=cr>.

More information:

[How to Create a Web Form Group](#) (see page 286)

Form Edit Window

After you open an existing form or are asked to create one, the Form Edit window appears. There are two tabs in the edit window, the Design tab and the Source tab. The Design tab is available for detail forms, list forms, and menu bar forms, and shows the controls on the form laid out more or less as a user would see them. It is not an image of how the form looks to an end user. To see this, select Tools, Preview.

The Source tab is a Notepad-style editor allowing you to review and edit the source code for a form. Some forms are editable only in the Source tab. For those forms, the edit window opens up on the Source tab, and the Design tab is disabled.

The title bar of the edit window shows the name of the form, its interface, and (if appropriate) its form group. You can open edit windows for more than one form at the same time.

Important! When you create or edit a detail or list form, make sure to use the "list_" and "detail_" prefixes" to name the HTML file. For example, use "list_test.html" and "detail_test.html." Adding this prefix lets you correctly preview a form. When you save a detail template with a custom name, you must also manually edit the <PDM_WSP> tag. For example, <PDM_WSP mode=edit preview="test.html+OP=CREATE_NEW" factory=cr>.

Edit List and Detail Forms in Design View

The Design view tab displays the controls on a form, arranged in the same tabular form they would be displayed to a client. You can rearrange controls dragging and dropping them. To delete a control, click the control and select Edit, Delete.

Design view is used for editing controls. It does not show the form as it would be viewed by an end user. To view a form as it would be viewed by the end user, select Tools, Preview. The main differences between Design view and the end user view are the following:

- Fonts and styling are not used in Design view.
- Each control shows its associated attribute name in Design view.
- The Web Screen Painter shows only CA Service Desk Manager controls (those defined by <PDM_MACRO> statements). It does not show content defined by standard HTML tags or JavaScript.
- The Web Screen Painter shows all controls on the form, regardless of conditionals (PDM_IF statements). This allows you to edit anything that might be on the form. The Web Screen Painter shows conditional controls themselves as red text, such as If or Else.

Properties Dialog

To change the properties of a control (including its label), display the Properties dialog by clicking the control on the form and selecting Controls, Properties. All Properties dialogs contain fields for the Attribute (column) name; the Caption (label), and the Column Span (number of columns in the grid). The remaining fields in a Properties dialog vary with the type of control.

To change a value in a Properties dialog, enter the new value in the appropriate place. Changes take effect as soon as you click outside the field, as well as when you close the Properties dialog.

The Web Screen Painter displays a brief summary of the significance of a property in a note that appears at the bottom of the Properties form when you select the property.








Insert a Control

You can use the following ways to insert a control on a form:







- Drag-and-drop the desired control from the Control Palette on the left side of the main Web Screen Painter window to the desired spot on the form.
- Click a position on the form where you want to place the new control and select the desired control from the Control menu.
- Copy an existing control and paste it to the form.

After the new control is properly placed, display and edit its properties.


The controls that can be inserted on both list and detail forms are the following:

Control	Icon	Description
Insert Row		Causes the selected control to be the last in its current row (moves following controls to the next row).
Delete Row	N/A	Deletes all controls on the same row as the currently selected control.
Textbox		Inserts a single or multi-line textbox for editing a string or text field.
Dropdown		Inserts a drop-down list for editing a field validated against a table.
Lookup		Inserts a lookup control for editing a field validated against a table. The control consists of a textbox with a hyperlink in the label that pops up a select form.
Button		Inserts a button.
Hierarchical Lookup		Similar to a Lookup control, except that it is used for a field with a hierarchical selector (such as request category).
Date		Inserts a date field. The control consists of a textbox with a hyperlink in the label that pops up a date selector.

The following additional controls are available for detail forms only:

Control	Icon	Description
Checkbox		Inserts a check box.
HTML Editor		Inserts an HTML editor for a text field that contains HTML.
Read Only Textbox		Inserts a non-editable text field.
Read Only Lookup		Inserts a non-editable lookup field. The field is displayed as a hyperlink to pop up the detail form defining it.
Read Only		Inserts a non-editable date field.
Notebook		Inserts a notebook. There can only be one notebook on a detail form, so this control can be inserted only on forms that do not already contain a notebook.

The following additional control is available for list forms only:

Control	Icon	Description
List		Inserts a list. There can only be one list on a list form, so this control can only be inserted on new list forms.

Notebook Designer

Many detail forms contain a notebook with two or more tabs. You can use the Notebook control to add a notebook to a detail form that does not already contain one. The Properties dialog for a Notebook control is replaced by the Notebook Designer. To open the Notebook Designer, double-click the Notebook control.

The Notebook Designer allows you to add, insert, and delete notebook tabs and to change their captions. You can also use the up and down arrow buttons to rearrange tabs by changing the position of the currently selected tab. The New Row check box specifies whether or not the selected tab begins a new row in the notebook header.

CA Service Desk Manager supports two types of notebook tabs:

- A *deferred tab* is loaded only when selected by the user in the notebook header. Its Notebook Designer entry specifies a URL. This URL can be either a standard Web URL beginning `http://` or a CA Service Desk Manager URL beginning `OP=`. To specify a deferred tab, select the Deferred check box.
- A *standard tab* is loaded at the same time the form is loaded. Its Notebook Designer entry specifies the file name of an HTML file that defines the contents of the tab, which must be bracketed by `<pdm_form>` tags in the file. To specify a standard tab, clear the Deferred check box.

The contents of a standard tab are defined in a separate HTML file. To edit this file using the Web Screen Painter, double-click the hyperlink file name that the Web Screen Painter displays on the body of the tab in Design view. The Web Screen Painter opens another form edit window for the HTML file defining the tab.

List Designer

CA Service Desk Manager list forms typically consist of a search filter at the top of the form, and a list at the bottom of the form. The search filter section of a list form resembles a detail form, and you edit it in the same way. The only difference is that fewer controls are available in this window. The list forms support only the textbox, drop-down list, date, lookup, command button and hierarchical lookup controls.

The Web Screen Painter displays the list section of a list as an empty rectangular box with the list headings at the top. The Properties dialog for a List control is replaced by the List Designer. To display the List Designer for a list, double-click the list control.

The List Designer allows you to add, insert, and delete attributes and to change their captions (column headers). You can also use the up and down arrow buttons to rearrange attributes by changing the position of the currently selected attribute. The New Row check box specifies whether the selected attribute begins a new row of attributes in the data for a single database row.

Note: For more information about the other fields on this form, see the *Web Screen Painter Help*.





Conditional Controls

CA Service Desk Manager supports conditionally including content on a form. For example, the Request Detail form (`detail_cr.html`) uses a lookup control for its change attribute only for new requests. For existing requests, it uses a read-only lookup that displays in Design view.

The Web Screen Painter shows both Textbox controls for the change attribute side by side, even though the user will only see one at a time. It shows the conditional control as a word in red text, such as, for example, If or Else. You can view the Properties dialog for a conditional control the same way as you display the Properties dialog for a typical control.

The conditional controls displayed by the Web Screen Painter correspond to PDM_IF and its associated tags. PDM_IF: Conditional Processing discusses these tags, including the syntax of a conditional.

The Web Screen Painter shows four types of conditional controls:

Control	Icon	Description
If		Begins a conditional.
Elif		(Optional) Specifies an alternative condition (else if). There can be any number of elif controls.
Else		(Optional) Specifies an alternative. If provided, must be the last control before endif.
Endif		(Required) Ends the conditional.

Preview Forms

To see how a form would look to an end user, select Tools, Preview. The Web Screen Painter copies the modified form to the server, where it is stored in a directory accessible only for your Web Screen Painter preview requests, and then submits a URL to display the form in a browser window.

A Web Screen Painter preview window allows you to view a form as it would be seen by an end user. Although it resembles a standard CA Service Desk Manager window, and most buttons and menus are functional, it is not a standard session, and you should not attempt to use it that way. The following are the principal limitations of a preview session:

- It is normally read-only. That is, although you can try out functions such as editing data, all database update requests are ignored; you cannot change the database in a Web Screen Painter preview session. The browser window indicates this in two ways:
 - The red Web Screen Painter icon on the top left corner indicates that the browser window shows a read-only preview session. It is possible for the CA Service Desk Manager administrator to allow updates in the preview window. However, we recommend that you do not do this. If your administrator has configured the Web Screen Painter this way, the icon is yellow (caution).
 - Any occurrence of the word “Save” on a button or menu is changed to “noSave” to indicate that no database update will occur.

- Not all functions are available. The Web Screen Painter preview always shows you the form or tab you are working on. However, many forms are intended to be reached by a specific path through the application, and their environment may not be set up correctly when they are shown directly. If you click a button or attempt to use a feature that has not been properly set up, the Web Screen Painter displays a message explaining that the function is not available in preview mode.
- The Web Screen Painter always shows a detail form in edit view, and populates it with data from your database (it uses the most recently added row from the appropriate table that you are authorized to view). To see the read-only view of the form, click the noSave button.
- The Web Screen Painter always shows a list form listing a single row from the database, with its search filter closed. You can view and change the search filter and repeat the search as necessary to preview the form.

The default behavior for the Web Screen Painter preview is to display a detail form in edit view, or a list form in list view. You can modify this behavior for a particular HTML form with the PDM_WSP tag as described in the PDM_WSP: Control WSP Preview.

Edit in Source View

Sometimes it is necessary or useful to look at source code for a form. This can be helpful for editing forms other than list or detail forms, or for editing HTML or JavaScript form elements that are not displayed in Design View. To switch to Source view for a form, click the Source tab to display the source code for the form.

If a control is currently selected, the Web Screen Painter automatically moves the cursor to the beginning of the source code that defines that control.

The Source view editor is a basic text editor, similar to the standard Windows Notepad editor, except that Source view is color-coded. HTML and other keywords are highlighted and colored. You can control the font and the color coding used in Source view by choosing Options from the Tools menu. The Options dialog shows the font used in Source view and the default color for eight HTML and JavaScript elements. To change a color, click the ellipsis button next to it and select the desired color from the palette.

Edit Menu Bars

Forms with names beginning *menubar_* define a menu bar. The Design view for a menu bar displays the menu at the top. You can click a menu item to lower the menu, but cannot otherwise edit the menu bar directly in Design view. To edit a menu bar, double-click the menu item to display the Menu Designer.

Note: Menus (and menubar forms) are used only in the analyst interface. The customer and employee interfaces use a "launch bar" containing actual links, not drop-down lists. To customize the customer or employee launch bar, edit form `std_body_site.html` from the appropriate interface.

The Menu Designer allows you to add, insert, and delete menus and their items and to change their captions. You can use the up and down arrow buttons to rearrange menus and items by changing the position of the currently selected item.

Both the Add and Insert buttons insert a new menu item: Add places the new menu item at the end of the menu, while Insert places it before the currently selected item.

To insert a new item on the menu bar

1. Add or insert a menu item.
2. Click the left arrow button to convert it to a menu bar item.
3. Click the right arrow button to if you want to reverse this action.

Note: For information about the fields on the form, see the *Web Screen Painter Help*.

Functions Useful in Menu Items

CA Service Desk Manager provides a menu bar on almost every form to control its functions. The menu bar is generated by an HTML form with a name of the form `menubar_xx.html`. We recommend that you use the Web Screen Painter to customize existing menu bars and define new ones.

The following predefined functions may be useful for scripts invoked by menu items:

upd_frame(form)

Loads a new form into the main window content frame.

create_new(factory, use_template, width, height [,args])

Pops up a form to define a new record.

`Popup_window(name, form[, width, height [,features [,args]]])`

Pops up a new window.

showDetailWithPersid(persid)

Pops up a detail record.

The following terms and definitions apply to the previous functions:

form

This is either an HTML file name of the form `xxx.html` or an operation code (for example `CREATE_NEW`).

factory

This is the name of a database object.

use_template

This is either true or false.

width

This represents the desired form width or zero for default.

height

This represents the desired form height or zero for default.

features

This is a list of window features, in the same format used with the standard window.open function.

args

This is one or more args of the form "keyword=value" for the operation specified for form.

persid

This is a persistent ID in the form factory:ID.

Edit Stylesheets

You can use the Web Screen Painter to edit or create CSS (cascading stylesheet) files.

To edit a stylesheet

1. Select File, Open.
The File Open dialog appears.
2. Select CSS Stylesheet from the Interface or File Type drop-down.
A list of stylesheets is displayed.

To create a stylesheet

1. Select File, New.
The New Form dialog appears.
2. Select CSS Stylesheet from the Interface or File Type drop-down list and click New.
In either case, the Web Screen Painter displays the Source view of the stylesheet. You can edit directly in Source view, or display the Style Designer by selecting Tools, Style Designer.

The top section of the Style Designer allows you to control the classes within the stylesheet. The Style Classes drop-down in the upper left of the Style Designer allows you to select a class to edit. The Add, Rename, and Delete buttons allow you to create a class, or rename or delete an existing one.

There are three tabs on the Style designer. The Font and Color tab allows you to select attributes for text formatted by the style class, and preview how it will look. The Font Preview section at the bottom of this tab shows you how the style will look.

The Position tab allows you to control positioning, and the Other tab allows you to control visibility, display, overflow, and cursor attributes. There are a number of style attributes, such as margin and border that can neither be seen nor edited in the Style Designer. These must be edited in Source view.

When you click OK in the Style Designer, the Web Screen Painter reformats the stylesheet and updates the Source view. You can continue to edit in Source view or bring up the Style Designer again.

Note: For performance reasons, CA Service Desk Manager stylesheets are delivered in two forms: Individual files (such as `search_filter.css`) and combination files grouping a number of individual files with comments and excess white space removed (such as `analyst_styles.css`). The Web Screen Painter always edits the individual files; you cannot edit a combination file directly. When you publish stylesheet changes, the Web Screen Painter automatically builds the associated combination file if necessary.

Edit HTML and JavaScript

You can use the Web Screen Painter Source view to edit HTML and JavaScript forms. To edit either, select File, Open. The Web Screen Painter displays the File Open dialog. Select HTML or JavaScript on the Interface or File Type drop-down list to display the list of files available to edit.

Note: For performance reasons, some CA Service Desk Manager JavaScript files are delivered in two forms: Individual files (such as `window_manager.js`) and combination files grouping a number of individual files with comments and excess white space removed (such as `std_head.js`). The Web Screen Painter always edits the individual files; you cannot edit a combination file directly. When you publish script changes, the Web Screen Painter automatically builds the associated combination file if necessary.

Saving Changes

At any time, you can save changes you have been editing. To save changes to a particular file, select its edit window, and then select File, Save. To save changes to all files you are editing, select File, Save All.

Note: The Web Screen Painter always saves changes on the server, not on your local PC (unless your local PC is the server). When you save a file, it becomes accessible to other Web Screen Painter users in a preview session, but is invisible to typical CA Service Desk Manager users. This is because the Web Screen Painter saves all files in the site/mods/wsp directory (UNIX) or the site\mods\wsp directory (Windows), and this directory is not used by a typical CA Service Desk Manager session.

Delete Changes Before Publishing

If you are not satisfied with changes you have made, you can delete them before publication. Deleting changes deletes a new form or leaves an existing form in its current state.

To delete changes, select File, Delete Form.

Requests to delete a form take effect when you publish changes.

To cancel a pending delete request, select Undelete Form from the File menu.

The delete request is canceled. You cannot cancel changes after publication; the only way to change a published form is to edit it again.

Delete Forms After Publishing

Only site-modified forms can be deleted. Requests to delete a previously published form take effect when you publish changes.

To cancel a pending delete request, select File, Undelete Form.

You undo changes to a form after publication; the only way to change a published form is to edit it again.

Publish Changes

When you are satisfied with changes, you can make them available to all CA Service Desk Manager users by publishing them. Publishing updates all CA Service Desk Manager servers with new or revised forms.

To publish changes

1. Select File, Publish.

If you have any unsaved changes, the Web Screen Painter prompts you to save them, and then displays a confirmation dialog showing all pending Web Screen Painter changes (including those saved in previous sessions, or saved by other Web Screen Painter users). By default, all changes are selected for publication. You can change the selection of changes to be published by clicking them.

2. Click OK when you are satisfied with the selection.

The Web Screen Painter makes the selected changes available to all users.

Test to Production Migration

One of the design goals for the Web Screen Painter was to make it safe to develop and test forms modifications on a production database. Such features as a Web Screen Painter-only directory tree on the server, dedicated Web Screen Painter server processes, and read-only preview sessions support this goal. However, many users prefer to develop their forms modifications in an independent test system and then migrate the forms to a separate production system after they are complete as follows:

1. Copy any HTML forms to be migrated from the appropriate subdirectory of site/mods/www/html on the test system to the same subdirectory of site/mods/wsp/project on the primary server of the production system.
2. Copy any CSS, JavaScript, and HTML files to be migrated from the appropriate subdirectory of site/mods/www/wwwroot on the test system to the same subdirectory of site/mods/www/wwwroot/wsp/project on the primary server of the production system.
3. Use the Web Screen Painter on the production system to publish the forms. Using the Web Screen Painter for publishing ensures that the new or updated forms are distributed to all production servers.

You can use any file copying method supported by your operating system to perform the copying described in steps 1 and 2 above. Windows users should substitute backslash (\) for slash (/) in the directory paths shown.

HTML Templates (HTMPL Form)

Forms in the CA Service Desk Manager web interface are delivered as HTML templates, in files with a suffix of .html. These are called HTMPL forms in the remainder of this document.

An HTMPL form contains standard HTML (including JavaScript) plus language extensions that are interpreted by a CA Service Desk Manager server daemon (or service) called the web engine that delivers standard HTML to the browser. These extensions are:

- References to server variables. These are indicated by a name beginning with a dollar sign. They can be the values of columns in the CA Service Desk Manager database, references to web engine configuration properties, or other server information.
- Special tags directing the web engine to perform tasks on the server, such as read information from the CA Service Desk Manager database. These tags have names of the form <PDM_...> or <pdm_...>.

Note: You do not need to understand the HTMPL extensions or even HTML itself to be able to customize CA Service Desk Manager forms with the Web Screen Painter.

More information:

[Server Variables](#) (see page 308)

[HTMPL Tags](#) (see page 287)

Template Naming Conventions

The following naming conventions are used to identify the four basic types of HTMPL files, where xxx is the object:

Template Type	Name
List (search filter and results)	list_XXX.html
Combined read-only and edit detail form (analyst interface)	detail_XXX.html
Read-only detail form	detail_XXX_ro.html
Edit detail form	detail_XXX_edit.html

You can find the definitions of the objects and their properties in the following locations:

- (UNIX) `$NX_ROOT/bopcfg/majic/*.maj`
- (Windows) `installation-directory\bopcfg\majic*.maj`

For information about the objects and attributes that define CA Service Desk Manager, see the *CA Service Desk Manager Technical Reference Guide*.

HTMPL Directories

The *Administration Guide* describes the web interfaces supplied with CA Service Desk Manager. There are different sets of HTMPL files supplied to implement these interfaces, as shown in the following table:

Operating System	Directory Containing HTMPL Files
Windows	<code>Installation-directory\bopcfg\www\html\web\interface</code>
UNIX	<code>\$NX_ROOT/bopcfg/www/html/web/interface</code>

In this table, *interface* is the name of the interface (analyst, customer, or employee).

Note: There is no separate directory for guest interface files; by default, this interface uses the employee interface files. You can change the guest user interface by changing the access type associated with user `System_Anonymous`. Both the customer and employee files dynamically modify themselves depending on whether the current user is a known user or a guest, using the `<PDM_IF>` template command described in this document.

There are three additional interface subdirectories under the `html` directory:

default:

Contains HTMPL files common to all interfaces. When searching for a file, the web engine looks first in the directory corresponding to the current user's interface, and then in the default directory.

pda/analyst: (UNIX)

pda\analyst: (Windows)

Contains HTMPL files used by the mobile device interface. In Unicenter Service Desk r11.0, the mobile device interface is provided only for analysts.

web/interface/legacy: (UNIX)**web\interface\legacy: (Windows)**

Contains HTML files from your previous release of CA Service Desk Manager that are no longer used. This directory is automatically created if you upgrade from a previous release when you install CA Service Desk Manager. You can delete the legacy directory when none of its files are referenced by your customized files.

We strongly recommend that you do not directly modify the supplied HTML files. Instead, either use the Web Screen Painter, or manually copy the file you want to modify to the site mods directory, and modify it there. The CA Service Desk Manager web server looks for a new form in the appropriate site mods directory before checking the distribution directory. The standard site mods directories for each of the interfaces are as follows:

Operating System	Directory For Site-Modified HTML Files
Windows	<i>installation-directory\site\mods\www\html\interface\interface</i>
UNIX	<i>\$NX_ROOT/site/mods/www/html/interface/interface</i>

Note: If you change the form and save it into the *install directory\site\mods\www\html\interface* directory, the form will be seen by everyone, regardless of the form group to which they belong. If you save it into the *install directory\site\mods\www\html\interface\interface* directory, only those Contacts that are defined as belonging to that form group will see the changed forms.

In the previous table, *interface* is the name of the interface (analyst, customer, or employee). There is no separate directory for guest interface files; this interface uses the employee interface files. The advantage of storing your modified HTML files in the site mods directory is that this directory is preserved when you install CA Service Desk Manager maintenance or a new release. In addition, keeping your modified files in site mods while preserving the originals ensures that you always have a correct copy of the originally distributed HTML file.

Each web interface page has a primary function, as indicated in the following table that lists the major HTML templates. However, you can add <PDM_FORM> blocks to any template to directly access any web interface supported operation. For example, you can modify the main menu to include fields for submitting an issue without using the intermediate page, or you can add search criteria fields and a search button to a list form:

Web Page	HTML Template
Main form	menu_frames.html
Display/create/update a change order	detail_chg.html

Web Page	HTML Template
Display a list of change orders	list_chg.html
Display/create/update and issue	detail_iss.html
Display a list of issues	list_iss.html
Display/create/update a request	detail_cr.html
Display a list of requests	list_cr.html
Display announcement detail information	detail_cnote.html
Display a list of announcements	list_cnote.html
Login	login.html

Note: For a complete list of templates, view the contents of the directories in the table at the beginning of this section.

Web Form Groups

You can collect customized web pages into one or more form groups. Form group directories are in the following directories:

Windows

install-directory\site\mods\www\html\web\interface
 install-directory\site\mods\www\wwwroot\subdirectory

UNIX

\$NX_ROOT/site/mods/www/html/web/interface
 \$NX_ROOT/site/mods/www/wwwroot/subdirectory

Each form group is a subdirectory under these directories. You specify the customized form directory in the Customization Form Group field of the access type.

When a user requests a form, the web engine looks first in the appropriate customized form group directory, then in the standard directory for the user's web interface, and finally in the default directory. You can define more than one access type for the same web interface, each with a different customized form group. This lets you define a few specialized forms for different types of users, and still take the majority of the forms from the standard interface.

A similar process occurs when a web page requests a file from one of the subdirectories of wwwroot (css, html, img, or scripts). The webengine examines an HTML reference of the form CAisd/img/xxx.gif and converts it to one of:

- /CAisd/sitemods/img/formgroup/xxx.gif
- /CAisd/sitemods/img/xxx.gif
- /CAisd/img/xxx.gif

selecting the first one where it finds xxx.gif.

How to Create a Web Form Group

Complete the following steps to create a web form group:

1. If you want a form group besides the predefined Analyst, Customer, or Employee form groups, create a form group by selecting Save As from the File menu in the Web Screen Painter and clicking the Add Form Group button on the Save Form As dialog. For example, if you want to provide two separate customized versions of the Analyst interface, you might create form groups called Analyst1 and Analyst2 to handle these. You might also define a new form group if the interface you are defining does not logically fit into one of the predefined form groups.
2. On the web interface (not a Web Screen Painter preview session), select Security, Access Types from the Administration menu. Then click an access type (or create one) and use the Customization Form Group drop-down list on the Access Type Detail window to assign a form group to an access type. CA Service Desk Manager determines the access type when a contact logs in and uses customization form group to determine where to look in the site mods directory structure for customized forms. If the web engine does not find a form in the form group directory, it looks first in the standard directory for the user's access type, and then in the default directory.
3. In the Web Screen Painter, select Save from the File menu, or manually copy the customized HTML files to the following directory:

In Windows: *installation-directory\site\mods\www\html\web\form_group_name* directory

In UNIX: *\$NX_ROOT/site/mods/www/html/web/form_group_name*

After you set up a new web form group and copied any supporting files to the appropriate subdirectories, you must restart the web service before the changes take effect.

HTMPL Tags

PDM_EVAL: Insert the Value of a Pre-Processor Variable

The `pdm_eval` tag is used to insert the value of a pre-processor variable into the input of the webengine parser. If used inside a macro, its effect is deferred until the macro completes.

The `pdm_eval` tag works similarly to `pdm_include` or `pdm_macro`. It inserts the text into the parser at the point of the tag, exactly as if the value of its variable had been coded in place of the tag.

`pdm_eval` has the following syntax:

```
<PDM_EVAL TEXT=PRE.name>
```

name

(Required) The name of the pre-processor variable whose value is to be inserted into the webengine's input.

PDM_FORM: Start an HTML Form with a Session ID

`<PDM_FORM>` and `</PDM_FORM>` can be added to any web interface HTML template to create an HTML form including two hidden fields for the server variables SID (session ID) and FID (form ID). The optional OP operand creates an additional hidden field for one of the supported operations, as with the `PDM_LINK` tag. Except for the automatically-generated hidden fields, `<PDM_FORM>` and `</PDM_FORM>` are used in the same way as the standard HTML `<form>` and `</form>` tags (and generate these tags as part of their expansion).

PDM_FMT: Format Text from a Server Variable

The `<PDM_FMT>` and `</PDM_FMT>` tags are used to format blocks of text inserted by server variables (`$args.xxx`) as directed by its arguments.

Note: `<PDM_FMT>` is ignored for literals, including `$prop.xxx` variables.

The following table describes these tags:

Property	Description																						
ESC_STYLE= NONE C HTML JS JS2 URL	<p>Specifies the escape type of the formatted text. Valid values are:</p> <p>NONE Default setting. Specifies that no special treatment be given to any character in the content body.</p> <p>C Give special treatment to the characters ' , " , \ , \r , ` , and \n , which are meaningful in C programs. These characters will be escaped.</p> <p>HTML Give special treatment to the following characters, which are meaningful in HTML text:</p> <table><tr><td>&</td><td>becomes &amp;</td></tr><tr><td>'</td><td>becomes &apos;</td></tr><tr><td>"</td><td>becomes &quot;</td></tr><tr><td><</td><td>becomes &lt;</td></tr><tr><td>></td><td>becomes &gt;</td></tr></table> <p>JS Give special treatment to the following characters, which are meaningful in JavaScript text:</p> <table><tr><td>'</td><td>becomes %27</td></tr><tr><td>"</td><td>becomes %22</td></tr><tr><td>/</td><td>becomes %2F</td></tr><tr><td>\</td><td>becomes %5C</td></tr><tr><td>\r</td><td>becomes %0D</td></tr><tr><td>\n</td><td>becomes %0A</td></tr></table> <p>JS2 Same as JS, but give no special treatment to the character, / , and give special treatment to two additional characters:</p> <ul style="list-style-type: none">- % becomes %25- Line breaks are suffixed with %0A <p>URL Translate all characters other than letters, digits, and '@*-_.#' to '%xx' , where xx is the hexadecimal coding of the translated character.</p>	&	becomes &	'	becomes '	"	becomes "	<	becomes <	>	becomes >	'	becomes %27	"	becomes %22	/	becomes %2F	\	becomes %5C	\r	becomes %0D	\n	becomes %0A
&	becomes &																						
'	becomes '																						
"	becomes "																						
<	becomes <																						
>	becomes >																						
'	becomes %27																						
"	becomes %22																						
/	becomes %2F																						
\	becomes %5C																						
\r	becomes %0D																						
\n	becomes %0A																						

Property	Description
JUSTIFY=LEFT CENTER RIGHT TRUNCATE WRAP LIN	<p>Specifies the justification of the formatted text. Valid values include:</p> <p>TRUNCATE</p> <p>Default setting. Eliminates HTML formatting by replacing '<' and '>' with &lt; and &gt;</p> <p>Note: For more information, see the following information about KEEPLINKS and KEEPTAGS.</p> <p>LEFT CENTER RIGHT</p> <p>Produces exactly WIDTH characters, truncated or padded with spaces as necessary, with any embedded new lines replaced by a single space, and the output text delimited by [set the pre variable for your book] and </pre> tags. The WIDTH argument must be specified as a positive integer.</p> <p>WRAP</p> <p>Same as LEFT, except that text wrapping honors word boundaries (line breaks are not placed within words).</p> <p>LINE</p> <p>Same as TRUNCATE, except that it also replaces all embedded line breaks with
 tags.</p>
KEEPLINKS=YES NO	If KEEPLINKS=YES is specified, the action of JUSTIFY=LINE or JUSTIFY=TRUNCATE is modified to preserve HTML anchor tags (Action:) while converting all other '<' and '>' characters. Mutually exclusive with KEEPTAGS.
KEEPNL=YES NO	The normal action of PDM_FMT is to convert all embedded new lines and any following spaces to a single space. If KEEPNL=YES is specified, embedded new lines are preserved. This argument is ignored for JUSTIFY=LINE.
KEEPTAGS=YES NO	If KEEPTAGS=YES is specified, the action of JUSTIFY=LINE or JUSTIFY=TRUNCATE is modified to preserve all HTML tags. Mutually exclusive with KEEPLINKS.
PAD= YES NO	If PAD=NO is specified, PDM_FMT does not convert empty strings to a single space. This is the normal action when WIDTH is non-zero, or JUSTIFY is TRUNCATE or WRAP.
WIDTH= <i>nn</i>	When non-zero, specifies that the text should be formatted to exactly WIDTH characters.

<PDM_FMT> without WIDTH or JUSTIFY does no formatting on the enclosed text, but surrounds the text with [set the pre variable for your book]and </pre>.

For example, to produce a multi-line description, enter the following:

```
<PDM_FMT WIDTH=50 JUSTIFY=WRAP>$args.description</PDM_FMT>
```

To produce multi-column output, enter the following:

```
<PDM_FMT><PDM_FMT WIDTH=20 JUSTIFY=LEFT>$cst.last_name</PDM_FMT>  
<PDM_FMT WIDTH=20 JUSTIFY=LEFT>$cst.first_name</PDM_FMT>  
<PDM_FMT WIDTH=20 JUSTIFY=TRUNCATE>$cst.middle_name</PDM_FMT>  
</PDM_FMT>
```

PDM_IF: Conditional Processing

These tags are used to conditionally include text. <PDM_IF> blocks can be placed anywhere in an HTML file - in HTML, in JavaScript, and even within HTML tags. <PDM_IF> and <PDM_ELIF> (else if) both take a simple conditional clause as their properties rather than name-value pairs. If the clause is true, the text after the tag to the closing tag is included in the file; if the clause is false, the server discards the text between the tag and the closing tag. The closing tag can be <PDM_ELIF>, <PDM_ELSE>, or </PDM_IF>.

The <PDM_ELSE> and <PDM_ELIF> tags are optional. If both are specified, all <PDM_ELIF> tags must precede <PDM_ELSE>. There can be any number of <PDM_ELIF> tags between <PDM_IF> and <PDM_ELSE> (or </PDM_IF> if <PDM_ELSE> is omitted).

The syntax of the conditional in <PDM_IF> and <PDM_ELIF> is as follows:

- 0 is false; any other number is true
- "" is false; "any-string" is true
- "value op value" evaluates the left and right values against each other according to *op*. If both values consist of digits (optionally preceded by - or +), the comparisons are done numerically. Otherwise, they are done lexically (ASCII collation). Valid *op* values include:

op Value	Description
==	Equal to
!=	Not equal to
>=	Equal to or greater than (must be written as \>= or >=)
<	Less than (must be written as \< or <)
>	Greater than (must be written as \> or >)
<=	Equal to or less than (must be written as \<= or <=)
&	Performs a bit-and of the left and right values. True if any bits are set; false if none are set.

op Value	Description
%	Returns true if the left value is an even multiple of the right value, and false otherwise (useful for building two-dimensional tables).
:	Performs a byte-oriented pattern match like the UNIX grep command. It returns true if the left value contains the regular expression defined by the right value.

Example:

```

<PDM_IF $count \>= 10> ...
<PDM_ELIF $count &lt; 5> ...
<PDM_ELSE> ...
</PDM_IF>

```

There can be more than one conditional in a PDM_IF statement. Conditionals are separated by connectors, either && (and) or || (or). There is no precedence for either connector. The web engine examines a conditional from left to right until it reaches a connector. If the initial condition is true and the connector is ||, it considers the entire condition to be true without further evaluation. If the initial condition is false and the connector is &&, it considers the entire condition to be false without further evaluation. Otherwise, it considers the condition undetermined, and evaluates the conditional from after the connector.

PDM_INCLUDE: Inserting from a Different File

The <PDM_INCLUDE> tag is used to insert text from a second file into an HTML file. The server replaces the <PDM_INCLUDE> tag with the contents of the second file.

Included files can contain <PDM_INCLUDE> tags. There is no limit to the depth of nesting.

The <PDM_INCLUDE> tag supports the following properties:

Property	Description
FILE=filename	(Required) Specifies the file to include. The web engine searches the directories used for HTML files, as defined in the current user's access type.

Property	Description
FIXUP=[YES NO]	<p>(Optional) Indicates whether the file should be interpreted by the web interface like a normal HTML template file, such as expanding variables beginning with dollar signs (\$) and interpreting other CA Service Desk Manager tags, such as PDM_LIST, and PDM_FORMAT. The value YES, indicates that the file should be treated as a regular HTML template file, and the value NO means that the included file should be treated as literal text. The default is YES.</p> <p>Note: For compatibility with previous releases, the values TRUE or 1 can be substituted for YES, and the values FALSE or 0 can be substituted for NO. These values are deprecated, and should not be used in new pages.</p>
propname=value	<p>Specifies that property propname should have the specified value. The property's value can be accessed within the included file by prefixing propname with \$prop. For example, the following specification would allow the included file to reference \$prop.menubar:</p> <pre><PDM_INCLUDE ... menubar=no></pre> <p>Global properties can also be specified in the web.cfg configuration file. For information about web.cfg, see the <i>Administration Guide</i>.</p> <p>Note: For compatibility with previous releases, property values specified on <PDM_INCLUDE> can be referenced without the preceding 'prop.', in the form \$propname. This usage is deprecated and should not be used in new pages.</p>

PDM_JSCRIPT: Conditionally Include a JavaScript File

The <PDM_JSCRIPT> tag is used to conditionally include a JavaScript file on a form. This tag has two forms:

```
<PDM_JSCRIPT file=xxx.js [include=yes|no]>
```

Pdm_jscript with file=xxx.js specifies that JavaScript file xxx.js is required by this form. The webengine adds the file to a list of JavaScript files required by the form. Processing of the tag occurs while the form is being parsed, and is not affected by pdm_if. That is, a pdm_jscript tag referencing a file adds that file to the list of JavaScript files if it occurs anywhere in the file or in an included file, or in a macro.

The optional argument *include=no* can be specified to instruct the webengine to ignore the tag. This argument provides conditional processing for the tag, and is primarily useful when the tag is invoked in a macro. For example, the dtlTextbox macro specifies the following:

```
<PDM_JSCRIPT file=spellcheck.js include=&{spellchk}>
```

This indicates that any form containing a dtlTextbox macro that specifies spellchk=yes requires the JavaScript file spellcheck.js.

The second form of the pdm_jscript tag is the following:

```
<PDM_JSCRIPT insert=here>
```

Pdm_jscript with insert=here requests the webengine to insert standard HTML <script> tags for all required JavaScript files. The webengine processes this form of the tag during the HTML generation phase, so that it is affected by pdm_if. A pdm_jscript tag with insert=here is part of std_head_include.html, so it is present on virtually every form.

Note: The webengine inserts script tags only the first time it encounters pdm_jscript insert=here.

PDM_LINK: Create a Hyperlink Invoking an HTML Operation

<PDM_LINK> and </PDM_LINK> can be added to any web interface HTML template to create links a link that invokes an HTML operation. The <PDM_LINK> tag generates the standard HTML tag and has similar arguments, except that it allows specification of a CA Service Desk Manager operation in place of a URL.

The format is as follows, where *operation* is one of the supported operations:

```
<PDM_LINK OP=operation> ... </PDM_LINK>
```

Example:

```
<PDM_LINK OP=MENU> Menu </PDM_LINK>
```

```
<PDM_LINK OP=CREATE_NEW FACTORY=iss> Submit Issue </PDM_LINK>
```

```
<PDM_LINK OP=LOGOUT> Logout </PDM_LINK>
```

PDM_LIST: Format a List of Database Rows

The `<PDM_LIST>` and `</PDM_LIST>` tags are used to delimit repeating sections of HTML for multi-record output. Everything between `<PDM_LIST>` and `</PDM_LIST>` is repeated once for each record to be output. There are two types of PDM_LISTs:

- Lists taken from an object attribute that implies a list. For example, the properties attribute of the request object is the list of properties associated with that request. This type of PDM_LIST always has a SOURCE property.
- Lists with an explicit where clause. This type of PDM_LIST always has a WHERE property.

An object attribute `<PDM_LIST>` takes the following properties:

Property	Description																						
ESC_STYLE= NONE C HTML JS JS2 URL	<p>Specifies the escape type of the formatted text. Valid values are:</p> <p>NONE Default setting. Specifies that no special treatment be given to any character in the content body.</p> <p>C Give special treatment to the characters ' , " , \r , ' , and \n, which are meaningful in C programs. These characters will be escaped.</p> <p>HTML Give special treatment to the following characters, which are meaningful in HTML text:</p> <table> <tr><td>&</td><td>becomes &amp;</td></tr> <tr><td>'</td><td>becomes &apos;</td></tr> <tr><td>"</td><td>becomes &quot;</td></tr> <tr><td><</td><td>becomes &lt;</td></tr> <tr><td>></td><td>becomes &gt;</td></tr> </table> <p>JS Give special treatment to the following characters, which are meaningful in JavaScript text:</p> <table> <tr><td>'</td><td>becomes %27</td></tr> <tr><td>"</td><td>becomes %22</td></tr> <tr><td>/</td><td>becomes %2F</td></tr> <tr><td>\</td><td>becomes %5C</td></tr> <tr><td>\r</td><td>becomes %0D</td></tr> <tr><td>\n</td><td>becomes %0A</td></tr> </table> <p>JS2 Same as JS, but give no special treatment to the character /, and give special treatment to two additional characters:</p> <ul style="list-style-type: none"> - % becomes %25 - Line breaks are suffixed with %0A <p>URL Translate all characters other than letters, digits, and '@*-_.#' to '%xx', where xx is the hexadecimal coding of the translated character.</p>	&	becomes &	'	becomes '	"	becomes "	<	becomes <	>	becomes >	'	becomes %27	"	becomes %22	/	becomes %2F	\	becomes %5C	\r	becomes %0D	\n	becomes %0A
&	becomes &																						
'	becomes '																						
"	becomes "																						
<	becomes <																						
>	becomes >																						
'	becomes %27																						
"	becomes %22																						
/	becomes %2F																						
\	becomes %5C																						
\r	becomes %0D																						
\n	becomes %0A																						
LENGTH= <i>nn</i>	Specifies the number of rows of output (defaults to all).																						

Property	Description
<code>PREFIX=prefix</code>	Specifies the prefix on references to attributes from records in the list. These are referenced in the form <i>\$prefix.attr_name</i> in the text between <code><PDM_LIST></code> and <code></PDM_LIST></code> . The PREFIX property is optional in an object variable list. If PREFIX is omitted, the value of SOURCE is also used for the prefix.
<code>SEARCH_TYPE=DISPLAY GET_DOB</code>	<p>Specifies the method the server should use to build the list form:</p> <p>DISPLAY specifies the server should issue a single query for the entire form</p> <p>GET_DOB specifies the server should issue separate queries for each row of the form</p> <p>The choice affects list performance, and depends on the complexity of the list (the number of joins required to display it) and the characteristics of your DBMS. GET_DOB has more predictable performance than DISPLAY, and is the default.</p>
<code>SORT=index-name</code>	Specifies the index name to use for sorting. The default value of this argument is DEFAULT (which means the first sort index for the underlying factory).
<code>SOURCE=source</code>	Specifies the object variable defining this list. This field is required. Do not put a dollar sign (\$) in front of <i>source</i> on the PDM_LIST statement itself. If the PREFIX property is not specified, <i>source</i> is also used as the prefix for references to attributes from records on the list, in references of the form <i>\$source.attr_name</i> . When used in a reference, <i>source</i> does require a preceding dollar sign.
<code>START=nn</code>	Specifies the first output row (defaults to zero).

Example:

```

<table border>
<tr>
<th>Child Change Order Number</th>
<th>Summary</th>
</tr>
<PDM_LIST SOURCE=args.children>
<tr>
[assign the value for TD in your book]$args.children.chg_ref_num</td>
[assign the value for TD in your book]$args.children.summary</td>
</tr>
</PDM_LIST>
</table>

```

Because no prefix was specified, references to attributes of the listed records are prefixed by `$args.children`, the source value.

A where clause `PDM_LIST` takes the following properties:

Property	Description
<code>FACTORY=name</code>	Specifies a class of object to be searched. This property is required.
<code>LENGTH=nn</code>	Specifies the number of rows of output (defaults to all).
<code>ORDER_BY=attr-name</code>	Specify the attribute name to sort by. It can contain the DESC (descending) or ASC (ascending) modifiers.
<code>PREFIX=prefix</code>	Specifies the prefix on references to attributes from records in the list. These are referenced in the form <code>\$prefix.attr_name</code> in the text between <code><PDM_LIST></code> and <code></PDM_LIST></code> . The PREFIX property is required in a where clause list.
<code>START=nn</code>	Specifies the first output row (defaults to zero).
<code>WHERE=where-clause</code>	Specify the where clause for the search. It can contain (dotted) attributes. This property is required.

For example:

```
<table>
<tr>
<th>Child Change Order Number</th>
<th>Summary</th>
</tr>
<PDM_LIST PREFIX=list FACTORY=chg WHERE="status = 'OP'">
<tr>
[assign the value for TD in your book]$$list.chg_ref_num</td>
[assign the value for TD in your book]$$list.summary</td>
</tr>
</PDM_LIST>
</table>
```

PDM_MACRO: Insert Text from a Macro File

The PDM_MACRO Tag

The <PDM_MACRO> tag is used to insert a macro file into an HTML file. Its functionality is similar to PDM_INCLUDE, with two important differences:

- A file included by PDM_MACRO has a formal argument list, with required arguments and arguments with default values.
- A file included by PDM_MACRO always comes from the directory specified for the configuration property MacroPath, regardless of the current user's access type.

NAME=*macroname*

(Required) Specifies the macro to include. The web engine affixes the suffix “.mac” and searches for the file in the path specified by configuration file property MacroPath.

Other properties may be required, depending on the macro included. A macro file has the general layout:

```
comments
#args
name1 [= value1]
name2 [= value2]
...
#data
data to insert
```

The following descriptions explain the file layout, line by line:

- **comments**—The only valid statements in a macro prior to the `#args` statement are comments. Comments are indicated by either a `#` sign or a `//` as their first non-blank character or characters.
- **#args**—Must be coded exactly as shown, with the `#` sign in column one and no other information on the line. This statement begins the args section, which can contain argument definitions and comments.
- **name [= value]**—Defines an argument for the macro. Only arguments explicitly mentioned in the args section are valid for the macro. A value specified for an argument in the args section is that argument's default value. Arguments without a default value are required, and must be supplied by the caller on the `<PDM_MACRO>` statement itself.
- **#data**—Must be coded exactly as shown, with the `#` sign in column one and no other information on the line. This statement begins the data section, which is the part of the macro inserted into the file using `PDM_MACRO`. Everything in the data section is inserted into the calling file, including lines that would be comments prior to the data section.
- **data to insert**—The data to insert into the calling file. This data can contain references to arguments in the form:
- **&{arg_name}**—These references are replaced with the value of the argument supplied by the caller, or with the default value if the caller did not supply a value.

The web engine normally reads a macro file only once, the first time it is used, and then stores the parsed macro in its own memory. This improves performance, but can be inconvenient if you are developing a macro. Use the configuration file property `SuppressMacroCache` to prevent this behavior and cause the web engine to discard all macros in its memory each time it begins processing a new form.

To Comment Out PDM_MACRO Tags

To comment out `<PDM_MACRO>` tags, enter an exclamation point in front of the `P` as follows: `<!PDM_MACRO>`. To prevent the browser from processing the commented out portion of the form, place `<PDM_IF 0>` before the `<!PDM_MACRO>` tag, and `</PDM_IF>` after the line you commented out.

Example:

```
<PDM_IF 0>
```

```
<!PDM_MACRO NAME=dtlDropdown hdr="Status" attr=status lookup=no
evt="onBlur=\\\"detailSyncEditForms(this)\\\"">
<PDM_MACRO NAME=dtlDropdown hdr="Priority" attr=priority lookup=no
evt="onBlur=\\\"detailSyncEditForms(this)\\\"">
```

```
</PDM_IF>
```

Predefined Macros Used by the Web Screen Painter

There are a number of predefined macros included with CA Service Desk Manager. The majority of these insert JavaScript text to create an element on a web form. Use the Web Screen Painter to create and modify forms using these macros.

