CA Unicenter Service Desk Integrations

ca

Transforming
IT Management

LEGAL NOTICE

This publication is based on current information and resource allocations as of its date of publication and is subject to change or withdrawal by CA at any time without notice. The information in this publication could include typographical errors or technical inaccuracies. CA may make modifications to any CA product, software program, method or procedure described in this publication at any time without notice.

Any reference in this publication to non-CA products and non-CA websites are provided for convenience only and shall not serve as CA's endorsement of such products or websites. Your use of such products, websites, and any information regarding such products or any materials provided with such products or at such websites shall be at your own risk.

Notwithstanding anything in this publication to the contrary, this publication shall not (i) constitute product documentation or specifications under any existing or future written license agreement or services agreement relating to any CA software product, or be subject to any warranty set forth in any such written agreement; (ii) serve to affect the rights and/or obligations of CA or its licensees under any existing or future written license agreement or services agreement relating to any CA software product; or (iii) serve to amend any product documentation or specifications for any CA software product. The development, release and timing of any features or functionality described in this publication remain at CA's sole discretion.

The information in this publication is based upon CA's experiences with the referenced software products in a variety of development and customer environments. Past performance of the software products in such development and customer environments is not indicative of the future performance of such software products in identical, similar or different environments. CA does not warrant that the software products will operate as specifically set forth in this publication. CA will support only the referenced products in accordance with (i) the documentation and specifications provided with the referenced product, and (ii) CA's then-current maintenance and support policy for the referenced product.

Certain information in this publication may outline CA's general product direction. All information in this publication is for your informational purposes only and may not be incorporated into any contract. CA assumes no responsibility for the accuracy or completeness of the information. To the extent permitted by applicable law, CA provides this document "AS IS" without warranty of any kind, including, without limitation, any implied warranties of merchantability, fitness for a particular purpose, or non-infringement. In no event will CA be liable for any loss or damage, direct or indirect, from the use of this document, including, without limitation, lost profits, lost investment, business interruption, goodwill or lost data, even if CA is expressly advised of the possibility of such damages.

COPYRIGHT LICENSE AND NOTICE

This publication may contain sample application programming code and/or language which illustrate programming techniques on various operating systems. Notwithstanding anything to the contrary contained in this publication, such sample code does not constitute licensed products or software under any CA license or services agreement. You may copy, modify and use this sample code for the purposes of performing the installation methods and routines described in this document. These samples have not been tested. CA does not make, and you may not rely on, any promise, express or implied, of reliability, serviceability or function of the sample code.

TITLE AND PUBLICATION DATE

CA Unicenter Service Desk Integrations
Publication Date: September, 2008

ACKNOWLEDGEMENTS

Maria Lazar

Randal Locke

Venu Gopal Paleti

Vikas Piprade

Terry Pisauro

Richard J. Schneider

Cheryl Stauffer

Steve Troy

CA PRODUCT REFERENCES

This document references the following CA products:

- CA Unicenter Network and Systems Management (NSM) r11 through r11.1

- CA Desktop Management Suite (DMS) r11 through r11.2

- CA Unicenter Asset Portfolio Management (UAPM) r11.0 through r11.3

- CA Unicenter Patch Management r11.0 through r11.2

- CA Identity Manager r8.0

- CA SiteMinder® r6.0

- CA SupportBridge™ Live Automation r6.0

- CA Clarity™ r7.5.3

- CA Harvest Change Manager r7.1, SP1

- CA Unicenter Service Catalog r11 through r11.2

- CA SQM (Service Quality Management) r4.0

- CA CMDB r11.0 through r11.1

DOCUMENTATION CHANGES

The following is a list of new chapters and new or revised topics in the September, 2008 update to this Green Book:

■   Check the List of Open Tickets for an Asset (see page 95)—This topic has been updated to reflect the correct procedure to add a new option to the Asset Detail window in UAPM.

FEEDBACK

Please email us at greenbooks@ca.com to share your feedback on this CA Green Book. Please include the title of this Green Book in the subject of your email response. For technical assistance with a CA product, please contact CA Technical Support at http://ca.com/support.

Contents

Chapter 5: Desktop Management – Integrating with CA DMS   65

Chapter 6: Patch Management – Integrating with CA Unicenter Patch Management   79

Chapter 7: Financial Asset Management – Integrating with UAPM   89

Chapter 8: Service Management – Integrating with CA Unicenter Service Delivery and CA Unicenter Service Catalog   99

Chapter 9: Live Automation Management – Integrating with CA SupportBridge   111

Chapter 10: Password Management – Integrating with CA Identity Manager   121

Chapter 11: Application Change Management – Integrating with CA Software Change Manager for Distributed (formerly CA Harvest)   127

Chapter 19: Integrating with Third-Party Products   223

Appendix A: Code Examples 289

Index 311

Chapter 1: Introduction

This book describes a series of CA Unicenter® Service Desk integration capabilities with CA solutions and third-party products. These integrations can extend the capability of the service desk into other systems, and also help enhance the management and coordination of service support business processes.

The following information is provided for each integration:

■   An overview that includes the benefits the integration provides.

■   A real business process example illustrating the business challenge.

■   The approach that CA uses to solve the business challenge.

■   Recommended best practices for the integration.

■   Steps to set up, configure, troubleshoot and test the integration.

Who Should Read This Book?

This book provides the support technician, software architect, software developer, software engineer, and system administrator with the information necessary to integrate CA Unicenter® Service Desk with a variety of CA solutions and third-party products. It is intended for technically-oriented