Detail Form Macros

button

Inserts a graphic button.

dtlCheckbox

Inserts a check box on a detail form.

dtlDate

Inserts a date field on a detail form.

dtlDateReadOnly

Inserts a read-only date field on a detail form.

dtlDropdown

Inserts a drop-down list on a detail form.

dtlEnd

Ends a detail form.

dtlEndTable

Ends a table within a detail form.

dtlForm

Begins a detail form.

dtlHTMLEditBox

Inserts a detail form field that is a text box containing an HTML editor.

dtlHier

Inserts a detail form field that is a text box validated against an external table with a hierarchical lookup.

dtlLookup

Inserts a detail form field that is a text box validated against an external table.

dtlLookupReadOnly

Inserts a detail form field that is a read-only hyperlink to an external table.

dtlReadonly

Inserts a read-only text field on a detail form.

dtlStart

Begins the first table in a detail form.

dtlStartExpRow

Begins an expandable row on a detail form.

dtlStartRow

Begins a normal row on a detail form.

dtlTextbox

Inserts a text box on a detail form.

contactLookup

The contactLookup macro creates a contact lookup. This macro has the following arguments:

```
contactLookup("&{header}","&{frameName}","&{factory}","&{lookupName}");
```

header

Identifies the lookup header.

frameName

(Required) Identifies the form name.

factory

Specifies the factory.

Default: agt

lookupName

(Required) Identifies the lookup name.

You can also enable and disable this element by using the following:

```
contactLookupDisable( Name, bDisable )
```

bDisable=

- true

Disables the element.

- false

Enables the element.

dtlCheckboxReadonly

The dtlCheckboxReadonly macro specifies a readonly checkbox field on an HTML detail form. The macro has the following arguments:

```
detailCheckboxReadonly("&{hdr}", "&{attr}", "&{colspan}", "$args.&{attr}", "&{on}", "&{off}");
```

hdr

Specifies the text of the header.

Default: "\$args.&{attr}.DISPLAY_NAME"

attr

(Required) Specifies the name of the attribute.

on = "X"

Specifies the value shown on readonly form when field is checked.

off = ""

Specifies the value shown on readonly form when field not checked.

colspan = 1

Specifies the number of columns on the form.

On both the readonly and edit forms, the field is displayed as specified in "on" and "off" arguments.

Note: This macro is similar to dtlCheckbox.mac, except that it is always read-only, even in Edit mode.

List Form Macros

IsCol

Specifies a column in a list form.

IsEnd

Ends the list portion of a list form.

IsStart

Begins the list portion of a list form.

IsWrite

Inserts text into the repeating section of a list form.

sfDate

Inserts a date field in a search filter.

sfDropdown

Inserts a drop-down list on a search filter.

sfEnd

Ends a search filter.

sfHier

Inserts a search filter field that is a text box validated against an external table with a hierarchical lookup.

sfLookup

Inserts a search filter field that is a text box validated against an external table.

sfStart

Begins a search filter.

sfStartRow

Begins a row within a search filter.

sfTextbox

Inserts a textbox into a search filter.

Menubar Macros

endMenu

Ends a menu within a menu bar.

menuItem

Defines a global item on a menu.

endMenubar

Ends a menu bar.

menuItemLocal

Defines an item on a menu invoked in the context of the current window.

menubarItem

Defines a menu within a menu bar.

startMenu

Begins a menu within a menu bar.

startMenubar

Starts a menu bar.

PDM_NOTEBOOK: Create a Notebook

Several of the forms in the CA Service Desk Manager analyst interface use a notebook control. A notebook allows several sets of fields to be displayed in the same physical area of the screen, with only one set visible at a time. The user selects the set of fields that is visible by clicking a named tab at the top of the notebook, or by pressing the access key combination `Alt+n`, where *n* is the number of the tab. An example of a form using a notebook is the Issue Detail (`detail_iss.html`). We recommend that you use the Web Screen Painter to modify the contents of notebooks, or insert a notebook into a form that does not already contain one.

The `</PDM_NOTEBOOK>` tag marks the end of a notebook. We recommend that you provide this tag in order to ensure compatibility with future releases. However, in this release it produces no output code and is optional.

PDM_PRAGMA: Specify Server Information

The `<PDM_PRAGMA>` tag is used to specify information used by the web engine, such as form release and version. It does not generate any HTML code, and can be placed anywhere in a form. Possible arguments are:

Argument	Description
RELEASE=value	Specifies the CA Service Desk Manager release number corresponding to this form. This value is "110" on all Unicenter Service Desk r11.0 forms. It is accessible within the form in the <code>\$prop.release</code> variable.
SITEMOD=value	Specifies a site-defined string identifying the modifications applied to this form. It is accessible within the form in the <code>\$prop.sitemod</code> variable.
VERSION=value	Specifies a CA-defined string identifying the version number of this form. It is accessible within the form in the <code>\$prop.version</code> variable.
OVERIDE=YES NO	Specifies whether or not values in this <code>PDM_PRAGMA</code> statement override values in previous <code>PDM_PRAGMA</code> statements.

CA uses `PDM_PRAGMA` statements to document form versions. All Unicenter Service Desk r11.0 forms include the following `PDM_PRAGMA` statement:

```
<PDM_PRAGMA RELEASE=110>
```

In addition, the `std_head.html` form includes the following JavaScript statement:

```
cfgFormRelease = "$prop.release" - 0;
```


The PDM_PRAGMA statement and the cfgFormRelease variable allows the CA Service Desk Manager web interface to distinguish Unicenter Service Desk r11.0 forms from previous release forms. Releases prior to Unicenter Service Desk r6.0 did not support the PDM_PRAGMA statement.

Normally, only PDM_PRAGMA statements in the highest-level file of a form (that is, a file not brought in by PDM_INCLUDE) are used to set \$prop.release, \$prop.sitemod, and \$prop.version. In addition, a PDM_PRAGMA statement will not override a non-empty value set by a previous PDM_PRAGMA statement. You can specify OVERRIDE=YES to specify that a PDM_PRAGMA statement can override previous PDM_PRAGMA statements, or that a PDM_PRAGMA statement in an included file can be used.

PDM_SCOREBOARD: Build a Scoreboard Tree

The <PDM_SCOREBOARD> tag is used to generate the scoreboard shown on the left side of the main form. It takes the following property:

TARGET=value

Specifies the name of the target frame for lists requested by clicking a node on the scoreboard. Lists are loaded into the target specified, which can be any value supported for the target attribute of a link. The default value is *_self* (the window containing the PDM_SCOREBOARD tag).

Any HTML form including a <PDM_SCOREBOARD> tag must also include the fldtree.js JavaScript file. This file can be included with the following statement in <HEAD> section of the form:

```
<SCRIPT LANGUAGE="JavaScript" SRC="$CAisd/CAisd/fldtree.js"></SCRIPT>
```

In addition, it is desirable to include a link with the name scoreboard_asof_data to display the effective date of the numbers in the tree. See the distributed file scoreboard.html for an example of the use of this tag.

The queries included on the scoreboard are defined by the contents of the User_Query table (object name usq) for the current user. A record in this table defines each line on the tree (folder or node).

Initially, users have no entries in their User_Query table. A user with no User_Query entries receives the default set of scoreboard queries associated with their access type. A user with administrative authority can also customize the default scoreboard for an access type.

PDM_SET: Set the Value of a Server Variable

The <PDM_SET> tag is used to assign a value to a server variable. It has the following syntax:

<PDM_SET arg.name[+]=value>

arg

(Required) Specifies the variable type, and must be arg for normal use.

Note: There is no \$ character.

Name

(Required) Specifies the name of the variable.

+

(Optional) Specifies that the value should be appended to the existing value of the variable. There cannot be any spaces before or after.

=

(Required) Must be specified exactly as shown, with no spaces before or after.

value

(Required) Specifies the text to be assigned or appended to the variable.

The PDM_SET tag can also be used in the preprocessor phase to create or update a preprocessor variable.

More information:

[Web Engine PreProcessing](#) (see page 333)

PDM_TAB: Create a Tab within a Notebook

The <PDM_TAB> tag is used to define a notebook tab. We recommend that you use the Web Screen Painter to modify the contents of notebooks, or insert a notebook into a form that does not already contain one.

PDM_WSP: Control WSP Preview

The <PDM_WSP> tag is used to control the Web Screen Painter preview feature. It does not generate any HTML code, and can be placed anywhere in a form.

By default, the Web Screen Painter determines how to preview a form by examining the form name:

- For detail forms (names of the form detail_*factory*.html), the Web Screen Painter displays the form in edit view, with data from the most recently created row of the appropriate table. If there is no data you are allowed to view in the table, the Web Screen Painter displays the form set up to create a row. The Web Screen Painter preview sessions are typically prohibited from updating the database. The Web Screen Painter displays forms in edit view to allow you to preview all features. However, CA Service Desk Manager ignores a Save request from a read-only preview session. The web engine changes the text on the Save button to noSave as a visual reminder of this.
- For list forms (names of the form list_*factory*.html), the Web Screen Painter displays the form in list view, with the list displaying data from the most recently created row of the appropriate table. If there is no data you are allowed to view in the table, the Web Screen Painter displays the form in search view, with the filter open.
- For other forms, the Web Screen Painter displays the form with no database context.

You can change this default behavior by placing a PDM_WSP tag anywhere on the form. For example, you can display a notebook tab form on its associated detail form, or provide prerequisite arguments for forms normally invoked with an environment provided by another form. Possible arguments are the following:

Property	Description
FACTORY= <i>value</i>	Specifies the Object Engine factory used by this form.
PREVIEW= <i>name</i> .html <i>value</i> no	Specifies the preview URL. This can be an HTML file name, in the form xxx.html; a CA Service Desk Manager URL (used unaltered if it begins with "OP="); or the keyword "no", indicating the form cannot be previewed. A value not beginning OP= is modified by replacing a reference of the form { <i>factory</i> } or { <i>factory</i> :} with an ID or persistent ID (respectively) of the most-recently created row from the referenced factory that the current user is authorized to view.
WHERE= <i>value</i>	Specifies a where clause used to search for a representative row or rows to show on the previewed form.

Property	Description
MODE= <i>value</i>	<p>Specifies the mode of the constructed URL. Can be the following:</p> <ul style="list-style-type: none">■ GENERAL. General format. Determine the mode by examining the preview argument: detail_xxxx.html - READONLY list_xxxx.html - LIST any other - GRONK■ READONLY. Detail file in read-only view.■ EDIT. Detail file in edit view.■ LIST. List file.■ GRONK. Unspecified file. In this situation, gronk the file.

Server Variables

CA Service Desk Manager information is included in the HTML template using variables beginning with a dollar sign (\$). Each page is created with some variables that are documented in the template file. These variables can be put on the page or used in conditional statements:

- Simple Variables
- Property Variables
- Environment Variables
- Business Object Variables
- List Variables

Simple Variables

Simple variables specify flags that are passed to the web page. To access a simple variable, use the variable name preceded by a dollar sign (\$). This makes the value of the variable available. For example, two such variables are \$CAisd and \$cgi. Putting \$CAisd in a template results in the substitution of the main CA Service Desk Manager web server installation directory, whereas \$cgi refers to the URL of the pdmweb.exe program. Simple variables are documented in the upper section of the HTML file that uses them.

The following shows a list of variables that can be used in all the HTML files:

\$ACCESS.group

The user access privilege object contains the privilege settings on the function group *group* for the current login user. For example, \$ACCESS.admin holds the privilege value for the admin functional group. Valid privilege values are:

- 0-NO ACCESS
- 1-VIEW
- 2-MODIFY

This variable is not available in the login form.

\$cgi

The URL of the pdmweb.exe program.

\$cst

The data object of the current login user. This variable is not available in the login form. You can reference individual attributes of this object with the form *\$cst.attrname*; for example, *\$cst.first_name*.

\$CAisd

The URL of main CA Service Desk Manager web server installation directory.

\$MachineName

The MachineName defined in the web.cfg file.

Note: For information about web.cfg, see the *Administration Guide*.

\$ProductName

The product name defined in the NX.env file.

\$SESSION

The session object saves all session variables including session ID (\$SESSION.SID) and all variables defined in the web.cfg file.

Note: For information about web.cfg, see the *Administration Guide*.

\$USER_STATE

User-defined state information.

Property Variables

Property variables represent a property of the configuration file, web.cfg. You can access any entry in the web.cfg file (including user-defined entries) within an HTML template file by prefixing it with “\$prop.”

For example, one of the lines in web.cfg, which specifies the number of entries displayed in a single page on a list form is as follows:

```
ListPageLength 10
```

You can refer to this variable in an HTML template with the specification:

```
$prop.ListPageLength
```

If you use the <PDM_INCLUDE> special tag to incorporate another file into a template file, you can specify additional properties as attributes of the <PDM_INCLUDE> tag. You can reference these properties in the included file in the same way as web.cfg properties. A property specified as a <PDM_INCLUDE> attribute that has the same name as a web.cfg property overrides the web.cfg property within the included file.

For example, the following <PDM_INCLUDE> tag creates a property called \$prop.menubar that can be referenced within the std_body.html file:

```
<PDM_INCLUDE FILE=std_body.html menubar=no>
```

Note: You can refer to configuration file property xxx in two ways: \$prop.xxx or \$SESSION.xxx. Both return the same value. However, the \$prop.xxx syntax is preferred because it involves less server overhead.

In addition to properties from web.cfg, there are several predefined properties that can be accessed with \$prop. These are:

\$prop.browser

A string identifying the browser in use. This will be “IE” for Internet Explorer.

\$prop.combo_name

A string containing the current user's name, in the form “last_name, first_name middle_name.”

\$prop.factory

A string containing the factory associated with the current form, such as “cr” for requests or “iss” for issues.

\$prop.FID

A string containing the numeric form ID of the current form.

\$prop.form_name

A string containing the name of the current HTML template, in the form xxx.html.

\$prop.form_name_1

A string containing the substring of the form name before the first underscore. For example, for the form detail_chg_edit.html, form_name_1 would be "detail."

\$prop.form_name_2

A string containing the substring of the form name after the first underscore and before the last underscore (or dot). For example, for the form detail_chg_edit.html, form_name_2 would be "chg."

\$prop.form_name_3

A string containing the substring of the form name after the last underscore and before the dot. For example, for the form detail_chg_edit.html, form_name_3 would be "edit." For the combination detail form, which has a file name of the form detail_xxx.html, \$prop.form_name_3 is set to the current view, either "ro" or "edit".

\$prop.release

A string containing the release level of the form. The PDM_PRAGMA statement contains more details on this property.

\$prop.SID

A string containing the numeric session ID of the current session.

\$prop.sitemod

A string containing the site-defined modification name of the form. The PDM_PRAGMA statement contains more details on this property.

\$prop.user_type

A string containing "analyst," "customer," "employee," or "guest."

\$prop.version

A string containing the version of the form. The PDM_PRAGMA statement contains more details on this property.

More information:

[PDM_PRAGMA: Specify Server Information](#) (see page 304)

Environment Variables

Environment variables represent an entry within the NX.env configuration file. You can reference any entry in NX.env within an HTMPL template file by prefixing it with “\$env.”

For example, one of the lines in NX.env, which specifies the host name of the CA Service Desk Manager server is as follows:

```
@NX_SERVER=hostname
```

You can refer to this variable within an HTMPL template file with the specification:

```
$env.NX_SERVER
```

Business Object Variables

Business object variables represent a CA Service Desk Manager object, such as an issue or a request. To access an object, you need to start with the variable name, followed by a period (.), followed by whatever attribute names you want to display. For example, on an issue where, by convention, the object is represented by the variable args, you can display the description, the open date, the assignee's phone number, the number of activities on the issue, and the description of the first activity, as shown by the following:

```
$args.description  
$args.open_date  
$args.assignee.phone_number  
$args.act_log.length  
$args.act_log.0.description
```

You can use braces to delimit the variable name if it is not surrounded by white space. For example, \$foo bar and \${foo}bar are both valid. You can also use the variable args to access non-attribute values (for example, \$args.KEEP.*name* as described in Supported Operations).

It is possible that a non-attribute variable may not be defined. For example, it may be possible to get to a form from two different places, only one of which provides a value for \$args.KEEP.foo. You can provide a default value for a \$args reference with the following syntax, where the string after the colon is substituted for the reference if *variable* is undefined:

```
${args.variable:default}
```


Time Zone Date Variables

Time zone date variables are a special case of business object variables. They provide a means to convert universal dates (UTC) represented as integers to string dates adjusted for the time zone of the user's browser. The variable for representing integer dates is:

`$args.attr_name_INT_DATE`

Example: `$args.open_date_INT_DATE`

Factory Data Variables

Factory data variables are a special case of business object variables. A factory data variable is replaced by information about a referenced object. There are seven such variables available:

`$args.attr_name.COMMON_NAME`

The common name (externally readable string) of the table referenced by the attribute. For example, on the Request Detail form, the value of `$args.assignee.COMMON_NAME` is the assignee's combo name ("last, first, middle").

`$args.attr_name.COMMON_NAME_ATTR`

The attribute name of the common name in the table referenced by the attribute. For example, on the Request Detail form, the value of `$args.assignee.COMMON_NAME_ATTR` is "combo_name".

`$args.FACTORY_attr_name`

The name of the factory associated with the specified attribute. For example, on the Request Detail form, the value of `$args.FACTORY_assignee` is "agt".

`$args.LENGTH_attr_name`

The maximum length of the attribute. For example, on the Request Detail form, the value of `$args.LENGTH_summary` is 240.

`$args.attr_name.REL_ATTR`

The rel attr (foreign key) of the attribute. For example, on the Request Detail form, the value of `$args.assignee.REL_ATTR` is the value of the assignee's ID field.

`$args.attr_name.REL_ATTR_ATTR`

The attribute name of the rel_attr in the table referenced by the attribute. For example, on the Request Detail form, the value of `$args.assignee.REL_ATTR_ATTR` is "id".

\$args.REQUIRED_attr_name

A string, either “0” or “1” indicating whether the referenced attribute is required.

\$args.attr_name.SELECTIONS

A list of valid selections for *attr_name*. This value is an empty string if *attr_name* is not a reference to another table, or if the size of table referenced by *attr_name* exceeds the value of the configuration file property `SellListCacheMax`. Otherwise, the SELECTIONS variable is a string containing the common name and rel attr of all the entries in the referenced table. Successive values are separated by the string “@, @”, so the variable's value has the form:

"cname1@, @rel_attr1@, @cname2@, @rel_attr2"

\$args.factory_SEL_UNDER_LIMIT

A string, either “0” or “1”, indicating whether the current number of rows in the table corresponding to *factory* is less than the value of the configuration file property `SellListCacheMax`. This variable is deprecated in favor of the SELECTIONS variable, which should be used in all new forms.

Factory data variables containing a dotted reference (COMMON_NAME, REL_ATTR, and SELECTIONS) can be used with a dotted reference of any length. For example, on a Request Detail form \$args.assignee.organization.COMMON_NAME is replaced by the external name of the assignee's organization.

List Variables

List variables are used to iterate through data. They are accessed using list tags as described in PDM_LIST: Format a List of Database Rows.

Server Operations

Supported Operations

The following operations are supported to let you integrate the CA Service Desk Manager web pages with your web pages:

CREATE_NEW

Provides a generic interface to let the user create a row in a specified table. The object name must be specified, and by default a template named *detail_xxx_edit.html* is used for object xxx. You can override the .html file by specifying the HTML property.

Required specifiers:

FACTORY=*object-name*

Optional specifiers:

ALG_PRESET=preset_expression
ALG_PRESET_REL=preset_expression
CREATE_ALG=activity_log_type
HTMPL=zdetailxxx_factory.htmpl
KEEP.attr_name=value
PRESET=preset_expression
PRESET_REL=preset expression
SET.attr_name=value
use_template=1 | 0 (0 is the default)

Note: To use the HTMPL specifier with CREATE_NEW, the referenced form must have a name conforming to the naming convention *zdetailxxx_factory.htmpl*. The name must begin with the string *zdetail*, followed by any alphanumeric characters (including a null string), followed by an underscore and the factory name.

ENDSESSION or LOGOUT

Ends the current logged-in session. ENDSESSION is the preferred operation.

GENERIC_LIST

Provides a generic interface to allow the user to display a list from any table in the database. The object name must be specified, and by default a template named *list_xxx.htmpl* is used for object *xxx*. You can override the .*htmpl* file by specifying the HTMPL property.

Required specifiers:

FACTORY=object-name
KEEP.attr_name=value

DISPLAY_FORM

Provides a generic interface to let the user display any customized form.

Required specifiers:

HTMPL=htmpl_file

Note: DISPLAY_FORM replaces JUST_GRONK_IT. Existing implementations can continue to use JUST_GRONK_IT, which functions exactly like DISPLAY_FORM. DISPLAY_FORM is the preferred operation.

MENU

Displays the main menu page, which is defined in the web.cfg file in the Menu property.

Optional specifiers:

HTMPL=menufile
menufile is the name of an alternate main menu file.

PAGE_EXTENSION

Allows the webmaster to specify additional extensions to the interface.

Required specifiers:

NAME=*html_file*

html_file is one of the file names listed in the configuration file
UserPageExtensions directive.

Optional specifiers:

REQUIRES_LOGIN=1

If present, a login page appears first if the user is not currently logged in. If omitted or set to zero, the file is shown without checking if the user is currently logged in.

RELOG

Displays the login page.

SEARCH

Provides a generic interface to allow the searching of any table in the database. This operation assumes that an appropriate search_XXX.html has been created, where XXX is the *object-name*, as defined in the .maj files in the majic directory in bopcfg.

Note: For more information, see the *CA Service Desk Manager Technical Reference Guide*. By default, the results of this search are displayed in list_XXX.html, but this can be overridden by specifying the HTML property.

Required specifiers:

FACTORY=*object-name*

QBE.op.attr_name=*value*

Optional specifiers:

ALG_PRESET=*preset_expression*

ALG_PRESET_REL=*preset_expression*

CREATE_ALG=*activity_log_type*

HTML=*list_html_file*

KEEP.attr_name=*value*

SEC_REFRESH

Refreshes the user access information from the security subsystem. A hyperlink for this operation is provided to users who have MODIFY privileges (for the admin functional group) on the menu screen. After updating user access privileges with the security program, this operation provides a means to refresh access information. (This operation refreshes the security information for all users.)

Note: Security refresh is an asynchronous process. When the security refresh is done, a message shows in the standard log file (stdlog).

SET_MENU

The behavior of this operation is the same as MENU when MENU is used with the HTMLPL variable. The only difference is that this operation also sets the default menu form to the menu form specified with the HTMLPL property.

Required specifiers:

HTMLPL=*htmlpl_file*

Note: This operation overrides the MENU set in the web.cfg until the web service is restarted.

SHOW_DETAIL

Provides a generic interface to allow the user to display a read-only detail of a row in a specified table. The persistent ID name must be specified (from which the object name is inferred). By default, a template named detail_XXX_ro.html is used for object XXX. The .html file can be overridden by specifying the HTMLPL property.

Required specifiers:

PERSID=*persistent-id*

Optional specifiers:

ALG_PRESET=*preset_expression*

ALG_PRESET_REL=*preset_expression*

CREATE_ALG=*activity_log_type*

HTMLPL=*readonly_detail_html_file*

UPDATE

Provides a generic interface to editing any table. The ID and object name must be passed in and a detail form that the user can edit is displayed to the user. By default, the user has exclusive access to the record for two minutes, and is guaranteed to get changes into the database if they are submitted in this time.

Required specifiers:

PERSID=*persistent-id* or

SET.id=*id-of-row-to-update* FACTORY=*object-name*

Optional specifiers:

NEXT_PERSID=*persistent-id* (of record to display after successful update)

KEEP.*attr_name=value*

KEY.*attr_name=value*

HTMLPL=zdetailxxx_*factory*.html

Note: To use the HTMLPL specifier with UPDATE, the referenced form must have a name conforming to the naming convention zdetailxxx_*factory*.html. The name must begin with the string “zdetail”, followed by any alphanumeric characters (including a null string), followed by an underscore and the factory name.

Note: For information about web.cfg, see the *Administration Guide*.

Operation Variables

This table lists the variables that can be set for each of the operations in the supported operations:

Variables	Description	Operations
ALG_PRESET ALG_PRESET_REL	Specifies values for one or more of the attributes of the activity log created as a result of the CREATE_ALG variable. If CREATE_ALG is not specified, ALG_PRESET and ALG_PRESET_REL are ignored.	CREATE_NEW SEARCH SHOW_DETAIL
CREATE_ALG	<p>Specifies the activity log type of an activity log to be created as a side effect of the operation. Use the ALG_PRESET or ALG_PRESET_REL variables to specify values for the attributes of the new activity log.</p> <p>The timing of creation of the activity log depends on the operation, as follows:</p> <p>CREATE_NEW The activity log is created when the new record is saved. If the new record is not saved, no activity log is created.</p> <p>SEARCH The activity log is created when a record is selected from the list form. If the record is viewed instead of selected (that is, the user explicitly selects the View command from the list form's mouse-over menu), no activity log is created.</p> <p>SHOW_DETAIL The activity log is created before the record is displayed.</p>	CREATE_NEW SEARCH SHOW_DETAIL
FACTORY	Specifies the class of object to be searched, created, or updated. You can use any name specified as an OBJECT in the *.maj files in \$NX_ROOT/bopcfg as listed in the CA Service Desk Manager Technical Reference Guide.	CREATE_NEW GENERIC_LIST SEARCH UPDATE

Variables	Description	Operations
HTML	<p>Allows the HTML author to override the default template naming convention and explicitly specify the HTML file to display, instead of the default template.</p> <p>Note: When the HTML specifier is used with CREATE_NEW or UPDATE, the name of the referenced form must conform to the naming convention <code>zdetailxxx_factory.html</code>, where <code>xxx</code> are any characters, and <i>factory</i> is the factory name.</p>	CREATE_NEW DISPLAY_FORM JUST_GRONK_IT MENU SEARCH SET_MENU SHOW_DETAIL UPDATE
KEEP.name	Specifies the value that can be saved and passed between pages.	CREATE_NEW GENERIC_LIST SEARCH UPDATE
KEY.attr_name	Similar to the SET.attr_name, except that this specifies a lookup on attr_name, which must be a reference to another table or object.	UPDATE
NEXT_PERSID	Specifies the persistent ID of the record to be displayed next.	UPDATE
PERSID	<p>Specifies the persistent ID of a record to be displayed. You can specify this in either of the following ways:</p> <p>Directly, with a persistent ID consisting of a factory name, a colon (:), and a unique integer database ID. For example, PERSID=chg:1234, specifies the change order with database ID 1234.</p> <p>Indirectly, with a persistent ID consisting of a factory name, a colon (:), an attribute name, a second colon (:), and a value. This form of PERSID specifies the record of the specified factory that has an attribute of the specified value. For example, PERSID=chg:chg_ref_num:demo:3 specifies the change order with reference number demo:3.</p>	SHOW_DETAIL UPDATE
PRESET PRESET_REL	Specifies values for one or more of the attributes of the record created as a result of the CREATE_NEW variable. If CREATE_NEW is not specified, PRESET is ignored.	CREATE_NEW

Variables	Description	Operations
QBE.op.attr_name	<p>Specifies the values to use when performing a search. These values are identified using a QBE keyword, where <i>attr_name</i> identifies any attribute name on a ticket that can be set and <i>op</i> indicates to search where the attribute:</p> <p>EQ is equal to the value</p> <p>NE is not equal to the value</p> <p>GT is greater than the value</p> <p>LT is less than the value</p> <p>GE is greater than or equal to the value</p> <p>LE is less than or equal to the value</p> <p>NU is null</p> <p>NN is not null</p> <p>IN matches the SQL LIKE expression</p> <p>KY contains the text entered</p> <p>If you do not define any QBE variables, the standard search window is displayed.</p>	SEARCH
SET.attr_name	Specifies an attribute name to use when a ticket is created, where <i>attr_name</i> identifies any attribute in a ticket that can be set. The attribute names will vary depending on the underlying object. All objects and their attributes can be found in the *.maj files in the majic directory in bopcfg as listed in the CA Service Desk Manager Technical Reference Guide.	CREATE_NEW UPDATE
SET.id	Specifies the database ID of the row to be updated.	UPDATE
SKIPLIST	When set to 1, searches that result in 1 hit do not display the search result list. Instead, the read-only detail is displayed directly.	SEARCH
use_template	When set to 1, the SEARCH operation will return a list of templates. The returned template selected will be used in the CREATE_NEW operation to populate a new record. This variable is valid for change orders, issues, and requests.	CREATE_NEW SEARCH

More information:

[Syntax of PRESET, PRESET_REL, ALG_PRESET, and ALG_PRESET_REL](#) (see page 321)

Syntax of PRESET, PRESET_REL, ALG_PRESET, and ALG_PRESET_REL

The PRESET, PRESET_REL, ALG_PRESET and ALG_PRESET_REL keywords in the URL specify initial values for attributes of the ticket and its activity log, respectively. There are two possible formats:

[ALG_]PRESET=attr:value

Indicates that the specified attribute of the ticket or activity log should be set to the specified value. For example, the following specification sets the description of the new ticket to "Hello:"

```
PRESET=description:Hello
```

[ALG_]PRESET_REL=attr:obj.relattr:testattr:value

Indicates that the specified attribute of the ticket or activity log should be set to a value copied from another database table. The value is copied from the *relattr* attribute of the *obj* whose *testattr* has the specified *value*. For example, the following specification sets the analyst attribute of the new ticket to the ID of the contact with user ID xyz123:

```
PRESET_REL=analyst:cnt.id:user:xyz123
```

When this format is used, the implied query must retrieve a unique record. If more than one contact has a user ID of xyz123 (or none), the example PRESET specification has no effect.

The PRESET, PRESET_REL, ALG_PRESET and ALG_PRESET_REL keywords can occur as many times as desired in a URL, allowing the setting of multiple attributes. Alternatively, a single keyword operand can specify multiple values separated by @@. If the '@@' separator is used, you cannot mix value formats for [ALG_]PRESET and [ALG_]PRESET_REL keywords. For example, the following example shows two different ways of specifying values for ticket description, summary and analyst:

```
PRESET=description:Hello+PRESET=summary:HelloThere+PRESET_REL=analyst:cnt.id:user:xyz123
PRESET=description:Hello@@summary:HelloThere+PRESET_REL=analyst:cnt.id:user:xyz123
```

For requests, issues, incidents, problems, and change orders, both PRESET and PRESET_REL support a keyword attribute ASSET to link an object to an asset. The ASSET attribute updates the affected_resource attribute of a request, incident, or problem, or the asset LREL of an issue or change order.

Link Examples

The following link examples do not include the path to CA Service Desk Manager. All CA Service Desk Manager URLs begin with coding of the following form:

```
http://hostname[:port]/CAisd/pdmweb.exe
```

In this example, *hostname* is the name of your server and *port* (optional) is the port number if you are using Tomcat. This coding is shown as an ellipsis (...) in the following URL examples:

- To create a request with an affected end user with the userid tooda01, use the following example URL:

```
...?OP=CREATE_NEW+FACTORY=cr+PRESET_REL=customer.cnt.id:userid:tooda01
```

- To display a list of all requests assigned to userid tooda01, use the following example URL:

```
...?OP=SEARCH+FACTORY=cr+QBE.EQ.assignee.userid=tooda01
```

- To display the detail form for request 1234, use the following example URLs:

```
...?OP=SHOW_DETAIL+FACTORY=cr+PERSID=cr:ref_num:1234 (read-only view)
```

```
...?OP=UPDATE+FACTORY=cr+PERSID=cr:ref_num:1234 (update view)
```

Note: You can bypass the logon challenge by using Web Services for authentication. For information about the `getBopsid()` method, see the CA Service Desk Manager Technical Reference Guide.

Advanced Customization

You must be aware of various aspects of customizing web pages if you elect to use tools other than Web Screen Painter to modify HTML, or if you have unusually complex customization requirements. However, we strongly recommend that you work with the Web Screen Painter to customize CA Service Desk Manager web pages before trying any other approach. The Web Screen Painter is capable of doing almost any customization you need, and it automatically handles housekeeping issues, such as placing updates in the site mods directory, and distributing published files to all servers.

The Web Engine and Its Cache

When customizing web pages, it is helpful to understand the structure of the CA Service Desk Manager web server. The web interface uses either a J2EE servlet container, such as Tomcat, or a standard HTTP server, such as Apache or Microsoft Internet Information Server (IIS). When a user requests a CA Service Desk Manager web page, the HTTP server invokes the supplied program `pdmweb.exe`.

After it starts, `pdmweb.exe` sets up a connection with a CA Service Desk Manager daemon (or Windows service) called the web engine. The web engine interprets the user's request. Most requests require the web engine to look up a template (HTML) file and translate it into standard HTML. Usually, the translation process requires the web engine to communicate with a CA Service Desk Manager server to read or update the database, and include database information in the generated HTML. After the HTML is complete, the web engine sends it to `pdmweb.exe`, which in turn sends it back to the user's browser.

To maximize performance, the web engine typically reads each HTML file only once. After parsing the file and determining how to translate it to HTML, the web engine stores the parsed file in its cache, significantly reducing the processing time the next time the file is requested. While the cache is beneficial in a production environment, it can be inconvenient in development, as it means that changes to HTML files do not take effect until either the web engine is recycled or the `pdm_webcache` utility is used. In a development environment, you can avoid this behavior by specifying the configuration file property `SuppressHtmlCache`. However, we recommend that you do not suppress the HTML cache in a production environment because it severely impacts overall performance of the web engine.

Note: For more information, see the *Administration Guide*.

The web pages served up by `pdmweb.exe` are generated by reading HTML files and using them to generate HTML. HTML template files are identified by a file suffix of `.html`. You can modify these template files, and thereby customize the CA Service Desk Manager web pages.

The `pdm_webcache` Utility

Use the `pdm_webcache` utility to remove one or more HTML forms from the web engine cache. This forces the web engine to fetch these forms from the disk the next time they are used, allowing changes to forms to take effect.

```
pdm_webcache [-f form-name] [-g form-group] [-i interface] [-p process] [-v]
```

-f form-name

Specifies the name of the form to be removed from the cache, such as `detail_cr.html`. You can use `'%'` (or `'*'`) as a wildcard character to select more than one form. For example, the specification:

```
-f detail%
```

selects all detail forms.

This argument is optional. If it is omitted, all forms in the cache are selected.

-g form-group

Specifies the name of the form group to be removed from the cache, such as `Analyst`. You can use `'%'` (or `'*'`) as a wildcard character to select more than one form group. For example, the specification:

```
-g Anal%
```

selects all form groups beginning with "Anal".

This argument is optional. If it is omitted, all form groups in the cache are selected.

-i *interface*

Specifies the name of the web interface to be removed from the cache, such as analyst, customer, or employee. You can use '%' (or '*') as a wildcard character. For example, the specification:

-i a%

selects the analyst interface.

This argument is optional. If it is omitted, all interfaces in the cache are selected.

-p *process*

Specifies the name of the web engine process whose cache is to be modified, such as web:local.

This argument is optional. If it is omitted, all web engines are selected.

-v

Specifies verbose output. When this argument is specified, pdm_webcache lists the full name of every form removed from the cache, in the form:

interface:form-group:form-name

This argument is optional. If it is omitted, pdm_webcache reports only a count of forms removed from each web engine's the cache.

How to Modify HTML Templates

Typically, you can make two types of changes to the HTML templates:

- You can make modifications that will be visible to the user but will not be altered by the web interface prior to display. For example, you could add a GIF file for your company logo to the web interface pages (a “pass through”) by adding the reference to the appropriate template file or you could add JavaScript to your page to validate input. Any changes you make to the HTML file that are not contained within a PDM tag, as defined in the following, are passed unchanged in the HTML returned to the user.
- You can modify the replaceable sections of the templates. For example, you can add new application data to the request detail page.

Several kinds of template entries let you do the following:

- Display information from CA Service Desk Manager to the user.
- Set up a query page.
- Create links to other CA Service Desk Manager pages using link tags.

Files That Should Not be Modified

Certain HTML templates and JavaScript files contain information required by many CA Service Desk Manager web forms. The information in these templates is both release dependent and critical to the successful operation of the CA Service Desk Manager web interface. Therefore, these files are always replaced when a new version of CA Service Desk Manager is released; changes made to them are not upgraded.

The templates affected by this restriction are as follows:

ahdtop.html

Contains styles, scripts, and JavaScript variables used throughout the CA Service Desk Manager web interface. This file is part of the main frameset of the web interface, and is always present during a session. All CA Service Desk Manager forms have access to the JavaScript variable ahdtop that references the window containing ahdtop.html.

menu_frames.html

Defines the HTML frameset used by the CA Service Desk Manager main form.

msg_cat.js

Contains the text of all messages used in CA Service Desk Manager JavaScript files.

reports.html

Contains data required for web reports.

std_body.html

Contains standard information used at the beginning of the BODY section of most HTML templates.

std_footer.html

Contains standard information used at the end of the BODY section of most HTML templates.

std_head.html

Contains standard information used at the beginning of the HEAD section of almost all HTML templates.

styles.html

Contains CSS styles used throughout the CA Service Desk Manager web interface.

Although you cannot modify these files directly, you can add additional information to them. Each restricted file xxx.html (except for menu_frames.html and reports.html) has a corresponding xxx_site.html file that you can customize. For example, you can add additional information to ahdtop.html by customizing ahdtop_site.html, or add new messages by customizing msg_cat_site.js.

The xxx_site.html file corresponding to each restricted file is loaded after the main file so you can override or change JavaScript in the main file. Use caution when adding information, as badly designed changes to these files can cause unexpected problems throughout the CA Service Desk Manager web interface.

More information:

[Directories Used by Your HTTP Server](#) (see page 329)

Guidelines for New HTML Files

You can add your own HTML files to the CA Service Desk Manager web interface. Follow these guidelines to help ensure your HTML file works well with the rest of the CA Service Desk Manager interface:

1. Include the following statement somewhere in the <HEAD> section of the file. This statement should follow the <TITLE> statement (if any). It defines several JavaScript global variables required by CA Service Desk Manager web interface, and also registers your page with the CA Service Desk Manager window manager:

```
<PDM_INCLUDE FILE=std_head.html>
```

2. Include the following attribute as part of the <BODY> tag of the file. This attribute helps the CA Service Desk Manager window manager keep track of your page:

```
onUnload="deregister_window()"
```

3. Include the following statement at the beginning of the <BODY> section of your file. The "menubar=no" argument is optional; if specified, it suppresses the CA Service Desk Manager menu bar:

```
<PDM_INCLUDE FILE=std_body.html [menubar=no]>
```

4. Include the following statement at the end of the <BODY> section of your file.

```
<PDM_INCLUDE FILE=std_footer.html>
```

How to Add User Defined State Information

Many customers want to be able to embed their own state information in the CA Service Desk Manager web pages, and have CA Service Desk Manager pass the state information to all subsequent pages it serves up to the user's session. This information can be interrogated with conditional statements in the HTML files.

State information for a user's session is accomplished by setting the special attribute `USER_STATE` in your links or forms. After it is submitted into the CA Service Desk Manager web engine, every page that is presented to the user will have the HTML variable `USER_STATE` available and set to the value last submitted for `USER_STATE`.

The following examples show how you might set up an entry into CA Service Desk Manager from some other part of your site, such as from pages that are oriented to your sales force:

- Using a hyperlink

```
<a href="/CAisd/pdmweb.exe?USER_STATE=Sales">Service Desk</a>
```

- Using a form with a hidden field

```
<form action="http://yourhost.com/CAisd/pdmweb.exe">  
<input type=hidden name=USER_STATE value=Sales>
```

Click the button for the Service Desk

```
<input type=submit>  
</form>
```

Then you can customize your HTML forms based on the state information:

```
<PDM_IF "$USER_STATE" == "Sales">  
    custom information for sales audience
```

```
<PDM_ELIF "$USER_STATE" == "Engineering">  
    custom information for engineers
```

```
<PDM_ELSE>  
    information for everyone else
```

```
</PDM_IF>
```

How to directly create a Request from a Template

It is possible to create a Request directly from a Template using an URL.

Example

```
http://machinename/CAisd/pdmweb.exe?FACTORY=cr+OP=CREATE  
NEW+PERSID=cr:3106+use_template=1
```

where cr:3106 is the persid of the template.

How to Configure a Quick Close Ticket with Preset Information

In the Quick Profile View, you can create a Quick Close ticket, for example, a Quick Close Incident. Add a preset string to the URL when you create a Quick Close ticket to add a description, a summary, or other field information automatically.

To add a preset string to the Quick Close ticket feature, complete the following steps:

1. Copy the `ahdtop_site.html` file from `NX_ROOT/bopcfg/www/html/default` to `NX_ROOT/site/mods/html/www/default`.
2. Edit the [ahdtop_site.html file](#) (see page 328) with the appropriate variable (depending on the type of Quick Close ticket).
3. Log in to CA Service Desk Manager.
4. Create a Quick Close ticket.

The preset information is added to the ticket.

Create a Quick Close Ticket With Preset Options

In Quick Profile you can create a Quick Close ticket, for example, a Quick Close Incident. The Quick Close option lets you open and close a new ticket in one step, for example, when you open and resolve the incident in the same session. You can add a preset string to the URL when you create a Quick Close ticket to add a description, a summary, or other field information automatically.

To create a quick close ticket with preset options

1. Copy the `ahdtop_site.html` file from `NX_ROOT/bopcfg/www/html/default` to `NX_ROOT/site/mods/html/www/default`.
2. Edit the `ahdtop_site.html` file to add the appropriate variable (depending on the type of Quick Close ticket) with the preset string.

- Quick Close Incident—`var quick_close_preset_in`
- Quick Close Problem—`var quick_close_preset_pr`
- Quick Close Request—`var quick_close_preset_cr`
- Quick Close Issue—`var quick_close_preset_iss`

For example, the following string sets the description to `HelloIncident` and summary to `HelloIncidentSummary` for a Quick Close Incident.

```
var quick_close_preset_in = "PRESET=description:HelloIncident@@summary:HelloIncidentSummary";
```

3. Log in to CA Service Desk Manager.
4. Select View, Quick Profile in the Service Desk tab.
The Quick Profile Contact Search page appears.
5. Complete one or more of the search fields for the contact, and click Search.
The Quick Profile Contact List populates with those contacts that match your search criteria.
6. Select a contact.
The right pane displays the information for that contact.
7. Click Quick Close.
The ticket is created with the preset information.

Directories Used by Your HTTP Server

The default installation of CA Service Desk Manager defines two virtual directories to your HTTP server:

- The `CAisd` virtual directory points to the following directory in your CA Service Desk Manager installation:

In Windows: `installation-directory\bopcfg\www\wwwroot`

In UNIX: `$NX_ROOT/bopcfg/www/wwwroot`

- The `CAisd/site` virtual directory points to the following directory in your CA Service Desk Manager installation:

In Windows: `installation-directory\site\mods\www\wwwroot`

In UNIX: `$NX_ROOT/site/mods/www/wwwroot`

Subdirectories under these virtual directories are:

Subdirectory	Stores
css	Style sheets
help	Web interface help
html	HTML files
img	Graphic files
scripts	JavaScript
sitemods	Site-defined customizations

If you decide to create a customized version any of the files in the css, html, img, or scripts directories, we strongly recommend that you do not update the file in /CAisd. Instead, store the file in the appropriate subdirectory of /CAisd/sitemods. For example, if you decide to modify a style sheet in /CAisd/css, store your customized version in /CAisd/sitemods/css. When the web engine parses an HTML file, it automatically modifies file names beginning with \$CAisd to point to sitemods if the file exists in a subdirectory of sitemods.

Using the /CAisd/sitemods directory has these advantages:

- It allows you to keep a record of the distributed files you have changed.
- It gives you easy access to the original version in case there is a question or a problem.
- It makes the process of installing maintenance or a new release easier, since CA Service Desk Manager installation never places anything in the /CAisd/sitemods directory.

Note: There is no /CAisd/sitemods/help subdirectory. Because the help data is in standard HTML files (not HTML templates), the web engine cannot dynamically change file references. If you need to customize help, you must make your changes in /CAisd/help.

The HTML subdirectory contains a few heavily used files that do not need to be processed by the web engine and can improve performance when cached on the browser. If you create a customized version of any of these files, carefully check the file for references to other customized files. Because there is no web engine processing, you must manually insert a reference to sitemods where appropriate.

Download PDF Attachments

When you download and try to view a PDF attachment in CA Service Desk Manager, the PDF file may not display correctly, or a blank window may appear after you upgrade to Adobe Acrobat release 7.0 or 8.0. With CA Service Desk Manager, you can display the PDF file correctly by completing the following steps:

1. Set the *forceDecompressOnDownload* parameter to YES in
\$NX_ROOT\bopcfg\www\CATALINA_BASE\webapps\CAisd\WEB-INF\web.xml.

Note: On Linux, \$NX_ROOT is /opt/CAisd

2. Restart the CA Service Desk Manager services.

Looking Up Information in Reference Tables

Input fields on a detail form editing a database record are named SET.attr_name. When the record is saved, data from SET fields are copied directly to the underlying record. Thus, an input field for an attribute that references another table should contain the REL_ATTR (foreign key) of that table. This is normally the id, persistent_id, or code of the reference record.

Users do not directly provide REL_ATTR values, and the SET fields for attributes referencing another table are hidden. The visible field on the form is named KEY.attr_name, and it contains the common name of the referenced record. A common name must be converted into a REL_ATTR to update the record. There are several times when this might be done:

- For fields with a drop-down list, the SET value is provided directly by the drop-down.
- For fields with a lookup when the user clicks the lookup and selects an item, the SET value is copied from the selected item.
- For fields with a lookup where the user provides a partial key that uniquely identifies the record and then clicks the label, the browser requests the SET value from the server and copies both it and the full key back to the form.
- If the Autofill configuration file property is provided or defaulted, and the user both provides a partial key that uniquely identifies the record and clicks Notebook to exit the field, the browser requests the SET value from the server and copies both it and the full key back to the form.

Otherwise, when the record is saved with a KEY value and no SET value, the web engine resolves the value during the save. If any KEY values cannot be resolved to a unique SET value, the save is prevented, and the edit form is redisplayed.

If a form has been redisplayed as a result of a save that failed due to a lookup resolution failure, the following variables are available in the HTML for each attribute field for which a lookup was performed:

LIST_attr

Contains all the matches found. Typically this is specified as the right-hand side of the SOURCE= field in a <PDM_SELECT> statement.

FLAGS_attr

This is set to one of the following values:

0

Display initial search field.

1

More than one and fewer than MaxSelectList were found (typically a <PDM_SELECT> list would be displayed in this case).

2

No matches were found.

3

Too many matches were found (more than MaxSelectList).

SEARCH_STATUS_attr string

Contains the TooManyMatches text string from the web.cfg file.

Note: For information about web.cfg, see the *Administration Guide*.

Specifying Lookups on Contacts

When specifying a contact (last name, first name, middle name) in an editable form, you can delimit the contact name with commas (,) or blank spaces, but not both. Commas are preferable because names often have embedded spaces, which cause problems.

Since a combination of commas and blank spaces is not allowed, the presence of commas implies that all parts of the name are comma-separated; if no commas are present, names are delimited by spaces.

Since the information is eventually passed to an SQL query, the percent symbol (%) serves as a wildcard character. For example, 'P%, J%' would match 'Public, John', 'Penxa, Jane', and any other names whose last name begins with P and first name begins with J. (Case-sensitivity depends on the underlying database.) Similarly, 'P% J%' would bring up the same names.

However, 'P%, Jon D' would not bring up all contacts with a first name of Jon, a middle initial of D, and a last name beginning with P, because the presence of one comma means all delimiters are commas. Therefore, the last name would be looked up as 'P%' and the first name would be looked up as 'Jon D'. To avoid this error, specify 'P%, Jon, D' instead.

Web Engine PreProcessing

The web engine goes through two phases when processing an HTML file:

- The preprocessing phase, when it reads the HTML file and any referenced files (including files referenced by PDM_INCLUDE and PDM_MACRO tags). The output from preprocessing is an entry in the web engine's internal cache.
- The generation phase, where it reads the form from its cache and generates HTML. The output from generation is HTML delivered to the browser.

The pre-processing phase is typically done once for each form in the lifetime of the web engine. The generation phase is done each time a form is requested.

You can use the PDM_SET and PDM_EVAL tags during the preprocessor phase to generate and store information, such as HTML text, that the web engine can use in the generation phase.

Preprocessor Variables

Preprocessor variables begin with the string "\$PRE.". They are created and updated with the PDM_SET tag. This tag has the following syntax when used with a preprocessor variable:

```
<PDM_SET PRE.name[+]=value>
```

This tag assigns or updates a preprocessor variable, creating it if necessary. It is processed when the web engine encounters it while reading a form. Only the invariant PDM_IF statements affect PDM_SET of a preprocessor variable; others are ignored.

Invariant PDM_IF Detection

When parsing a form, the web engine detects invariant PDM_IF statements. An invariant PDM_IF is one whose argument consists entirely of literals, environment variables, constant properties, and preprocessor variables. When the web engine detects an invariant PDM_IF, it evaluates its condition immediately. This has the following effects:

- PDM_SET and PDM_EVAL tags that are bypassed by an invariant PDM_IF are ignored. All other pdm_eval tags and PDM_SET tags referencing preprocessor variables are executed when processed, even if they are within a non-invariant PDM_IF.
- Form variable references bypassed by an invariant PDM_IF are ignored, and their value is not fetched when the form is used. You can use this technique to improve the performance of a form. For example, if a form contains the following, the web engine fetches the value of \$args.def before it displays the form:

```
<PDM_IF "$env.NX_OTB_MARKET == 'itil' && '$args.a' == 1">  
<h1>This is form $args.def</h1>  
</PDM_IF>
```

However, if the following segment has been written, the web engine determines that the first PDM_IF is invariant, and retrieves the value of \$args.def only if \$NX_OTB_MARKET is "itil".

```
<PDM_IF "$env.NX_OTB_MARKET == 'itil'">  
<PDM_IF "$args.a" == 1">  
<h1>This is form $args.def</h1>  
</PDM_IF>  
</PDM_IF>
```

PDM_EVAL: Insert Text from a Preprocessor Variable

The PDM_EVAL tag inserts the value of a preprocessor variable into the input to the web engine parser. If used inside a macro, its effect is deferred until the macro completes.

The PDM_EVAL tag works similarly to PDM_INCLUDE or PDM_MACRO. It inserts the text into the parser at the point of the tag, exactly as if the value of its variable had been coded in place of the tag.

PDM_EVAL has the following syntax:

```
<PDM_EVAL text=PRE.name>
```

where PRE.name specifies the name of the preprocessor variable whose value is to be inserted into the web engine's input

Execution of the PDM_EVAL tag can be controlled by invariant PDM_IF statements.

Important! On UNIX, the LIBPATH needs to be set before running the utility. Use pdm_task to perform this task. For example, before running the utility, input "pdm_task pdm_eval".

Free-Form Customization of Detail Forms

Using JavaScript on Detail Forms

You can use the Web Screen Painter to add your own fields to a detail form, or rearrange or change edit characteristics of fields provided on the form by default. However, sometimes you want to customize a form beyond simply adding new fields to a grid. There are a number of a JavaScript functions provided with CA Service Desk Manager to make it easy to merge your own customizations into a combination detail form and give it any appearance you want. These functions are summarized as follows:

- You can place any HTML whatsoever prior to the DetailForm() statement or after the endDetail() statement without affecting the operation of the detail form at all.
- You can use the detailEndTable() function to close the table that lays out detail form elements in a grid. After you have done this task, you can lay out your own HTML in any desired format. In this case, your HTML is inside the detail form, and any form fields within it are submitted to the web engine when the user clicks Save. You can use the detailNextID() function to generate ID fields for your HTML elements that allow them to participate in mouse-less navigation of the detail form. You can see several examples of this technique in the notebook tabs, such as xx_alg_tab.html.
- You can follow your own HTML with a dtlStartRow macro to restart standard detail form formatting. This starts a second grid, whose fields will not necessarily be aligned with the first. This technique is used in every notebook tab.

- If you want to insert a custom element at the end of a row, you can use the `detailWriteRow()` function to write out the contents of a row without closing it. You can see an example of this technique in the code that generates the “24 Hour” button in `detail_cr.html` and `detail_iss.html`.
- If you want to explicitly specify the contents of an element in a row without closing out the table that lays out the grid, you can use the `detailRowHdr()` function to specify the header text and the `detailSetRowData()` function to specify the data text. You can see an example of this technique in the code that generates the timer field in `detail_cr.html` and `detail_iss.html`.
- If you provide a function to validate a field's value (normally in an event handler), and want its results reported during browser-side validation (so that an erroneous field is redrawn with a thick red border, and an error message appears in a yellow band at the top of the form), use the function `detailReportValidation()`. You can see an example of this in the `validate_duration()` function used to validate the duration fields in `xx_candp_tab.html`. The `validate_duration()` function is in the file `val_type.js`.
- If you want review the HTML generated for a detail form, you can use functions `docWrite()` and `docWriteLn()` in place of the standard functions `document.write()` and `document.writeLn()`. Then if you invoke the function `holdHTMLText()` anywhere in the <HEAD> section of your form, CA Service Desk Manager will pop up a debugging form containing a TEXTAREA with all of the HTML generated for the form, which you can review or copy and paste into a validation tool.

While you are composing your modifications, remember that the combination detail form is displayed in both a read-only and an edit view. If your customizations apply specifically to one view or the other, you can test the current view in one of two ways:

- In JavaScript, the expression `_dtl.edit` is true in the edit view and false in the read-only view.
- In either JavaScript or open HTML, the statements:

```
<PDM_IF "$prop.form_name_3" == "edit">
```

(code used only in the edit view)

```
</PDM_IF>
```

or

```
<PDM_IF "$prop.form_name_3" == "ro">
```

(code used only in the read-only view)

```
</PDM_IF>
```

can be used to bracket code intended only for the edit or read-only view, respectively.

detailEndTable()

This function closes the HTML table that lays out the detail form elements in a grid. It has no arguments.

You can start a new grid with the `dtlStartRow()` macro. However, elements in a new grid are not necessarily aligned with elements in a previous grid.

detailNextID([colspan,][lastelement])

This function returns a string of the form:

```
" ID=df_nn_nn TABINDEX=n onFocus=func onBlur=func"
```

Inserting this string into an HTML element causes the element to follow the conventions of CA Service Desk Manager mouse-less navigation, including accessibility with the arrow keys and turning pale yellow when focused. The returned string begins with a space and ends with no space.

colspan

Specifies the number of columns in the grid occupied by the element. This argument is optional; it defaults to one if not provided. If omitted, the element is assumed to occupy one column of the grid. This affects arrow key behavior. The *colspan* argument can be omitted even if the *lastelement* argument is provided.

lastelement

A Boolean value specifying whether the element for which the ID being generated is the last one in its row. If omitted, the element is assumed to be followed by other elements. This affects arrow key behavior.

detailNextLinkID()

This function returns a string of the form:

```
" ID=dfink_nn_nn TABINDEX=0 onFocus=func onBlur=func"
```

Inserting this string into an HTML element defining a link element causes the element to follow the conventions of CA Service Desk Manager mouse-less navigation, including accessibility with the up arrow key from the base element and turning pale yellow when focused. The returned string begins with a space and ends with no space.

This function takes no arguments.

detailReportValidation(field, has_error, emsg)

This function reports the result of external field validation. If validation is reported to have failed, the field is redrawn with a thick red border and the error message provided is shown in a yellow band at the top of the form. The user is not permitted to save the record until a subsequent call to detailReportValidation() reports the field as error-free.

The detailReportValidation() function is functional only for fields registered for browser-side validation. All fields created with detail form macros are automatically registered for validation. You can register other fields with the detailSetValidateFunction().

field

(Required) Specifies the form element object containing the field. The easiest way to obtain this is to pass this argument to the event handler performing the validation. Another way is to use the standard JavaScript function document.getElementById().

has_error

(Required) A Boolean or integer value specifying whether the field is in error. Setting a field in error prevents the user from saving the record, causes the field to be highlighted with a thick red border, and places the error message supplied as the third argument in a yellow band at the top of the form. Setting a field as not in error reverses these changes.

emsg

A text string specifying the message to display in the yellow band at the top of the detail form when the *has_error* flag is set. This argument is required if *has_error* is set.

detailSetValidate(hdrtext, is_required, maxsize)

This function specifies that the most recent field created with an ID supplied by detailNextID() is subject to browser-side validation. Validation for required fields and for fields with a maximum size is automatic. Other forms of validation may be provided through JavaScript functions or event handlers calling detailReportValidation().

You should call detailSetValidate() only for form fields you have defined yourself whose ID was created by detailNextID(). The detailSetValidate() function must be called immediately after creating a field that you want validated. It is unnecessary (and will caused unexpected results) to call detailSetValidate() for fields created by detail form macros.

hdrtext

(Required) Specifies a string used to identify the field in error messages.

is_required

(Required) A Boolean or integer value specifying whether the field is required. CA Service Desk Manager automatically verifies that all required fields are provided whenever the user attempts to save a record.

maxsize

An integer specifying the maximum length of data allowed for the field. CA Service Desk Manager automatically verifies that all fields with a *maxsize* value have a length within limits whenever the user attempts to save a record. This argument is required. To suppress *maxsize* validation, specify a value of 0.

detailRowHdr(*hdrtext*, *colspan*, *is_required*)

This function stores text for the header (TH) element of an item in the grid. The text is not actually written to the form until a `detailWriteRow()` function or `dtlStartRow` macro is invoked.

hdrtext

Specifies the text in the header element. This argument is required.

colspan

Specifies the number of columns in the grid occupied by the element. This argument is optional; it defaults to one if not provided. If omitted, the element is assumed to occupy one column of the grid. This affects arrow key behavior. The *colspan* argument must be provided if the *is_required* argument is provided.

is_required

Specifies whether the *hdrtext* should be displayed in the style corresponding to a required field. The argument can be a Boolean, a number, or a string. A number or a string is interpreted as false if zero and true otherwise. This argument is optional; if omitted, the *hdrtext* is styled as a non-required field.

detailSetRowData(*text*)

This function stores HTML text for the data (TD) element of an item in the grid. The text is not actually written to the form until a `detailWriteRow()` function or `dtlStartRow` macro is invoked. The single argument is the HTML text of the element to be stored.

detailWriteRow()

This function writes the HTML stored for the current row. This creates two HTML table rows, one for the header (TH) elements and one for the data (TD) elements. The function also writes the [assign the value for TD in your book] tag that begins a new data element. The TD tag is automatically closed by the dtlStartRow macro, so it is unnecessary (and incorrect) to provide the [assign the value for TD in your book] tags in HTML text that follows detailWriteRow(). This function has no arguments.

Understanding List Forms

The following information provides background information about the internals of CA Service Desk Manager list forms. We recommend that you use the Web Screen Painter Design View to modify these forms.

CA Service Desk Manager list forms are defined with the following macros (invoked with the PDM_MACRO tag):

IsStart

Begins a list

IsCol

Defines a column in a list

IsWrite

Inserts text into the pdm_list part of a list

IsEnd

Ends a list

The general form of a list using these macros is the following:

```
<pdm_macro name=IsStart>
<pdm_macro name=IsCol hdr=hdr1 attr=attr1>
<pdm_macro name=IsCol hdr=hdr1 attr=attr1>
<pdm_macro name=IsEnd>
```

This results in text similar to the following example in the output HTML:

```
var rs = new ResultSet();           From IsStart
rs.startList(); From IsStart
rs.header("hdr1"); From IsCol
rs.setData("attr1","options"); From IsCol
rs.header("hdr2"); From IsCol
rs.setData("attr2","options"); From IsCol
<PDM_LIST SOURCE=list> From IsEnd
rs.data(attr1) From IsCol/IsEnd
rs.data(attr2) From IsCol/IsEnd
</PDM_LIST> From IsEnd
```

Note: There are two distinct sections to the output list: the setup section before the `<PDM_LIST>` tag, and the actual list between the `<PDM_LIST>` and `</PDM_LIST>` tags. The `IsCol` macro makes use of preprocessor variables and the `<PDM_SET>` tag to output data to both sections of the list. The entire list section of the list is created by a `<PDM_EVAL>` tag generated by the `IsEnd` macro.

To insert your own JavaScript in the setup section of the list, simply include it where needed. Use the `IsWrite` macro to insert your own code into the list section of the list.

More information:

[Edit List and Detail Forms in Design View](#) (see page 271)

The `IsWrite` Macro

The `IsWrite` macro specifies text for the list section of a list (the portion between the `<pdm_list>` and the `</pdm_list>` tags). Text specified for the text argument of this macro is deferred, and not written to the output HTML until the `IsEnd` macro.

`IsWrite` [both=no|yes]

text="xxx"

both

Specifies that the text operand is to be written both immediately to the output HTML and to the deferred text buffer. This can be useful to output JavaScript to conditionally bypass both the setup and the list information output by a subsequent `IsCol` macro. Optional; defaults to no.

text

Specifies the text generated by this macro. Text specified is deferred until the `IsEnd` macro.

It is often desirable to include `pdm` tags and references to form variables in the text output by an `IsWrite` macro. To prevent these from being interpreted by the web engine during parsing of the `IsWrite` macro itself, follow these syntax rules:

- If the `IsWrite` macro generates a `pdm_tag`, omit the surrounding “<” and “>” delimiters of the tag. For example, to insert a `<pdm_else>` statement into the list section of the list, code:

```
<PDM_MACRO NAME=IsWrite text="pdm_else">
```

The web engine automatically inserts the “<” and “>” before producing the text when it detects that the first four characters are “pdm_” (or “PDM_”).

- If the IsWrite macro generates a reference to a form variable, code an @ character in place of the \$ character that designates the variable. For example, to generate a reference to the list variable \$list.persistent_id, code:

```
<PDM_MACRO NAME=IsWrite text="@list.persistent_id">
```

The web engine automatically converts the “@” to “\$” before producing the text. To produce a literal @ sign, precede it with a backslash.

Edit in List Customization

Several list forms, such as the Request and Issue lists, include an Edit in List button. When this button is available and a result set is displayed, the user can click Edit in List to replace the search filter with a small edit form. The edit form allows the user to update records directly on the list form. The user can even update everything selected in the list by placing the desired new data in the edit form and clicking Change All.

Editing list data involves no communication with the server until the user clicks Save. When the user clicks Save, all the updates (marked by yellow highlights on the form) are sent to the server, which applies all the changes in a single operation, returning a status message and redisplaying the list.

You can customize this feature by controlling whether the Edit in List button is available on a particular list form, and by controlling the fields that appear in the edit form displayed when the user clicks Edit in List.

To place an Edit in List button on a list form, including the following statement somewhere in the <HEAD> section of the form:

```
<SCRIPT LANGUAGE="JavaScript" SRC=$CAisd/CAisd/list_edit.js></SCRIPT>
```

Simply adding this statement puts the button on the form. However, the button is disabled unless JavaScript statements specifying the contents of the edit form are also included in the form. These statements must be placed immediately prior to the results set specification, and have the following format:

Statements	Comments
startListEdit(_search_filter);	Specify exactly as shown
listEditStartRow();	Specify exactly as shown
listEditField("attr"[, "hdr"]);	Specify zero or more
listEditReadonly("attr"[, "hdr"]);	Specify zero or more
endListEdit();	Specify exactly as shown

The `endListEdit()` statement must be followed by the `ResultSet()` statement that begins the results set. You specify the fields in the edit form and their sequence on the form by coding one or more `listEditReadOnly()` or `listEditField()` statements.

`startListEdit(_search_filter);`

This statement begins the list edit form. It must be coded exactly as shown.

`listEditStartRow();`

This statement begins a new row of fields on the list edit form. It must be coded exactly as shown. You must place a `listEditStartRow()` statement immediately after the `startListEdit()` statement. You can optionally include additional `listEditStartRow()` statements among the `listEditField()` and `listEditReadOnly()` statements that specify the fields on the form.

`listEditField(attr_name[,hdr]);`

This statement specifies an attribute to be included on the list edit form.

attr_name

Specifies the name of the attribute to be included in the edit form (including dots, if appropriate). All attributes specified for a list edit form must also be in the results set. The *attr_name* specified must be identical to that specified in the `rs.showData()` or `rs.showDataWithLink()` that adds the attribute to the results set.

The attribute appears on the edit form in the same format that it appears in the search filter. If the attribute is not in the search filter, it is edited in a 20-character text box.

attr_name is a required argument.

hdr

Specifies the text of the header on the field in the edit form. This argument is optional; if omitted, the header text is taken from the search filter. If *hdr* is omitted and the attribute is not in the search filter entry for *attr_name*, the header text defaults to the attribute name surrounded by question marks.

`listEditReadOnly(attr_name[,hdr]);`

This statement specifies a non-editable attribute to be included on the list edit form. Its arguments have the same significance as those for `listEditField()`.

`endListEdit();`

This statement ends the list edit form. It must be coded exactly as shown.

Integrating with Your Own Web Pages

You can integrate the CA Service Desk Manager web interface functionality with your web pages to present a seamless interface for your users.

Note: The web engine, which is the executable that acts as the gateway between the web server and the CA Service Desk Manager server, allows multiple simultaneous connections from a given user. More than one frame at a time can have an open connection to the CA Service Desk Manager web engine process.

You can integrate the web interfaces in the following ways:

- By creating links from any of your web pages to the appropriate CA Service Desk Manager web page without having to go through the web interface menu page.
- By adding HTML forms to your web pages that collect input and perform supported operations directly, without displaying any CA Service Desk Manager web data entry pages.
- By creating web form groups that can be used to associate HTML web-based forms to users through their access type. Similar to the form groups used by the administrative interface, web form groups can be used to customize your HTML pages.

More information:

[Supported Operations](#) (see page 314)

Linking to CA Service Desk Manager Functions

You can link directly to major CA Service Desk Manager functions without displaying the main page. You typically do this by accessing the pop-up window for the new window containing the CA Service Desk Manager information. You can also replace your web page with the CA Service Desk Manager page.

In both cases, the product displays the requested page in the same way that the user sees it in a typical session, but without the main page and scoreboard. If you are an analyst, display the main page and scoreboard by selecting File, Restore Scoreboard, which is available only on pages displayed by bypassing the main page.

To create a link that bypasses the main page, specify a URL of the following form:

```
http://hostname[:port]/CAisd/pdmweb.exe?OP=operation+var=value+...
```

In this example URL, *hostname* is the web server host computer; *port* is the port number (typically 8080) required only if you are using Tomcat as your http server; *operation* is one of the supported operations, and *var=value* is one or more of the variables allowed with the operation.

For example, a link that loads the form for creating a request can be specified as the following:

```
<A HREF=http://hostname/CAisd/pdmweb.exe?OP=CREATE_NEW+FACTORY=cr>Define Request</A>
```

More information:

[Link Examples](#) (see page 321)

Posting Forms to CA Service Desk Manager

You can also access CA Service Desk Manager functionality by adding HTML forms to your web pages that refer to supported operations. If the form is submitted with sufficient information to perform the operation, such as creating a request, the operation is performed without displaying a form to collect additional input.

When you add an HTML form to your web page:

- The ACTION for the form is the URL for pdmweb.exe.
- The METHOD is POST.
- Either the name of the SUBMIT button should be one of the supported operations, or you should have a hidden field named OP whose value is one of the supported operations.

For example, to create an HTML form that loads the page for creating a request, specify the following code:

```
<FORM ACTION=/CAisdCAisd/pdmweb.exe METHOD=POST>
<INPUT type=HIDDEN NAME=FACTORY VALUE=iss>
.
.
.
<INPUT type=SUBMIT NAME=CREATE_NEW VALUE=" OK ">
</FORM>
```

JavaScript Customization

The CA Service Desk Manager web interface makes extensive use of JavaScript and includes a number of JavaScript files in the `/CAisd/scripts` directory. If you decide to customize any of these script files, place the modified version in `/CAisd/sitemods/scripts`, as described in [Directories Used by Your HTTP Server](#) (see page 329).

For performance reasons, the JavaScript files delivered in the `/CAisd/scripts` directory are compressed, with comments and unnecessary white space removed. This compression can make them difficult to read. You can find uncompressed versions of all JavaScript files in one of the following directories:

- (UNIX) `$NX_ROOT/sdk/scripts`
- (Windows) `$NX_ROOT/sdk/scripts`

If possible, avoid creating customized versions of entire JavaScript files, because each file contains a number of functions and you may only want to modify one function. In most cases, you can override individual functions by placing a modified version in the JavaScript file `sitemods.js`. We strongly recommend that you take this approach when modifying JavaScript.

`sitemods.js`

A skeleton *sitemods.js* file is distributed with CA Service Desk Manager. All distributed HTML files include this file at the end of their `<head>` section, making it the last JavaScript file loaded. Because it is the last file, any functions defined in it override functions with the same name included earlier. This lets you provide your own version of a distributed JavaScript function without directly modifying distributed code.

This approach is not effective for functions invoked at load time in the `<head>` section, such as those in `menubar.js` and `ahdmenus.js`.

However, you can customize most JavaScript functions by completing the following steps:

1. Place a modified version of the function in `sitemods.js`.
2. Store the updated copy of `sitemods.js` in `CAisd/site/mods/www/wwwroot/scripts`.

More information:

[Edit Menu Bars](#) (see page 276)

Modifying Context Menus

A number of forms within CA Service Desk Manager use context menus, accessed by right-clicking an object. Using the Web Screen Painter, you can modify context menus to add, remove, or modify their items.

Note: For more information about adding menu items, see the *Web Screen Painter Help*.

Updating and Creating Change Orders as Employee User

By default, a user can only view change orders from the Employee web interface. Use the following steps to enable creating and updating change orders by employees:

1. Sign on to the web as the Administrator, and select the Administration tab.
2. Select Access Type from the Security menu.
The Access Type List appears.
3. Select the Employee link to display the Employee Access Type Detail window.
4. Set the Change Orders to "modify" under the Function Access tab and save.
5. Click the Back button to return to the Administration tab, and then select Data Partitions, Data Partition List.
6. Click Employee to display the Data Partition Detail window. On the Constraints List portion of the window, review the Type column for following Change_Request Tables:
 - Pre-Update
 - Create
7. For each Table that you want to edit, click the Table name to display that table's Data Partition Constraint Detail window.
8. Click the Edit button.
9. Edit the constraint as follows:
change "id = 1" to "affected_contact = @root.id".
10. Click Save.

Now when you login to the web interface as an employee user, the *Create Change Order* link appears.

Add a "Closed Change Orders" link to the Employee Scoreboard

You can use the product to add a Closed Changes node option to the Employee web interface scoreboard.

To add a Closed Changes node

1. Log in to the product as an Administrator.
2. Click the Service Desk tab.
3. Select File, Customize Scoreboard.
The Customize Scoreboard dialog appears.
4. Click the Role option and select Employee in the drop-down list.
5. Under Add New Node, click the Node's Stored Query link.
The Stored Query List dialog appears.
6. Search and select Closed Changes from the Stored Query list. This is typically displayed as code CHGUBIN7.
7. Specify a location for the new node by selecting an item in the scoreboard tree on the left.
8. Click Add New Node.
The new node named Closed Changes is added to the scoreboard tree.
9. Click Finished.

Download Attachments

When you download an attachment in CA Service Desk Manager, it automatically displays the attachment in the browser window without prompting for a response from you. This action can be dangerous if a virus is associated with the attachment.

With CA Service Desk Manager, you can force a save-as dialog that prompts you to respond if you want to save the attachment on the disk or open it. Saving an attachment can be a secure method because you can save the attachment on the disk and scan it before you can actually open it. You also have the option to force the save-as dialog only on certain attachment types.

You can force the save-as dialog to appear through the web.xml servlet configuration file. The web.xml file is located at the following paths:

Windows:

`$NX_ROOT\bopcfg\www\CATALINA_BASE\webapps\CAisd\WEB-INF\web.xml`

Linux: `$NX_ROOT` is `"/opt/CAisd"`

Event Log Data Storage Customization

The system environment variable `@NX_EVENT_LOG_EXCLUDE`, which is set in the `NX.env` file and requires a restart of the CA Service Desk Manager services, lets you control the amount of data that is stored in the event log (event_log table). This variable lets you store only the events you want to track, report on, and use as part of the Recent Activity that can be launched as a button from the Quick Profile page.

In this variable, commas separate list items (for example, `@NX_EVENT_LOG_EXCLUDE = FAQ,KD_OPEN`). For example, if you use the LOGIN,LOGOUT events from the following table (`@NX_EVENT_LOG_EXCLUDE` value of LOGIN,LOGOUT), the product does not record login and logout events.

Refer to the following information when customizing data to store in the event log using this variable.

Event	Enum	By	Sets	Comments
LOGIN	1	CA Service Desk Manager		Specifies that the User logs into the system.
LOGOUT	2	CA Service Desk Manager	numdata1	Specifies that the User logs out, where numdata1=logout reason: 0 —normal 1 —timeout 2 —abnormal
CR_CREATE	3	CA Service Desk Manager	sd_obj_type, sd_obj_id, kd, numdata1	Indicates that the User creates a request, where numdata1=id of affected end user.
ISS_CREATE	4	CA Service Desk Manager	sd_obj_type, sd_obj_id, kd, numdata1	Indicates that the User creates a change order, where numdata1=id of the affected end user.
CHG_CREATE	5	CA Service Desk Manager	sd_obj_type, sd_obj_id, kd, numdata1	Indicates that the User creates an issue, where numdata1=id of the affected end user.
EMAIL	6	Knowledge Management	kd	Specifies that the Analyst emails a document.
LINK	7	Knowledge Management	kd, sd_obj_type, sd_obj_id	Indicates that the User accepts a solution, and links it to a ticket.
UNLINK	8	CA Service Desk Manager	sd_id, sd_obj_type, sd_obj_id	Specifies that the User unlinks a solution from a ticket.

Event	Enum	By	Sets	Comments
SEARCH	9	Knowledge Management	numdata1,	Indicates that the User searches knowledge, where numdata1= CI_ASKED_QUES id.
FAQ	10	Knowledge Management	numdata1	Indicates a FAQ search, where numdata1= O_INDEXES id (category).
DT_NAVIGATE	11	Knowledge Management	kd, numdata1, textdata1	Indicates that the User navigates a decision tree, where numdata1= ES_NODES ID textdata1=path.
KD_BOOKMARK	12	Knowledge Management	kd	Indicates that the User bookmarks a KD.
KD_COMMENT	13	Knowledge Management	kd, numdata1	Indicates that the User adds a comment to a KD, where numdata1= O_COMMENTS id.
KD_CREATE	14	Knowledge Management	sd_ob_type, sd_obj_id, kd	Specifies that a User creates a document. CA Service Desk Manager IDs are used when a KD is created using submit knowledge from a request or an issue.
KD_OPEN	15	Knowledge Management	kd, numdata1	Indicates that a User opens a KD, where numdata1=BU_TRANS ID.
KD_RATE	16	Knowledge Management		Indicates that a User rates a KD, where numdata1=BU_TRANS ID.
KD_NEW	17	Knowledge Management	numdata1	Specifies that a User clicks on the New Documents folder in the Knowledge tab.
NX_ATTACH_AUDIT_TO_NEW_TICKET	18	CA Service Desk Manager		When a User opens a new ticket, all events for the current session appear by default on the Event Log tab of the ticket. 0 —Only events relevant to the ticket appear on the Event Log tab. 1 —All events for the current session appear on the Event Log tab of the ticket.

Event	Enum	By	Sets	Comments
TICK_OPEN	19	CA Service Desk Manager		Indicates that the ticket was viewed.
TICK_SEARCH	20	CA Service Desk Manager		Indicates the user searched for tickets and links the number of searches.
KD_PRNT	21	Knowledge Management	kd	Indicates the knowledge document was printed.

CA Business Intelligence Reports Customization

You can customize CA Business Intelligence reports, starting at the point after fields and tables have been defined in the CA Service Desk Manager schema.

Note: For information about performing schema modifications, see the [Schema Designer Modification Overview](#) (see page 254).

Before you begin, verify that you have done the following:

- Installed CA Business Intelligence and configured CA Business Intelligence so it works correctly with CA Service Desk Manager.
- Established user permissions, roles, authentication options, and data partitions security for your Reporting environment.

Note: For information about setting up CA Business Intelligence security, see the *Administration Guide*.

CA Business Intelligence Infrastructure

CA Business Intelligence (CA BI) is an enterprise reporting infrastructure that enables you to create, maintain, store, schedule, and distribute reports for CA Service Desk Manager users and roles. BusinessObjects Enterprise XI, Release 2 and its associated tools, coupled with BusinessObjects Crystal Reports XI are the backbone of the architecture. BusinessObjects Enterprise tools are contained in a CA Service Desk Manager created package, merging CA Service Desk Manager reporting essentials into an industry leading business intelligence framework.

Note: Although Crystal reports are delivered as the primary component of CA BI, the report creation and maintenance tool, Crystal Reports XI, is not delivered. Crystal Reports XI is a separately licensed product that can be purchased from BusinessObjects and used in conjunction with CA BI.

Reporting Components

Following are the primary components included in the CA Business Intelligence infrastructure:

- **CA Service Desk Manager Database/ Domsrvr / ODBC Driver**—Report data is stored in a SQL Server or Oracle CA Service Desk Manager database. BusinessObjects reporting applications (Crystal Reports and Web Intelligence) access the database using an ODBC driver that connects directly with the CA Service Desk Manager object engine (domsrvr). All CA Service Desk Manager security, including data partition and tenancy restrictions, is automatically applied to reports.
- **Central Management Server**—The Central Management Server (CMS) is the central repository that stores all objects used in every reporting process.
- **Central Management Console**—The Central Management Console (CMC) is the main administrative facility for BusinessObjects. It provides access to all BusinessObjects administration functions. Using the CMC, you can deploy reports and assign user access and folder permissions for InfoView.
- **BusinessObjects Universe**—The universe provides a business representation of a data warehouse or transactional database. It describes the classes (tables) and objects (columns) which are used in reports. The CA Service Desk Manager universe is installed and configured during the installation. At the completion of the installation, the universe connection is assigned to various groups and users in CA Service Desk Manager.
 - **Designer**—The Designer is a tool in BusinessObjects Enterprise that lets you modify the CA Service Desk Manager Universe, which is a metalayer between CA Service Desk Manager schema, and BusinessObjects reporting tools. The Import/Export Wizard facilitates object population or extraction within the CMS.

- **Default Predefined Reports**—Predefined reports are web-based CA Service Desk Manager and Knowledge Management reports developed with either BusinessObjects Web Intelligence (WebI) or Crystal Reports. The reports can be used as models for defining site-specific reports.
- **InfoView**—BusinessObjects InfoView is a web interface that allows authorized CA Service Desk Manager users to interact with web-based predefined reports by viewing, running, and scheduling report types including, but not limited to, WebI and Crystal Reports. Reports are contained in folders in the public section in InfoView.
- **Ad Hoc Reports**—Ad hoc reports are created and administered from InfoView using a WebI plugin-based interface. This tool is intended for users who want to create basic reports easily without writing queries.

Development Environment

Updating the CA BI infrastructure with CA Service Desk Manager schema changes is an administrative function. An administrator who is promoting modified schema into the reports must set up the environment, apart from their production environment.

Some of the tools used by CA BI require a Windows-based architecture. This means that installations of Linux/UNIX must configure CA BI on a Windows computer to interact with the Linux/UNIX production environment CA BI installation. If you are using Windows servers in production, you should configure an additional Windows computer for your development environment.

Tools

Updating the CA BI infrastructure with CA Service Desk Manager schema changes is an administrative function. To promote modified schema changes into the reports, you must include the following tools in your development environment:

Business Objects Designer

This full client Windows tool is installed on the CA BI production server as part of the base CA BI installation for windows. When the CA BI server is a non-Windows architecture, or when login access to the production CA BI application server is undesirable, you must create a separate CA BI installation on a Windows (development) server. A development CA BI server installation allows you to access the production CA BI objects remotely, no matter the architecture of the CA BI production server installation.

BusinessObjects Web Intelligence

This web-based report creation tool is used for modifying and creating Web Intelligence (WebI) reports. You can access the WebI tool through the BusinessObjects InfoView interface. Administrative permissions for the WebI and InfoView tools are available within CA BI, specifically using the BusinessObjects Central Management Console (CMC) tool.

CA Service Desk Manager ODBC Driver

Provided with the CA BI installation is the CA Service Desk Manager ODBC Driver. This component enables WebI and Crystal Reports to access CA Service Desk Manager data while enforcing data partition security. The ODBC Driver is installed as part of the base CA BI installation on the CA BI application server. It is also available as a client installation so that it can be used on a computer not running CA BI along with the Crystal Reports XI client.

Note: For information on defining data partitions security for your reporting environment, see the *Administration Guide*.

How to Create a Development Environment

To create a development environment, do the following:

1. Secure a server with a supported Windows operating system.
2. Install and configure CA Service Desk Manager.
3. Install and configure CA Business Intelligence.
4. Change the default ODBC DSN Name from casd_XXXXX to casd_YYYYY where YYYYY is exactly the same as the DSN on the production implementation.

Important! Regardless of the actual connection properties, the DSN name must be identical on the development and production implementations.

5. (Optional) Install and configure Crystal Reports XI.

Note: It is not required to install Crystal Reports on the same computer as CA BI. Crystal Reports can be installed on a different computer, as long as the CA Service Desk Manager ODBC Driver is also installed on the Crystal Reports computer, and the DSN Name is modified to be identical to the production implementation, regardless of the actual connection properties. For more information about installing an individual copy of the CA Service Desk Manager ODBC driver, separate from the CA BI installation, see your ODBC Driver documentation.

6. [Create the Framework](#) (see page 355).

Framework

After the tools are available in your development environment, the next step is to create a framework that will allow schema changes to be preserved through product upgrades.

Important Do not modify the default development CA Service Desk Manager universe installed with CA BI. Otherwise, your schema changes may be overwritten during patch and upgrade processes. Modifying the CA Service Desk Manager universe will eventually result in lost schema changes within the CA BI infrastructure.

The BusinessObjects universe is the metalayer that describes the schema within the CA BI infrastructure. Instead of changing the CA supplied universe, you can create a customer-specific universe linked to the CA Service Desk Manager universe. Using this approach, you can maintain local schema changes with minimal effort during the upgrade process, and CA Service Desk Manager will be able to provide upgrades to the base universe.

CA Service Desk Manager customers familiar with BusinessObjects universe documentation will be aware of other documented procedures available from BusinessObjects that allow tying universes together. The process discussed here, however, is the only process that is supported by CA for maintaining customer modifications.

The default universe is named "CA Service Desk Manager" and it is stored in the "CA Universes" folder within the Central Management Console (CMC). This default universe is the "kernel" universe in a structure where universes are linked.

The CA Service Desk Manager universe can be named anything you choose. The name will be displayed to report writers when they are building reports, so make sure the name is meaningful. The customer universe is the "derived" universe in a structure where universes are linked.

Within this framework, any number of derived universes can be maintained, but only one is required for maintaining schema changes. Multiple derived universes may be used for ease of maintenance or security requirements, but such decisions are solely at the discretion of your production support needs.

In any multiple derived universe environment, ensure that you do the following:

- Maintain the z_ naming convention for the universe file name on all universes.
- Use the CA Service Desk Manager connection, then store the universe in the CA Customer Universe folder.
- Do not delete the link to the kernel universe.

Create a Framework for Promoting Schema Changes to CABI

To create a framework for promoting schema changes to CA Business Intelligence

1. Open the BusinessObjects Designer.
Select File, New from the Designer menu.
The Universe Parameters window appears.
2. Click the Definition tab and enter a meaningful name for this universe in the Name field.
3. (Optional) Enter a description in the Description field.

4. Select CA Service Desk Manager from the Connection drop-down list.
5. Click the Add Link button on the Links tab.
The Universe to Link dialog appears.
6. Expand the CA Universes folder and complete these tasks:
 - a. Open the CA Service Desk.unv file. The Universe to Link dialog closes and the CA Service Desk Manager universe appears on the Links tab.
 - b. Click OK to close the Universe Parameters dialog.
Designer may take a few minutes to process the link and create the derived universe.
7. After the derived universe is created, perform these tasks:
 - a. Modify the following parameter(s) as appropriate.
 - Select Parameters from the File menu.
 - Click the Parameter Tab.
 - Specify ANSI92 = YES.
 - b. Click the Controls tab and set the following fields to a value appropriate to your implementation, then click OK to save the values and close the parameter dialog:
 - Limit size of result set
 - Limit execution time
 - Limit size of long text objects (minimum of 4000).
 - c. Define hierarchies. Note that customer hierarchies are not imported.
 - Select Tools, Hierarchies.
 - Multi-select all custom hierarchies, then click the Add arrow button. All hierarchies are moved to the right side.
8. From the Designer menu, click File, Save.
The Save As dialog appears.
9. In the File Name field, select any descriptive file name, and proceed the file name with "z_". For example, a universe named "ACME Anvil Co" might default to: "ACME_Anvil_Co.unv." Change this file name to "z_ACME_Anvil_Co.unv" before saving.
10. Export the derived universe to the CMS as follows:
 - a. Select File, Export from the Designer menu.
 - b. From Domain field drop-down list, select <Browse>, then locate and select CA Customer Universes.

- c. Click OK to export the universe to the local CMS

The Universe Successfully Exported dialog appears.

The framework now exists to promote custom schema changes throughout CA BI.

11. Log into BusinessObjects InfoView as an administrative user and do the following:

- a. Select Public Folders.
- b. From the InfoView toolbar, click New, Folder.
- c. In the Folder Name field, provide a description meaningful to report users, such as "Organization Name Reports."
- d. Click OK to see the folder created under Public Folders.

This creates the minimum framework to use and store reports by your organization. Any number of subfolders and objects can be added to this folder structure.

Schema Changes to the Infrastructure

After the CA BI development environment is established and schema changes have been published to CA Service Desk Manager using the documented process for customizing schema data, the schema changes are ready to be promoted through the CA BI infrastructure. You can make the new schema available for report creation and modification.

Add Schema Changes to Derived Universe

Promoting schema changes into the CA BI infrastructure is as straight-forward as adding the new schema object to the derived universe.

Note: Before you begin, verify that the appropriate steps have already been completed, and the new schema objects have been added to the CA Service Desk Manager flexible schema.

To add schema changes to the derived universe

1. Open the BusinessObjects Designer, and import the derived universe to a local file system as follows:
 - a. Select File, Import from the Designer menu.

The Universe Successfully Imported dialog appears.
 - b. Click OK.

2. Refresh the structure of the derived universe as follows:

- Select View, Refresh Structure from the Designer menu.

The following questions appear:

- "Do you want to refresh the out of date columns in selected tables?" Click OK.

Note: If the message "No update needed" appears, it means the CA Service Desk Manager object layer has not been appropriately updated with the new schema. Review the steps for publishing schema changes to CA Service Desk Manager.

- "Refresh structure: The structure has been successfully modified." Click OK.

New columns appear in the universe structure on the right side of the window, making new object(s) available for use within the derived universe.

The objects are available to the CA BI tools after they are moved from the right pane to the left pane. When you add objects to the left pane, make sure that you follow the [common schema modifications](#) (see page 359) standards.

3. Drag and drop the new object(s) to the desired location in the left pane.

4. Click Save.

5. Select File, Export from the Designer menu.

The Universe Successfully Exported dialog appears.

6. Click OK.

Changes added to the derived universe schema are exported to the local CMS.

7. From the Designer menu, select Tools, Check Integrity.

- a. In the dialog that appears, select the Parse Objects check box. (Do not change other settings.)
- b. Click OK. The integrity check is started.

Note: No parse errors should be reported. If errors are found, modify your objects in the left pane so that they do not produce parse errors.

8. Click OK to close the dialog.

9. Export the derived universe to the CMS as follows:

- a. Select File, Export from the Designer menu.
- b. From Domain field drop-down list, select <Browse>, and then locate and select CA Customer Universes.
- c. Click OK to export the universe to the local CMS.

The Universe Successfully Exported dialog appears.

10. Save your changes and export the CA Service Desk Manager universe.

The changes are now available in your CA BI reporting environment, including Web Intelligence and Crystal Reports.

Common Schema Modifications

You can implement schema modifications in the Universe. To familiarize you with the process, the following table lists the common schema modifications that you might encounter.

When a field type is defined in Web Screen Painter as...	Follow these rules when using the field in the Universe: Right click the attribute and select...
INTEGER	Object Properties, Definition Tab, Type = Number
STRING	Object Properties, Definition Tab, Type = Character
DATE	Object Properties, Definition Tab, Type = Date Object Format, Number Tab: Choose category "Date/Time"; Choose Format mm/dd/yyyy hh:mm:ss AM/PM
DURATION	Object Properties, Definition Tab, Type = Number; Object Properties, Definition Tab, Select = PdmSeconds(object.attr)
SREL	Create a CA Service Desk Manager attribute alias.
BREL	Not Applicable
QREL	Not Applicable
DERIVED	Use an appropriate data type and object format for the value stored in the derived field, if desired. The Derived field can produce any result, so there is not a specific standard to follow.

When a field type is defined in Web Screen Painter as...	Follow these rules when using the field in the Universe: Right click the attribute and select...
Special Case: Local This is not a data type defined within Web Screen Painter, but instead a data type used by the universe sometimes to indicate an unsupported data type.	The Local field is displayed in the right pane of the universe with type "L". These fields can be dragged, but not dropped into a class on the right universe pane. Most often, fields data types such as binary, are not supported by the Universe. However, they can be added to the left pane of the universe by creating an object and placing the PdmString (object.attribute) in the "SELECT" window of the Edit Properties dialog.

Reports and Folder Structures

Included with the CA BI installation are several Crystal Report XI and WebI report objects. The reports are contained in the following CA Service Desk Manager folder: CA Reports\CA Service Desk Manager.

Important! Do not modify the CA Service Desk Manager universe and report objects contained in the CA Service Desk Manager folder structure.

Consider the following information about reports and the folder structures:

- The steps for creating a framework explain how to add a folder in the InfoView public section, which is specific to the end user. Within this folder, a user can create additional subfolders and report objects.
- In an implementation where each user is authorized access to CA BI by their unique CA Service Desk Manager login ID, users can save reports for personal use within the My Folders section. BusinessObjects enforces security on this folder by showing these objects only to the logged in user.
- In an implementation where all users have a single reporting user ID for accessing CA BI, the My Folders section is available to all users.

Create a Web Intelligence Report

To create a Web Intelligence report within CA Business Intelligence

1. From the CA Service Desk Manager Reports tab, click the InfoView button.
The InfoView home page appears.
2. Click New, Web Intelligence Document from the menu bar.

3. Select the derived universe that you created when you defined your development framework.

The Web Intelligence report creation tool appears.

Note: Save your document at regular intervals. If the connection session times out, your report modifications will be lost. For information on how to increase the Web Intelligence connection session time-out value, see the *Administration Guide*.

4. From the Web Intelligence toolbar, select Save, Save as.

The Save Document dialog appears.

5. In the General section, specify a meaningful name for this report in the Title field.
6. In the Location section, select the appropriate folder.
7. (Optional) modify the properties as desired.
8. Click OK to save the report.

The report appears in the specified folder and is available to all report users.

Modify a Web Intelligence Report

To modify a CA Business Intelligence report that was delivered in the CA Reports\CA Service Desk folder structure

1. From the CA Service Desk Manager Reports tab, click the InfoView button.

The InfoView home page appears.

2. In the left pane, navigate the CA Reports folder structure, and open the desired Web Intelligence report.

3. Click the name of the report so that the report runs and displays a result.

4. From the Web Intelligence toolbar, select Document, Save as.

The Save Document window appears.

5. In the Location section, select the appropriate folder.

6. Click OK to save the report in the new location.

7. Select Document, Edit.

8. Click Edit Query (the name of the universe CA Service Desk Manager appears on the Data tab).

9. Click the Properties tab. If necessary, click the down-facing arrows next to the Universe, so that the CA Service Desk Manager text is displayed with an ellipse (...).

10. Click the ellipse (...) button to display the Universe dialog.

The Other Available Universes window appears.

11. Select the name of your universe and click OK.

Web Intelligence will automatically map all known fields from the CA Service Desk Manager universe to your universe and display the Change Source dialog. Green check boxes appear next to each mapped field. If all fields are mapped correctly, click OK to confirm the change. If any fields are displayed with a red "X", click the ellipse (...) button next to the field name, and select the appropriate field.

12. From the Web Intelligence toolbar, click Edit Report, and select the Properties tab.

13. Expand the General node.

14. Click the ellipse (...) button next to the Document Properties value.

The Document Properties dialog appears.

15. In the Document Options section, select the Refresh on Open check box.

16. Click Save and then close Web Intelligence.

The report is associated with the appropriate universe and can be modified as needed.

Create a Crystal Report

To create a Crystal report

1. Launch Crystal Reports XI.

2. Select File, New, Blank Report.

The Database Expert dialog appears.

3. Expand the Create new Connection node and click Universes.

The Business Objects Enterprise dialog appears.

4. Log on to BusinessObjects Enterprise using your administrator credentials.

5. Navigate to the folder containing the derived universe.

6. Select the derived universe and click Open.

The Business Objects Query Panel dialog appears.

7. In the Universe tree structure, drag and drop the appropriate attributes into the Select and Filter sections of the Query panel.

8. When the query building process is complete, the standard Crystal Reports designer tool is presented.

9. Build and run the report.

Note: For detailed information on building and running reports, see Crystal Reports documentation.

10. Save the report in the Business Objects Enterprise repository as follows:

- a. Select File, Save as.
- b. In the Save As dialog, select Enterprise.
- c. Navigate to the folder created when you defined your development framework, and save the new report in BusinessObjects Enterprise.

The new report is now available in the enterprise, and can be modified as needed.

Modify a Crystal Report

To modify Crystal Reports XI Reports

1. Open Crystal Reports XI.
2. Select File, New, Blank Report.
The database expert dialog displays.
3. Click to expand Create new Connection.
Click Universes.
The Business Objects Enterprise dialog displays.
4. Log on to Business Objects Enterprise as administrator.
5. Click to navigate the folder housing the derived universe and click to select the derived universe.
Click Open.
The Business Objects Query Panel dialog displays.
6. Navigate the universe tree structure to find, drag, and drop attributes to the select and filter portions of the query panel.
When the query building process is complete, the report writer is presented with the standard Crystal Reports designer tool.
7. Build and run the report as per Crystal Reports instructions.
8. When ready, save the report to the Business Objects Enterprise repository.
 - a. Choose File, Save as
The Save as dialog displays.
 - b. On the left side of the save as dialog, click the Enterprise icon.
 - c. Navigate the folder structure starting with the customer-specific folder created earlier in this document and click Save to save this report in Business Objects Enterprise.
9. The new report is now available in the enterprise, and can be modified as needed.

Move New Crystal Reports into CA Business Intelligence

To move new Crystal reports into CA Business Intelligence

1. Open Crystal Reports XI.
2. Select File, Open.
The Open dialog appears.
3. From the left side, click Enterprise.
The Log on to BusinessObjects Enterprise dialog appears.
4. Log on to BusinessObjects Enterprise using your administrator credentials.
Navigate the folder structure beginning with "CA Reports\CA Service Desk" until the target report displays.
5. Select the target report and click Open.
The report appears in Crystal Reports.
6. Click File, Save As.
The Save As dialog appears.
7. Specify the folder that you created when you defined your development framework.
8. Click Save.
9. From the Crystal Reports menu, select Database, Database Expert.
The Database Expert dialog appears.
10. From the left pane, select Create New Connection, Universe.
11. Select the derived universe.
12. Drag and drop all attributes used by this report into the new query and click OK.
13. In every formula and cell, replace the original attributes with attributes from the new query.
14. After all attributes are replaced, do the following:
 - a. Select Database, Database Expert from the Crystal Reports menu.
 - b. In the Database Expert dialog, select the original query, and click the left facing arrows in the center of the two panes to remove the old query.
 - If all attributes have been properly replaced, the original query will be removed without issue and you can continue modifying the report.
 - If any original attributes remain in the report, the following message appears: "The report is using fields from one or more tables. Continue?"
 - Click Cancel and return to the report.

- Continue replacing all original attributes with attributes from the new query until the original query can be removed without any warning messages.

The report is ready for customization.

How to Move Existing Access Reports into CA Business Intelligence

Microsoft Access predefined reports are no longer developed or provided with CA Service Desk Manager. You can, however, move your existing Access reports into the CA Business Intelligence reporting environment by performing the following steps:

1. Add the appropriate CA Service Desk Manager schema changes to CA Business Intelligence.
2. Use Web Intelligence or Crystal Reports to rewrite the report.

Move from Development to Production

At this point, a derived universe has been added, folder structures have been created and reports have been created and modified. Therefore, you can move the development structure into production. In this step, you will use the BusinessObjects Import Wizard.

To move from development to production

1. From the Start menu, select All Programs, BusinessObjects XI Release 2, BusinessObjects Enterprise, Import Wizard.
The Import Wizard interface appears.
2. In the Source Environment screen, select Business Objects XI R2 and fill in the necessary credentials for the development system.
3. In the Destination Environment screen, select *one* of the following options:
 - **The production system.** This option moves all selected objects immediately into the production system.
 - **An external file that can be imported into the production system at a later time.** This option stores all objects in a Business Intelligence Archive Resource (BIAR) file.
4. In the Select Objects to Import screen, clear all settings, then select the following check boxes: "Import folders and objects" and "Import Universes".
Note: Ignore any warnings that might appear.
5. Select the folder structure containing your specific folder and uncheck any objects that should not be moved.

6. In the Import Options for Universes and Connections screen, click the following option: "Import the universes and connections that the selected Web Intelligence and Desktop Intelligence documents use directly."
7. On the Universe Folder and Universes screen, expand the appropriate folder where the universe is stored.

Note: Ignore any warnings that might appear.

8. Click Finish.

If the folders and reports were moved directly to the production computer, you can view the changes in InfoView. If the objects were placed in a .biar file, use the Import Wizard at a later time, selecting the .biar file as the Source Environment, and the production server as the Destination Environment.

Customizing Legacy Reports

CA Service Desk Manager lets you customize legacy reports or design your own reports. You can:

- Customize legacy Summary, Detail, and Analysis reports to contain exactly the information you need, such as additional fields.
- Produce a new report with any information available from the database in a format that is useful to you.
- Pass arguments of variable information into the report by including command line arguments. Arguments can be values or expressions, such as the current value of a field or an SQL WHERE clause expression.
- Generate reports at the command line, from a script file, or from a menu option.

To generate a custom report

1. Design the report:
 - Decide what data you want to include in the report.
 - Create a report template that contains SQL-like queries, expressions, and functions to manipulate data, and statements to format the data for the printed page.

2. Generate the report from:
 - The command line
 - A CA Service Desk Manager menu option
 - A script file

Note: If you have a third-party database system you can use its report generating tools to create reports with data from the CA Service Desk Manager database. CA Service Desk Manager provides several database views that simplify the process of creating customized reports using third-party database systems. See the documentation for your database system for information about reporting on databases. For more information about database views, see the *Administration Guide*.

More information:

[Customize Crystal Reports](#) (see page 390)

Custom Report Design

To design a custom report, you must have a basic understanding of the following concepts:

- Writing SQL queries.
- Programming, especially in C.
- Creating special programs or script files that you may need to execute before you execute the report template program. For example, you may want to create a program that prompts the user to enter an argument, such as the conditions for a WHERE clause.

Note: Before you create a custom report, be sure to check if the report you need is already provided. CA Service Desk Manager provides a wide variety of Crystal and Microsoft Access reports, and runtime versions of these products to let you run the reports. For more information about reports, see the *Administration Guide*.

Selecting Information for the Report

To help you select data from the CA Service Desk Manager database for customized reports, see the *CA Service Desk Manager Technical Reference Guide*. It lists database tables, fields, descriptions, and other database information.

How to Create a Report Template

A report template is a file that, when executed by a CA Service Desk Manager report program, generates a report of a particular design. A report template contains variable expressions, functions, and statements that define how the data is fetched, calculated, and printed.

To create a report template, create a file containing the following types of report statements:

Block statements

Defines the CA Service Desk Manager database tables from which data will be fetched and the actions that are to be performed on the fetched data.

Layout statements

Defines how the data variables and literal text display on the report output.

Note: Store all your .rpt files in a new directory, \$NX_ROOT/site/mods/rpt (UNIX) or *installation-directory\site\mods\rpt* (Windows). This directory preserves them when you upgrade to a new release of CA Service Desk Manager.

More information:

[Report Template Reference](#) (see page 376)

[Example: Report Template](#) (see page 372)

Block Statements

Block statements provide the report template with its framework. They define the data to be manipulated and control the execution of the report. Block statements begin with a name that must be unique throughout the report template. They then have the following two sections:

Data query section

Contains SQL SELECT, WHERE, and SORT clauses to define which data is fetched from the database.

Output program section

Defines the actions that are to be performed on the fetched data. It contains variable declarations, functions, and other block statements, including nested statements, which can be used to create conditional reports. It can also contain layout statements, which format and print the data as ASCII text.

A simplified version of the syntax of a block statement that shows the relationship between the two sections follows:

```
BLOCK blockname ("SELECT clause", "WHERE clause")
```

```
    SORT clause {output program statements}
```

The BLOCK in the Reference section discusses the detailed version of the syntax, along with a description of each clause and parameter.

Layout Statements in Report Templates

Layout statements define how variables and literal text will appear on the report output:

- You can use the PAGE HEADER and PAGE FOOTER statements to place information at the top and bottom of each report page.
- You can nest HEADER, HEADER2, FOOTER, and PRINT statements within the braces section of the parent BLOCK statement to create titles and summary totals for the various *reporting sections* (parts of the report output).

Note: When nesting, be careful not to confuse the braces used in layout statements with the braces that encompass the nested statements within a parent BLOCK statement.

- You can include literal text to create labels and line drawing characters to enhance the appearance of the report.

The layout statements are as follows:

PAGE HEADER

Places information at the top of each report page. It is placed outside the BLOCK statement.

PAGE FOOTER

Places information at the bottom of each report page. It is placed outside the BLOCK statement.

HEADER

Places information at the top of each reporting section. It is placed inside the BLOCK statement.

HEADER2

Places continuation header information at the top of each succeeding page of a reporting section, if that reporting section extends over multiple pages. It is placed inside the BLOCK statement.

FOOTER

Places information at the bottom of each reporting section. It is placed inside the BLOCK statement.

PRINT

Places the data in a reporting section. It is placed inside the BLOCK statement.

You can also use the following predefined variables in layout statements:

- CT prints the current time
- CD prints the current date
- PG prints the page number

Data Fields

Specifies any variable in a layout statement that results in a piece of data when you generate the report. Use the following guidelines when placing fields in your report template:

- Enclose data fields in square brackets ([]).
- The field's square brackets define its print space on each line of output. This space is the number of characters delimited by the square brackets, including the brackets. If the output of a variable is longer than the print space, the output is truncated. To ensure that the field has enough print space, you can add trailing spaces between the variable name and the closing bracket. For example, these trailing spaces allow for contacts with long names:

```
[contact      ]
```

- For output that is less than one line, the field can be closed with a greater than right angle bracket (>). This extends the print space to the right margin. For example, the right angle bracket used in a HEADER statement allows the current date to print without being truncated:

```
[CD          >
```

Note: When the field is more than one line and the variable is flagged as MULTILINE, the right angle bracket (>) acts exactly the same as the right square bracket (]). If the print statement for a MULTILINE variable is closed with the right angle bracket (>), characters wrap on white space to stay within the field defined by the left bracket ([) and the right angle bracket (>). Also, if the variable is not MULTILINE, the right angle bracket (>) causes all the data to be displayed on the current line regardless of its length.

- A field in a layout statement can refer to a previously defined variable or a column name.
- To reference a variable or column name in another block statement, use the following syntax:

```
blockname::column | variable-name
```

Literal Text

Literal text allows you to include supplementary information in your report. It will appear on the report output exactly as specified in the template. To include literal text in a layout statement, place it on any line after the opening brace ({) and before the closing brace (}). Do not enclose it in quotes or square brackets.

In this example, “ACME Company” and “Page: ” are interpreted as literal text by the CA Service Desk Manager report program:

```
PAGE HEADER {  
    ACME Company    Page: [PG]  
}
```

More information:

[Report Template FOOTER Statements](#) (see page 384)

[Report Template PAGE FOOTER Statements](#) (see page 387)

[Report Template HEADER Statements](#) (see page 385)

[Report Template HEADER2 Statements](#) (see page 386)

[Report Template PAGE HEADER Statements](#) (see page 388)

[Report Template PRINT Statements](#) (see page 389)

Variable Expressions in Report Templates

Every value that you want to appear on the report output can be assigned to a variable. Variable expressions let you:

- Manipulate CA Service Desk Manager data
- Use functions to perform calculations on fetched values

The following example creates a variable named *desc* to reference the contents of the *chg_desc* field in the Change Order window. The MULTILINE flag allows the variable to print in its entirety over multiple lines:

```
desc = description MULTILINE;
```

The following example prints the description. The output will be as long as the length defined in the brackets. If you want a longer description to appear, increase the number of spaces in the brackets.

```
PRINT { [desc ] }
```

Example: Report Template

The following Affected Contact Report template shows how to create a report template. It produces a report that lists open change orders with the same affected contact:

```
PAGE HEADER {
    As Of: [CD>
    [CT>
}
PAGE FOOTER {
    Page: [PG>
}
BLOCK chg ("SELECT \
    chg_ref_num, description, priority, \
    status, category, assignee \
FROM Change_Request",
    "WHERE #Change_Request.status = 'OP' \
AND #Change_Request.requestor = #ca_contact.id \
AND #ca_contact.last_name = ? \
AND #ca_contact.first_name = ? \
AND #ca_contact.middle_name = ? ", $1, $2, $3)
{
    BLOCK st ("SELECT sym FROM Change_Status",
        "WHERE code = ? ", chg::status) {}
    BLOCK (strlen(category)) cat ("SELECT sym FROM Change_Category",
        "WHERE code = ? ", chg::category) {}

HEADER {
    OPEN CHANGE ORDERS WITH SAME REQUESTOR/FROM CONTACT
CHANGE ORDER Summary    Pri Status Category    Assignee
}
HEADER2 {
CHANGE ORDER Summary    Pri Status Category    Assignee
-----
}

    num = chg_ref_num;
    desc = description MULTILINE;
    pr = deref (priority);
    stat = st::sym;
    catgry = cat::sym;
    asgn = deref (assignee);

PRINT {
[num ][desc ][pr ][stat ][catgry ][asgn ]
}
}
```

Page Header

Specifies what to print on the top of each page of the report. CD and CT are predefined variables that give the current date and time. They will appear in the header on the top of each page. Each of these fields ends with an angle bracket, which allows the field to expand towards the right margin. Because “As Of:” is outside of a field and because it is on a line after the opening brace, it will appear as literal text on the report output.

```
PAGE HEADER {                As Of: [CD>
                               [CT>
}
```

Page Footer

Includes the page number with “Page: ” as literal text.

```
PAGE FOOTER {
                Page: [PG>
}
```

Note: Since PAGE HEADER and PAGE FOOTER statements produce global headers and footers, they are not included in a BLOCK statement.

Reporting Section

Creates a reporting section for the main BLOCK statement, along with its nested statements. A reporting section is usually only part of the data in the report, but this report has only one reporting section. The unique name of this block is chg.

The SELECT clause selects the columns to be included in the data for the report FROM three tables, but only where conditions specified by the WHERE clause are met.

The last three AND expressions in the WHERE clause contain question marks, which act as argument placeholders that take the values of the \$1, \$2, and \$3 arguments, in order. Thus \$1 is for ca_contact.last_name, \$2 is for ca_contact.first_name, and \$3 is for ca_contact.middle_name. The \$1, \$2, and \$3 arguments obtain the values of command line arguments.

```
BLOCK chg ("SELECT \
...",
"WHERE \
...\
AND #ca_contact.last_name = ? \
AND #ca_contact.first_name = ? \
AND #ca_contact.middle_name = ? ", $1, $2, $3)
```

Reporting Section Headers

Specifies that the opening brace starts the output program part of the BLOCK statement: its statements tell what to do with the data fetched by the SELECT and WHERE clauses. This example has nested HEADER and HEADER2 statements that will apply to this reporting section only. HEADER2 prints only if the report output is on multiple pages.

```
{
  ...
  HEADER {
    OPEN CHANGE ORDERS WITH SAME REQUESTOR/FROM CONTACT
    CHANGE ORDER Summary      Pri  Status Category Assignee
  }
  HEADER2 {
    CHANGE ORDER Summary      Pri  Status Category Assignee
    -----
  }
}
```

Variable Assignments

Specifies variable expressions that act on the data specified by the SELECT clauses. They assign variables to the values of columns and to the results of expressions. These variables match the fields in the PRINT statement that follows.

The MULTILINE flag on the *desc* variable causes them to print or display on multiple lines rather than being truncated. The *deref* function is used to return the string expression contained in the referenced columns.

```
num = chg_ref_num;
desc = description MULTILINE;
pr = deref (priority);
stat = st::sym;
catgry = cat::sym;
asgn = deref (assignee);
```

Printing

Contains the fields to be printed. This statement could have also included literal text of lines that could enhance the appearance of the report. The final ending brace matches the opening brace of the output program section of the BLOCK statement.

```
PRINT {
[num ][desc      ][pr] [stat] [catgry] [asgn      ]

}
}
```

More information:

[The Report Command](#) (see page 375)

How to Generate Reports

After you create the report template, you can generate the report by running the CA Service Desk Manager report program. The program can be executed from:

- The command line
- A CA Service Desk Manager menu option
- A script file

Note: If you are working on a UNIX server, you can include the report output redirector (rptuiDsp) parameter with the report command to display a dialog with options for sending the report to the screen, a file, or the printer.

More information:

[How to Display a Dialog \(UNIX Only\)](#) (see page 376)

The Report Command

To generate a report from the command line in UNIX, you must use the CA Service Desk Manager report command:

```
pdm_task report [-h][-e][-f][-F ffstring][-p pagelength] filename [command-line arguments]
```

Note: The report command is preceded by the pdm_task command, which sets necessary environment variables. If the report is designed to accept command line arguments, you must enter one for each argument in the report template.

On Windows, use the rpt_srv command:

```
rpt_srv report-title
```

The following example includes the three command line arguments (Smith, Jane, and L) needed for the Affected Contact Report described in the report template example earlier in this section:

```
pdm_task report /reports/myrpt.rpt Smith Jane L
```

If an argument is empty, you must use a null string. For example, if Jane Smith did not have a middle initial, the syntax would be:

```
pdm_task report /reports/myrpt.rpt Smith Jane ""  
rpt_srv \reports\myrpt.rpt Smith Jane L
```

How to Display a Dialog (UNIX Only)

You can include the report output redirector (rptuiDsp) parameter with the report command to display a dialog. The dialog displays the options to print the report to a file, display it in an Xterm window, or send it to the printer.

Example:

```
pdm_task rptuiDsp report /reports/myrpt.rpt Smith Jane L
```

This example adds the title “Inventory Report” to the dialog:

```
pdm_task rptuiDsp report /reports/myrpt.rpt Smith Jane L "title:Inventory Report"
```

Report Template Reference

You can use variable expressions, functions, and statements in a report template.

Report Template Variable Expressions

Variable expressions define the data to be printed or displayed. They are placed in a layout or block statement.

Syntax

Syntax refers to the rules governing the formation of statements in a programming language. The structure of this string is as follows:

```
variable-name = expression [flags]
```

Flags

Flags format the result of a variable expression. Use these flags to format text fields:

MULTILINE

Displays on multiple lines rather than truncating.

RIGHT

Right justifies.

Use these flags to format numeric fields:

BLANKZERO

Functions as null-value fields, which do not print a zero.

BOOL

Converts zero to no or non-zero to yes.

REAL

Displays as floating point (default is integer).

ZEROFILL

Shows leading or trailing zeros

Use these flags to format date and time fields:

DATE

Shows only date portion of date/time.

DAYS

Displays durations with days.

HOURS

Displays durations with hours.

MINUTES

Displays durations with minutes.

SECONDS

Displays durations with seconds.

TIME

Shows only time portion of date/time.

Example

```
desc = description MULTILINE
```

Remarks

Variable names must be unique within a BLOCK statement and must not duplicate any column in the SELECT clause for the block. The same variable name can be used in different BLOCK statements but it cannot be repeated within a BLOCK statement.

Follow these syntax rules when including expressions in your report template:

- Use any valid C expression.
- Do not enclose variable or column names in quotes.
- Enclose string constants in single or double quotes.
- You can refer to a nested block, but only if it contains exactly one row.

- To include a column name that is the same as a keyword, precede the column name with a backslash (\). For example, ALIAS is a keyword and \alias is a column name.
- Use the dollar sign (\$) to reference environment variables, such as \$name, and to reference command line arguments, such as \$n, where *n* is the position of the argument on the command line.
- To specify the number of command line arguments, use \$#. For example, the following expression means that if the number of command line arguments is greater than one, use the additional argument as an argument; otherwise, set the value of the argument to an empty string. The report template itself is considered a command line argument. Therefore, the number of arguments is at least one.

```
$# > 1 ? $1 : "
```

- Use ## to concatenate two strings, for example:

```
title = "This is the " ## "first line. "
```

```
long_name = fn
```

```
irst_name ## last_name
```

- The following casts are supported:
 - (number)
 - (string)
 - (date_time)
 - (duration)
- To reference a variable or column name in another block, precede the name with its block name and two colons. For example:

```
blockname::column | variable-name
```

Report Template Functions

The following functions can be used in your report template:

is_null (*expr*)

This function returns true if the expression is null.

```
false = 0
```

```
true = is_null (false)
```

sqrt (*expr*)

This function calculates the square root of the expression.

```
nine = 9
```

```
three = sqrt (nine)
```

pow (*expr1*, *expr2*)

This function raises *expr1* to the power *expr2*.

```
two = 2
```

```
three = 3
```

```
eight = pow (two,three)
```

log (*expr*, *expr*)

This function calculates the natural log of the expression.

```
ten = 10
```

```
result = log (ten)
```

catname (*expr*, *expr*, *expr*)

This concatenates three strings representing a contact name into a string with commas, according to rules in the field format file.

```
last = "Murphy"
```

```
first = "Fred"
```

```
middle = "P"
```

```
contact_name=catname (last, first, middle)
```

strlen (*string*)

This function returns the length of the string.

```
buffer = "A thirty character long string"
```

```
thirty = strlen(buffer)
```

strindex (string, pattern [, start_index])

This function returns the index of the first pattern match, or the next pattern match after the start_index, in the string. Returns -1 if there is no match.

```
buffer = "A thirty character long string"
zero = strindex(buffer, "[A-Z] ")
two = strindex(buffer, "[a-z] ")
```

substr (string, pattern [, length])

This function returns the portion of the string after the first pattern match. If length is defined, it limits the length of the output string. Returns a string of zero length, if there is no match.

```
buffer = "A thirty character long string"
last_word = substr(buffer, "[a-z]*$ ")
first_capital_letter = substr(buffer, "[A-Z] ",
1)
```

substr (string, index [, length])

This function returns the portion of the string after the index. Its length is defined and limits the length of the output string. Returns a string of zero length, if there is no match.

```
buffer = "Summary: The network card displays a
code of ..."
summary = substr(buffer, 9)
30_char_summary = strindex(buffer, 9, 30)
```

The remaining functions (pseudofunctions) perform on a block of data rather than on variable expressions. These functions are usually placed in a BLOCK statement to get information about a nested BLOCK statement's data.

count (*block-name*)

Returns the number of rows in the block specified in the BLOCK statement. The block-name must be a simple string.

```
BLOCK sample ("SELECT id FROM Contact") {  
  entries = count (sample)  
}
```

sum (*block-name, expr*)

Executes the expression for each row of the specified block and sums the result.

```
BLOCK sample ("SELECT actual_cost, est_cost FROM Change_Request") {  
  difference = sum (sample, est_cost-actual_cost)  
}
```

average (*block-name, expr*)

Executes the expression for each row of the block and returns the average of the result.

```
BLOCK sample ("SELECT actual_cost, est_cost FROM Change_Request"){  
  avg_difference = average (sample, est_cost-actual_cost)  
}
```

prev (*expr*)

Returns the previous value of the expression. This function should be used with caution so its value does not overwrite the latest value by accident.

downtime (*sla_schedule, expr1, expr2 [, delay-block, expr, expr]*)

Invokes an SLA downtime calculation. The first argument must be a string that identifies a workshift. The other arguments are start and end times:

expr1 is the start date/time of the event

expr2 is the end date/time of the event

In this example, the wrkshft BLOCK fetches the work shift schedule, the evt_dly BLOCK statement fetches the delays and the downtime function uses these records to calculate the downtime.

```
BLOCK attev ("SELECT start_time, fire_time, event_tmpl, obj_id FROM
Attached_Events")

{
BLOCK evt ("SELECT persid, sym, work_shift FROM Events ", "WHERE persid = ?", attev::event_tmpl) {}
BLOCK wrkshft ("SELECT sched FROM Bop_Workshift", "WHERE persid = ?", evt::work_shift) {}

BLOCK evt_dly ("SELECT start_time, stop_time FROM Event_Delay", "WHERE obj_id = ?", attev::obj_id) {}
total_downtime = downtime(wrkshft::sched,
attev::start_time, attev::fire_time,
evt_dly,
evt_dly::start_time, evt_dly::stop_time);
}
```

deref (column-name)

Returns the string representation of the pointer by performing an automatic lookup in the appropriate table.

```
BLOCK chg ("SELECT organization FROM Change_Request") {
org = deref (organization)
}
```

Because this pseudofunction involves lookups, it is valid only if it is the only thing in the expression. For example, this is valid:

```
model = deref (nr_model)
```

This is not valid:

```
model = "model" ## deref (nr_model)
```

Note: Forward references to variables or blocks are not allowed.

Report Template BLOCK Statements

Block statements define the database tables from which data will be fetched, and can include actions to perform on the fetched data.

Syntax

Syntax refers to the rules governing the formation of statements in a programming language. The structure of this string for BLOCK is as follows:

```
BLOCK blockname (
  "SELECT [ALIAS.] field_name [, field_name ...]
  FROM table_name [, table_name ...] "
  [,"WHERE where_clause" ][, arguments ,])
  [SORT "sort clause" ]
{
output program statements
}
```

Parameters

blockname

Identifies the block. Each *blockname* must be unique.

SELECT clause

Follows the *blockname* and is delimited by double quotes. Lists the columns to be fetched, followed by the keyword FROM, followed by the tables from which the columns are to be fetched. It is required. Here is an example with three tables specified:

```
"SELECT open_date, chg_ref_num \
last_name, first_name \
FROM Change_Request, \
ca_contact"
```

You cannot include an SQL alias, such as:

```
"SELECT open_date As OpenDate"
```

WHERE clause

(Optional) Follows the SELECT clause and further qualifies the information selected. It may be a string constant or an expression evaluating to a string. If the WHERE clause is an empty string, all records are returned. WHERE clauses can contain replacement arguments (which refer to variables or command line arguments) using the syntax of a question mark (?). The following WHERE clause could follow the previous SELECT clause:

```
"WHERE #Change_Request.open_date >= ? \
AND #Change_Request.active_flag = 1 \
AND #ca_contact.last_name = ? ", $1"
```

Note: The WHERE clause must be separate from the SELECT clause because the WHERE clause can be an expression evaluating to a string, whereas the SELECT clause is exclusively a string constant. This gives you more flexibility and data manipulation capabilities in producing your report.

SORT clause

(Optional) Follows the SELECT and WHERE clauses and sorts the fetched rows of data. The SORT clause is formatted like the SQL ORDER BY clause. Here is an example:

```
SORT "open_date"
```

Output program statements

Controls execution of the report. Before the data query, the HEADER statement, if included, prints the header test for the block. The data query then runs. If data is returned, each statement executes in the order written, with one exception. Block functions, like sum and average, behave as though they were at the end of the output program. In fact, their values are not stable until execution begins on the next data record.

Important! The output program depends on the success of the data query. If no data is returned from the query, then, except for the HEADER statement, the output program will not execute.

Example

This BLOCK statement assumes that an argument will be passed that holds an integer equal to the change order priority. The WHERE clause first checks the number of arguments passed (\$#). If one is present, it is used to evaluate the expression to produce the WHERE clause; otherwise a null WHERE clause is substituted (" ").

```
BLOCK chg ("SELECT priority FROM Change_Request",  
$# > 1 ? "WHERE priority = " ## $1 : "") {}
```

Example

```
FOOTER {  
  Summary Information:  
    Total Failures: [Fail_count >  
    Total Downtime: [Downtime >  
}
```

Remarks

HEADER, HEADER2, FOOTER, PRINT and variable expressions can be placed in the braces. Any statement will be executed for each row selected.

Note: PAGE HEADER and PAGE FOOTER statements cannot be placed in a BLOCK statement.

Report Template FOOTER Statements

This layout statement places information at the bottom of a reporting section.

Syntax

Syntax refers to the rules governing the formation of statements in a programming language. The structure of this string for FOOTER is as follows:

```
FOOTER {parameters}
```


Parameters

The parameters are as follows:

CD

A predefined variable used to display the current date.

CT

A predefined variable used to display the current time.

PG

A predefined variable used to display the current page number.

column | variable-name

This field can be a variable from an earlier variable expression or a reference to a column in the SQL clause of a BLOCK statement.

literal-text

Any text that is not a predefined variable or a column or variable name is interpreted as literal text. Literal text that you include in the FOOTER statement appears in the exact horizontal location where you enter it.

Remarks

FOOTER statements are printed at the bottom of a reporting section. A typical use might be to present summary information or statistics. You can include a FOOTER statement in a BLOCK statement.

A field's content occupies the exact space delineated by the square brackets. Any excess characters are truncated. However, you can close a field with an angle bracket (>) to permit its content to expand in its entirety towards the right margin.

To reference a variable or column name in another BLOCK statement, use the following syntax:

blockname::column | variable-name

Report Template HEADER Statements

This layout statement places information at the top of a reporting section.

Syntax

Syntax refers to the rules governing the formation of statements in a programming language. The structure of this string for HEADER is as follows:

HEADER {parameters}

Parameters

For a list and explanation of the valid parameters for this statement, see [Report Template PAGE HEADER Statements](#) (see page 388).

Example

```
HEADER {  
    Contact Summary Report  
    Contact Name Contact Alias Organization  
}
```

Remarks

HEADERS are printed at the beginning of a reporting section and can be included in a BLOCK statement. HEADERS are typically used to present section and/or column headings.

A field's content occupies the exact space delineated by the square brackets. Any excess characters are truncated. However, you can close a field with an angle bracket (>) to permit its content to expand in its entirety towards the right margin.

Note: If the print statement for a MULTILINE variable is closed with the right angle bracket (>), characters wrap on white space to stay within the field defined by the left bracket (|) and the right angle bracket (>). Also, if the variable is not MULTILINE, the right angle bracket (>) causes all the data to be displayed on the current line regardless of its length.

To reference a variable or column name in another BLOCK statement, use the following syntax:

blockname::column | variable-name

Report Template HEADER2 Statements

This layout statement places continuation HEADER information at the top of each succeeding page of a reporting section, if that reporting section extends over multiple pages.

Syntax

Syntax refers to the rules governing the formation of statements in a programming language. The structure of this string for HEADER2 is as follows:

```
HEADER2 {parameters}
```

Parameters

For a list and explanation of the valid parameters for this statement, see [Report Template HEADER2 Statements](#) (see page 386).

Example

```
HEADER2 {  
    Contact Summary Report (continued)  
    Contact Name  Contact Alias  Organization  
}
```

Remarks

A HEADER2 statement can be included in a BLOCK statement.

A field's content occupies the exact space delineated by the square brackets. Any excess characters are truncated. However, you can close a field with an angle bracket (>) to permit its content to expand in its entirety towards the right margin.

To reference a variable or column name in another BLOCK statement, use the following syntax:

blockname::column | variable-name

Report Template PAGE FOOTER Statements

This layout statement places information at the bottom of each report page.

Syntax

```
PAGE FOOTER {parameters}
```

Parameters

With the exception that you cannot use column and variable names, the parameters for this statement are the same as those for FOOTER. For a list and explanation of the valid parameters for this statement, see [Report Template PAGE FOOTER Statements](#) (see page 387).

Example

```
PAGE FOOTER {  
    Page Number: [PG>  
}
```

Remarks

A field's content occupies the exact space delineated by the square brackets. Any excess characters are truncated. However, you can close a field with an angle bracket (>) to permit its content to expand in its entirety towards the right margin.

To reference a variable or column name in another BLOCK statement, use the following syntax:

blockname::column | variable-name

Report Template PAGE HEADER Statements

This layout statement places information at the top of each report page.

Syntax

Syntax refers to the rules governing the formation of statements in a programming language. The structure of this string for PAGE HEADER is as follows:

PAGE HEADER {*parameters*}

Parameters

With the exception that you cannot use column and variable names, the parameters for this statement are the same as those for FOOTER. For a list and explanation of the valid parameters for this statement, see [Report Template PAGE FOOTER Statements](#) (see page 387).

Example

```
PAGE HEADER {  
  Date of Report: [CD>  
  Time of Report: [CT>  
}
```

Remarks

PAGE HEADERS are printed at the top of every report page. They can be defined at any point within the report template file, but they cannot be included within a BLOCK statement.

A field's content occupies the exact space delineated by the square brackets. Any excess characters are truncated. However, you can close a field with an angle bracket (>) to permit its content to expand in its entirety towards the right margin.

To reference a variable or column name in another BLOCK statement, use the following syntax:

blockname::column | variable-name

Report Template PRINT Statements

This layout statement places data in a reporting section.

Syntax

Syntax refers to the rules governing the formation of statements in a programming language. The structure of this string for PRINT is as follows:

PRINT {*parameters*}

Parameters

Refer FOOTER for a list and explanation of the valid parameters for this statement.

Example

```
PRINT {  
[num ] [desc      ] [pr] [stat] [catgry] [asgn      ]  
}
```

Remarks

Place PRINT where you want the data for a reporting section to appear in the report. You can include a PRINT statement in a BLOCK statement.

A field's content occupies the exact space delineated by the square brackets. Any excess characters are truncated. However, you can close a field with an angle bracket (>) to permit its content to expand in its entirety towards the right margin.

Note: If the print statement for a MULTILINE variable is closed with the right angle bracket (>), characters wrap on white space to stay within the field defined by the left bracket (|) and the right angle bracket (>). Also, if the variable is not MULTILINE, the right angle bracket (>) causes all the data to be displayed on the current line regardless of its length.

To reference a variable or column name in another BLOCK statement, use the following syntax:

blockname::column | variable-name

Customize Crystal Reports

Before you can display any of these reports, the following conditions apply:

- You must make the Crystal reports available to the Crystal Report Selector by copying them to the Crystal directory: `$NX_ROOT/bopcfg/rpt`.
- Your database client must be up and running, with connectivity established to the database server running on the same or another computer. If you are using a CA Service Desk Manager Client to run your Crystal or Access reports you need to have installed a database client for the specific database and have established connectivity with the database server in order to run these reports.

After creating any custom Crystal reports, perform the following:

1. Copy custom Crystal reports to the following crystal directory:
`$NX_ROOT/bopcfg/rpt`
2. Add the file names of custom Crystal reports to the following configuration file:
`crystal.cfg`

You can then access the Crystal reports by clicking Start on the taskbar, and then choosing Reporting, Service Desk Reporting (Crystal Reports) from the CA Service Desk Manager menu (accessible from the Programs menu). The Service Desk Reporting (Crystal) window appears.

Important! CA Service Desk Manager clients cannot be upgraded. Therefore, if you create and use Crystal reports on the CA Service Desk Manager Server and you plan to upgrade your version of CA Service Desk Manager, you need to copy all custom reports to a different location so you will not lose them. Following the upgrade, copy the reports back to the `$NX_ROOT/bopcfg/rpt` Crystal directory and modify the `crystal.config` file to make them accessible from the Report Selector.

Chapter 7: Populating CA CMDB

This section contains the following topics:

[Database Population](#) (see page 391)

[How GRLoader Populates the Database](#) (see page 391)

[Federation Adapter Data Components](#) (see page 392)

[Data Population Glossary](#) (see page 392)

[Create a Data Source Name](#) (see page 393)

[How to Import Data Using ADT](#) (see page 394)

[Federation Adapters](#) (see page 400)

Database Population

Populating CA CMDB with the configuration items and relationships in your IT infrastructure is a part of using the application efficiently. You can populate CA CMDB with data manually using the built-in Configuration Item Editor or by importing items from other asset management tools.

Note: For information about the Configuration Item Editor, see the *Online Help*.

How GRLoader Populates the Database

Populating the database by loading configuration items (CIs) and relationships is a multistep process as follows:

1. Input data containing information about CIs and their relationships is converted to XML.
2. The CA CMDB GRLoader program uses the XML data as input.
3. GRLoader loads the data into the database.

Note: For information about GRLoader, see the *CA CMDB Technical Reference Guide*.

4. CA CMDB provides ADT to convert the data from its native format into the correct XML format. In addition, CA CMDB includes several Federation Adapters that use ADT facilities.

More information:

[Start ADT Server and Run the Script](#) (see page 398)

[Use GRLoader to Import the Data](#) (see page 399)

Federation Adapter Data Components

A Federation Adapter consists of a data definition component and a data mapping component.

Data definition

Describes the metadata for the input and output data. Metadata specifies where the input or output data is located, how to access it, and what fields it contains.

Data mapping

Maps the fields in the input fields to the XML output fields. Output from the Federation Adapter is an XML document located on your storage media.

Input for Federation Adapters

You can use an ODBC data source, an XML file, or a flat file as the source for input to the Federation Adapters. For example, use the following formats for input data sources:

- Oracle databases
- Ingres databases
- Microsoft Access databases
- Excel Spreadsheets
- DB2 MVS databases
- DBASE databases
- AS/400 databases
- UDB databases
- CSV (comma separated value) files
- Text files

Note: The field names in the XML document do not match exactly the field names in the CA CMDDB object model. You can map the fields in the input data to fields in the output data, and then transform this data as necessary. ADT provides a mapping capability that lets you move the data from the source to the target.

Data Population Glossary

ADT program

An *ADT program* is a visual representation of the data mappings and relationships between tables.

ADT script

An *ADT script* is a text representation of an ADT Program.

Compiled ADT script

A *compiled ADT script* is an executable version of an ADT script, stored in the IDB tables within the MDB.

Graphical Mapper

The *Graphical Mapper* is an ADT component which is primarily concerned with how to move data from the input table to the output table. The Graphical Mapper creates a script which the script manager reads.

GRLoader

GRLoader is a program which reads in CI and relation items from an XML document and loads the data into the MDB tables.

IDB

IDB is the ADT "internal database" set of tables within the MDB database.

MDB

MDB is a database which contains all CA data across all products.

Profile

A *profile* is specified in the Graphical Mapper, for each input or output table. Consult the appropriate ADT Interface guide for details on what information you enter into the profile.

Script Manager

Script Manager is an ADT component responsible for managing profiles and scripts.

Create a Data Source Name

You can create a Database Source Name (DSN) that contains the information about a specific database for the ODBC driver to use the ADT Generic ODBC data source.

To create a database source name

1. Select Settings, Control Panel, Administrative Tools, ODBC Data Sources from the Start menu.

The ODBC Data Source Administrator dialog appears.

2. Click Add, and select the driver that supports the source of your data on the System DSN tab.
3. Click Finish.

A dialog appears that lets you specify a data source name and other setup information. The dialog appearance depends on the driver you selected.

4. Complete the fields as appropriate for your driver and click OK.
5. For a SQL Server 2005 database, select the SQL Native Client, and click Finish.
6. Continue following the on-screen instructions.
7. Verify that Perform Translation for Character Data is selected and click Finish.

Note: Typically, no changes are necessary on this dialog.

8. Click Test Data Source to test the database connectivity.

When the test has completed successfully, click OK.

The data source name is created.

How to Import Data Using ADT

The following process describes how you use ADT to populate configuration items (CIs) in CA CMDB.

1. Open the ADT Mapper.
2. Set up the input table.
3. Set up the target data file.
4. Generate the script.
5. Set up profiles using the script manager.
6. Start the ADT server and run the script.
7. Use GRLoader to import the data.

The source of the CI information can be a Microsoft Excel spreadsheet, such as one located in the following directory:

`<root>\cmdb\data\federationAdapters\cidata.xls`

`<root>` specifies the default location:

- `c:\program files\ca\CA CMDB` for a standard installation
- `c:\program files\ca\servicedesk` for an integrated installation

Open the ADT Mapper

You use the ADT Mapper to import CI data and locate an existing program that meets the importing requirements.

To open the ADT Mapper

1. Select Programs, CA, Advantage Data Transformer, Mapper from the Start menu.

The Advantage Data Transformer Mapper starts. The Connect to Metadata Store page appears.

2. Complete the following fields and click OK:

Select IDB data source name

Specifies the data source name (DSN) that was created during the ADT installation.

Enter user id

Specifies the ADT Administrator user ID, infopump.

Enter password

Specifies the password you specified during the ADT installation.

The Browse Metadata Store page appears.

3. Double-click the program file that contains the program you want to run. In this case, select load_ci_from_xls. This file is the adapter for loading CI data from an Excel spreadsheet.

A page appears with panels that let you move data from the input table (left) to the output table (right).

Important! Data moves from the input table (left) to the output table (right) on a field by field basis. The lines connecting the two tables represent the field mappings between the tables.

4. (Optional) Right-click the output table title bar and select “filter columns” to select only those fields actively mapped by the program.

A simplified list appears.

The left panel lists the items in the input source file. The right panel lists the output file items. The output for all Federation Adapters is an XML file. This file is used as input to the CA CMDB GRLoader program.

5. (Optional) Right-click the left title bar and select Target Table to convert a table from an input source to an output source. If Target Table is selected, the table is defined as an output table.

Important! All field mappings are deleted when you converting a table to or from a target table. When you create a table, identify it as a target table or not a target table, so you do not lose any work.

Set Up the Input Table

Set up and verify the definition of the input table. The Profile associated with the Table Properties contains the definitions necessary to access data. These definitions specify the type of data, the location of the data, and security parameters.

To set up the input table

1. On the ADT Mapper page, right-click the left panel and select Properties.

The Properties for the input source file display.

2. Click the Profile tab.

The Select Profile page appears.

3. Specify or select a profile name. You can click the ellipse (...) to the right of the profile name to select a profile.

The profile specifications appear.

4. Complete the page fields.

Important! Create an ODBC DSN and change the profile before setting up the input table. Create additional profiles if needed. In an ODBC environment, you need one DSN and one profile for each database.

5. Click OK.

The Select Profile page closes.

6. Click OK.

The Table Properties page closes.

The input table is set up.

More information:

[Create a Data Source Name](#) (see page 393)

Set Up the Target Data File

Set up and verify the target data file (an XML file).

To set up the target data file

1. On the ADT Mapper page, right-click the title bar of the right panel, labeled All, and select Properties.

The properties page appears.

2. Click the Profile tab.

The properties page displays a profile suitable for XML output.

3. Click the File Information tab.

The target location for the XML output file is specified in the File name field.

The Federation Adapters creates the following default location for XML files:

```
<root>\cmdb\data\userdata
```

4. (Optional) Override the default location. Select Specify the Instance Specific File Name, click Browse, and navigate to the location where you want to save the target data file.
5. Click OK.

The target data file is set up.

Set Up Profiles Using the Script Manager

The Script Manager component of ADT has the following roles:

- Manages ADT scripts and profiles
- Changes or creates Profiles from within Script Manager
- Executes scripts from the Script Manager environment

Important! The profile editor that you use to set up profiles displays the same tabs and the same input fields, regardless of interface type. Only certain input fields are valid for some interfaces. To determine the appropriate fields for each interface type, see the appropriate Advantage Data Transformer Interface Guide. Entering configuration information for an input field that does not apply to an adapter can cause the script to produce errors when executed. For example, for profiles with the Generic ODBC interface type, set the Select Server field on the Server/Database tab to the ODBC DSN. However, do not enter any information into the Select Database or Enter Connection/Form Information fields.

To set up profiles using the Script Manager

1. From the Start menu, select Programs, CA, Advantage, Data Transformer, Script Manager.

The Script Manager appears.

2. Double-click CMDB_FederationAdapters.

A page shows the ADT profiles that are distributed with CA CMDB.

3. Double-click cmdb_profile_xls and click the Interface tab.

The Profile Editor appears.

4. Select Generic ODBC interface instead of the database-specific interface whenever possible.
5. Click the Server/Database tab and verify that the Select server field is set to the name of the ODBC connection defined in the Profile of the input table. If the ODBC DSN does not exist, create one.

The Profile setup is complete.

More information:

[Create a Data Source Name](#) (see page 393)

Generate the Script

Before running a Federation Adapter or any ADT program, generate a script. Scripts and their associated profiles are the basis of the visual programs created in the mapper.

To generate a script

1. On the ADT Mapper page, right-click anywhere in the space between the input and output files, and select Generate Script.

Note: Generating a script from the ADT Mapper does not automatically compile the script. Compile the script using the ADT Script Manager.

2. Watch for error messages displayed in the lower right-hand corner of the page, in the output page.

More information:

[Start ADT Server and Run the Script](#) (see page 398)

Start ADT Server and Run the Script

You start the ADT Server and run the import script to create an XML document.

Note: To install the ADT Server as a Windows service during installation and configuration, begin with Step 3.

To start the ADT server and run the import script

1. From the Start menu, select Programs, CA, Advantage, Data Transformer, Server.

The server starts and prompts you for the user ID and password.

2. Enter the user ID and password you created in Configure ADT.
The ADT Mapper page appears.
3. From the ADT Mapper page, select Tools, Script Manager.
The ADT Script Manager starts.
4. Right-click the script that you generated in Set Up the Input Table and select Compile.
5. When the script is compiled, right-click the script again and select Run.
The script runs and the XML document is created.
6. Use Internet Explorer or Notepad to open the XML file and verify the content.
Note: Refer to Set Up the Target Data File for the name and location of the XML file.

Use GRLoader to Import the Data

Import data by using the GRLoader program provided with CA Service Desk Manager, which creates CIs based on the data in the XML file that you created in Start ADT Server and Run the Script).

To use GRLoader to import data

1. From the Start menu, select Run.
2. Type cmd.
A DOS command window appears.
3. Enter the following command:

```
Grloader -u <username> -p <password> -s http://<cmdb_servername>:8080  
-i <xml_document>
```

GRLoader creates CIs using the data in the XML file. If errors are found during this process, an error file is created, listing the CIs that could not be imported and the reason.

GRLoader import completes.
4. Start CA CMDB and verify that the CI data has been correctly populated.
5. Start the CMDB Visualizer to verify that relationship data has been populated correctly.
The data is imported and verified.

Note: For more information about GRLoader, see the *CA CMDB Technical Reference Guide*.

Federation Adapters

CA CMDB provides the following Federation Adapters:

Load_ci_from_xls

Loads any family of CIs from an Excel spreadsheet.

Load_generic_template

Loads any family of CIs from any data source.

Load_hardware_server

Loads hardware server CIs into the MDB.

Load_relations_from_xls

Loads a table of CI relationships from an Excel spreadsheet.

Load_SMS_from_view

Loads Microsoft Systems Management Server (SMS) data from a view created at the source database.

Load_UAM_from_view

Loads asset management data from a view created at the source MDB.

Prerequisites for Running a Federation Adapter

Verify that the following parameter values are correctly set before executing a CA Service Desk Manager Federation Adapter:

- An ODBC DSN exists for each database or spreadsheet.
- A profile exists for each database/spreadsheet. For a Generic ODBC interface, specify the following:
 - Server: DSN
 - Login information (if any)
 - Interface: Generic ODBC
- For each database table, the correct profile is specified on the Properties dialog.
- For each XML output file, specify the following on the Properties dialog.
 - The correct XML profile on the Profile tab.
 - The correct output file name on the File
- The script is generated from within the ADT Mapper.

- The script is compiled from within the ADT Script Manager.
- The ADT Server is started.

Note: For other interfaces, other parameters can apply. Consult the ADT Interface documentation for details.

CA CMDB Federation Adapter Checklist

Before using ADT to execute a CA CMDB Federation Adapter by, verify the following:

- The Program has “Display output on console” property selected.
- For spreadsheets only, verify the following:
 - All spreadsheet data is alphabetic
 - Dates are formatted as strings (left justified)
 - Numbers are formatted as character strings (left justified)
 - Column names do not have embedded spaces or use keywords like DESC
- ODBC DSN exists for each database or spreadsheet.
- For each input database table, properties/profile specify the correct profile name.
- A profile exists for each database/spreadsheet. For a Generic ODBC interface, specify the following profile:
 - Server: DSN
 - Logon information (if any)
 - Interface: Generic ODBC

Note: For instructions on using other database interfaces, consult the ADT interface guides.

- For each XML output file, verify the following:
 - Properties/profile specify the XML profile
 - Properties/file information specifies the correct output file name
- From within the Graphical Mapper, the script is generated and compiled.
- The ADT Server is started.
- The CA Service Desk Manager Server service is started.

Do the following after completing the previous items:

- Run the script from the script manager (refresh the display to see newly generated programs).
- See the script start, rows read, and script stop message in the Server console.
- Review the console log messages for errors.
- Open the XML file using an XML Editor or Internet Explorer.
- Run GRLoader to import the CIs and Relationships.

Family and Class Assignments

Apply a classification scheme to each CI; this scheme involves assigning each CI the following attributes:

- Family—A collection of configuration items having similar attributes
- Class—A subset of configuration items within a family

You can create assignments in the following ways:

- Populate manufacturer data identifying family and class for each hardware asset.
- Include nonblank values for family and class in the respective columns in the input file.

The GRLoader does not import any CI being loaded which cannot resolve family and class to an existing family and class.

Source Data Mappings

CI information can be imported from multiple types of data sources. An example data source is provided with CA Service Desk Manager in the cidata.xls spreadsheet. This spreadsheet contains mappings for each of the family-specific input tables listed in the file.

Load CIs from an Excel Spreadsheet

The ability to load CI information from a spreadsheet provides the following:

- A good learning tool for your first use of ADT
- An easy way to load data that was maintained on a spreadsheet on some ones desk.

You can use this adapter to import any data for any family of CI; it can be used as a general framework for importing CIs of any family. Unless you are familiar with defining ODBC tables in an Excel spreadsheet, copy cidata.xls and make all your modifications to this file. If you want to start with a new or blank spreadsheet, perform the necessary insert/name operations in the spreadsheet to define the tables.

To load CIs by using an Excel spreadsheet

1. Navigate to the following location and copy the blank spreadsheet that is included with CA CMDB:

```
<root>\cmdb\data\federationAdapters\cidata.xls
```

2. Add or remove rows of data to edit the cidata.xls spreadsheet. If you want to change the order of the columns or remove columns, use the ADT Mapper to scan the new table.

Important! All cells in the spreadsheet must be formatted as Text. Failure to format all cells as Text can result in incorrect data conversion and corrupted data.

The cidata.xls spreadsheet has input areas defined for every CI family, and definitions for the following additional tables:

Common_attributes

Lists common attributes which every family includes; the intersection of attributes every family includes.

All_attributes

Lists attributes which one or more families includes; the union of attributes every family includes. When using this adapter, insert rows into the all_attributes data area that is defined within the spreadsheet. Only data in the all_attributes data section are loaded; other data is ignored.

Relation

Enables the definition of relationships between CIs.

Note: Predefine some field values so they are accepted as valid input. For example, define Contacts and Locations before a CI can reference them.

3. Save and close the spreadsheet.
4. Create an ODBC DSN to the spreadsheet.
5. Change the cmdb_profile_xls profile to refer to the DSN that exists for the CA CMDB Federation Adapter. Specify the DSN name that you created previously in the server tab in the cmdb_profile_xls profile.

6. Review the output table definition in the output table properties to verify the output table destination.
7. Generate and compile the script.
8. Run the compiled script to transform the Excel spreadsheet into an XML document.
9. Use the GRLoader utility to read in the XML document and load in the data.

Load Hardware Servers from an Excel Spreadsheet

The Federation Adapter is a specialized version of the `load_ci_from_xls` adapter. Only those fields associated with a Hardware Server are contained in the program. No other fields are mapped.

Instead of using the `common_attributes` table in the spreadsheet, the `hardware_server` table is used as input. Instead of using the `All` table for output, you use the `HardwareServer` table definition.

Instead of inserting CI data into the `all_attributes` table in the spreadsheet, rows are inserted only in the `hardware_server` section.

Note: The adapter only loads data in the `hardware_server` table. All other data is ignored.

All other instructions for [loading CIs from an Excel spreadsheet](#) (see page 402) are followed, changing the names of the input tables and output tables.

We recommend, for simplicity, that the input data have the `class` and `family` columns populated. Consider adding the `set_hardwareServer` transformation, found on the transformation tab, to the mapping program.

Load Relations from an Excel Spreadsheet

You use the Federation Adapter to load relationship information into the MDB from a spreadsheet. The CIs association with each other must exist before the relationship can be formed between them.

Only input that is contained in the `relation` table in the `cidata.xls` spreadsheet is imported.

Note: The GRLoader properly sequences the MDB updates so that all CIs in a batch are defined before relationships are added. Take special care to verify that the CIs in the relationship are uniquely described in the spreadsheet. Refer to the reconciliation section of GRLoader for details. Follow the instructions that were used to import `load_cidata_from_xls`, inserting the data row into the relationship area.

How to Load Microsoft SMS Data

The Federation Adapter is used to load Microsoft Systems Management Server (SMS) data into the MDB from a SQL Server database that contains SMS data. A view is created on the source database to select data from several tables and simplify the data mapping process. The SMS data is located in a different database than the one containing the MDB.

Note: Only hardware servers are imported using this adapter. If you want to import other data, create a custom adapter.

To create the view, the following sample job can be modified as necessary and executed:

```
<root>\cmdb\data\federationAdapters\SQL_SMS_View.sql
```

After the view that contains SMS data has been created, complete the following:

1. Create an ODBC DSN to the view. If you have modified the structure of the view, create custom programs to map data.
2. Change the cmdb_profile_xls profile to refer to the DSN that you created.
3. Review the output table definition Hardware Server in the Mapper to verify the output table destination.
4. Run the script to read the SMS database and transform it into an XML document.
5. Use the GRLoader utility to read the XML document and import the data.

More information:

[Custom Federation Adapters](#) (see page 407)

How to Load CA APM Data

The primary input to the CA Asset Portfolio Management (CA APM) Loader program is a table or view that contains an extract of the CA APM import data. This data is contained in the CMDb_Export_Asset_Data database table or view. In some cases, the import data into the MDB is located in the same database as the target. However, in many circumstances, the import data resides in a different database, for example, when importing data across subsidiaries. In either case, create a view on the same database as the source data.

The CA APM data view does not contain the class and family attributes, and can come from a system with a different classification scheme. For example, if the source of the asset data is a different company, a different classification system can be in effect.

The CA_MODEL_DEF table contains a list of models, which are matched against the data in CMDB_Export_Asset_Data. If there is a match, the family and class from the model are assigned to the asset being imported. If there is no match, define a new model for the asset. Consider either copying entries from the source CA_MODEL_DEF table to the target MDB, or updating the CA_MODEL_DEF table with entries for all new hardware makes and model numbers.

The SQL script to create this view is in the following location:

```
<root>\cldb\data\federationAdapters\SQL_UAM_View.sql
```

After you create the view, do the following:

1. Create an ODBC DSN to the spreadsheet.
2. Use the ADT Mapper/Scanner to scan in the definition of the common_attributes table contained in the spreadsheet.
3. Change the cldb_profile_xls profile to refer to the DSN previously created.
4. Review the output table definition to verify the output table destination.
5. Generate the script using the ADT Mapper, then compile it in the Script Manager.
6. Run the script to transform the Excel spreadsheet into an XML document.
7. Use the GRLoader utility to read the XML document and load the data.

How to Use the Generic Template

The generic template is provided as a starting point for creating Federation Adapters. By itself, this template is not executable. You can use this program to create any CI or any relationship by adding in the appropriate data sources and field mappings to import CIs from several families at the same time.

To use the generic template to create your own Federation Adapter, do the following:

1. Create an ODBC DSN to the database containing the source CI table.
2. Analyze the source input table, using the ADT Mapper and Scanner.
3. Open the generic template in the ADT Mapper.
After the source table is scanned, it can be included in an ADT program.
4. Click the Data tab and navigate to the table that was scanned. Drag the icon representing this table onto the program palette.
5. Create a mapping of the field relationships between the input table you scanned and the output table which the template places there by the template. Drag each field from the input table to the output table.
6. Create a profile for the input table.
7. Review the profile for the output table.

8. Review the output table definition to verify the output table destination.
9. Run the script to create the XML document from the input table.
10. Use the GRLoader utility to read the XML document and load the data.

Custom Federation Adapters

When your input data does not exactly match the format of the built-in CA CMDB Federation Adapters, use ADT to create a custom Federation Adapter. The following example for accessing data describes this task.

Note: Consult the appropriate ADT documents for exact details about creating custom Federation Adapters.

Example: Create Custom Federation Adapters

In the example, you have an ODBC-compliant Microsoft Access database that is contained in the following file:

c:\My Documents\db1.mdb

The db1.mdb file contains two tables: one with information about CIs, and the other containing information about relationships between the CIs. Assume that the database contains data about CIs in multiple families. Data from both tables is loaded into CA CMDB.

The following Microsoft Access [ci : Table] table describes the data that contains CI information. The column names are shown:

Field Name	Data Type
ID	AutoNumber
myname	Text
myfamily	Text
myclass	Text

The following Microsoft Access [relations : Table] table describes the data that contains relationship information. The column names are shown:

Field Name	Data Type
ID	AutoNumber
myprovider	Text

Field Name	Data Type
myrelation	Text
myconsumer	Text

Chapter 8: Using the MDR Launcher

This chapter describes how to define MDRs, import data, map CIs back to their source, and display federated data for a CI.

This section contains the following topics:

[The MDR Launcher](#) (see page 410)

[MDR Terminology](#) (see page 411)

[MDR Mapping](#) (see page 412)

[MDR Launching](#) (see page 412)

[CMDBf Viewer](#) (see page 412)

[Define an MDR to CA CMDB](#) (see page 413)

[Federation Using GRLoader](#) (see page 422)

[Map Between MDR CIs and CA CMDB CIs](#) (see page 424)

[How To Configure MDRs for CMDBf Viewer](#) (see page 425)

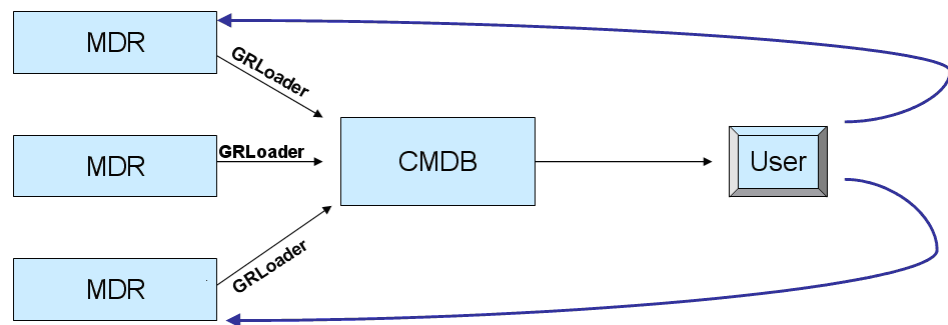
[Launching the MDR Web Browser Interface](#) (see page 426)

[CA Cohesion Integration](#) (see page 426)

The MDR Launcher

One of the main purposes for implementing CA CMDB is to aggregate data from multiple data sources (known as MDRs). However, a CI must always include a reference back to its MDR origin.

CA CMDB provides facilities for importing and loading CIs and also for associating the CIs with their origins. In addition, by using the MDR Launcher capability when viewing a CI in the CA CMDB, you can navigate seamlessly back into the system from which the CI originated, as shown in the following diagram.



Using the MDR Launcher, it is possible to implement a “closed loop” change management process such as the following one:

1. Create a change record.
2. Implement the change.
3. Verify the change by checking the MDR source.
4. Update the CMDB to indicate the change has been made.

From a Problem Management process perspective, you can use the MDR Launcher in the following way:

1. Detect a problem.
2. Determine the severity and pervasiveness of the problem by utilizing the CI relationship data to determine what dependent CIs are affected.
3. Determine possible causes of the problem by researching provider CIs.
4. Perform an in-depth analysis if necessary using the highly detailed information available in the MDR, and use the MDR to take corrective action.

MDR Terminology

The following terms are used in CMDB-MDR integration:

A management data repository (MDR) represents software or data that contains source information about a CI. An MDR generally contains more unrefined CI information than the CMDB, which contains a managed subset of that data.

An *MDR class* (MDR_CLASS) is used to group MDRs that are processed similarly by CA CMDB. There are three special MDR classes: **COHESION**, **GLOBAL**, and **cmdbf**.

An *MDR name* (MDR_NAME) is the name that an MDR uses to reference itself. Verify that the `mdr_name` and `mdr_class` value combination must be unique within your enterprise.

A Federated asset ID (FEDERATED_ASSET_ID) is a unique MDR identifier for a CI.

Different CI families typically use different respective MDRs as data providers. However, a single CI can have multiple MDR data providers. For example:

CI Family	MDR_CLASS
Contact	human resources system
	telephone directory
	single sign-on authentication system
Document	document management system
Air Conditioning	document management system
	contract management system
	air conditioning control system
Mainframe	tape management system
	DASD management system
	performance management system
	job scheduler
Storage	storage management system
	asset management system
Location	asset management system
	education calendar
	office directory
Network	network management systems
	problem management system

There can be multiple MDRs in each MDR class, and each MDR can contribute data to multiple CIs. A given CI can receive data from zero or more MDRs. A CI also can have data contributed to it independently. For example, one mainframe CI may have data which was donated by Disk Management System 1, while another mainframe CI can have data donated by Disk Management System 2 and Job Scheduler 2. CA CMDB manages the relationships among CIs and all their related MDRs.

MDR Mapping

Every MDR has a unique way of identifying the CIs that it manages. Those identifiers are seldom synchronized across MDRs. For example, when referencing a specific Contact CI, different MDRs can use the national identity number, telephone number, license number, or Employee ID; all refer to the same person. The process of associating these disparate identifiers with the same unique identifier (UUID) maintained in the MDB is named *mapping*. Mapping occurs automatically when data is imported using GRLoader when the CI contains the <mdr_name> <mdr_class> and <federated_asset_id> tags. Mapping can also be accomplished manually through the Administration functions in the user interface. A CI that has no mappings associated with it is named *unfederated*. Every CI is automatically mapped to global MDRs using the UUID as the federated_asset_id.

MDR Launching

When you view a CI by using the CA CMDB user interface, you can click a series of buttons to directly launch an MDR user interface. There is one button per MDR mapping for the focus CI. This launching is typically used when you want to verify that a change request has been completely successfully or to obtain additional information about a CI when that data is not collected by the CMDB.

CMDBf Viewer

CA Service Desk Manager provides the CMDBf Viewer to display the results of CI federation across MDRs. From a CI Detail page (or the CI right-click menu on the CI List), click CMDBf Viewer to see CI attributes of federated CMDBs and MDRs in parallel. On the Federated View page, you can click Retrieve to update the information from any of the federated MDRs. For better readability, CA CMDB metadata files can reconcile MDR attribute names and CA CMDB attribute names.

Note: This feature requires MDRs that support Query. You configure the MDR CMDBf Endpoints to display their results on Federated View. For more information, see the *Implementation Guide*.

Define an MDR to CA CMDB

Before a CI can be associated with an MDR, you define the MDR to CA CMDB.

To define a new MDR Provider

1. On the Administration tab, navigate to CA CMDB, MDR Management, MDR List.
2. Click Create New.

The Create New MDR Definition page appears.

3. Complete the fields:

Tenant

Identifies the tenant owner of this MDR (if multi-tenancy is installed).

Button Name

Specifies the button label to appear on the CI Detail page. This name must be unique for each MDR. Required for “launch in context” and CMDBf Viewer.

MDR Name

Specifies the string to match the XML data that is sent in the `mdr_name` field. While the MDR can use any string, the host name is used frequently. This name together with the `mdr_class` form a unique name for the MDR. Required for “launch in context” and CMDBf Viewer.

MDR Class

Specifies the class that must match the data that is sent in the `mdr_class` field in the XML. While this name can be anything, it must together with the `mdr_name` field form a unique identifier for the MDR. Global MDRs are defined with an MDR Class of **GLOBAL**.

- CA Cohesion ACM MDRs must specify an MDR class of **COHESION**, which automatically sets the Path, Parameters and URL to be Launched fields to the required CA Cohesion ACM launch-in-context values.
- CA Asset Portfolio Management r11.3.4 MDRs must specify an MDR name of APM and MDR class of **GLOBAL**, which sets the Path, Parameters and URL to be Launched fields to the required CA Asset Portfolio Management r11.3.4 launch-in-context values.
- CA APM r12.5 MDRs must specify an MDR name of ITAM and MDR class of **GLOBAL**, which sets the Path, Parameters, and URL to be Launched fields to the required CA APM 12.5 launch-in-context values.
- For CMDBf Viewer, MDR Class must be **cmdbf**.

Active

Denotes this MDR definition as active or inactive. Inactive MDR definitions are logically deleted, but they can be made active again by using the Search utility.

Owner

Specifies the contact responsible for this MDR.

Description

Specifies a description in free-form text.

Hostname

Specifies the host name, DNS name, or IP address of the host, which contains the web server which hosts the web page to be launched. Required for "launch in context".

Port

Specifies the TCP/IP port used by the MDR web server to serve up web pages. Port 80 is the default. Required for "launch in context".

Path

Specifies the portion of the URL that precedes the question mark (?) character. This information can be obtained from your MDR documentation.

- For `mdr_class` of Cohesion, the value is set automatically to "CAisd/html/cmdb_cohesion.html" and cannot be changed.
- For `mdr_name` of APM and `mdr_class` of GLOBAL, the value is set automatically to `apm/frmObject.aspx` and cannot be changed.
- For `mdr_name` of ITAM and `mdr_class` of GLOBAL, the value is set automatically to `ITAM/Pages/Asset.aspx` and cannot be changed.

Parameters

Specifies the portion of the URL that follows the question mark (?) character. This information can be obtained from the MDR documentation.

- For `mdr_class` of Cohesion, the value is set automatically to "hostname={hostname}+port={port}+family={family}+name={name}+secret={password}+federated_asset_id={federated_asset_id}" and cannot be changed.
- For `mdr_name` of APM and `mdr_class` of GLOBAL, the value is set automatically to `ObjectID={cmdb_asset_id}&obj=11&FUNCTION=1&WinID=OBFRASSET{cmdb_asset_id}&WinContainerID=` and cannot be changed.
- For `mdr_name` of ITAM and `mdr_class` of GLOBAL, the value is set automatically to `ParentClass=Asset&assetid={cmdb_asset_id}&TicketID={itam_ticketid}` and cannot be changed.

Userid

Specifies the MDR user logon, if required. This value is substituted into the URL wherever {userid} is found. If blank, userid defaults to whomever is currently signed on.

For CA Cohesion ACM, "Shared Secret" is the secret used to access CA Cohesion ACM, if required. This value is substituted into the URL wherever {password} is found.

Note: For more information, see the *CA Cohesion ACM Implementation Guide*.

Shared Secret

Specifies information that is shared between CA CMDB and the MDR. This value is substituted into the URL wherever {password} is found. For CA Cohesion ACM MDRs, the value must match the value of the "com.cendura.security.oneclickauth.secret". For more information about creating a shared secret, see "Integrating with CA CMDB" in the *CA Cohesion ACM Implementation Guide*. Required for CMDBf Viewer.

CMDBf Namespace

Specifies the `federated_asset_id` that is passed to the query as a local ID. For CA CMDB, the value is `http://cmdb.ca.com/r1`.

CMDBf Timeout

(Optional) Specifies time limit for CMDBf endpoint query. Default is ten (10) seconds.

URL to be Launched

Default value of `http://{hostname}:{port}/{path}?{parameters}`. For some MDRs, it can be overridden if necessary to accommodate MDR-specific requirements. Required for "launch in context".

For `mdr_name` of APM and `mdr_class` of GLOBAL, the value is `http://{hostname}:{port}/{path}?{parameters}`

For `mdr_name` of ITAM and `mdr_class` of GLOBAL, the value is `http://{hostname}:{port}/{path}?{parameters}`

For `mdr_class` of Cohesion the default value is `http://cmdb_hostname:cmdb_port/{path}?{parameters}`

where:

cmdb_hostname is the host name, DNS name or IP address of the CA CMDB web server. Defaults to the current hostname that is currently accessing the CA CMDB web server.

cmdb_port is the TCP/IP port of the CMDB web server. Defaults to the current port number used to access the CA CMDB web server.

Note: If you have enabled SSL support for CA Cohesion ACM, set the URL to:

`http://hostname:port/{path}?{parameters}+https=yes`

For information about enabling CA Cohesion ACM HTTPS support, see the CA Cohesion ACM online help topic [Creating the HTTPS Certificate and Enabling HTTPS](#).

CMDBf Endpoint

Specifies the Query Service endpoint for the MDR. Required for CMDBf Viewer and retrieving updated MDR data. If you use CA CMDB as an MDR provider, the value is `http://cmdb_hostname:cmdb_port/axis/services/QueryPort`.

Click Save.

The MDR is defined.

MDR URL Definitions

The URL to be launched has the default value of `http://{hostname}:{port}/{path}?{parameters}`. This expression can be modified if necessary to accommodate any MDR-specific considerations. The URL is required for "launch in context".

For `mdr_name` of APM or ITAM and `mdr_class` of GLOBAL, the default value is:

`http://{hostname}:{port}/{path}?{parameters}`

For `mdr_class` of Cohesion, the default value is:

`http://{cmdb_hostname}:{cmdb_port}/{path}?{parameters}`

cmdb_hostname

Specifies the hostname, dnsname or IP address of the CA CMDB web server.
Defaults to the current hostname currently accessing the CA CMDB web server.

cmdb_port

Specifies the TCP/IP port of the CA CMDB web server. Defaults to the current port number that is used to access the CA CMDB web server.

If you have enabled SSL support for CA Cohesion ACM, set the URL to the following:

`http://{hostname}:{port}/{path}?{parameters}+https=yes`

Note: For information about enabling CA Cohesion ACM HTTPS support, see the CA Cohesion ACM online help.

MDR Launch URL

The MDR launch URL has the following default value:

`http://{hostname}:{port}/{path}?{parameters}`

You can modify this expression to accommodate MDR-specific considerations. The URL is required for "launch in context".

- For `mdr_name` of APM or ITAM and `mdr_class` of GLOBAL, the default value is:

`http://{hostname}:{port}/{path}?{parameters}`

- For `mdr_class` of Cohesion, the default value is:

`http://{cmdb_hostname}:{cmdb_port}/{path}?{parameters}`

cmdb_hostname

Host name, DNS name or IP address of the CA CMDB web server. Defaults to the host name currently accessing the CA CMDB web server.

cmdb_port

TCP/IP port of the CA CMDB web server. Defaults to the current port number that is used to access the CA CMDB web server.

If you have enabled SSL support for CA Cohesion ACM, set the URL to:

`http://{hostname}:{port}/{path}?{parameters}+https=yes`

Note: For information about enabling CA Cohesion ACM HTTPS support, see [Creating the HTTPS Certificate and Enabling HTTPS in the Cohesion online help](#).

Defining Launch Parameters for URL Substitution

When defining an MDR, the following parameters can be used to construct its URL for display. These parameters are substituted with their appropriate values at run time. These variables must be specified in the fields described previously.

{hostname} is the MDR host name from the MDR definition.

{alarm_id} is the IP address of the selected CI.

{federated_asset_ID} is the unique identifier of the selected CI in the MDR.

{cmdb_asset_id} is the asset ID for the CI.

{port} is the MDR port number from the MDR definition.

{userid} is the user ID from the MDR definition. If blank, userid defaults to whomever is currently signed on.

{password} is the shared secret from the MDR definition.

{mdr_name} is the mdr_name from the MDR definition.

{mdr_class} is the mdr_class from the MDR definition.

{class} is the class of the selected CI.

{family} is the family of the selected CI.

{path} is the path as described in the MDR definition.

{name} is the name of the selected CI.

{model} is the model of the selected CI

{manufacturer} is the manufacturer of the selected CI

{itam_ticketid} is the ticket id to log into CA APM

Example: Launching an MDR

A CA CMDB user is looking at a Server CI named `server1`, which has a map to an internally developed application named Comet. Comet uniquely identifies `server1` as `server:server1`.

Comet is defined as an MDR with the following properties:

- Hostname: CometServer
- Port: 80
- Path: index.php
- Parameters: item={federated_asset_id}
- Launch_url: http://{hostname}:{port}/{path}?{parameters}

In CA CMDB, when the user clicks the Comet button on the Attributes tab of the `server1` CI, a web browser window opens the following URL:

`http://CometServer:80/index.php?item=server:server1`

Parameters for URL Substitution

When defining an MDR, you can use the following parameters to construct its URL for display. These parameters are substituted with their appropriate values at runtime. These variables must be specified in the MDR field definitions.

{hostname}

Specifies the MDR host name from the MDR definition.

{alarm_id}

Specifies the IP address of the selected CI.

{federated_asset_ID}

Specifies the unique identifier of the selected CI in the MDR.

{cmdb_asset_id}

Specifies the asset ID for the CI.

{port}

Specifies the MDR port number from the MDR definition.

{userid}

Specifies the user ID from the MDR definition. If blank, `userid` defaults to whomever is currently signed on.

{password}

Specifies the shared secret from the MDR definition.

{mdr_name}

Specifies the mdr_name from the MDR definition.

{mdr_class}

Specifies the mdr_class from the MDR definition.

{class}

Specifies the class of the selected CI.

{family}

Specifies the family of the selected CI.

{path}

Specifies the path as described in the MDR definition.

{name}

Specifies the name of the selected CI.

{model}

Specifies the model of the selected CI

{manufacturer}

Specifies the manufacturer of the selected CI

Example: Use Parameters for URL Substitution

A CA CMDB user is looking at a Server CI named server1, which has a map to an internally developed application called Comet. Comet uniquely identifies server1 as server:server1.

Comet is defined as an MDR with the following properties:

- Hostname: CometServer
- Port: 80
- Path: index.php
- Parameters: item={federated_asset_id}
- Launch_url: http://{hostname}:{port}/{path}?{parameters}

In CA CMDB, when the user clicks the Comet button on the Attributes tab of the server1 CI, a web browser window opens the following URL:

http://CometServer:80/index.php?item=server:server1

Federation Using GRLoader

When using GRLoader, the following XML tags must be populated in every CI in the XML document. These tags apply to every MDR family.

- <mdr_name>
- <mdr_class>
- <federated_asset_id>

If this information is not present in the XML, “launch in context” is not possible because the origin of the CI cannot be determined.

To identify the source of a CI, you can modify the XML before it is input into GRLoader. You can use a text editor to modify the XML and make a global change, or this task can be done programmatically.

Note: For more information about MDR identification and GRLoader, see the *CA CMDB Technical Reference Guide*.

Note: CA Cohesion ACM automatically provides mdr_name, mdr_class and federated_asset_id.

Federate a CI

If CIs are loaded into CA CMDB before their corresponding MDR is defined, they are unfederated, which means that they are not yet connected to an MDR and do not yet support “launch in context”.

To federate a CI

1. Define the required MDR.
2. Do one of the following:
 - Manually map the CI.
 - Re-run the CA Cohesion ACM report which created the CI, specifying Allow update of existing CIs on the report.

Note: For more information about CA Cohesion ACM Reports, see the *CA Cohesion ACM Product Guide*.

3. Re-run GRLoader, specifying the same input file that was used to create the CIs.
The CA CMDB reconciliation engine merges the MDR information into the existing CIs.
4. Create an XML document that describes the CI and its MDR and run GRLoader in Update mode.
The CA CMDB reconciliation engine merges the new information into the existing CI. The existing CI is federated.

Example: Specify CI Location

You must verify that the reconciliation engine is provided with enough information to locate the CI you want to update. In the following example, end tags are removed and spaces added for readability.

```
<ci>
  <name>    server3
  <mac_address>
  <serial_number>
  <asset_num>
  <dns_name>
  <mdr_name> mdr_one
  <mdr_class> Cohesion
</ci>
```

Define Multiple MDRs to a CI Using GRLoader

You can define multiple MDRs to a CI by using GRLoader.

To define multiple MDRs to a single CI, the XML document can repeat the <ci> node, with each duplicated <ci> node specifying a different mdr_name and mdr_class. In other words, each MDR can contribute its attributes independently of any other MDR that contributes data to the CI.

Example: Define Multiple MDRs to a CI

If MDR1 and MDR2 both contribute data to the server2 CI, the XML document looks something like the following example. In the example, end tags are removed and spaces added for readability.

```
<ci>
  <name>    server2
  <mdr_name> mdr1
  <mdr_class> Cohesion
  <diskspace> 500 gb
  <disktype> SCSI-3
</ci>
```

```
<ci>
  <name>    server2
  <mdr_name> mdr2
  <mdr_class>Service Assure
  <sla>
</ci>
```

CA CMDB reconciles the two previous CIs to the same CI and associates each of the two MDRs to that single CI.

Note: The CIs can be imported in one or two runs of GRloader.

Map Between MDR CIs and CA CMDB CIs

After you define a CI manually using the File, New Configuration Item... option, you must manually define the mapping between this CI and the CI in the federated MDR. There are two ways to associate a CI with an MDR:

- Editing the CI.
- Using the Federated CI Mapping node on the CA CMDB Administration tab.

To create a mapping by editing the CI

1. Display the CI Detail page of the CI that you want to associate with an MDR.
2. Click Edit.
3. Display the Attributes tab.
4. Click Add MDR.

The CI is associated with the MDR.

To create a mapping using the Federated CI Mapping page

1. Click the CA CMDB Administration tab.
2. Open the Management Data Repository node.
3. Select the Federated CI Mapping node.

The Federated CI Mapping List appears.

4. Click Create New.

The Create New Federated CI Mapping For page appears.

5. Associate the CI with the MDR by completing the Federated CI Mapping fields:

CI Name

Specifies the name to use to identify the configuration item.

Federated Asset ID

Specifies the string identifier used by the source MDR to identify this CI. The identifier is determined by the MDR software.

MDR Name

Specifies the name that identifies the MDR (and its MDR button).

Active

Denotes whether this mapping is active or not. Mappings cannot be deleted, only inactivated.

6. Click Save.

The mapping between this CI and the CI in the federated MDR is defined.

How To Configure MDRs for CMDBf Viewer

Before you can use the CMDBf Viewer, set up your federated MDR providers to point to the CMDBf query service, as follows:

- External MDRs must provide a query service that can handle InstanceIdConstraint query.
- Button Name, MDR Name, and MDR Class are required to show the CMDBf Viewer button on the CI Detail page.
- MDR Class must be defined as **cmdbf**
- For CA CMDB, CMDBf Namespace must be set to **http://cmdb.ca.com/r1**. For other CMDBs and MDRs, see the appropriate documentation.
- Timeout is optional. Default is ten (10) seconds.
- To display a working Retrieve button on the Federated View, you must define CMDBf Endpoint, Userid, and Shared Secret.

Note: An existing CA CMDB system can be set up as a CMDBf provider by specifying an CMDBf Endpoint of "http://servername:port/axis/services/QueryPort" where:

- hostname is the computer where CA CMDB is installed (localhost or computer name).
- port is the port where CA CMDB is configured.

Launching the MDR Web Browser Interface

After a mapping between a CI and an MDR is created, a button is placed automatically on the Attributes tab. If there are multiple MDRs associated with this CI, multiple buttons appear.

When you click an MDR button, a new page opens and the fully substituted MDR URL that is defined in the MDR definition appears.

Example: Launch CA Cohesion

When a CI has been correctly associated with its MDR, a Cohesion button appears on the Attributes tab. If a button does not appear on the Attributes tab, review the mapping for the CI that is displayed. Verify that there is a mapping for this CI and that the target MDR has a URL that can be launched. Clicking the button to launch the MDR causes a new window to open to the target URL to launch CA Cohesion.

CA Cohesion Integration

Consider the following when integrating CA Cohesion ACM with CA CMDB:

- Integrating Cohesion with CA CMDB

Note: For information about Cohesion-CA CMDB integration, see the *CA Cohesion ACM Implementation Guide*.

- Importing CIs from a Cohesion MDR

Note: For information about how to import CIs from a Cohesion MDR, see the online help that is available from CA Cohesion ACM Reports Report Templates tab.

- Launch-in-Context for Cohesion MDRs

For launch-in-context integration to work best with CA Cohesion ACM, we recommend that you use the CA CMDB Administration tab to define the Cohesion MDR *before* running the Cohesion CMDB Report.

Note: Because CA Cohesion ACM does not support a unique Federated asset ID for NIC or File System CIs, Cohesion does not support MDR Launcher for NIC or File System CIs. Therefore, a Cohesion-based NIC or a File System CI does not display an MDR launch button even when it was imported successfully.

Chapter 9: Managing Web Services

Important! For additional information on web services, see the CA Service Desk Manager Technical Reference Guide.

This section contains the following topics:

[Web Services Management](#) (see page 427)

[CA Service Desk Manager Components](#) (see page 428)

[Tips for Web Services Clients](#) (see page 428)

[Web Services Configuration](#) (see page 431)

[Web Services Security](#) (see page 433)

[Error Handling](#) (see page 435)

[Web Services Installation](#) (see page 438)

[External Specifications](#) (see page 438)

[Objects](#) (see page 457)

[ITIL Methodology](#) (see page 463)

[Use the Web Services](#) (see page 465)

Web Services Management

Web Services are a set of data exchange standards that enable communication between products, even if they are on different operating environments. This ability is analogous to browsing the Web on a personal computer—all remote websites are accessible regardless of whether they are hosted on Solaris, AIX, Windows, and so on. In the same manner, Web Services allow products to communicate over HTTP to various servers regardless of operating environment. For example, a Microsoft Office product can communicate with a program on a UNIX server, and a Java Server Page can access a server hosted on a Windows server. This platform-neutral communication allows for powerful integrations.

The Web Services take advantage of this technology, allowing almost any product to access CA Service Desk Manager and Knowledge Management. Web Services clients can create tickets, update assets, search the knowledge base, and so forth.

CA Service Desk Manager Components

The installation files for this version of J2EE Web Service are initially installed by the CA Service Desk Manager and Knowledge Management installation and can be located in the following directory:

```
<NX_ROOT>/sdk/websvc/R11
```

The installation files for the backward compatible J2EE Web Services are located in the following directory:

```
<NX_ROOT>/sdk/websvc/60
```

where <NX_ROOT> is the root installation path for CA Service Desk Manager.

Tips for Web Services Clients

A sample Java client application for the Web Services in the *samples* directory of the CA Service Desk Manager installation assists developers with web services client application development.

Many of the Web Services methods require arrays as input parameters (for example, the method `createIssue()` permits an empty array for the 'propertyValues'). Sometimes these arrays are optional, but the service requires an empty array to be passed. As a tip for using Visual Studio .NET to access Web Services, specify an empty array with one of the following arrays:

C# language

```
String[] emptyArray = new string[0];
```

Visual Basic .NET

```
Dim emptyArray As String() = {}
```

Java

```
ArrayOfString attr = new ArrayOfString();  
attr.setString(new String[0]);  
ArrayOfString is a proprietary class.
```

`emptyArray` can then be passed to array parameters that accept empty arrays.

Note: The CA Service Desk Manager Web Services use the Apache implementation of standards established by the World Wide Web Consortium (W3C). Ideally, a client on any type of operating environment can access the services, but vendor implementations vary. Many programming environments provide a tool to generate proxy classes from a Web Services Description Language (WSDL) description.

More information:

[Java Clients](#) (see page 429)

Java Clients

The TableOfContents.doc in \$NX_ROOT/samples/sdk/websvc lists several Java sample programs.

Each sample program contains notes on how it can be compiled and run using the script files run_java_test.bat.txt (Windows) and run_java_test.sh.txt (UNIX). These scripts demonstrate how to use org.apache.axis.wsdl.WSDL2Java to generate the CA Service Desk Manager web services client-side stub files.

The -w parameter is required to properly generate the stub files when using Axis 1.4. Running WSDL2Java as shown will generate the stub files in subdirectory com/ca/www/UnicenterServicePlus/ServiceDesk. The following files are generated:

- ArrayOfInt.java
- ArrayOfString.java
- ListResult.java
- USD_WebService.java
- USD_WebServiceLocator.java
- USD_WebServiceSoap.java
- USD_WebServiceSoapSoapBindingStub.java.

Import these classes with the following statement:

```
import com.ca.www.UnicenterServicePlus.ServiceDesk.*;
```

Many Web Service methods have parameters of type ArrayOfString, a proprietary class. For example, the createRequest() method's attrVals, propertyValues and attributes parameters are all ArrayOfString parameters.

To set the values in an ArrayOfString variable, instantiate the variable and then use setString() as follows:

```
ArrayOfString attrVals = new ArrayOfString();  
attrVals.setString(new String[]{"customer", customerHandle, "description", "description text"});
```

To set it to empty

```
attrVals.setString(new String[0]);
```

Use a variable of type `ListResult`, another proprietary class, as the return value from the `List` methods: `doQuery()`, `getRelatedList()`, `getNotificationsForContact()`, `getPendingChangeTaskListForContact()` and `getPendingIssueTaskListForContact()`. A `ListResult` contains `listHandle` and `listLength` elements, which can be retrieved using `getListHandle()` and `getListLength()` as shown in this example:

```
ListResult doQueryResult = new ListResult();
doQueryResult = USPSD.doQuery(sid, "iss", "active = 1");
int listHandle = doQueryResult.getListHandle();
int listLength = doQueryResult.getListLength();
```

The `getListValues()` method uses the `listHandle`, retrieving the values from a subset of the list.

The `Handles` parameter of the `freeListHandles()` method is an `ArrayOfInt`, another proprietary class. Call `freeListHandles()` using the `listHandle` taken from a `ListResult`:

```
ArrayOfInt handleList = new ArrayOfInt();
handleList.setInteger(new java.lang.Integer []{ new java.lang.Integer(listHandle) });
USPSD.freeListHandles(sid, handleList);
```

Some methods have pass by reference parameters of type `javax.xml.rpc.holders.StringHolder`. For example, `createRequest()` has two parameters of this type, `NewRequestHandle` and `NewRequestNumber`.

```
StringHolder NewRequestNumber = new StringHolder();
StringHolder NewRequestHandle = new StringHolder();
String result;
result = USPSD.createRequest(sid, creatorHandle, attrVals, propertyValues, template, attributes,
NewRequestHandle, NewRequestNumber);
```

The Request's handle and reference number (`ref_num`) can then be obtained from `NewRequestHandle.value` and `NewRequestNumber.value` respectively.

Web Services Configuration

The CA Service Desk Manager Web Services may be configured with entries in special web configuration files. The names and descriptions of the configuration options are summarized as follows:

Option Name	Description
design_mode_stubs	Sets the Web Service to 'design mode' (CA Service Desk Manager only).
require_secure_logon	Requires the login() and loginService() web methods to be called with a secure protocol, such as https.
require_secure_connection	Requires that every web method be called with a secure protocol.
disable_user_logon	Disables both login() and loginService() web methods, so only loginServiceManaged() can be used to log in.

Note: The configuration settings can be set in the deploy.wsdd file. For Unicenter Service Desk r11.0, this file is located in this subdirectory: <NX_ROOT>/sdk/websvc/R11. For post GA 6.0, it is located in this subdirectory: <NX_ROOT>/sdk/websvc/60. We recommend that you create a back up copy of either file before making any changes to it.

CA Service Desk Manager has added protection to the integrity of the Tomcat server on which it is running by checking the length of the attribute values that are passed to Web Service methods. By default, web service calls will return an Axis Fault if the length of an attribute value is greater than 900,000 bytes.

The following parameters are set in the deploy.wsdd file:

- **fatal_max_string_length.** Sets the length of the largest attribute value that will be accepted by a web service method.
Default: 900,000 bytes
- **validate_parameters.** Sets whether the attribute value length checking will be performed. Set to 0 to turn the validation off.
Default: 1 (on)
- **exception_methods.** Displays a comma-delimited list of Web Service methods that are exempt from the attribute value length validation.

More information:

[Redeploy the Web Services](#) (see page 432)

Redeploy the Web Services

New configuration settings take effect when CA Service Desk Manager Web Services is redeployed. Complete the following steps to redeploy the Web Services:

1. Open a command prompt and set the CLASSPATH environment variable to include the required Axis jar files, which are supplied in <NX_ROOT>/java/lib. For example, to set it on Windows, use the following:

```
set AXISHOME=%NX_ROOT%\javallib
set classpath=
%AXISHOME%\axis.jar;%AXISHOME%\jaxrpc.jar;%AXISHOME%\saaj.jar;%AXISHOME%\commons-logging.jar;%AXISHOME%\commons-discovery.jar;%AXISHOME%\wsdl4j.jar;%AXISHOME%\log4j-1.2.8.jar;%classpath%
```

2. Change the directory to <NX_ROOT>/sdk/websvc/R11 (or <NX_ROOT>/sdk/websvc/60 for GA Version 6.0 Web Services), and run the following commands:

```
java org.apache.axis.client.AdminClient undeploy.wsdd
java org.apache.axis.client.AdminClient deploy.wsdd
```

3. Recycle Tomcat by recycling the CA Service Desk Manager service. You may avoid shutting down the entire CA Service Desk Manager system by recycling Tomcat by simply using the following commands:

```
pdm_tomcat_nxd -c stop
pdm_tomcat_nxd -c start
```

The Web Service is now redeployed. You can verify that the service actually deployed by viewing the Axis services listing page at the following default URL:

`http://<servername>:<port>/axis/services`

Note: The exact URL depends upon your installation settings.

Web Services Security

There are important security considerations in deploying web services. The default configuration when using HTTP is insecure, as it is for all information in web service calls sent between the client and the server in plain text over the network using the HTTP protocol. This includes not only application data, such as ticket descriptions and contact names, but also web service session identifiers (SID); and depending upon the web service application login methods used, it may include passwords. Administrators deploying web services are highly encouraged to review this information carefully and to take additional configuration steps at the application and network levels to secure their web service environment.

Important! The default web service configuration used with HTTP is insecure and vulnerable to security threats, which can include password discovery, session fixation, and data spying, among others.

There are three interrelated key security considerations in deploying Web Services:

- What (application level) access authentication schemes should this deployment support?
- What additional networking level security features does this deployment require?
- How will these requirements be enforced through web service configuration options?

The following describes each security feature:

- **Web Service Application Level Authentication Schemes**—To access Web Services, a web service client application must be authenticated with the web service application. Web Services provides two schemes of access authentication. The first is by username/password, and the other is by Public Key Infrastructure (PKI) technology. Both work with the Access Control and Management component in Web Services, using access policy. Access authentication and access management are the most important security features of Web Services.

Authentication with username/password methods may be disabled using the following security configuration command:

```
disable_user_logon
```

Before enabling this option, the administrator needs to determine if each web service client for which an enterprise is requesting Web Services access, can actually provide support for the alternative authentication method, which is the PKI-based login method. The key advantage to the PKI technology is that Web Services client applications do not require *maintained* system user accounts, that is; the maintenance, storage, and transmission of their passwords.

- **Networking Level Security Configuration**—In both authentication schemes, username/password and Public Key Infrastructure (PKI), notice that the session identifier returned from the specific login method (as well as all subsequent information), are transmitted in plain text when using HTTP. Furthermore, if the username/password authentication scheme is used, the password is sent unprotected (in plain text) from the web service client application to the Web Services. During product development, the W3C did not have recommended standards for web services security. Subsequently, WS-Security is not used by these Web Services implementations to provide a security context. Instead, point-to-point transport layer security (SSL/TLS) and other network level security mechanisms (for example, IPSec), are recommended to protect the otherwise plain text transmission of the application-level authentication exchange(s), and subsequent session identification and data.

Important! We recommend using SSL (or https) when deploying Web Services to protect the application-level authentication exchanges and subsequent transmissions of session identification and data.

- **Web Service Configuration**—To allow administrators to enforce communications protocol-level security at the level of the Web Services application, the following two security configuration commands are supported:

`require_secure_login`

This security feature requires you to use SSL (or https) for calling the `Login()` and `LoginService()` methods. This feature also provides a handy method for protecting the username and password, while avoiding the overhead of SSL for the rest of the web services.

Important! If you use the `require_secure_login` command, the Web Services application will not confirm that communications protocol-level security is enforced for methods other than `Login()` and `LoginService()`. Unless other precautions are taken, the other Web Services methods may be invoked insecurely, causing greater vulnerability to security threats.

`require_secure_connection`

This security feature requires you to use SSL to access any part of the web service. If https is required but not used, then a SOAP Fault with code `UDS_SECURE_CHANNEL_REQUIRED` is returned.

Note: For information about how to configure SSL, see your J2EE Servlet Container documentation.

More information:

[External Specifications](#) (see page 438)

Error Handling

If an error occurs with a Web Services method, a SOAP Fault is returned. The SOAP Fault is the standard means of returning exception information for Web Services.

The Fault message contains standardized <Message> and <Code> elements, but the most informative is the <Detail> element. The <Detail> element contains <ErrorCode> and <ErrorMessage> elements. The <ErrorCode> element returns an enumerated error code specific to either the CA Service Desk Manager or Knowledge Management product. The <ErrorMessage> contains an English string describing the errors. The <ErrorMessage> elements are more suitable for aiding the developer and more appropriate messages should display to users.

For example, the following illustrates a SOAP Fault when a bad parameter is supplied to the CA Service Desk Manager getObjectValues() method:

```
<soap:Fault>
  <faultcode>soap:Client</faultcode>
  <faultstring>Error on fetch with attribute list:persistent_id,first_name,last_nameParamErrorHere</faultstring>
  <detail>
    <ErrorCode>1001</ErrorCode>
    <ErrorMessage> Error on fetch with attribute list: persistent_id,first_name,last_nameParamErrorHere
  </ErrorMessage>
  </detail>
</soap:Fault>
```

If you are using a client built with Microsoft .NET managed code, a failed Web Services method call raises a "SOAPException" exception. All errors cancel the operation invoked.

In some cases, errors may be written by the servlet container and therefore, display in the servlet container logs. In other cases, error information may be written to CA Service Desk Manager logs. These logs are located in the following subdirectories:

- In the /bopcfg/www/CATALINA_BASE/logs subdirectory of CA Service Desk Manager installation
- In the /log subdirectory of the CA Service Desk Manager installation and to all logs that have the prefix "stdlog".

Note: We recommend that you constantly monitor these logs, as the server may log its own errors without reporting them to the CA Service Desk Manager Web Services.

Lock Errors

CA Service Desk Manager objects are locked during updates. Methods that update objects (such as, `updateObject()` or `transfer()`) may return the following lock error code:

UDS_LOCK_ERR

This code indicates that another user is updating the record. Often the locking user's handle is returned in the `ErrorMessage` element.

Time Outs

A method may take a long time to process if the CA Service Desk Manager server is heavily loaded. In rare cases, a method may never return because a separate process failed to reply or some other error occurred. To guard against excessive blocking, every Web Services method times out after a number of seconds. Web Services method time-out is a CA Service Desk Manager server time-out, *not* a Web server time-out, network time-out, and so on.

If a method times out, it returns the following error code:

UDS_TIMEOUT_ERR

The operation is not aborted! The server may have received the request and will process it successfully, although slowly. This type of problem may occur when using the `doSelect()` method to retrieve several thousand records.

Note: For information about the `doSelect` method, see the *CA Service Desk Manager Technical Reference Guide*.

Note: The Web Services will delay a few seconds the first time it is accessed after the J2EE application server is recycled. This happens because the application is initializing, loading DLLs, libraries, and so on, and occurs only with the first Web Services method call. All subsequent calls return much faster.

Error Codes

The following table lists the possible values for the `<ErrorCode>` value in a SOAP Fault returned from a Web Services call:

Error Name	Value	Description
UDS_OK	0	Successful.
UDS_FAILURE	1	General failure, check system logs.

Error Name	Value	Description
UDS_BAD_PARAM	1000	A bad parameter was passed to a method. This error occurs if a required parameter is missing, the wrong type was passed, or an invalid value was used.
UDS_INTERNAL_ERR	1001	Signals that an internal error occurred. A description is found in the return array and the system logs.
UDS_LOCK_ERR	1002	An attempt was made to update an object locked by another user or process. Usually the ID of the contact responsible for locking the object is returned in the return data.
UDS_UPDATE_ERR	1003	An error occurred updating an object. Make sure all required attributes were set and check the system log.
UDS_CREATION_ERR	1004	An error occurred creating an object. Make sure all required attributes were set and check the system logs.
UDS_NOT_FOUND	1005	A search method failed to find any matches or failed to find an object specified. This can happen if a bad or invalid handle is passed to any method.
UDS_SESSION_TIMEOUT	1006	The current method timed out, the CA Service Desk Manager server may be heavily loaded or the method itself was bad.
UDS_SERVER_GONE	1007	The CA Service Desk Manager server connection is lost, UDS methods will no longer function and all list references are lost.
UDS_FETCH_ERR	1008	An error occurred while retrieving list data.
UDS_BAD_SESSION	1010	An invalid SID was used.
UDS_CNTXT_TIMEOUT	1011	The SID timed out.
UDS_SECURE_CHANNEL_REQUIRED	1012	The Web Services (or a web service method) requires a secure channel (for example: SSL) for access, but an unsecured channel is being used.
UDS_SECURITY_VIOLATION	1013	The attempted operation violates CA Service Desk Manager security and was aborted.
UDS_OVER_POLICY_LIMIT	3002	The attempted request is refused because it exceeds the limit defined in the policy.

Web Services Installation

The web services are installed during the CA Service Desk Manager installation for both the primary and secondary servers. In order for web service clients to use a URL on a secondary server, a web engine must be added to the secondary server.

Note: For information about adding and configuring web directors and web engines, see the *Administration Guide*.

Web services will, by default, use the primary server's object manager named *domsrvr*. To use any other object manager, set and install the *webservice_domsrvr* Options Manager option. The object manager can be on a primary or a secondary server.

Note: For information about setting and installing the *webservice_domsrvr* option, see the *Online Help*.

How to Activate Design-Time

The CA Service Desk Manager Web Services includes a method stub configuration feature for developers. When activated, the Web Services ignores the CA Service Desk Manager server and returns simulated data for method calls so that Web Services calls can be made without running a CA Service Desk Manager server.

To activate design-time in the Java version, do the following:

1. Edit `deploy.wsdd` to uncomment the sections for "design_mode_stubs".
2. You must reverse the deployment and redeploy the server.
3. Restart the application server.

The design-time feature is activated.

Note: The design-time feature applies to CA Service Desk Manager Web Services methods only.

External Specifications

CA Service Desk Manager lets you specify user access authentication and the features available by access control and management.

User Access Authentication

CA Service Desk Manager Web Services provides two access authentication schemes. They are associated with the new access control and management feature, which uses an access policy.

User Name/Password

Verifies the User Name/Password, as described in previous releases of the product.

Public Key Infrastructure (PKI) Technology

Verifies that the person requesting the access has ownership of a certain private key.

Important! If you plan to use an application that accesses this version of CA Service Desk Manager Web Services, we recommend strongly that you first define a Web Service Access Policy, complete with its code value, in CA Service Desk Manager. A default access policy with a policy code of DEFAULT is available when CA Service Desk Manager is installed and configured.

More information:

[Define an Access Policy](#) (see page 448)

User Name/Password Authentication

If you plan to use the User Name/Password type of access authentication, the user application needs to invoke one of following two web services methods to gain access to CA Service Desk Manager Web Services.

Note: The login user that you specify in the username parameter (not the proxy contact specified in the policy) is responsible for activities initiated in a session. All function group security and data partition is enforced for this login user.

login (Username, Password)

This method is provided for backward compatibility, where access authentication is performed on the username and password supplied. A SID (session ID) is returned only if the access is authenticated. All subsequent web services calls need to include this SID. Default access policy is then applied to all subsequent web services accesses labeled with the SID.

Username and password are required fields that require plain text when you define them.

loginService (Username, Password, Policy)

This method is similar to the previous login function in that access authentication is performed on the username and password supplied. A SID is returned only if the access is authenticated. However, a specific access policy, as identified in the third parameter, is applied to control and manage all subsequent Web Services accesses. Empty content in the policy parameter automatically applies the default policy.

Username and password are required fields that require plain text when you define them. Policy is required, but can be empty, and you must use plain text. Use the policy code defined in a policy.

How a login is validated depends on the contact's assigned *access type*. The access type object is hosted by CA Service Desk Manager and sets the validation type. You can use the product to view the access type record, and you can also use the `getAccessTypeForContact()` web method to retrieve any access type object information.

Note: For more information about access types, see the *Administration Guide*.

Public Key Infrastructure (PKI) Authentication

If you plan to use the PKI authentication, realize that the content of the login request is encrypted with a private key that can only be decrypted by its matching public key. The response of the login request is returned as plain text.

Generally, each application accessing CA Service Desk Manager Web Services is assigned with a policy. CA Service Desk Manager Web Services stores detailed information about a policy, along with the public key of a digital certificate. An application, as the policy holder, uses the private key of the digital certificate and the policy code (as policy identifier) to assemble a login request.

loginServiceManaged (Policy, Encrypted_Policy)

CA Service Desk Manager Web Services performs the user authentication by locating the policy through the plain text policy code, retrieving the policy holder's public key associated with the policy, decrypting the encrypted policy code, matching the decrypted content with the policy code, and finally, opening a session with a back-end server. The plain text session ID (SID) is returned and can be used for subsequent method invocations. Only the policyholder holds the private key that matches the policy's associated public key stored in CA Service Desk Manager.

All subsequent web services calls must include the returned session ID (SID). The Proxy contact specified in the policy is responsible for all web services activities initiated in this session. All function group security and data partition is enforced for the proxy contact.

Important! The `Encrypted_Policy` parameter should be in the BASE64 text format. The user application must perform proper conversion from the binary format.

Policy is a required field. When you define it, use plain text policy code as defined in a policy. `Encrypted_Policy` (the digital signature of the policy code encrypted with the policy holder's private key) is required. When you define `Encrypted_Policy`, use the algorithm SHA1 with RSA to obtain the digital signature.

Implement loginServiceManaged in Java

The following shows how to generate Certificates and then use these generated Certificates to access the CA Service Desk Manager web services.

In the following example, the login process completes using the CA Service Desk Manager Certificate and then performs two common web services calls. The `getBopsid()` web services method call allows you to obtain a token that is linked to a specific user. This token can be used to login to the CA Service Desk Manager web interface as the linked user without being prompted for a password. This allows seamless integration to be enabled between different applications.

Important! The generated BOPSID token expires after 30 seconds, so it must be used promptly.

Note: Use the AXIS Tool known as WSDL2Java to generate the required stub classes. You can get this tool from <http://ws.apache.org/>.

Important! There is a known issue when using the 1.4 version of the AXIS tool. For more information, see the *Release Notes*.

To implement loginServiceManaged in Java

1. Start the CA Service Desk Manager service.
2. Run `pdm_pki -p DEFAULT`.

DEFAULT.p12 is created in the current directory. This policy will have the password equal to the policy name (in this case DEFAULT).

Note: This command will also add the Certificate's public key to the field `pub_key` field (`public_key` attribute) in the `sapolicy` table/object.
3. Log into CA Service Desk Manager and select the Administration tab.

Navigate to Web Services Policy, Policies.

The Web Services Access Policy List appears.
4. Click DEFAULT.

The Update Web Services Access Policy appears.
5. Complete the Proxy Contact field (in this example, ServiceDesk) and confirm that the DEFAULT policy record Has Key field displays "Yes."
6. Copy DEFAULT.p12 (from the directory where command `pdm_pki` is executed), the JSP file called *pkilogin.jsp* and the HTML file called *pkilogin.htm* (from the `$NX_ROOT\samples\sdk\websvc\java\test1_pki` directory) to the following directory:

`$NX_ROOT\bopcfg\www\CATALINA_BASE\webapps\axis`
7. Open the HTML form (from the axis directory). For example, `http://localhost:8080/axis/pkilogin.htm`

Complete the appropriate fields.

Note: The Directory field identifies the location of the Certificate file. Modify the path to the correct location.
8. Click Log me in!

The results page opens.

9. Click the BOPSID URL.

Important! Click this immediately! The BOPSID has a limited life token of about 30 seconds.

The format of a URL using a BOPSID is as follows:

`http://<server name>:<port>/CAisd/pdmweb.exe?BOPSID=<BOPSID value>`

Note: In order to use the `loginServiceManaged` method for a Java client program running on AIX, you may need to replace a pair of security policy files within your `JAVA_HOME`. Go to <http://www.ibm.com> and search for "developerworks java technology security information AIX". In the "developerWorks : Java technology : Security" document, follow the link to "IBM SDK Policy files". Download the unrestricted policy files, `local_policy.jar` and `US_export_policy.jar`. Use these files to replace the original files in your `JAVA_HOME/lib/security` directory."

Configuration for the PKI Authentication Type

To configure for PKI authentication, you must first create an access policy. The process flow is as follows:

Create an Access Policy

The administrator performs this task using the product (Web Interface only), and as part of the process, needs to assign a unique text code to each access policy.

Obtain a Digital Certificate with a Public/Private Key Pair and Associate it with the Access Policy

For PKI access authentication, a user application needs to obtain a digital certificate that contains both a public key and private key pair. An administrator can obtain the digital certificate through third-party Certificate Authority (CA) or security products that support digital certificates. CA Service Desk Manager also provides a server-side utility that can generate a digital certificate. It is located in `<NX_ROOT>/bin` directory as follows:

```
pdm_pki -p policy_code [-l certificate file] [-f] [-h]
```

-p

Identifies a unique policy code.

-f

Allows the utility to replace the existing public key with a new public key.

-l

Loads the public key stored in a X509 V3 certificate.

-h

Displays help on the command line window.

If you obtain a digital certificate through a third-party, CA, or security products, import it to where the CA Service Desk Manager server is located, and then associate it to an access policy. The administrator of the user application should obtain a digital certificate file that includes the content of an X509 V3 certificate in DER/ASN.1 format.

In addition, the certificate should contain only the public key of the public/private key pair. Using the `-l` option, the administrator should invoke the `pdm_pki` utility to load the certificate. The utility then loads the certificate, extracts the public key, converts the public key to BASE64 text format, and saves it with the access policy specified by the policy code.

When a digital certificate is generated by the `pdm_pki` utility, the administrator invokes the command in CA Service Desk Manager without the `-l` option. The utility then generates a public and private key pair (keys are RSA1024 bit keys). The public key is converted to BASE64 text format where it is stored along with the access policy specified by the policy code. An X509 V3 certificate is also created to hold the public key along with other information (the default pass phase is set as the policy code). Finally, the X509 V3 certificate is packaged with the private key to a standard portable certificate format of PKCS12. It is then saved in a file with a file name of *policy_code.p12*, depending on the policy code supplied. This file can then be exported to clients.

Note: If an access policy has already been associated with a public key of a certificate, users need to specify the `-f` option when calling the `pdm_pki` command in order to overwrite the existing public key with a new public key.

More information:

[Define an Access Policy](#) (see page 448)

Login to Web Services

The following describes the process flow for logging in to Web Services configured with PKI authentication:

Process	Description
Load the Digital Certificate and Extract the Private Key	<p>The digital certificate must be stored in secure storage on the user side, where it can be retrieved and used for logging in to Web Services.</p> <p>Example of secure storages include the following:</p> <ul style="list-style-type: none"> ■ Windows Certificate Store ■ Java Certificate Store (managed by java_keytool utility) ■ Certificate store (created by other CA security products). <p>A user application should be able to load the digital certificate and extract the private key using appropriate APIs, depending on user environments.</p>
Create a Digital Signature of the Plain Text Policy Code with the Private Key	<p>After the private key is extracted from the digital certificate, it can be used to generate a digital signature of policy code. Creating a digital signature encrypts a digest of a text with a private key. The digest algorithm must be standard SHA1, and the encryption algorithm should be RSA. Also, the binary digital signature should be converted to BASE64 text format before it can be used for logging in to Web Services. Depending on user environments, appropriate API calls should be used to archive this information.</p>
Invoke the Web Service Call	<p>A user application should invoke the Web Services method loginServiceManaged(), along with the plain text policy code and the BASE64 text formatted digital signature of the policy code.</p>
Obtain the Returned SID	<p>If the access request is authenticated, a plain text SID is automatically returned.</p>

After a SID is generated, it establishes a successful binding between a Web Service session and an access policy. The user application can invoke other web services methods with this SID, and all of its access to Web Services becomes controlled and managed by this access policy.

Session and Authorization

A successful validation returns a SID that is associated with the validated username, whether it is the user name supplied for login or the proxy contact specified in a policy. Because of this process, each CA Service Desk Manager user is assigned security rights that you may want enforced in your web service application.

For example, a specific user may have a Data Partition restricting which Requests the user can view. When using a SID for the user to get Request information, the CA Service Desk Manager system ensures the data partition is enforced.

Function Group security is also applied. For example, a user may not have access to the Call Manager function group. Invoking any web services methods, such as viewing or creating Requests, is denied because access is denied to the Call Manager function group.

When your application is finished doing work for a user, call the Logout() method to invalidate the SID.

Each SID expires after a period of inactivity. That is, a SID expires if the interval between method calls is greater than a certain timeout value. The timeout interval is set in Options Manager and is specified by the following CA Service Desk Manager option:

`'webservice_session_timeout'`

If this value is set to zero (0), a SID never times out. If this option is missing or not set, the default is one hour. If a Web Service method is called with an expired SID, a Fault is returned with an error code of UDS_SESSION_TIMEOUT the first time it is referenced, and UDS_BAD_SESSION each time thereafter.

To keep a SID active, call any web service method before the time out is reached. To keep the SID active without working the server, call serverStatus().

Web Service Option

You can direct traffic to a domsrvr other than the primary domsrvr for a web service session. Navigate to these options by going to Options Manager, Web Service. The following options control the web service session:

webservice_session_timeout

Sets the timeout value (in minutes) for Web Service sessions. If the time between successive Web method calls is greater than the value specified here, the session ID is marked expired and is no longer valid.

To prevent sessions from expiring due to activity, set this Option's value to 0. Sessions may still be invalidated by other methods, such as logoff routines.

webservice_domsrvr

Specifies the name of the object engine to be used by Web Services. If not installed, Web Services will use "domsrvr".

The value of the option must be a string beginning with the characters "domsrvr:"

Note: These options require restarting the CA Service Desk Manager server and the Windows IIS Service.

Access Control and Management

To minimize the potential problem of web services ticket flooding and to maintain the stability of the CA Service Desk Manager server, this version of CA Service Desk Manager Web Services uses an Access Control and Management system. It works primarily to handle the excessive service activities initiated by trusted user applications that can result from programming errors or exceptions. It also works as a barrier for controlling access to CA Service Desk Manager Web Services from malicious attackers. An administrator of a web service application is able to create and define an access policy in CA Service Desk Manager that controls access to CA Service Desk Manager Web Services from a web service application.

Note: A default access policy with a code of DEFAULT is provided. The default access policy contains no access restrictions and is only applied to sessions authenticated through username and password.

Define an Access Policy

To create any web services access policy, an administrator has to define an access policy.

To define the access policy

1. Click the Administration tab.
2. In the tree on the left, click Web Services Policy, Policies.

The Web Services Access Policy List page appears.

3. Click Create New.

The Create New Web Services Access Policy dialog appears.

4. Enter the information for the new access policy:

Note: The default value of -1 in any operation counter indicates that no restrictions apply to the corresponding operation. A value of 0 (zero) indicates that the corresponding operation is not allowed.

Symbol

(Required) Identifies a symbolic name of the access policy.

Code

(Required) Indicates the unique text that identifies this access policy.

Status

(Required) Identifies the status of an access policy. An inactive policy is not used.

Proxy Contact

Identifies the contact to use for all web services operations and CA Service Desk Manager security.

Default

Identifies the default policy. Set this policy as the default policy. Only one active default policy is allowed to exist. Creating a default policy automatically sets the current default policy to a non-default status.

Has Key

(Read-Only) indicates whether a public key has been associated with this policy. This field is updated when a public key is associated with a policy through the pdm_pki utility.

Allow Impersonate

Identifies the allow impersonate privilege. If this field is set, the policyholder can invoke the impersonate() web services method and create a web services session in the name of the user to be impersonated. Additional access authentication is not performed when creating the session. However, only when the access_level of the new user's access type is less than or equal to the grant_level of the proxy user's access type, can this method be successfully called.

Description

Indicates the detailed description of this access policy.

Ticket Creation

Indicates the number of ticket (call request, change order, and issue) insertion operations allowed per hour.

Object Creation

Indicates the number of CA Service Desk Manager object (other than ticket object) insertion operations allowed per hour.

Object Updates

Indicates the number of CA Service Desk Manager object update operations allowed per hour.

Attachments

Indicates the number of attachment-related operations allowed per hour.

Data Queries

Indicates the number of data query operations allowed per hour.

Knowledge

Indicates the number of knowledge-related operations allowed per hour.

5. Click Save.

Web Services Methods by Category

Each CA Service Desk Manager Web Services method belongs to a specific category. The following lists each category and their corresponding methods:

Ticket Creation

- createTicket()
- createQuickTicket()

- createRequest()
- createChangeOrder()
- createIssue()

Object Creation

- logComment()
- createAsset()
- addAssetLog()
- createAssetParentChildRelationship()
- createObject()
- createWorkFlowTask()
- createActivityLog()
- notifyContacts()
- addBookmark()
- addComment()
- createFolder()

Object Updates

- addMemberToGroup()
- removeMemberFromGroup()
- closeTicket()
- createLrelRelationships()
- removeLrelRelationships()
- deleteWorkFlowTask()
- updateObject()
- transfer()
- escalate()
- attachChangeToRequest()
- detachChangeFromRequest()
- changeStatus()
- clearNotification()
- updateLrel()
- deleteBookmark()
- updateRating()

Attachments

- createAttmnt()
- createAttachment()
- attachURLLink()
- deleteAttmnt()
- deleteComment()
- removeAttachment()

Data Queries

- impersonate()
- serverStatus()
- getBopsid()
- getConfigurationMode()
- getHandleForUserid()
- getAccessTypeForContact()
- getPermissionsGroup
- getObjectTypeInfoInformation()
- getRelatedList()
- getRelatedListValues()
- getGroupMemberListValues()
- getPendingChangeTasksForContact()
- getPendingIssueTasksForContact()
- getWorkFlowTemplates()
- getWorkflowTemplateList()
- getTasksListValues()
- getNotificationsForContact()
- getPolicyInfo()
- getAssetExtensionInformation()
- getLrelValues()
- getObjectValues()
- doSelect()
- doQuery()
- getPropertyInfoForCategory()

- `getValidTaskTransitions()`
- `getListValues()`
- `getListInfo()`
- `findContact()`
- `getAttmntInfo()`
- `getAttmntList()`
- `getBookmarks()`
- `getCategory()`
- `getComments()`
- `getContact()`
- `getDecisionTrees()`
- `getDocument()`
- `getDocumentTypes()`
- `getFolderInfo()`
- `getFolderList()`
- `getLrelLength()`
- `getPriorities()`
- `getRepositoryInfo()`
- `getStatuses()`
- `getTemplateList()`

Knowledge

- `createDocument()`
- `deleteDocument()`
- `doSelectKD()`
- `faq()`
- `attmntFolderLinkCount()`
- `getAttmntListPerKD()`
- `isAttmntLinkedKD()`
- `getDocumentByIds()`
- `getKDListPerAttmnt()`
- `getQuestionsAsked()`

- `modifyDocument()`
- `rateDocument()`
- `search()`

When an access policy is updated by CA Service Desk Manager, Web Services dynamically updates the corresponding policy information. Active Web Services sessions controlled under this policy remain controlled with configurations in the policy. New Web Services sessions for this policy to manage and control, take the latest configurations in effect.

Note: For information about each method, see the *CA Service Desk Manager Technical Reference Guide*.

Define an Error Type

Error types are assigned when creating tickets and an access policy defines one set of these error types. A CA Service Desk Manager Web Services user application may use low-level web methods to create a ticket (request, change order or issue), specifying one of these types to categorize the error addressed in the ticket. Error types can be used only with the high-level `createTicket()` method. Low-level methods, such as `createRequest()`, do not use error types.

More information:

[Simplified Web Services Access](#) (see page 456)

Web Services Error Types

CA Service Desk Manager Web Services also provides a defined set of default error types, which are created for *every* policy. These default types, designated as *internal* error types, can be deactivated, but cannot be deleted. In the product, you can use the Web Services Access Policy Detail page to see the default error types provided when a new policy is created.

The following information describes each internal error type:

ACCESS_ERROR

Indicates that the system failed to connect to or find a resource, such as a file, website, and so on.

EXCEPTION_FATAL

Indicates that the application is shutting down unexpectedly.

EXCEPTION_RUNTIME

Indicates that the application code encountered an exception.

LOGIN_ERROR

Indicates that the operator failed to gain access to the application.

Additional Error Types

The administrator of an access policy can add additional error types as described in the following information:

Error Type	Description
Ticket Template	Identifies a template of a Incident or Error, issue, or change order that you use to create a ticket when this error type is reported. Note: The owning policy's contact is used as the end user. The Ticket Type and Ticket Template Name define the ticket template.
Default	Indicates if this error type is the default for the policy. Only one default is allowed per policy. Note: A new default error type overwrites the existing default error type associated with the policy.
Active	Represents an active error type. Note: An inactive type does not create tickets.
Internal	Identifies the field as read-only, which indicates whether this error type is an internal, default error type.
Symbol	Indicates the symbolic name of the error type.
Code	Identifies the unique text identifier of the error type.
Description	Describes the detailed description of the error type.
Duplicate Handling	Defines the action to take when the product detects that an identical ticket already exists.

Error Type	Description
Return Data	Identifies the user-defined return message you can specify for the web method "createTicket()" to return to client applications. Return data might be used for indicating an action the application should take (Application Data Return), or a message (User Data Return) to display to the end user.

More information:

[Duplicate Ticket Handling](#) (see page 455)

Duplicate Ticket Handling

The Web Service Access Policy can detect and handle duplicate tickets, which is helpful for preventing ticket flooding. A ticket created with the potential of being a duplicate applies if all of the following conditions are true:

- At least one ticket of the same type (cr, iss, or chg) already exists and is ACTIVE.
- The existing ticket was created by the web service.
- The existing ticket was created with the same Policy and Error Type as the ticket being created.
- The "create date" of the existing ticket is within a specified threshold (for example, it was opened less than 2 days ago).

Note: The create date field is configured using the Maximum time interval for searching duplicates.

- The duplicate ID matches the one provided by users when invoking the createTicket() method.

Users can also assist in preventing duplicates by classifying tickets as being unique or different, based on criteria known to the user. To do this, add an optional string parameter to the createTicket Web Services call. If duplicate handling is on, the string parameter is inspected after other duplicate handling criteria match to determine whether this is a unique or duplicate call to this method.

Duplicate Ticket Results

If the create ticket action results in a duplicate, the existing Error Type may be configured to do one of the following:

Reconfigured Error Type	Results
Create the New Ticket and Ignore Duplicates	A new ticket handle and number are returned (default).
Do Not Create a New Ticket; Add an Activity Log to the Existing Duplicate Instead	The ticket handle and existing ticket number are returned.
Do Not Create a New ticket; Add an Entry to the CA Service Desk Manager Standard Log Instead	A ticket handle and existing ticket number are returned.
Create a New Ticket and Attach it as a Child to the Duplicate	A new ticket handle and number are returned.

Simplified Web Services Access

CA Service Desk Manager Web Services provides an abbreviated set of high-level web services methods that are simplified versions of existing web services methods. The majority of users applications do not have to completely rely on a large set of web services methods before requesting service desk services through CA Service Desk Manager Web Services. Working closely with user-defined access policies and using default parameters defined in the policies, this set of high-level web services methods can function with little knowledge of the CA Service Desk Manager object schema. Also, the high-level methods cover a common set of CA Service Desk Manager functionalities that most service-aware applications need.

The following describes the use of these high-level web services methods:

createTicket (SID, Description, Error_Type, Userid, Asset, DuplicationID)

You must specify an error type for the reported error if you use this method. The error type should contain the ticket template appropriate for the ticket you want to create. It should define the action to take in the case of a duplicate ticket, specify the data outputs, and finally, it must be associated to the access policy that is defined for the user application.

When this method is invoked, CA Service Desk Manager Web Services locates the current access policy and the error type required for the ticket creation. The following shows the sequence that CA Service Desk Manager Web Services uses for locating the proper error type:

- If a specific error type code is provided as input and it matches a error type that is associated to the policy, this error type is used, regardless of whether it is internal.

- If an error type is not specified or the previous step fails to locate an error type, the default error type is used if there is one defined for the policy.
- If a default error type is not defined for the policy or the previous step fails, the default error type defined for internal error types is used.

After an error type is defined, CA Service Desk Manager Web Services uses it to create a ticket. The proxy user defined in the access policy is used for the ticket creation if the userid is empty, and asset information is added to the ticket (if the input is not empty). After the ticket is created, CA Service Desk Manager Web Services returns both user data and application data, as specified by the error type.

closeTicket (SID, Description, TicketHandle)

Users can call this function to close an open ticket. It simply sets the status of an open ticket to 'close' and adds the input description to the activity log.

logComment (SID, TicketHandle, Comment, Internal_Flag)

Adds an entry with the input comment to the activity log for the open ticket.

getPolicyInfo (SID)

Lets users obtain the policy information that controls the current web services session. You can use this information as an indicator of server capacity for this user application. Users may want to adjust their web services calls to fit into the capacity.

By having this set of simplified Web Services APIs, a majority of users are spared the tremendous effort of understanding the complete set of web services API and CA Service Desk Manager schema. Using them simplifies and accelerates the process of creating service-aware enabled applications for these users.

Objects

CA Service Desk Manager treats each entity, such as a contact or an issue, as an *object*. These high-level objects are defined in majic (.maj) and mod (.mod) files on the CA Service Desk Manager server in the following directory:

/bopcfg/majic

Customized objects are defined in the following directory:

/site/mods/majic

Objects are essentially high-level wrappers around a database table.

An object's type (sometimes referred to as *factory*) defines the object. For example, request objects belong to the 'cr' type. Each object's type is defined by the "OBJECT" declaration in a majic file.

Note: All objects shipped with CA Service Desk Manager are enumerated in the CA Service Desk Manager Technical Reference Guide.

An object has *attributes*, which are essentially columns in a database table (do not confuse these with XML attributes). Web Services offers many methods to retrieve values for attributes. Many methods require you to name attributes for setting or retrieving values. You must use the attribute name assigned in the majic or mod file that defines the object, which can be different from the actual database name. Client sites can add additional attributes as a customization.

Note: For a list of all the attributes for each object, see the *CA Service Desk Manager Technical Reference Guide*.

Web Services uniquely identifies an object by its *handle*, which is a string value of the form *objectType:ID*, where *objectType* is the object's type (factory) name, and ID is a unique value. The ID value matches that of the 'id' attribute found in every CA Service Desk Manager object. Because the 'id' attribute is almost always indexed in the DBMS, using the ID portion of the object handle is especially valuable for forming efficient queries. Each object, regardless of its type, stores this value in an object attribute named "persistent_id".

Note: In prior releases, the ID portion of the handle was always a string of integers. In Unicenter Service Desk r11.0 and later, the ID portion may also be the string representation of a UUID, typically 32 characters.

The following information lists the object and factory names of the entities that use UUIDs:

Object Name	Factory Name
Contact	cnt
Asset	nr
Organization	org
Location	loc
Company/Vendor	ca_cmpny
Model	mfrmod

Handles are *persistent*; a handle representing a particular object is always unique for its lifetime, even across database migrations. Clients may want to take advantage of this persistence when working with fairly static objects, for example, Status or Contact Types.

Object handles are the key to using CA Service Desk Manager Web Services. Many methods, especially those that update data, require handles. Most methods that return object data also include the object's handle.

System Updates and Caching

Web Services caches information for object types. Type information is not cached until the type is referenced for the first time, and will cause a small delay.

To avoid any server or caching delays, you may want to run a primer client to activate the Web Services and cache the most popular object type information. The easiest way to cache the object type information is to perform repeated calls to `GetObjectTypeInfo()`. The object types to consider for this technique could be one of the following:

Object Type	Definition
cr	Request
chg	Change Order
iss	Issue
cnt	Contact
nr	Asset
wf	CA Workflow (Change Orders)
iss_wf	CA Workflow (Issues)
prp	Property (for Change and Issue)
prptpl	Property Template (for Change and Issue)
cr_prp	Request Property
cr_prptpl	Request Property Template

Add any additional object types your client code references.

Categories and Properties

The request, change order and issue objects all have a category field, which is used to classify the nature of the ticket. A category may have property objects, which are attached to the ticket when the category is assigned. Some of these may be marked *required*, which means a value must be supplied before the ticket can be saved (applies to both insert and update operations).

CA Service Desk Manager Web Services automatically supplies default values for any ticket created with the Web Services. The default value (currently, “-”) is obtained from the CA Service Desk Manager localized message catalog.

If you need to set property values at creation time, there are three ticket creation methods: `createChangeOrder`, `createIssue`, and `createRequest`. Each has a parameter with which you can pass in values for any properties. To discover which properties will be attached, you must find out the properties associated with the category you intend to assign to the ticket. The easiest method to use is `getPropertyInfoForCategory()`.

Note: For more information about the `getPropertyInfoForCategory()`, see the *CA Service Desk Manager Technical Reference Guide*.

To identify the valid values for a property, first find the property validation rule for the appropriate property template. To do this, request the `validation_rule` attribute when calling the `getPropertyInfoForCategory` method. Then, retrieve the associated `validation_type` for that rule. If the type is dropdown, you can then use the `getRelatedList` method to retrieve the values associated with the rule, using the "values" BREL attribute in the `prpval_rule` object.

Note: For more information, see the *CA Service Desk Manager Technical Reference Guide*.

To set property values after an update operation with `updateObject()`, you must query the property list after the update. `getRelatedList()` can help with this task.

Validation of property values through Web Service methods is not currently supported. For example, to assign property values to a validation rule with a validation type of drop-down option, you would have to write additional code to create property values while creating the drop-down option validation rule. Do not attach a property value to a check box validation rule.

Note: For more information about property validation rules, see the *Administration Guide*. For information about creating property validation rules through the CA Service Desk Manager interface, see the *Online Help*.

XML Object Returns

Many of the Web Services methods return an XML representation of CA Service Desk Manager objects. The Web Services uses a standard XML structure beginning with the following root element:

```
<UDSObject>
```

The format of the XML representation is described in the following table:

XML Element	Type	Description
<UDSObject>	N/A	Identifies the root node.
<Handle>	String	Identifies the object's handle.

XML Element	Type	Description
<Attributes>	Sequence	Identifies the attribute values. This holds zero or more elements for the object's attribute values.
<attrName0 DataType = "typeEnum">	String	<p>Identifies the <i>AttrName0</i>, which is an object attribute name as defined in the CA Service Desk Manager majic (.maj) or mod (.mod) file.</p> <p>This name may use dot-notation depending on the web method used.</p> <p>The element's value is the attribute's value. An empty element indicates a null/empty value for this object's attribute.</p> <p>The DataType attribute is an integer indicating the attribute's data type in the CA Service Desk Manager environment.</p> <p>Note: For information about the DataType attribute value, see the <i>Technical Reference Guide</i>.</p>

For example, a call to `getObjectValues()` can return information illustrated by the following:

```
<UDSObject>
  <Handle>cnt:555A043EDDB36D4F97524F2496B35E75</Handle>
  <Attributes>
    <Attribute DataType="2003">
      <AttrName>first_name</AttrName>
      <AttrValue>first name</AttrValue>
      <DisplayValue>Yaakov</DisplayValue>
    </Attribute>
    <Attribute DataType="2005">
      <AttrName>organization</AttrName>
      <AttrValue>342</AttrValue>
      <DisplayValue>Accounting Crew</DisplayValue>
    </Attribute>
  </Attributes>
  <Lists>
    <List name="mylist1">
      <UDSObject>...</UDSObject>
      <UDSObject>...</UDSObject>
    </List>
  </Lists>
</UDSObject>
```

Some methods, such as `doSelect()`, return a sequence of `<UDSObject>` elements contained inside a `<UDSObjectList>` element.

The `<Lists>` section holds zero or more `<List>` nodes. A `<List>` node holds zero or more `<UDSObject>` nodes. `<List>` elements are generally returned only when a specific request for list values is made.

When you want to return a list of values related to a specific object, you should use the *`getRelatedListValues`* method.

Note: For information about the *`getRelatedListValues`* method, see the *Technical Reference Guide*.

If a request is made just for a list with no attribute name, such as `actlog`, then the entire `<UDSObject>` is returned in the `<List>` section.

Specialized methods, like `getDocument()`, can of course be different. When a request is made for an attribute, the database value is returned. For SREL attributes, this may not be so useful. Requesting the assignee attribute of a Request returns an integer because the Contact REL_ATTR (foreign key) is its ID. For Unicenter Service Desk r11.0, the return data for attributes includes elements for the DBMS and common name value of SREL references.

Note: For information on setting SREL attributes with foreign key values, see the *Technical Reference Guide*.

ITIL Methodology

By default, the Web Services fully support the ITIL methodology. CA Service Desk Manager ITIL features let you take advantage of the ITIL methodology.

Incident or Problem Creation

CA Service Desk Manager supports ITIL methodology so that incidents and problems can be created using the CA Service Desk Manager Web Services. Both incidents and problems are held in the cr (Call_Req) object. Its *type* attribute distinguishes the record as an incident, problem, or request. To create an incident, problem, or request, call `createRequest` and specify the appropriate value for the *type* attribute.

The *type* attribute is a pointer (SREL) to the crt (Call_Req_Type) object, so you must pass a handle as the value.

The following code examples illustrate how to create an incident or problem by passing the correct crt object handle to the `createRequest` method. Setting the *type* attribute in the name-value pair that is passed as a parameter to `createRequest`, creates the tickets:

Example: Syntax for a Problem

```
attrVals = {"summary", "A new problem", "description", "new problem", "type", "crt:181"}
USPSD.createRequest(SID, creatorHandle, attrVals, template, new String[0], new String[0])
```

Example: Syntax for an Incident

```
attrVals = {"summary", "A new incident", "description", "new incident", "type", "crt:182"}
USPSD.createRequest(SID, creatorHandle, attrVals, template, new String[0], new String[0])
```

More information:

[Default Handles](#) (see page 466)

Query for Incidents or Problems

To retrieve incidents or problems, include the *type* attribute of the *cr* object in the *where* clause. The following example illustrates a *where* clause for retrieving all active incidents. This *where* clause could be used with methods that perform queries for 'cr' objects, such as *doSelect* and *doQuery*:

```
type.id = 182 AND active = 1
```

The '182' is the ID portion of the handle representing Incident types.

Note: For more information, see the *crt (Call_Req_Type)* objects table illustrated in [Default Handles](#) (see page 466). For more information about forming proper queries, see [WHERE Clause](#) (see page 241).

Attach an Incident to a Problem

The Web Services can create associations among ITIL tickets, such as associating one or more incidents to a problem. The *parent* attribute on a ticket is used to create parent-child relationships between the *cr* objects that are used for request, incident and problem objects. Setting the *parent* attribute of a ticket to point to another ticket creates the relationship.

For example, a newly created incident relates to an existing problem. To associate the incident to the problem and set the *parent* attribute of the incident, use *UpdateObject*. The following example code illustrates this by setting the *parent* attribute to the handle of an existing problem ticket:

```
attributeValues = {"parent", "cr:12346"}  
USPSD.UpdateObject(SID, incidentHandle, attributeValues, new String [0])
```

Attach a Problem to a Change Order

Incidents and problems can be linked to change orders with the *attachChangeToRequest* method. The following example code uses this method to simultaneously create a change order and attach it to a problem. In the example, "cr:12347" is the object handle of the problem—it passes a blank handle for the fourth parameter, which causes the method to create a change:

```
UPSPSD.attachChangeToRequest(SID, creatorHandle, "cr:12347", "", new String[0], "activity description")
```


Configuration Items

ITIL methodology uses the term *configuration item* (CI) to refer to hardware, software, and other IT resources. This term refers to the “nr” object stored in the CA-owned resource database table. All methods that use *asset* objects also work with CIs. This is only a difference in terminology.

Use the Web Services

The information in this section provides you with the fundamentals for using the CA Service Desk Manager Web Services. Example code using the Web Services exists in the following CA Service Desk Manager installation directory:

```
<NX_ROOT>/samples/sdk/websvc/java
```

The sample code is written in Java using Apache Axis for SOAP messaging.

Logins

Before any Web Services method can be used, a SID (session ID) must be obtained from one of these methods: `login()`, `loginService()`, and `loginServiceManaged()`. The first two methods require a username and password that are validated exactly the same as the CA Service Desk Manager web interface; the contact’s Access Type specifies the validation method. The third method requires a public/private key pair, where login request encrypted with the private key can only be decrypted through the public key, and vice versa.

More information:

[External Specifications](#) (see page 438)

How to Perform Common Tasks

The Web Services is a flexible and powerful API into CA Service Desk Manager, but it requires some knowledge of the object structure used by the product as follows:

1. Familiarize yourself with the information about objects and attributes in the *CA Service Desk Manager Technical Reference Guide*. This guide lists the attributes of each object in the system, which is essential because many of the Web Services methods require attribute names.

2. Review the Web Services methods, especially generic ones. For example, if your application must display all the activity logs for a request, first identify how the activity logs relate to the request. The *CA Service Desk Manager Technical Reference Guide* shows that the request object has two lists of Activity Logs: The `act_log` (which shows only non-internal logs), and `act_log_all` (which lists all activity logs).
3. Identify which Web Services methods are needed. To get lists attached to an object, use `getRelatedList()` or `getRelatedListValues()`.

Default Handles

Some default data provided by the product is frequently used. Instead of looking up handles for these objects, some of the commonly used ones are listed in the following tables.

Note: While the handles do not change, the legible symbols may be edited.

Contact Type (Object name: `ctp`)

Handle	Note
<code>ctp:2307</code>	The “Analyst” type
<code>ctp:2310</code>	The “Customer” type
<code>ctp:2305</code>	The “Employee” type
<code>ctp:2308</code>	The “Group” type

Impact (Object name: `imp`)

Handle	Note
<code>imp:1605</code>	Impact ‘None’
<code>imp:1600</code>	Low impact ‘5’
<code>imp:1601</code>	Medium-low impact ‘4’
<code>imp:1602</code>	Medium impact ‘3’
<code>imp:1603</code>	Medium-high impact ‘2’
<code>imp:1604</code>	High impact ‘1’

Priority (object name: `pri`)

Handle	Note
<code>pri:505</code>	Unassigned priority ‘None’

Handle	Note
pri:500	Low priority '5'
pri:501	Medium-low priority '4'
pri:502	Medium priority '3'
pri:503	Medium-high priority '2'
pri:504	High priority '1'

Severity (object name: sev)

Handle	Note
sev:800	Low severity '1'
sev:801	Medium-low severity '2'
sev:802	Medium severity '3'
sev:803	Medium-high severity '4'
sev:804	High severity '5'

Call Request Type (object name: crt)

Handle	Note
crt:180	Request
crt:181	Problem
crt:182	Incident

Query for Requests, Issues, or Change Orders Assigned to a Contact

One of the most common operations is retrieving the active requests assigned to an analyst (assignee). You can use one of several methods, such as `doQuery()` (to get a list reference), or `doSelect()` (to get the values immediately). Assuming the assignee's handle is already known, the where clause to use is as follows:

```
assignee.id = U'<assigneeID>' AND active = 1
```

In this where clause, `<assigneeID>` is the id portion of a contact handle, value, such as "555A043EDDB36D4F97524F2496B35E75".

This where clause works for requests, change orders and issues because they all have the 'assignee' and 'active' attributes, and they mean the same thing for all three object types. The 'active = 1' portion of the where clause restricts the search to active requests.

The Active Flag

Most CA Service Desk Manager objects have a field called 'active' or 'delete_flag'. This is actually an SREL pointer to the Active_Boolean_Table object or Boolean_Table object. Consider adding these fields to your queries to filter objects marked as Inactive by the system administrator. For querying purposes, search for 'delete_flag = 0' to find active records and 'delete_flag = 1' for inactive records. For example, the following pseudo-code demonstrates using doSelect() to retrieve values for all active Request Status objects:

```
doSelect(SID, "crs", "delete_flag = 0", -1, new String[0]);
```

To set an object to active or inactive, you need to pass the handle of the Boolean object representing either true or false. These handles do not change, so you can safely hard-code them. These are listed as follows:

Active_Boolean_Table	Boolean_Table
actbool:4551 = 'Active'	bool:200 = 'False'
actbool:4552 = 'Inactive'	bool:201 = 'True'

Retrieve Related List Length

When requesting attribute values from an object, such as with getObjectValues(), you can get the length of a related list by requesting the following attribute:

```
"<listName>.length"
```

For example, to get the number of Activity Logs for a certain request, pass the following to getObjectValues():

```
"act_log_all.length"
```

Note: This is the only way you can use list names in these types of methods.

Chapter 10: Integrating with Other Products

You can integrate CA Service Desk Manager with some CA products; however, we do not detail integrations with all CA products.

Note: For detailed information about additional integrations with CA Service Desk Manager that are not described in this section, see the *CA Unicenter Service Desk Integrations Green Book* at <http://ca.com/support>.

This section contains the following topics:

[CA Workflow Integration](#) (see page 469)

[CA NSM Integration](#) (see page 473)

[CA Portal Integration](#) (see page 514)

[Mainframe Product Integration](#) (see page 521)

CA Workflow Integration

A workflow denotes the tasks, procedural steps, organizations or individuals involved, required input and output information, and the tools necessary for managing and maintaining each step in a business process. The workflow service provides a total solution to help you manage the business processes.

In general terms, *workflow* is best described as the automation of a business process, in whole or part, during which documents, information, or tasks are passed from one participant to another for action according to a set of procedural rules.

CA Workflow is a generic, high performance, and scalable workflow management system, that allows for the definition, management, and execution of workflows, and provides a generic workflow solution.

CA Workflow is integrated into CA Service Desk Manager using web services.

CA Workflow Components

CA Workflow uses the following components that are delivered with the Workflow Embedding Kit:

- [Workflow Design Environment](#) (see page 470)
- [Workflow Server](#) (see page 471)
- [Worklist](#) (see page 471)

The CA Workflow Design Environment

You use the CA Workflow Design Environment to create and manage workflows. The CA Workflow Design Environment is your primary tool for creating and managing process definitions and workflow instances, and for controlling workflow participant interaction.

A company creates a business process to identify the resources, manual and automated activities, and activity relationships to realize its business goals. A workflow partially or fully automates the business process by defining the business process activities in a process definition. These workflow activities commonly include passing forms, tasks, or information from one resource to another, where the resource may be human or software.

A process definition is a representation of your business process. A process definition is comprised of nodes, events, roles, actors, work, and the criteria for process start and process end.

While process definitions represent what you want to happen in your business process, process instances represent what is actually happening. By running a process definition, you create a process instance. You can create multiple process instances of the same process definition, also called a process definition instance.

Start the CA Workflow Design Environment (Windows)

At any time, you can start the CA Workflow Design Environment to create and manage workflows. To start the CA Workflow Design Environment, select Start, Programs, CA, Service Desk, CA Workflow IDE.

Note: You can also start the CA Workflow Design Environment by double-clicking the *ide.bat* file, located in the `$NX_ROOT\site\Workflow\Client` directory. If the IDE classpath requires additions to accommodate application requirements, you must modify the APPCPATH parameter in the *ide.bat* and use the *ide.bat* to invoke the IDE.

Start the CA Workflow Design Environment (Linux)

At any time, you can start the CA Workflow Design Environment to create and manage workflows. To start the CA Workflow Design Environment, run the *ide.sh* file, located in the `$NX_ROOT/site/Workflow/Client` directory. The CA Workflow IDE is not supported on UNIX.

Note: To accommodate application requirements, the *ide.sh* file has an APPCPATH parameter which can be modified to add to the classpath.

CA Workflow IDE Parameters

When you access the CA Workflow Design Environment, a login screen displays. This login screen is pre-populated with user name, password, and URL connection. For streamlined access to the IDE, you can completely bypass the login screen by using a combination of the following command line parameters:

```
-u <username>  
-p <password>  
-url <PM URL>  
-t <eIAM Safe Session token>
```

You can invoke the IDE by providing either a user name and password (-u and -p) or by providing a token (-t). The -t token refers to an exported SafeSession (also referred to as an artifact) which is a String. This allows for parent products to re-use any existing SafeSessions they may have, rather than prompting for a user name and password again.

The -url is a required parameter that must be supplied regardless of whether you use the -u and -p userid/password method or the -t token method.

For example, [-u, -p, -url] or [-t, -url]

To use this facility to bypass the IDE logon screen, edit either the ide.bat file (for Microsoft Windows) or ide.sh (for Linux) to add these parameters to the list of parameters passed to the java command.

Note: This facility cannot be used with the ide.exe.

CA Workflow Server

The CA Workflow server exposes runtime execution and management services for workflow processes. The CA Workflow server consists of these components:

- Actor Adapters allow for expansion of workflow activity functionality
- The Process Engine executes process definition instances
- The Process Manager provides management for process definitions as well as process instances

All workflow clients, the Process Designer, the Worklist, and both the Java and Web Services API use services provided by the CA Workflow server.

Worklist

The Worklist is launched using a web browser (for example, `http://<hostname>:<port number>/wl`) or using API calls from an embedding product.

CA Workflow Access

CA Service Desk Manager provides workflow management using a common Workflow engine. The workflow system provides a graphical workflow definition environment, sophisticated branching mechanisms, and the ability to interact with individuals who do not have direct access to the CA Service Desk Manager application for gaining approvals. You can use CA Workflow authentication and add users to the CA Workflow groups.

CA EEM and CA Workflow

All logins to CA Workflow are authenticated by CA EEM. A user must have a CA EEM user record in order to access the CA Workflow IDE or Worklist application. The CA Workflow administrator, specified during CA Service Desk Manager configuration, has full access to CA Workflow. By default, this administrator is used by CA Service Desk Manager for the CA Workflow integration. This user account is set by the `cawf_username` and `cawf_password` Options in Options Manager. You must make sure the username and password set in these options are correct and the user has full access to CA Workflow resources within CA EEM.

Resource Classes

CA Workflow also uses CA EEM to restrict access to specific CA Workflow functions. The access is controlled by two Resource Classes:

IDE

The IDE resource has a single action named *login* for login access to the IDE. A user must have permission for this action to log in to the CA Workflow IDE application.

Process

The Process resource has the single action named *start* for the ability to start a process instance. A user must have permission for this action to start processes from within the Worklist web application. All users have access to the CA Workflow Worklist application to view and perform workitem tasks. This permission is only for starting new instances from the Worklist.

Note: These resource classes are defined with the CA Service Desk Manager application instance in CA EEM. When logging in to the CA EEM Web user interface, you need to specify the CA Service Desk Manager application instance in order to see the resources, policies and groups discussed here.

Add Users to CA Workflow Groups

Users who need to either log in to the IDE or start process instances need an authorization grant to the resources and the two actions. The CA Service Desk Manager configuration adds two policies to CA EEM that grant access to these resources. Two user groups are also added that are granted rights to the policies: CA Workflow Administrators and CA Workflow Process Initiators. Adding users to the CA Workflow Administrators group gives them access to the IDE. Adding users to the CA Workflow Process Initiators group allows them to start processes from the Worklist application.

To add/remove users from the groups on the computer where CA EEM is installed

1. Select Start, Programs, CA, Embedded Entitlements Manager, EEM UI.
The login window displays.
2. Select the CA Service Desk Manager application and enter the CA EEM administrator name and password.
The main CA EEM window appears.
3. Select Manage Identities.
4. Select the Users Search, enter search criteria, and click Search.
5. Select a user in the result list.
6. On the user details display, add/remove group membership in the Application Group Membership section.
If this section is not displayed, click Add Application User Details.
7. Click Save.
The users are added to the groups.

CA NSM Integration

The CA NSM Integration automatically installs when the CA Service Desk Manager server and CA NSM are installed on the same computer.

Windows

When the CA Service Desk Manager server and CA NSM are installed on different computers, you must also install and configure a CA Service Desk Manager secondary server or the standalone CA NSM Integration component on the CA NSM server.

UNIX

When the CA Service Desk Manager server and CA NSM are installed on different computers, you must install a CA Service Desk Manager secondary server on the CA NSM Server.

The CA NSM integration lets you do the following:

- Automatically control network management issues detected by CA NSM Event and Alert Management.
- Automatically coordinate critical management events detected by CA NSM with incident management.
- Determine impacts from your network administrators and service desk staff.
- Avoid requests or incident storms.
- Automatically update requests or incidents.
- Implement business rules as a best practice for network and service support management.
- Auto-dispatch new request or incident occurrences.
- Automate the interaction with CA Service Desk Manager to reduce the workload of the customer support staff by eliminating some manual tasks.
- Reduce the number of requests and incidents opened by users when a problem in the enterprise occurs because the request or incident has been created automatically as soon as it occurs.

How to Integrate with CA NSM

Complete the following steps to integrate CA Service Desk Manager and CA NSM:

1. Make sure that both CA Service Desk Manager and CA NSM have been properly installed and configured.

Important! Changing default settings when you integrate CA Service Desk Manager with CA NSM can cause unexpected results. For example, to avoid instantaneous creation of thousands of CA Service Desk Manager requests during integration, the following parameters in the `NX.env` file located in `$NX_ROOT` (UNIX) or `installation-directory` (Windows) default to NO:

```
@NX_TNG_OBJECT_UPDATED_SUBSCRIBE=NO
@NX_TNG_OBJECT_ADDED_SUBSCRIBE=NO
@NX_TNG_OBJECT_DELETED_SUBSCRIBE=NO
@NX_TNG_OBJECT_STATUS_UPDATED_SUBSCRIBE=NO
```

2. Integrate CA Service Desk Manager and CA NSM.
3. Create owned assets.
4. Monitor Event Console messages.
5. Filter events that you do not need.
6. Troubleshoot the integration, when necessary.

More information:

[Create Owned Assets](#) (see page 477)

[How to Monitor Event Console Messages](#) (see page 478)

[Filter Rule Considerations](#) (see page 479)

[Troubleshoot Integration](#) (see page 501)

Configure the Converter on UNIX

After integrating CA Service Desk Manager and CA NSM, you must also configure the converter using the `pdm_edit` utility as follows:

1. On the CA Service Desk Manager primary server installation, switch to the `$NX_ROOT/samples/pdmconf` directory.
2. Enter the following command to start the `pdm_edit` utility:
`pdm_perl pdm_edit.pl`
3. Answer the prompts according to your requirements, and when you get to the main menu, type **N** to select the Edit UNI Converters (UNIX_ONLY) option.
4. Type **A** to add, then enter the name or IP address of the CA NSM server computer (configured as the CA Service Desk Manager secondary server) when prompted for a host name.
5. The script prompts you for an IP address. Enter the IP address of the CA Service Desk Manager primary server computer.
6. Press Enter to return to the main menu, and then type **X** to save and exit.

This process creates a file called `pdm_startup.rmt` that stores your new configuration values.
7. Create a backup of the `pdm_startup.tpl` that resides in the `$NX_ROOT/pdmconf` directory of the CA Service Desk Manager primary server installation, then replace it with the newly created `pdm_startup.rmt` file.
8. Run `pdm_configure` on the CA Service Desk Manager primary server without making any changes. This puts your new configuration settings into effect the next time you start the CA Service Desk Manager server.

Important! Do not reinitialize your database when you reconfigure.

9. As the privileged user, run `pdm_proctor_init` on the CA NSM server to start the CA Service Desk Manager proctor.
10. As the privileged user, restart the CA Service Desk Manager services to start the CA Service Desk Manager daemons. Run `pdm_status` to display the status of the daemons.

Consider the following information as you make selections during the integration process:

- Before the process of “receiving a CA NSM event and creating a Request in CA Service Desk Manager” can work, you must install and configure all components, and ensure that they are active.
- An event would only become lost if the event converter service has been stopped. The event converter service queues CA NSM events when the CA Service Desk Manager system is down (meaning it has been paused instead of stopped from the Microsoft Windows Services Panel). When CA Service Desk Manager is restarted, it processes the queued events.
- The CA NSM event converter service queues events up to a maximum specified by the `NX_TNGCNV_QUEUE_SIZE` environment variable.
- When the CA NSM repository is rebuilt after integration with CA Service Desk Manager, CA Service Desk Manager menu entries are lost. To restore them, you must re-run the integration on the CA NSM Windows computer. To do this, run `integAHD.exe` located in *installation-directory\bin*.
- If the CA NSM event converter starts during CA NSM events generation, events that occur before the event converter is fully initialized are lost.

Post Integration Process

After integration, perform the following steps:

1. Create assets that are owned by CA Service Desk Manager which represent WorldView managed objects.
2. Monitor event console messages to define message records and message actions as needed to create requests automatically.
3. Make the necessary entries in the `topology.cfg` file, and define filter and writer rules to filter unwanted events.
4. Troubleshoot problems, if necessary.

More information:

[Create Owned Assets](#) (see page 477)

Create Owned Assets

Many CA products create assets in the CA MDB; however the assets are not automatically available to CA Service Desk Manager. Usually a CA Service Desk Manager organization only wants to track assets that the organization owns and that were acquired through a formal process. Some devices that are detected on the network by other CA products should most likely not become available to CA Service Desk Manager automatically. For example, the laptop computer of a visiting consultant is probably of no interest to the CA Service Desk Manager organization.

To make a discovered asset in the CA MDB available to CA Service Desk Manager, use the Discovered Asset selection dialog that is available from the Asset Search or Create New Asset forms in the Analyst Web Interface.

Note: The Discovered Asset selection dialog is available on the Web Client.

If the WorldView integration is being used, discovered Assets can be made available to CA Service Desk Manager from the `pdm_nsmimp` command-line utility. The CA NSM 2D/3D map and Unicenter Explorer let you access and create requests by right-clicking a managed object to display two menu options: Create Request and Request List. These menu options are added to CA NSM when you integrate with CA Service Desk Manager.

How to Register Discovered Assets

The `pdm_discimp` utility is used for batch registration of non-CA Service Desk Manager Discovered Assets. You can use this utility to search the CA MDB for assets that other software products registered, and register them as CA Service Desk Manager assets, so they can be used in CA Service Desk Manager. This utility is an interactive batch process.

The logic is similar to the Discovered Assets dialog that can be launched from Asset Search/List web form. This program queries the `ca_logical_asset`, `ca_asset`, and `ca_logical_asset_property` tables, using various parameters, and attempts to register new CA Service Desk Manager Assets from the discovered values.

Note: If the processing results in a blank Asset Label, the value found for the Host Name or DNS Name is used as the Asset Label. Assets must have at least a Label and Asset Class to be registered for use in CA Service Desk Manager.

The following queries are performed to select the appropriate records to process, because of the structure of the CA MDB and CA Service Desk Manager architecture.

1. A query retrieves the rows from a join between the `ca_logical_asset` and `ca_asset` tables that match label, serial number, tag, and hostname.
2. For each resulting row, a query is performed against `ca_logical_asset_property` to match `dns_name` and `mac_address`.

The asset from the first query is selected for registration when the second query results in rows being returned.

Note: This process can affect performance.

How to Monitor Event Console Messages

The basic CA Service Desk Manager installation automatically monitors status changes of monitored objects and the addition of new managed objects. You can set up your environment to use AHD.DLL to monitor console messages on Windows and send event messages to the CA Service Desk Manager server.

We recommend using AHD.DLL to send events to CA Service Desk Manager. However, other methods are available when the event console is running on a UNIX system.

Event Management in CA NSM lets you identify events to which you want to respond and specify one or more actions to initiate automatically. After you define a message and an associated action, the action is performed automatically whenever the event is encountered.

After integration with CA Service Desk Manager, when the event message occurs, you can send generic event data to CA Service Desk Manager to do the following:

- Create requests or update existing requests
- Create and post announcements on the CA Service Desk Manager scoreboard

To begin the integration, do the following:

1. Use AHD.DLL to create a CA NSM `msg_action` record.
2. Use the `cawto` command to pass that message record to the event console.

Note: Use the default filter and writer rules provided with CA Service Desk Manager until you are sure that your system is integrated. Do not modify or remove the default rules until you have witnessed CA NSM events creating requests.

Send Generic Event Data (UNIX Only)

To send generic event data to filter daemons in CA Service Desk Manager, use the `uniconv` daemon in a message action in CA NSM Event Management.

Generic event data can then be used to generate requests automatically in the same manner as `AHD.DLL` is used on Windows. This method is the preferred method on UNIX.

Post Announcements Automatically

When you integrate CA Service Desk Manager with CA NSM, you can create and post announcements on the CA Service Desk Manager scoreboard from CA NSM.

Filter Rule Considerations

After you set up integration correctly and events are creating or updating requests as defined, you can modify your filter rules to filter out unwanted events.

The following list provides some tips for coding filter rules:

- Anytime you change a filter rule, restart CA Service Desk Manager.
- Use the default writer rules while testing your filter rules.
- When coding the *node_ID*, *user_ID*, or *event_ID* parameters in the filter rule, match the case of the incoming event. To help ensure that you have the case specified correctly, create an event in CA NSM on the parameter you are testing. This action creates a request with the information you need. Match the case on the request with the case in your filter rule. This includes the case of incoming console messages.
- When you are coding a filter rule for a specific event, the incoming text on the event may have the following format: "Object_Status_Updated minor." Code your filter rule to look for the following text:

```
tngr::*::*:Object_Status_Updated.*minor.*:::(0,1)
```

- Blanks in the event may actually be unprintable characters. Code the filter to include one or more characters with a period and asterisk (`.*`), instead of the blank space (`' '`). Also include `.*` at the end of the event for any unprintable characters that may be at the end of the event text.

- To help ensure the filter works when you are coding more than one parameter on it, such as combining the *node_ID*, *user_ID*, and *event_ID*, code only one initially and verify that it works. Then, add the next parameter, verify that it works with the first parameter, and so on, until finished.
- For ease in debugging, order your filter rules from the most specific at the beginning of the file, to the most general at the end of the file.

Event writer rules follow the same set of debugging practices as filter rules. If you are using the CR_CREATE action and you have included a template, verify that the template exists.

How to Filter Events Automatically

You can configure CA Service Desk Manager to create requests automatically for error and event messages generated by network devices and other event sources. Defining how events are filtered lets you automatically detect when an event occurs and initiate the actions specified for handling that event. This method can improve overall operating efficiency and reduce the potential for error.

You can configure CA Service Desk Manager to do the following:

- Filter events based on host, content, type, and other information from CA NSM
- Create requests automatically for error and event messages generated by network devices
- Execute commands automatically in response to an event

To filter events automatically, do the following:

- Define filter rules that identify the events that require special handling
- Define event writer rules that identify the actions to perform automatically when these events are encountered

Filter and event writer rules use a generic event data structure to receive and send data. They are defined using a text editor and stored and edited on the CA Service Desk Manager server only.

Any event is eligible for processing by filter and event writer rules. Some examples of events you may want filter and event writer rules to process are the following:

- Hardware failures
- Failures of scheduled batch processes
- Excessive CPU usage
- High paging rates

- Unusual file activity
- Unusual security conditions

You can configure any number of filter daemons, event writer daemons, and event sources on different computers. The locations and relationships of these elements are stored in the topology file on the CA Service Desk Manager server.

Note: When integrated with CA NSM, you can use the `uniconv` daemon to automatically generate requests (this is the preferred method for UNIX). `uniconv` is used in a message action in CA NSM Event Management.

Generic Event Data

Information about events is communicated using a *generic event* data structure. The generic event data structure consists of the following data elements:

Source Type

Identifies the format for the rest of the event.

Node ID

Identifies the device name or ID.

User ID

Identifies the user name or ID (when applicable).

Major Source

Identifies the source application ID.

Minor Source

Identifies the agent of event or further delineation.

Date/Time

Identifies the event date and time.

Event ID

Identifies the source event string that triggered the event.

Event Data

Identifies the associated event data.

Severity

Identifies the measure of the event's importance.

Handle

Identifies the daemon-supplied string resulting from rules.

Handle Source

Identifies the daemon identifier that assigned the handle.

Handle Status

Identifies the status as create, update, or terminate.

Status Count

Identifies the number of updates.

Filter Rule Setup

Filters receive information from event sources that can then be passed to event writers. Because network devices generate hundreds of event messages, you can use filter rules to isolate those that can be used to create requests. Filter rules let you do the following:

- Determine which network events or traps to report and how.
- Control what happens when each type of event or trap is reported. Valid actions include ignoring, reporting on, and marking.

When integrated with CA NSM, filtering lets you manage events from all systems managed by CA NSM, including SNA and TCP/IP networks.

By filtering events, you can retrieve specific information about a particular node, user, or workstation, and then pass that information to event writers. You can define event filter rules to screen for your particular needs, helping you identify suspicious events and correct them before they cause problems.

Note: It is helpful to be familiar with UNIX regular expressions when writing filter rules. You can also use UNIX regular expressions on Windows.

Filter Rule Definitions

Filter rules use UNIX regular expression matching to determine if an event has a matching filter rule and blocks events that do not filter. A special filter (ID=5) passes the incoming generic event unchanged to the writer. The filter passes only filtered events and assigns a handle source of *filter* and handle status of *create*, *update*, or *terminate*.

CA Service Desk Manager uses the major source, event ID, node ID, and user ID fields from the generic event data structure to find a matching filter rule.

Important! Filter rules pass information from events to another daemon that uses event writer rules, which can automatically create requests.

The default filter rule file, `tngfilter_rule.dat`, is located in `$NX_ROOT/site/eh/IP` (UNIX) or `installation-directory\site\eh\IP` (Windows) on the CA Service Desk Manager server. `$NX_ROOT` or `installation-directory` is the directory where you installed CA Service Desk Manager and `IP` is the IP address of the computer on which the filter resides. Typically, this computer is the CA Service Desk Manager server.

This file also contains many comment lines that show you how to set up various filter rules. Comment lines begin with the pound sign (#).

Use a text editor to view, update, and save the filter rule file. However, do not use an editor that leaves extra formatting characters in the file. We recommend WordPad for Windows users and vi for UNIX users.

Syntax (Filter Rules)

The filter rule syntax is:

```
source_type:::node_ID:::user_ID:::event_ID:::filter
```

where:

Fields in the rule are separated by three colons (:::). You can use an asterisk (*) as a wildcard character, which means the value of the parameter is ignored when determining whether an incoming event matches the filter rule pattern.

Note: The first four parameters in the filter rule describe a pattern to compare against incoming events. This pattern is used to determine if the filter specified in the last parameter of the rule should process an incoming event.

Parameters (Filter Rules)

source_type

Specifies the type of source directing events to the filter. The major source field of the incoming event is compared to the value in this parameter. The type of converter that is sending events to the filter generally determines the value of this parameter. Valid values are:

uni

Identifies the UNIX CA NSM converter.

tng

Identifies the Windows CA NSM converter.

If an incoming event matches several rules when source types and event IDs are compared, then node IDs are compared.

node_ID

Specifies the node ID where the event originated. This parameter must exactly match the node ID of the incoming event, or this filter rule is not used. An asterisk (*) indicates that the node ID is ignored when determining if an incoming event matches this filter rule pattern.

Rules that match the node ID of the incoming event take precedence over rules that have asterisk (*) specified for node ID.

If an incoming event matches several rules when source types, event IDs, and node IDs are compared, user IDs are compared.

user_ID

Identifies a user associated with the event. Use this parameter to execute a particular action for events from a specific user. Many events will not have specific users associated with them. This parameter cannot contain a UNIX regular expression. You must supply an exact user ID or an asterisk. An asterisk (*) is the default and indicates that the user is ignored in selecting the rule.

Rules that match the user ID associated with the event take precedence over rules that have asterisk (*) specified for user ID.

event_ID

Specifies the event identifier. You can use a UNIX regular expression. A period and an asterisk (.*) matches any event name. For example, you can specify [Aa].* to match any event that starts with uppercase or lowercase "a."

If the *event_ID* parameter is simple text (not a regular expression), it must exactly match the generic event's event ID. Partial matching does not count. If the *event_ID* is a regular expression, the length of the regular expression match is used to determine how well the event filter rule's *event_ID* matches the generic event's event ID.

If an incoming event matches several rules when source types and event IDs are compared, node IDs are compared.

filter

Specifies which filter to use to process the incoming event and the parameters that the filter uses. The format is:

(filter_id, filter_parameter1, filter_parameter2, ...)

Valid values for *filter_id* are:

Value=0

Report all events

Value=2

Ignore events that occur infrequently

Value=3

Ignore outages shorter than the length specified in *filter_parameter1*

Value=4

Ignore bursts shorter than the length specified in *filter_parameter1*

Value=5

Pass all events to destination without modification

The number of filter parameters varies for each *filter_id*, as the following table shows:

filter_ID	filter_parameter1	filter_parameter2	filter_parameter3
0	0=no, 1=yes Indicates whether events should always be reported as separate problems.	None	None
2	Number of occurrences Specifies the number of times an event must occur during an interval to be important.	Interval Specifies the elapsed time, in seconds, during which an event must occur to be important.	0=no, 1=yes Indicates whether events should always be reported as separate problems.
3	Interval Specifies the elapsed time, in seconds, during which an event must occur to be important.	0=no, 1=yes Indicates whether events should always be reported as separate problems.	None
4	Interval Specifies the elapsed time, in seconds, during which an event must occur to be important.	0=no, 1=yes Indicates whether events should always be reported as separate problems.	None
5	None	None	None

Note: The first four parameters in the filter rule describe a pattern to compare against incoming events. This pattern is used to determine if the filter specified in the last parameter of the rule should process an incoming event.

Event Writers

Event writers specify what CA Service Desk Manager should do when it detects an important event from the network. The event writer uses converted events (events using the generic event data structure) from an event source and events that have passed through a filter to create requests.

Using event writers, you can do the following:

- Identify the type of event received from an event source
- Specify the action to perform when the event writer receives events of this type. Valid actions include:
 - Creating requests
 - Updating existing requests
 - Executing a command
- Indicate whether logging occurs

Event Writer Rule Definitions

When CA Service Desk Manager is installed, a default configuration is supplied that runs on the CA Service Desk Manager server. This default configuration sets up a single event source, filter, and writer, and has a filter rule that passes all events to the writer. The default writer rule creates requests for all events that it receives.

The default writer rule file, `tngwriter_rule.dat`, is located in `$NX_ROOT/site/eh/IP` (UNIX) or `installation-directory\site\eh\IP` (Windows) on the CA Service Desk Manager server. `$NX_ROOT` or `installation-directory` is the directory where you installed CA Service Desk Manager and `IP` is the IP address of the CA Service Desk Manager server.

You can view the contents of this file to see the default writer rule definitions and comments describing the format of the writer rules. The comments are at the beginning of the file, and the writer rules are at the end. Comment lines begin with the pound sign (#).

Use a text editor to view, update, and save the writer rule file. However, do not use an editor that leaves extra formatting characters in the file. We recommend WordPad for Windows users and vi for UNIX users.

Syntax (Event Writer Rules)

The event writer rule syntax is:

```
event_ID::device::user_ID::majorSrc::minorSrc::action::template::  
command::logging::event_token::user_parms
```

where:

Fields in the rule are separated by three colons (::). You can use an asterisk (*) as a wildcard character, which means the value of the parameter is ignored when determining matches for the event writer rule.

Parameters (Event Writer Rules)

event_ID

Specifies the event identifier to which the event writer rule applies. You can use a UNIX regular expression. A period and an asterisk (.) matches any event name. For example, you can specify [Aa].* to match any event that starts with uppercase or lowercase "a."

If the *event_ID* parameter is simple text (not a regular expression), it must exactly match the generic event's event ID. Partial matching does not count. If the *event_ID* is a regular expression, the length of the regular expression match is used to determine how well the event writer rule's *event_ID* matches the generic event's event ID.

device

Specifies the name of the object, which is typically a device or host associated with the event. A period and an asterisk (.) is the default and indicates that the source object is ignored in selecting the rule.

If an incoming event matches several rules when source types and event IDs are evaluated, devices are compared. Rules that match the device of the incoming event take precedence over rules that have asterisk (*) specified for *device*.

If you have defined rules for assigning device names, this parameter can be useful. For example, if you name devices located in the United States USxx, you can specify US.* in this parameter to execute a particular writer action for events from devices in the United States.

user_ID

Identifies a user associated with the event. Use this parameter to execute a particular action for events from a specific user. Many events do not have specific users associated with them. This parameter cannot contain a UNIX regular expression. You must supply an exact user ID or an asterisk. An asterisk (*) is the default and indicates that the user is ignored in selecting the rule.

If an incoming event matches several rules when source types, event IDs, and devices are evaluated, user IDs are compared. Those rules that match the user ID associated with the event take precedence over rules that have asterisk (*) specified for *user_ID*.

majorSrc

Automatically displays the major type of source directing events to the event writer. This parameter is required. Its value is determined by the event source. For events from CA NSM on Windows, the value must be "tng." For events from CA NSM on UNIX, the value must be "uni."

minorSrc

Automatically displays the minor type of source directing events to the event writer. For events from CA NSM, this parameter contains the event type. Use an asterisk (*) to accept all events.

action

Specifies the action that occurs when the event writer receives this type of event using one of the following values:

CR_CREATE

Write a new request for each event.

CR_UPDATE

Update an existing request or requests (if they exist), or create a request if no requests are found. By default, records are located by matching on the *log_agent* and *affected_resource* attributes. The user can override the defaults by specifying a list of any request attributes.

CR_UPDATE_ONLY

Like CR_UPDATE, except a request is never created when no matching requests are found.

COMMAND

Execute the command specified in the *command* parameter.

Note: If CA Service Desk Manager cannot access a request or change order, it attempts the update again after a fixed interval.

template

Specifies the name of a request template to use to create a request. This parameter is not required and is ignored if the action is not CR_CREATE.

Note: The request template must be created before the rule is defined.

command

Specifies the command to execute if the *action* parameter value is COMMAND. Substitution arguments, as the following table shows, can be incorporated into the command. The arguments are replaced by their real-time values when the command is executed:

&node

The device name or node identifier

&user

The user name or ID

&date

The event date

&time

The event time

&event-id

The source event string that triggered the event

&data

The associated event data

&sev

A measure of importance for the event

&major-src

The source application ID

&minor-src

The agent name or further delineation of the event

&handle

The daemon-supplied string resulting from rules

&src-handle

The daemon identifier that assigned the handle

&status-handle

The status of the handle (valid values are create, update, or terminate)

logging

Specifies whether logging occurs, using one of the following values:

NONE

No logging, other than normal error logging, occurs. NONE is the default value.

PDM

Logging occurs in the CA Service Desk Manager log (stdlog.0) in its internal generic event format.

SYS

Logging occurs in the UNIX syslog, which can be forwarded to the Unicenter Console. The event is assigned a message ID (CAPD13) to allow event processing from the Unicenter Console.

BOTH

Logging occurs in the CA Service Desk Manager log (stdlog.0) and the UNIX syslog.

event_token

A 30-character user-defined tag that identifies a specific request associated with an *event_id* (tng event message) or all messages like an *event_id* (for example, a wildcarded *event_id*). *event_token* is a request attribute and is stored in every request generated by the CA NSM interface. If no *event_token* is specified in the writer rule, the string "tng_generated" is used. This lets the user update all requests that match the *event_token* attribute. For example, two different messages for the same asset can update unique requests.

Each CR_UPDATE writer rule specifies the unique message parts and a unique *event_token*. The *event_token* is used to find and update the matching request. By default, an activity log containing the message is added to the matching request. In another example, the user can update the status attribute (for example, set status=CL (closed)) in an existing request by specifying the same *event_token* in the CR_UPDATE writer rule that was used when the request was created using a CR_CREATE writer rule.

For example, the first writer rule below causes the writer process (tngwriter) to create a call request with an event_token equal to 'SystemCritical' whenever it receives a NSM event identified by the string 'Event1'. The second writer rule causes the writer process to update the status value to 'CL' for all call requests with an event_token equal to 'SystemCritical' whenever it receives a NSM event identified by the string 'Event2'.

```
Event1:::*:*:*tng:::*:CR_CREATE:::::NONE:::SystemCritical::
```

```
Event2:::*:*:*tng:::*:CR_UPDATE:::::NONE:::SystemCritical::%SEARCH=EVENT_TOKEN;%STATUS=CL
```

user_parms

Contains the following types of information:

Request attribute values

Request attribute values are specified using the following syntax:
%ATTRIBUTE=value, where **ATTRIBUTE** is an attribute name identified in **text_api.cfg** that maps to a CA Service Desk Manager Majic call request attribute. This file is located in **\$NX_ROOT/site** (UNIX) or **installation-directory\site** (Windows).

Note: If you use multiple keyword/value pairs, separate each one with a semicolon (";").

For example, the writer rule below causes the writer process (tngwriter) to create a call request with assignee equal to 'mccda04' and customer equal to 'nsm' whenever it receives a CA NSM event identified by the string 'Event4'.

```
Event4::: *::*:tng::*:CR_CREATE:::::NONE:::::%ASSIGNEE=mccda04;CUSTOMER=nsm
```

A list of request attributes to match when updating existing request records

The syntax for the list of attributes to match is specified as follows:

%SEARCH=attribute1[, attribute2...], where **SEARCH** is a fixed keyword and **attribute1**, **attribute2**, and so on are **ATTRIBUTE** names specified in the **text_api.cfg**.

The default search list of attributes is "asset_name" (DEVICE or UUID) and LOG_AGENT. The **SEARCH** keyword adds attributes (to match on) to the default search list. The **SEARCH_EXPLICIT** keyword completely overrides the default search list. Only the list of attributes following the **SEARCH_EXPLICIT** keyword are used to search for a call request.

For example, the writer rule below causes the writer process (tngwriter) to update the status value to 'CL' for all call requests with assignee equal to 'mccda04' whenever it receives a CA NSM event identified by the string 'Event2'.

```
Event2::: *::*:tng::*:CR_UPDATE:::::NONE:::SystemCritical:::%SEARCH=ASSIGNEE;%STATUS=CL;%ASSIGNEE=mccda04
```

An attribute in the list of attribute values is used to search on if the attribute is in the **SEARCH** or **SEARCH_EXPLICIT** list. If it is not in the search list it is used to set or update the attribute's value in the call request. It cannot be used for both in the same writer rule.

Note: For more information about **text_api.cfg** and how CA Service Desk Manager uses it, see the *Administration Guide*.

Special parameter names that are replaced with their corresponding value from the CA NSM event structure

You can use the following special parameter names anywhere in the *user_parms* string:

&message

Message text associated with this CA NSM message.

&parm

AHD.DLL Parm field on CA NSM Message Action Screen.

&uuid

CA NSM universally unique identifier.

&device

Device (for example, host name) that generated the CA NSM message.

&majorsrc

The major type of source directing events to the event writer. For events from CA NSM on Windows, the value is "tng." For events from CA NSM on UNIX, the value is "uni."

&minorsrc

The minor type of source directing events to the event writer.

&node

Device (for example, host name) that generated the CA NSM message.

&addr

The IP address of the host that generated the CA NSM message.

&username

The user name on the host that generated the CA NSM message.

&date

An integer representing how long since 1970 the CA NSM message was generated.

&time

The Date and Time of the CA NSM message (for example, Tue Jul 4 10:23:37 2000).

&severity

The severity of the CA NSM message.

&tag

Tag data associated with the CA NSM message.

For example, the writer rule below causes the writer process (tngwriter) to create a call request with customer equal to the username value (&username) of the event message whenever the writer receives a NSM event identified by the string 'Event2'.

```
Event2:::*:*::tng:::*::CR_CREATE:::::NONE:::::%CUSTOMER=&username
```

Using *event_token* and *user_parms*, you can set initial values or update values of all attributes of the request and specify which fields to match when locating records to update. The only restriction is that the description attribute is never updated in an existing call request record. If a record update and the description field are specified, an activity log containing the text of the description is added to the existing record.

CA NSM Message Action Record: ahd.dll AHD_Call <parms...>

You can also specify data (<parms...>) to the AHD_Call on the CA NSM message action record that can be used to replace the &Parm parameter specified in your user_parms parameter in your writer rule. In order for the data specified on the AHD_Call to be processed in this way the data (<parms...>) must be preceded with a '%' character and the &Parm 'special parameter' must be included somewhere in the user_parms field of the writer rule.

The following is an CA NSM message action and a writer rule that work together to cause the writer process to create requests with assignee set to 'mccda04' whenever it receives a NSM event identified by the string 'Event3'.

```
Ahd.dll AHD_Call %ASSIGNEE=mccda04
```

```
Event3:::*:*::tng:::*::CR_CREATE:::::NONE:::::&Parm
```

CA Service Desk Manager customers upgrading from AHD4.5 and earlier may already be using the <parms...> field. For example, you may be using this data to match on Writer rules. You can continue this practice without change. If you want to use both 'old' and 'new' parameters then you must concatenate a '%' character to your 'old' <parms...> data and then follow with 'new' data.

Example:

ahd.dll AHD_Call old data

ahd.dll AHD_Call %new user_parms

ahd.dll AHD_Call old data%new user_parms

Data before the first '%' character is concatenated to the CA NSM event message which is placed into the Call Request description field. This is how CA Service Desk Manager has worked in the past. Data after the first '%' character is used to replace the &Parm parameter wherever it is specified in your user_parms parameter in your writer rule.

CA Service Desk Manager Log (stdlog.0) Syntax

The syntax for entries in the stdlog.0 file is:

```
genDate genTime genNode genProc PID level codefile linenum msgID
::msgDomain\msgNode::: ::msgGenDomain\msgGenUser:::domainID
::msgDate msgTime:::eventID arg tag:::IPaddr:filter:filterNum
::majorSrc::: ::msgSrc:::msgType:::msgSrcNum::: ::platform
```

where:

Fields in the rule are separated by three colons (:::).

genDate

Specifies the month and day (from the system clock) on which the log incident was generated.

genTime

Specifies the time (from the system clock) at which the log incident was generated.

genNode

Specifies the node name that generated the log incident. The value is typically the first eight characters of the generating node's DNS name. This value is always the same, because stdlog.0 only contains entries from processes running on the same node.

genProc

Specifies the name of the process (for example, ehwriter) that generated the log incident. The process name is operating environment-dependent, but should correspond to the name that appears in the Task Manager processes list (Windows) or in output from a ps command (UNIX).

PID

Specifies the numeric process identifier from the Task Manager processes list (Windows) or in output from a ps command (UNIX). The PID is critical when multiple processes are running with the same process name. For example, multiple database agents typically run concurrently with the same process name.

level

Specifies the programmer's estimation of the significance of the message. Possible values include FATAL, EXIT, RESTART, SIGNIFICANT, SEVERE, ERROR, MINIMUM, WARNING, INFORMATION, MILESTONE, TRACE, and VERBOSE.

codefile

Specifies the name of the code source file from which the message was generated.

linenum

Specifies the number of the line in the code source file at which the message was generated.

msgID

Specifies the ID assigned to a TNG event.

msgDomain

Specifies the name of the domain from which the message originated.

msgNode

Specifies the name of the node from which the message originated.

msgGenDomain

Specifies the domain from which the user identified by the msgGenUser value generated the TNG message.

msgGenUser

Specifies the user ID of the user that generated the TNG message.

domainID

Specifies the ID of the domain from which the TNG message was generated.

msgDate

Specifies the month and day (from the system clock) on which the TNG message was generated.

msgTime

Specifies the time (from the system clock) at which the TNG message was generated.

eventID

Specifies the source event string or sed-style regular expression that triggered the event.

arg

Specifies a variable entered in the text string of a message action. For example, if the text field on a message action is entered as "ahd.dll AHD_Call help me," the argument in stdlog.0 appears as "args=help me."

tag

Specifies tag data associated with the TNG message.

IPAddr

Specifies the IP address of the host that generated the TNG message.

filter

Specifies the process name for the filter rule file.

filterNum

Specifies the number of the filter rule file.

majorSrc

Specifies the major source (or converter type). This is a string that identifies the source application ID. That is, whether events come from Unicenter TNG for Windows (in which case the value is tng), from Unicenter for UNIX (in which case the value is uni), or from an internal daemon (in which case the value is -).

msgScr

Specifies the message source. Possible values are CNV (converter), FLT (filter), NOS (no source), and WRT (writer).

msgType

Specifies the message type. Possible values are CRT (create), DSC (discovery), TRM (terminate), and UPD (update).

msgSrcNum

Specifies the message source number.

platform

Specifies the operating environment from which the message originated. Possible values are AIX, AS400, DECOSF1, DGUX, DYNIX, HPUX, IRIX, MISERVER, MPRAS, MVS, NETWARE, SINIX, SOLARIS, SVR4MP, TANDEM, UNIXWARE, WNT (Windows), or any value defined by the UNIX agents.

Load Event Writer Rules

When you modify or define writer rules, you can load these new or modified rules without restarting the CA Service Desk Manager server or any of the support processes, such as the CA NSM converter or the filter and writer daemons. You can also use this utility to write existing writer rules to a file.

To do load event writer rules, use the wrtrule utility.

Syntax (Loading Event Writer Rules)

The syntax for loading event writer rules is:

```
wrtrule [-v] -c addr [-r rule_file] [-d dump_file]
```

Parameters (Loading Event Writer Rules)

-v

Specifies verbose mode, so feedback displays as the utility executes.

-c addr

The name of the event writer as shown by slstat.

-r rule_file

The name of the text file containing the events to load.

-d dump_file

The name of the file where you want the current event writer rules written. The following types of information are written to the file:

- Pending event messages
- Network resource information in the process of being retrieved
- Event writer rules

Note: Although both *-r rule_file* and *-d dump_file* are optional, you must specify one or the other. You can specify both, in which case new rules load first.

Note: If the event writer is handling one or more CA NSM event messages when it receives a request to load new writer rules, the load is delayed until the writer completes the current messages. For example, the writer may be processing an event because it is waiting for network resource (asset) information. In this case, the load message is delayed until this information has been retrieved and the request for a CA Service Desk Manager call request has been generated.

A request to write the current event rules to a file, however, is performed with no delay.

Maintain Filtered Events

The relationships between filter and event writer daemons and event sources on different computers are maintained in the topology file, *topology.cfg*, in the *\$NX_ROOT/site/eh* directory (UNIX) or *installation-directory\site\eh* directory (Windows) on the CA Service Desk Manager server. The topology file lets you determine and maintain the entire event handling system from one location. Instead of going to multiple clients, you can edit files in a single location and determine the topology of the entire event handling system.

Note: The *\$NX_ROOT/site/eh* directory (UNIX) or *installation-directory\site\eh* directory (Windows) on the CA Service Desk Manager server also includes filter and event writer rules.

Use a text editor to view, update, and save the topology.cfg file. We recommend WordPad for Windows users and vi for UNIX users. Do not use a text editor that leaves formatting characters in a file.

The format of the topology file is:

```
name cmd [dest-name] [converter-type]
```

where:

name

Specifies the host and the unique name of the event-handling daemon in the format *hostname:daemon-name* (for example, *ws2:uconv*). The *daemon-name* appears in *sstat*.

cmd

Specifies the name of the executable in the \$NX_ROOT/bin directory (UNIX) or *installation-directory\bin* (Windows), such as *tngcnv*, *uniconvert*, *filter_nxd*, or *ehwriter*.

dest-name

Specifies the daemon that receives the generic events from this daemon (for example, *ws2:filter1* and *ws3:wrtr*). Event writer daemons do not have destination daemons. All daemons should also have a record in the topology file.

converter-type

Specifies whether events come from CA NSM for Windows (*tng*), CA NSM for UNIX (*uni*), or from an internal daemon (*-*).

A sample topology file is as follows:

```
# maple:uniconv uniconvert maple:tngfilter uni
maple:tngcnv tngcnv maple:tngfilter tng
maple:tngfilter filter_nxd maple:tngwriter -
maple:tngwriter ehwriter - -
```

Example

Following is an example of how one organization implemented filtered events in an integrated installation of CA Service Desk Manager and CA NSM.

To automatically create requests when a critical status appears on the CA NSM WorldView map, the CA Service Desk Manager administrator (Ken) must create an event filter rule to identify the events to which he wants to respond. He must also create an event writer rule to specify the action to perform when events of this type are received.

Ken decides that he will use one simple filter rule at first. Then, as he becomes more familiar with the system, he will use a more complex set of rules. He decides to capture each critical event from U.S. servers. His system uses the convention that names all servers located in the contiguous United States as usaxxx, so this is very simple.

Ken first edits the `tngfilter_rule.dat` filter rule file in the `$NX_ROOT/site/eh/IP` directory on the CA Service Desk Manager server, where IP is the IP address of the CA Service Desk Manager server. He uses the UNIX vi editor, although he could use any text editor that does not add extraneous control characters (we recommend that Windows users use WordPad to edit files).

In the `tngfilter_rule.dat` file, the pound sign (#) is the comment character. Any characters after the pound sign are ignored when the rules are read. Most of the lines in Ken's `tngfilter_rule.dat` file are commented out, but he notices the following lines:

```
# Report All Events (separately)
uni:...:...:... (0,1)
tng:...:...:... (0,1)
```

These lines constitute an open filter; that is, a filter that will pass all events to the event writer.

Ken changes these lines to:

```
# Report All Events (separately)
# uni:...:...:... (0,1)
tng:...:...:...Object_Status_Updated.*Critical.*... (0,1)
```

Ken comments out the `uni` line because he is currently interested only in Windows CA NSM events. He enters `Object_Status_Updated.*Critical.*` in the `event_ID` field in the `tng` line because he wants the filter to pass only `Object_Status_Updated.*Critical.*` events. CA NSM generates `Object_Status_Updated.*Critical.*` events when the state of an object becomes critical.

After saving the filter rule file, Ken edits the `tngwriter_rule.dat` writer rule file found in `$NX_ROOT/site/eh/IP` on the CA Service Desk Manager server, where IP is the IP address of the CA Service Desk Manager server.

As with the filter rules, most of the lines in the file are comments. The last two lines in the file show the default writer rule definitions. Ken wants to pay attention to events from Windows CA NSM only, so he comments out the first of these two lines.

Ken formats the second line following these steps:

1. He has already set his filter to pass only critical events, so he leaves the event ID open with `.*`.
2. He wants to accept events from U.S. servers only, so he enters `usa.*` in the device field.
3. He wants to accept events from any user, so the user field remains asterisk (`*`).
4. He wants to write a new request for each critical event, so he leaves `CR_CREATE` in the action field.
5. He has already entered a suitable request template in the system, so he enters its name, `CriticalTemplate`, in the template field.
6. He does not want any other logging done, so the logging field remains `NONE`.

Here are the results of Ken's edit:

```
#.*...*:*:uni:*:*:CR_CREATE:*:*:NONE
.*...*:*:tng:*:*:CR_CREATE:*:*:CriticalTemplate:*:*:NONE
```

Ken saves the writer rule file, and then recycles the CA Service Desk Manager server. He is ready to receive events and write requests automatically.

Troubleshoot Integration

Errors can occur when integrating, configuring, and using CA NSM with CA Service Desk Manager. You troubleshoot these errors, using the following information:

- [2D/3D Map Error Messages \(Windows Only\)](#) (see page 501)
- [Review Server Configuration](#) (see page 503)
- [Filter Error Messages](#) (see page 504)
- [Check the Slump Connection](#) (see page 505)
- [Turn On Logging](#) (see page 506)

2D/3D Map Error Messages (Windows Only)

The integration portion of the 2D/3D map and Unicenter Explorer can produce CA NSM errors.

TNGWV object create failure= x

Reason:

This error typically occurs when the integration script is executed multiple times without first running de-integration (x represents a number).

Action:

To de-integrate, then integrate, CA Service Desk Manager with the CA NSM 2D/3D map and Unicenter Explorer, complete the following steps:

1. At the command line, enter:
`installation-directory\bin\deintahd.exe`
2. Open the CA NSM Object Browser. Locate and click Method to display a list of methods.
3. If any AHD Methods are listed in the name column, select Delete from the Object menu to remove them.
4. Click Popup_Menu to display a list of menus.
5. If any AHDManagedObject exists under the name column, select Delete from the Object menu to remove it.
6. With Popup_Menu still open, scroll down until ManagedObjects displays under the name column.
7. Search for any ManagedObject that has a method name containing AHD. If any exist, select Delete from the Object menu to remove them.
8. Click Jasmine_Menu_Action to display a list of Unicenter Explorer methods. If any USPSD menu actions are listed in the name column, select Delete from the Object menu to remove them.
9. Click Jasmine_Menu_Object to display a list of Unicenter Explorer menus. If any USPSD menu objects are listed in the name column, select Delete from the Object menu to remove them.
10. Exit Object Browser.
11. At the command line, enter:
`installation-directory\bin\integAHD.exe`

Your integration into the 2D/3D maps and Unicenter Explorer should now be successful. Right-click a managed object from the 2D/3D map or Unicenter Explorer, and verify that all of the CA Service Desk Manager menu options are displayed.

CAE0232E-Repository error code 22

Reason:

This message indicates that the path to the executable in the method is incorrect.

Action:

Add installation-directory\bin to your path or modify the exe_name field in the Method list to include the full path to the executable.

Review Server Configuration

When the CA Service Desk Manager server is on a UNIX or Windows computer and CA NSM is on a different Windows computer, the CA Service Desk Manager server must be up and running. Start the NSM Event Converter service on the CA NSM computer. To ensure the converter is up and running, check the processes using Task Manager. If tngcnv does not appear as a running process after the event converter service was started, check the most recent tngcnv.*n* or stdlog.*n* files (located in *installation-directory*\log). These files provide you with information about why the event converter is not running.

When the CA Service Desk Manager server and CA NSM server are on the same Windows computer, the CA Service Desk Manager Service must be up and running. Start the CA NSM Event Converter Service on the computer. To ensure the converter is up and running, check the processes using Task Manager. If tngcnv does not appear as a running process after the CA Service Desk Manager service was started, check the following:

1. Using WordPad, edit the pdm_startup file located in *installation-directory*\pdmconf. In this file, you will see text similar to the following:

```
[procset MAIN_PROCSET]
pdm_info
sw_ver_ctl
bpnotify_nxd
PDMBASE
PDMBOP
FILTERING
```

If a semicolon (;) is used to comment out FILTERING, remove the semicolon.

2. Moving backwards toward the top of the file, you should see the following text:

```
[ procset FILTERING ]
tngfilter
tngwriter
ehc
; UNICNV_REPLACE
; tngcnv:NT_ONLY
```

If a semicolon (;) is used to comment out any of the above options, remove the semicolon.

3. If you made any changes to this file, save them, then recycle CA Service Desk Manager server.

If the event converter still does not show up as a running process, check the most recent *tngcnv.n* file (located in *installation-directory\log*). This file provides you with information about why the event converter is not running.

Filter Error Messages

If your event converter (tngcnv) does not start or your converter is running but no requests are being created, your log files may contain an error indicating why it could not start or report events. You may see one or more of the following error messages:

Can't resolve host name to an IP address

Reason:

This message indicates that you are missing host entries on your CA Service Desk Manager computer, your CA NSM computer, or both.

This error might also indicate that DNS is not working or is not returning the correct IP address.

Action:

Check the %SystemRoot%/system32/drivers/etc/hosts file. Your host file must contain an entry for the CA Service Desk Manager computer and the CA NSM computer. After adding the host name, restart the converter (or the CA Service Desk Manager service). An example entry resembles the following:

```
127.0.0.1    localhost    # Local host loop back
141.202.211.11  usbegp11      # This is the NSM client box
141.202.211.12  usbegp12 ahdhost # This is the Service Desk host
              # (showing multiple entries)
```


Cannot find repository file

Reason:

This message indicates that the @NX_REPOSITORY parameter is not set in NX.env (located in *installation-directory*) on the client.

Action:

Edit NX.env, add your repository name, making sure that you match the name exactly, save the file, and then restart the event converter.

The dependency doesn't exist or has been marked for deletion

Reason:

This message occurs when there is a dependency (probably MSSQL) on the converter that should be removed.

Action:

Execute regedt32 and open the HKEY_LOCAL_MACHINE -> System -> Current Control Set -> Services -> AHD_Event_Converter tree. Remove the MSSQL dependency (or other as applicable) and restart your Windows server.

Destination unknown, Queuing events or No topology.cfg file entry for node; queuing event

Reason:

This message indicates that the topology.cfg file on the CA Service Desk Manager server must be corrected.

Action:

Correct the topology.cfg file as necessary, then recycle CA Service Desk Manager and the CA NSM converter. An example topology.cfg file for CA Service Desk Manager on a Windows computer named AHD1 and a CA NSM computer named TNG1 follows:

```
TNG1:tngcnv tngcnv AHD1:tngfilter tng
AHD1:tngfilter filter_nxd AHD1:tngwriter -
AHD1:tngwriter ehwriter - -
```

How to Check the Slump Connection

If the CA NSM converter is running and you still do not see any events being created, verify the slump connection to CA Service Desk Manager.

To verify the connection, do the following:

1. Use the *s/stat* command from the command line on the CA Service Desk Manager computer.

2. Look for a slump connection to tngcnv.

If you do not find one, the converter is not communicating with CA Service Desk Manager.

3. Restart the CA NSM converter, and then verify the connection again.
4. (If the connection still cannot be found) Check the *tngcnv.n* files (located in *installation-directory\log*) for errors regarding tngcnv and slump logon.

Turn On Logging

If the slump logon was successful and you still do not see new events, turn on logging and check the files for an indication that an event has come through CA NSM. You can create an event simply by changing the status on a managed object on the 2D/3D map.

To turn on logging, modify the file *NX.env* file located in *\$NX_ROOT* (UNIX) or *installation-directory* (Windows) to include the following line:

```
@NX_LOG_LEVEL_BSTRAP=VERBOSE
```

You can then monitor the *tngcnv.n* files (located in *installation-directory\log*) to determine if events are being passed to CA Service Desk Manager. You should see something similar to the following:

```
09/17 16:35:58:01 tngcnv 477 MILESTONE convtr.c 399
Sending 1:::dogwood:::9994011e-2f7e-11d1-a435-00c04fd478c9:::09/17/1997
:::16:35:41:::Object_Status_Updated minor:::BV:141.202.211.0:Segment.1
IP:141.202.211.14 MAC: CNT: LOC: DSC:::3:::tngcnv:1:::tng::WindowsNT:
::CNV:::CRT:::1:::
```

In this example, the message indicates that the status on a managed object has been updated to Minor.

If you find that events are being passed to CA Service Desk Manager and they appear in the log, double check to make sure that you are using the default filter and writer rules. If not, restore the original default filter and writer files, recycle the server, and create an event.

If you find that events are not being passed to CA Service Desk Manager, make sure that two CA NSM processes responsible for passing event information (*canotify.exe* and *caoprdrn.exe*) are executing. If they are not running, start them as described in the CA NSM documentation. If they are running, contact CA NSM Support.

CA Service Desk Manager Event Converter

This information only applies to the CA NSM integration in a Windows operating environment.

The CA Service Desk Manager Event Converter installs in two different ways when you run CA Service Desk Manager configuration on a Windows server:

- Installs as a Windows Service
- Installs as a part of the CA Service Desk Manager Daemon Server from where you can start and stop the converter

You can run the Event Converter on a Primary, Secondary or Client-only computer. While you can use either method to start the Converter on the Primary and Secondary servers, it can only be started as a Windows Service on a Client.

Note: If CA Service Desk Manager is shutdown for any reason, you may want to leave the Event Converter running so it can continue to receive messages from NSM. Any messages received from NSM while CA Service Desk Manager is down are queued and processed after CA Service Desk Manager is restarted.

Event Converter Removal (Daemon Manager)

The Daemon Manager not only starts and stops the Converter, but also restarts the Converter if it inadvertently stops. You can use *pdm_status* to see whether the Converter is running or not and to determine whether it is running on the local or on a remote (Secondary) server.

If you want to use the Windows Service Manager to start and stop the Event Converter, remove the Event Converter from the startup configuration of the Daemon Manager.

You can change the startup configuration on a Primary server by modifying the *tngcnv* line in the *pdm_startup* file so that it reads as follows:

```
; tngcnv.
```

Important! This change reverts if you run the CA Service Desk Manager configuration again. If you want to retain the change after running configuration, modify *pdm_startup.tpl*.

You can change the startup configuration on a remote (Secondary) server by using the *pdm_edit* facility.

Note: For more information about running *pdm_edit*, see the *Administration Guide*.

Event Converter Removal (Windows Service)

If you want to use the Daemon Manager to start and stop the Event Converter, use the following command to remove the Event Converter from the Windows Service:

```
tngcnv -u
```

You can reinstall the Event Converter as a Windows Service by using the following command:

```
tngcnv -i
```

Leverage NSM to CA Service Desk Manager Integration

The section provides an example of how to leverage the integration between CA NSM and CA Service Desk Manager. The example offers a sample approach for configuring this default functionality and is not intended to suggest that it is the only way to accomplish this task.

Sample Message Records/Actions with Limited Content

Two Message Records, CFNEW and CFUPDATE, written with a single message action each, are the cornerstone in this example for integrating NSM to CA Service Desk Manager (sample files are also included). For a clean, compact, robust and easy-to-maintain integration, you can leverage the strength of the following:

- tngwriter_rule.dat
- CR_UPDATE and CR_UPDATE_ONLY functions
- text_api keyword list

Note: The example shows how to create Message Records and Message Actions with little content. The content has been moved into the cawto commands that are illustrated in the examples that follow. You could alternately place the desired parameters and content directly into the “text” field of the message action, or even use a combination of the two approaches.

Sample Event Management Message Record

When you use this integration, you can use the Message Record - Detail page to display a sample Event Management Message Record that will react to an event containing “CFNEW” at the start of the event alert.

Sample Message Action List

When you use this integration, you can use the Message Action list that is associated with the “CFNEW*” Message Record. By double-clicking this item, you can see the detailed information stored as part of the Message Action. When you continue to use the integration, you can see a detailed view where you can see the “action” field set to “EXTERNAL” and the “text” field with the standard integration text value of “AHD.dll AHD_Call”.

Sample Message Record (configured to scan for events beginning with CFUPDATE)

When you use this integration, you can see the detail that results from a second Message Record, configured to scan for events that began with “CFUPDATE”. This Message Record also has an identical Message Action created for it (as illustrated by the previous “CFNEW*” Message Record).

Sample cawto Commands that Generate/Update Requests

To use the Message Records previously created, send a message through CA NSM Event Management by using the “cawto” utility/command provided with CA NSM. The “cawto” utility/command allows you to create or update a Request using the command formats shown in the two examples that follow.

Note: By using this approach, you can capture and reformat (for translation into an expected format for inclusion into CA Service Desk Manager) any alerts of interest coming into the Event Management console.

Issue the “opreload” command on the NSM Event management Console to refresh the NSM Event Message Records and Actions that are stored in memory after creating the previous NSM Message and Action rules.

Example 1: “cawto” formats for Generating and Updating a New Request

Use the following cawto format for generating a New Request:

```
CAWTO -n <NSM-server-name>
CFNEW;%STRING1=KEYWORD1;%CUSTOMER=ServiceDesk;%SUMMARY=Photos
required;%CATEGORY=Applications;%DESCRIPTION=Description: Photos required Justification: required
by regulations for entry
```

Use the following cawto format for updating a Request:

```
CAWTO -n <NSM-server-name>
CFUPDATE;%STRING1=KEYWORD1;%STATUS=Closed;%SUMMARY=Cancelled by ServiceDesk
```

Example 2: Alternative “Cawto” formats for Generating and Updating a New Request

Use the following alternative cawto format for generating a New Request:

```
CAWTO -n <NSM-server-name> CFNEW2;%EVENT_TOKEN=KEYWORD2;%SUMMARY=Server  
Installation Approved;%STATUS=Work in Progress;%CATEGORY=Hardware;%DESCRIPTION=Install  
new server and level V drops to existing server location within the data center. Also, install wiring on patch  
panel in router cabinet
```

Use the following alternative cawto format for updating a Request:

```
CAWTO -n <NSM-server-name>  
CFUPDATE2;%EVENT_TOKEN=KEYWORD2;%STATUS=Closed;%SUMMARY=Closed by Server  
Team
```

Sample Files to Use with the Integration

Sample files are available for your use when integrating NSM to CA Service Desk Manager.

topology.cfg

```
# <remote_node>:uniconv uniconvert <IP_ADDR_tngfilter>:tngfilter uni  
<IP_ADDR_tngcnv>:tngcnv tngcnv <IP_ADDR_tngfilter>:tngfilter tng  
<IP_ADDR_tngfilter>:tngfilter filter_nxd <IP_ADDR_tngwriter>:tngwriter -  
<IP_ADDR_tngwriter>:tngwriter ehwriter - -
```

where:

<IP_ADDR_tngcnv>

The IP Address of the server that is running the CA NSM Event Converter process.

<IP_ADDR_tngfilter>

The IP Address of the server that is running CA Service Desk Manager.

<IP_ADDR_tngwriter>

The IP Address of the server that is running CA Service Desk Manager.

tngfilter_rule.dat

Use the default version of this file, with no changes.

tngwriter_rule.dat

The tngwriter_rule.dat displays as follows:

```
# evt
id::dev::user::majorSrc::minorSrc::action::template::cmd::log::event_token::user_parms
```

where:

event id

The string or sed-style regular expression.

device

The string, sed-style regular expression, or '*' or empty.

user

The string, '*', or empty.

majorSrc

The string "uni" from uniconverter, or "tng" from tng converter.

minorSrc

The string or '*'.

action

The action options. The following options are available:

CR_CREATE

Write a new request for each event

CR_UPDATE

Updates an existing request or requests (if any exists) or creates a new request if no requests are found. By default, requests are located by matching on the log_agent and affected_resource (asset) fields. The user can override the defaults by specifying a list of any call request attributes.

CR_UPDATE_ONLY

This is like CR_UPDATE except that a new request is never created when no matching requests are found.

COMMAND

Executes <cmd>, identified in the cmd description in this table.

template

Specifies the name of a request template to use to create a request. This parameter is not required and is ignored if the action is not CR_CREATE.

Note: The request template must be created before the rule is defined.

cmd

The command passed to the shell—ignored on all but COMMAND action.

logging

The logging options, as shown by the following:

NONE

No logging (other than error logging).

SYS

Log incidents to the UNIX syslog (Unicenter message console).

PDM

Log incidents to application log (\$NX_ROOT/log).

BOTH

Log incidents to application log and syslog.

event_token

(Optional). This is a 30-character user defined tag that is used to identify a specific request associated with an event_id (tng event message) or all messages like an event_id (for example, a wildcarded event_id).

event_token is a request attribute that is stored in every request generated by the TNG interface. If no event_token is specified in the writer rule, the string "tng_generated" is used. This allows the user to update all requests that match the event_token attribute. For example, two different messages for the same asset can now update unique requests. Each CR_UPDATE writer rule specifies the unique message parts and a unique event_token. The event_token is used to find and update the matching request. By default, an activity log containing the message is added to the matching request.

In another example, the user can update the status attribute (such as, set status=CL (closed)) in an existing request by specifying the same event_token in the CR_UPDATE writer rule that was used when the request was created using a CR_CREATE writer rule.

user_parms

(Optional) This contains the following three types of information:

- Request attribute values
- List of request attributes to match on when updating existing request records
- Special parameter names that are replaced with their corresponding value from the TNG event structure

The request values and the list of attributes are specified using a %<KEYWORD>=<value> syntax. If you use multiple keywords/value pairs, you must separate each by a semicolon (";").

Request attribute values are specified using the syntax %<ATTRIBUTE>=<value>, where ATTRIBUTE is an attribute name identified in the text_api.cfg (located in \$NX_ROOT/site directory), which maps to an AHD majic request attribute.

The syntax for the list of attributes to match is specified as %SEARCH=<attribute1>[,<attribute2>...], where SEARCH is a fixed keyword and attribute1 (and so on), are ATTRIBUTE names specified in the text_api.cfg.

The following special parameter names can be used anywhere in the user_parms string:

&Message

The message text associated with this CA NSM message.

&Parm

The AHD.DLL Parm field on the CA NSM Message Action dialog.

&Uuid

The TNG universally unique identifier.

&Device

The device (for example, hostname) that generated the CA NSM message.

&Majorsrc

The major type of source directing events to the event writer. For events from CA NSM on Windows, the value is "tng". For events from CA NSM on UNIX, the value is "uni".

&Minorsrc

The minor type of source directing events to the event writer.

&Node

The device (for example, hostname) that generated the CA NSM message.

&Addr

The IP address of the host that generated the CA NSM message.

&Username

The user name on the host where the CA NSM message was generated.

&Date

The integer number representing the time since 1970 when the CA NSM message was generated.

&Time

The date and time of the CA NSM message. For example, Tue Jul 4 10:23:37 2000.

&Severity

The severity of the CA NSM message.

&Tag

The tag data associated with the CA NSM message.

As an example, using the examples outlined in Sample 2: Alternative “Cawto” formats for Generating and Updating a New Request the default Event Writer Rules file should be changed from:

```
*... *...:uni:...:CR_CREATE:.....NONE
```

To the following:

```
CFNEW.*... *...:tng:...:CR_UPDATE:.....NONE:.....&Parm;%SEARCH_EXPLICIT=STRING1
CFNEW2.*... *...:tng:...:CR_UPDATE:.....NONE:.....&Parm;%SEARCH=Event_Token
CFUPDATE.*... *...:tng:...:CR_UPDATE_ONLY:.....NONE:.....&Parm;%SEARCH_EXPLICIT=STRING1
CFUPDATE2.*... *...:tng:...:CR_UPDATE:.....NONE:.....&Parm;%SEARCH=EVENT_TOKEN;%STATUS=C
L
```

Note: The %SEARCH_EXPLICIT parameter is used to ensure that when an update is performed, the search looks for a matching Request by comparing the contents of the STRING1 field before proceeding with the update. For more information about text_api.cfg and how CA Service Desk Manager uses the Text API to create requests from CA NSM, see the *Administration Guide*.

More information:

[Event Writer Rule Definitions](#) (see page 486)

CA Portal Integration

You can access CA Service Desk Manager components through CA Management Portal and CA Portal.

Note: CA Management Portal and CA Portal are not shipped with CA Service Desk Manager, and must be purchased and licensed separately. CA Service Desk Manager provides only the most basic information for accessing CA Service Desk Manager through Portal Administration. For detailed information, see the *CA Portal* and *CA Management Portal Server Administration Online Help*.

Verify CA Service Desk Manager Web Interface Accessibility

After CA Service Desk Manager is installed on a system, make sure that you can access the web interface through the tomcat server. For Portal Integration to work, the CA Service Desk Manager Web Interface must be accessible through tomcat.

Note: For CA Service Desk Manager LINUX server installation, the LD_LIBRARY_PATH must be set to \$NX_ROOT/sdk/lib.

Install and Start CA Portal

For detailed information on installing, starting, and stopping CA Portal, see the *CA Portal Getting Started Guide* that applies to your installation.

Note: You can install CA Portal on the system where CA Service Desk Manager is installed, or on a separate system.

Include Portlets

You can use the product to include CA Service Desk Manager portlets in the portal.

To include portlets

1. Log in to CA Service Desk Manager and click Search from the Options Manager on the Administration tab.

The Options List window appears.

2. Click Portal_Safe_List.

The Portal_Safe_list Detail window appears.

3. Enter the *servername:portnumber* where the portal was installed in the Options Value Field.

Note: For more information about this option, see the *Online Help*.

4. Click Install.
5. Restart CA Service Desk Manager daemon.
6. Log into CA Portal as an administrator.

7. Create a user which is valid CA Service Desk Manager User. For more information about how to create a user, see the CA Portal documentation.

Note: The password used while creating this user in the portal could be different from the password used by this same user to log in to CA Service Desk Manager, as CA Service Desk Manager authenticates the users for this integration using a combination of the following, username, a valid CA Portal session and the CA Portal install value if it exists in the PORTAL_SAFE_LIST option.

8. Select Knowledge from the main CA Portal menu bar.

The Knowledge page appears.

9. Select Library from the left pane Knowledge bar.

The Library tree appears in the left pane.

10. Select (or create then select, if necessary) a folder in the Library tree, then select Publish File from the right pane Knowledge bar.

The Publish File form opens.

11. Type the following CA Service Desk Manager portlet URL in the Content text box on the General Information tab:

```
http://hostname:portnumber/CAisd/PortalServlet?USERNAME=$USER.username$&PORTALSESSION=$SESSION$&PORTALINSTALL=portalhostname:portalportnumber
```

Note: Substitute *hostname:portnumber* in the URL with the name and port of the web server on which CA Service Desk Manager resides. Substitute *portalhostname:portalportnumber* in the URL with the name and port of the web server on which Portal resides.

12. Enter CA Service Desk Manager in the Title text box.

13. Click Advanced.

The Advanced properties page of the Publish File form displays.

14. Enter *portal/variable-url* in the Content (mime) Type field and click Publish.

The published contents appear in the selected Library folder.

15. Configure the Workplace to display this portlet.

16. Log out and log in as the newly created user. You should be automatically logged into CA Service Desk Manager in the portlet you just created without logging in again.

Note: While running CA Service Desk Manager in the portlet, the preference Avoid Popups is not available in the Preference page and a popup window is always used regardless of the setting of the preference.

Connect to the CA Portal Server

To connect to the CA Portal server, open a web browser, and enter the following URL in the location or address field:

`http://<servername>:<port#>/servlet/portal`

<servername>

Specifies the server name (or IP address) of the portal server.

<port#>

Specifies the port number that the CA Portal server monitors. You specified the port number during the CA Portal server installation. The default port is 8080.

Configure CA Service Desk Manager to Use SSL with CA Portal

Note: For production purposes, we recommend that you obtain a certificate from a trusted certificate authority.

Before configuring CA Service Desk Manager to use SSL, check to see if the CA Portal and CA Service Desk Manager integration works without SSL. If the integration works without SSL, you can include portlets.

More information:

[Include Portlets](#) (see page 515)

Set Up SSL Using a Self-Signed Certificate

To set up the CA Service Desk Manager Portal Integration using a self-signed certificate

1. At the command line, enter:

```
%JAVA_HOME%\bin\keytool -genkey -alias tomcat -keyalg RSA
```

Answer the prompts appropriately and enter "changeit" as the password for both password prompts.

This sets up the certificate.

2. Edit the server.xml file located in:

```
$NX_ROOT/bopcfg/www/CATALINA_BASE/conf
```

3. Uncomment the following section and save:

```
<!--
<Connector className="org.apache.coyote.tomcat4.CoyoteConnector"
  port="8443" minProcessors="5" maxProcessors="75" enableLookups="true" acceptCount="100"
  debug="0" scheme="https" secure="true" useURIVValidationHack="false" disableUploadTimeout="true">
<Factory className="org.apache.coyote.tomcat4.CoyoteServerSocketFactory"
  clientAuth="false" protocol="TLS" /></Connector>
-->
```

4. Add keystoreFile attribute to server.xml. (When you run the command in step 1, a .keystore file is created in the user's home directory. Add the reference to the keystoreFile attribute and Save the file. Your server.xml should appear as follows.

```
<Connector className="org.apache.coyote.tomcat4.CoyoteConnector"

port="8443" minProcessors="5" maxProcessors="75" enableLookups="true" acceptCount="100"
debug="0" scheme="https" secure="true" useURIVValidationHack="false" disableUploadTimeout="true">

<Factory className="org.apache.coyote.tomcat4.CoyoteServerSocketFactory"

clientAuth="false" protocol="TLS" keystoreFile="location/.keystore" />

</Connector>
```

5. Restart CA Service Desk Manager.
6. To check the SSL functionality, point your browser to <https://hostname:8443>. This should display a Security Alert dialogue. Click Yes.

Note: SSL uses port 8443.

7. Replace the CA Service Desk Manager portlet to use HTTPS and port 8443.

<https://hostname:8443/CAisd/PortalServlet?>

USERNAME=\$USER.username&&PORTALSESSION=\$SESSION&&PORTALINSTALL=portalhostname:
portalportnumber

Connect to CA Service Desk Manager when CA Portal Uses SSL

You can import the CA Portal Server Certificate so that a trusted connection can be made between CA Service Desk Manager and CA Portal (when CA Portal is configured to use SSL).

To connect to CA Service Desk Manager when Portal Uses SSL

1. Verify that CA Portal is configured and works with SSL.

Note: For information about the verification process, see your CA Portal documentation.

2. Export the certificate from the computer on which CA Portal is installed by following these steps:

- a. Locate the server.xml file at the following location:

PORTAL_Install_Dir\jakarta-tomcat-4.1.29\conf.

- b. Note the keystore location and password (pwd), as illustrated in the following lines in server.xml. The default password is *changeit* (all lower case). If you used a custom password while creating the certificate during the portal setup, you will have to use the custom password. For information, see your CA Portal documentation. In the following steps and examples, *changeit* is the default password used:

```
<!-- Define a SSL Coyote HTTP/1.1 Connector on port 8443 -->
<Connector className="org.apache.coyote.tomcat4.CoyoteConnector"
  port="8443" minProcessors="5" maxProcessors="150"
  enableLookups="true"
  acceptCount="100" debug="0" scheme="https" secure="true"
  useURIVValidationHack="false" disableUploadTimeout="true">
  <Factory className="org.apache.coyote.tomcat4.CoyoteServerSocketFactory"
    keystoreFile="c:\Program Files\CA\SC\Unicenter Management Portal\UMPkeystore"
    keystorePass="changeit"
    clientAuth="false" protocol="TLS" />
</Connector>
```

- c. Navigate to the JRE bin directory (PORTAL_Install_Dir\jre\bin) on the portal server computer to access the keytool utility that you will use for exporting the PORTAL Server certificate to a file.

- d. Access the keytool utility, using the following command:

```
keytool -export -alias tomcat -file umpserver.cer -keystore "c:\Program Files\CA\SC\Unicenter
Management Portal\UMPkeystore"
```

Enter keystore password: changeit

Certificate stored in file <umpserver.cer>

Note: When prompted for the password, be sure to use the password obtained from step 2b. In the previous example, *changeit* is the password noted in step 2b. The keystore location is also obtained from step 2b.

3. Import the certificate obtained from the server to the computer containing the CA Service Desk Manager installation by using the keytool utility, as follows:

- a. On the CA Service Desk Manager computer, navigate to the JRE\bin directory directory, typically at the following location:

C:\Program Files\CA\SC\JRE\bin.

- b. The certificate should be imported into the Certification authority used by the CA Service Desk Manager Java Virtual Machine.

The following is an example of an import. In this example, the location of the Certificate authority is:

C:\Program Files\CA\SC\JRE\1.4.2_06\lib\security\cacerts

When prompted for a *pwd*, enter “changeit”. When prompted for *Trust this certificate*, enter Yes.

```
Keytool.exe -import -alias tomcat -trustcacerts -file umpserver.cer -keystore "C:\Program Files\CA\SC\JRE\1.4.2_06\lib\security\cacerts"
```

```
Enter keystore password: changeit
```

```
Owner: CN=ump001.ca.com, OU=unicenter, O=ca, L=islandia, ST=ny, C=us
```

```
Issuer: CN=ump001.ca.com, OU=unicenter, O=ca, L=islandia, ST=ny, C=us
```

```
Serial number: 43ecb469
```

```
Valid from: Fri Feb 10 10:42:33 EST 2006 until: Thu May 11 11:42:33 EDT 2006
```

```
Certificate fingerprints:
```

```
MD5: A1:AF:AE:92:39:2E:53:D5:1C:6D:FE:44:68:61:DD:5C
```

```
SHA1: 66:3A:BC:77:32:81:60:89:70:B9:EF:FB:74:3D:93:74:CD:8E:E2:D2
```

```
Trust this certificate? [no]: yes
```

```
Certificate was added to keystore
```

Note: When prompted for the password, use the password obtained from step 2b. In the previous example, *changeit* is the password noted in step 2b.

4. Edit the file portal-xml-api.xml under
\$NX_ROOT\bopcfg\www\CATALINA_BASE\webapps\CAisd\WEB-INF\xml\portal-xml-api.xml by completing the following steps:

- a. Replace http in the line:

```
<!DOCTYPE PORTAL SYSTEM "http://127.0.0.1:8080/servlet/media/xml/api/request.dtd">
```

With https:

```
<!DOCTYPE PORTAL SYSTEM "https://127.0.0.1:8080/servlet/media/xml/api/request.dtd">
```

- b. Save the file.
- c. If Portal_Safe_List has been installed, make sure you change the port number to 8443 and the computer name to include the domain name (for example, computername.ca.com:8443).

Important! Include the domain name in the computer name as the portal certificate contains the domain name. For more information, see your CA Portal documentation.

5. Recycle the CA Service Desk Manager server.
6. From CA Portal, connect to the CA Service Desk Manager Portlet using the following URL:

```
http://hostname:portnumber/CAisd/PortalServlet?USERNAME=$USER.username$&PORTALSESSION=$SESSION$&PORTALINSTALL=servername:8443
```

Note: Substitute *servername* in the URL with the name of the web server on which CA Portal resides. The server name in this URL should include the domain name, for example, servername.ca.com:8443. Substitute the hostname:portnumber in the URL with the name and port of the web server on which CA Service Desk Manager resides.

More information:

[Include Portlets](#) (see page 515)

Mainframe Product Integration

CA Service Desk Manager side data (.dat file) is associated with mainframe product integrations.

Load CA Service Desk Manager Side Data

The CA Service Desk Manager side data (.dat file) associated with mainframe product integrations is in a list which associates a .dat to the mainframe product name.

Note: The CA Service Desk Manager server is configured, by default, to use ITIL methodology. However, ITIL updates must be applied to the integration data. Use `pdm_userload -a itil_integXXX.dat` to apply the updates only after loading the respective integXXX.dat.

Use `pdm_userload -f integXXX.dat` to load CA Service Desk Manager side data to enable a particular integration. The files are delivered to `$NX_ROOT\data\integrations\`.

Note: For information about enabling the calling side (mainframe product side) of the integration, see the *CA Common Services for z/OS - CA Service Desk Integration Guide*.

CA Products Currently Using CAISDI

The following table lists the CA mainframe products currently using CAISDI and the associated .dat files:

CA Product	Primary Data File	ITIL Update File
CA Advantage EDBC	integEDBC.dat	
CA 1 Tape Management	integCA1.dat	itil_integCA1.dat
CA Allocate DASD Space and Placement	integAllocate.dat	itil_integAllocate.dat
CA Disk Backup and Restore	integDisk.dat	itil_integDisk.dat
CA TLMS Tape Management (CA TLMS)	integTLMS.dat	itil_integTLMS.dat
CA Vantage Storage Resource Manager (CA Vantage SRM)	integVantage.dat	itil_integVantage.dat
CA 7 Workload Automation (CA 7 WA)	integCA7.dat	itil_integCA7.dat

CA JARS Resource Accounting (CA JARS RA)	integJARS.dat integJARSMVS.dat	itil_integJARS.dat itil_integJARSMVS.dat
CA MIM Resource Sharing (CA MIM RS)	integMIM.dat	itil_integMIM.dat
CA OPS/MVS Event Management and Automation (CA OPS/MVS EMA)	integOPSMVS.dat	itil_integOPSMVS.dat
CA SYSVIEW Performance Management (CA SYSVIEW)	integSysview.dat	itil_integSysview.dat
CA NetMaster Network Management for TCP/IP (CA NetMaster NM for TCP/IP)	integNetMaster.dat	itil_integNetMaster.dat
CA NetMaster Network Management for SNA (CA NetMaster NM for SNA)	integNetMaster.dat	itil_integNetMaster.dat
CA NetMaster Network Automation (CA NetMaster NA)	integNetMaster.dat	itil_integNetMaster.dat
CA NetMaster	integNetMaster.dat	itil_integNetMaster.dat
CA NetMaster Network Operations for TCP/IP (CA NetMaster NO for TCP/IP)	integNetMaster.dat	itil_integNetMaster.dat
CA NetMaster File Transfer Management (CA NetMaster FTM)	integNetMaster.dat	itil_integNetMaster.dat
CA MICS Resource Management (CA MICS)	integNeuMICS.dat	itil_integNeuMICS.dat

CA Products Planning to Use CAISDI

The following table lists the CA mainframe products that would use CAISDI and the associated .dat files:

CA Product	Primary Data File	ITIL Update File
CA 2E	integ2e.dat	
CA Enterprise Workload Automation	integAutoSys.dat	
CA Bundl	integBundl.dat	
CA 11 Enterprise Workload Automation Restart and Tracking	integCA11.dat	
CA Connect	integConnect.dat	
CA Datamacs	integDatamacs.dat	
CA Date Simulator	integDate.dat	
CA Deliver	integDeliver.dat	itil_integDeliver.dat
CA Dispatch	integDispatch.dat	itil_integDispatch.dat
CA FAVER VSAM Data Protection	integFaver.dat	

CA FileAge	integFileAge.dat	
CA File Master Plus	integFileMaster.dat	
CA Filesave RCS Automated Recovery	integFileSave.dat	
CA Gen	integGen.dat	
CA GSS (Common component)	integGSS.dat	
CA InterTest	integInterTest.dat	
CA JCLCheck Enterprise Workload Automation	integJCLCheck.dat	
CA Jobtrac Job Management	integJobtrac.dat	itil_integJobtrac.dat
CA LPD Report Convergence	integLPD.dat	
CA Optimizer	integOptimizer.dat	
CA Optimizer/II	integOptimizerII.dat	
CA Plex	integPlex.dat	
CA Scheduler Job Management	integScheduler.dat	
CA Spool Enterprise Print Management	integSpool.dat	itil_integSpool.dat
CA SymDump	integSymDump.dat	
CA Verify	integVerify.dat	
CA View	integView.dat	itil_integView.dat

Appendix A: Samples Directory

This section contains the following topics:

[Contents of the Samples Directory](#) (see page 525)

Contents of the Samples Directory

You can modify several files in the \$NX_ROOT/samples directory for use with various external interfaces. These files are grouped into various subdirectories. None of the files in the samples directory are executable as originally shipped.

asset

assetx.sch

This file defines the minimum definition necessary for an asset extension table.

assetx.maj

This file is a sample extension table.

assetx.spl

This file illustrates how to define the required method for copying an asset.

myco_demo.caz

This file contains a sample Business Process View.

How to Modify the Message Catalog

To modify the message catalog, complete the following steps:

1. Refer to the format of pdm.xml that is located in \$nx_root\bin.
2. Create a customized copy of pdm.xml and place it in the \$nx_root\msg_catalog directory.
3. Add, modify, or add and modify messages in the XML message files from the previous step.

macro_lock

This file contains a spel fragment that can be run using a bop_cmd to turn off locks that are being held by macros.

ntf_meth

These are sample notification methods. As noted, modifications may have to be made before using them with your system. These scripts are all written to run with a Bourne shell interpreter.

Note: Under UNIX, these scripts will run as written. However, on Windows NT, either a third party Bourne shell interpreter must be installed on the server or the scripts need to be rewritten as complied c files, or .bat files. Another possibility is to install a Perl interpreter and translate the scripts to Perl.

hp.pdm_pager0 (UNIX Only)

hp.pdm_pager1(UNIX Only)

hp.sendpage0(UNIX Only)

hp.sendpage1(UNIX Only)

pager.p(UNIX Only)

sun.pdm_pagera(UNIX Only)

sun.pdm_pagerb(UNIX Only)

sun.sendpagea(UNIX Only)

sun.sendpageb(UNIX Only)

These files contain sample notification methods for pagers. These are only examples and will not necessarily work with different pagers.

pager_notify.pl

This file contains a sample perl script that can be modified and used as a notification method for pagers.

Note: This file is only a sample and may not work with different pagers.

pdmconf

web.xml.tpl

pdm_startup.tpl

pdm_edit_usage_notes.htm

alias_install.bat

web.cfg.tpl

pdm_startup.i.tpl

pdm_edit.pl**README_files**

All of these files are used by `pdm_edit.pl` to create startup files for a primary server and secondary servers that are configured to run a variety of daemons.

call_mgt

Contains samples for customization in request management.

gencr.frg

This file can be used in conjunction with `bop_cmd` to create requests from a command line. All notifications and activity log entries will occur, however no Request Form will display on the server when created. You must use the `-u` parameter to execute `gencr.frg` with the `bob_cmd` utility. Be sure to read the `gencr_readme.txt` file to learn the syntax, and how to modify it if necessary. The file should be placed in `$NX_ROOT/site/mods/interp`, if the directory does not exist, you should create it. Example: `bop_cmd -d domsrvr -u nsm -f gencr.frg "gencr('My Description')"`

iss_site.mod

This file can be used to enable activity logging of site-adapted fields in issues. This file should be placed in `$NX_ROOT/site/mods/majic` after it has been changed for the site-adapted fields.

cr_site.mod

This file can be used to enable activity logging of site-adapted fields in requests. This file should be placed in `$NX_ROOT/site/mods/majic` after it has been changed for the site-adapted fields.

chg_site.mod

This file can be used to enable activity logging of site-adapted fields in change orders. This file should be placed in `$NX_ROOT/site/mods/majic` after it has been changed for the site-adapted fields.

gencr_readme.txt

This file contains instructions on how to use the `gencr.frg` file.

chg_site.spl

This file may be modified to change the mapping of attributes when creating a change order from a request. This file should be placed in `$NX_ROOT/site/mods/majic` after the appropriate changes have been made.

audlog_site.mod

This file can be used to enable audit logging of site-adapted fields. This file should be placed in \$NX_ROOT/site/mods/majic after it has been changed for the site-adapted fields.

Notify_add.spl (UNIX only)

This file can be used to add the request's log agent, assignee and group to the request notification list. This file should be placed in \$NX_ROOT/site/mods/majic.

Notify_replace.spl (UNIX only)

This file can be used to add the request's log agent, assignee and group to the request notification list when they are changed. This file should be placed in \$NX_ROOT/site/mods/majic.

sdk

This directory contains a sample file for making CA Service Desk Manager web service calls.

TableOfContents.doc further explains what is available, found in the following directory:

\$NX_ROOT\samples\sdk\websvc

PKI_loginServiceManaged_JAVA_steps.doc and
PKI_loginServiceManaged_PERL_steps.doc explain how to configure ServiceDesk for digital certificate logins in the following directory:

\$NX_ROOT\samples\sdk\websvc\java\test1_pki

The following lists PERL and JAVA samples, examples and locations:

PERL samples

\$NX_ROOT\samples\sdk\websvc\perl\test1_pki

Example: loginServiceManaged() web service call

JAVA samples

\$NX_ROOT\samples\sdk\websvc\java\test1_pki

Example: loginServiceManaged() and getBopsid() web service call

\$NX_ROOT\samples\sdk\websvc\java\test2_basic

Example: Combined CreateRequest() and CreateChangeOrder() web service call

Appendix B: Loading Supplemental Content

sd_content.dat

Supplemental content for CA Service Desk Manager is available in sd_content.dat. This data file contains Change Category and Root Cause records. To load the data from a command window, go to \$NX_ROOT/data and run the following command:

```
pdm_load -f sd_content.dat
```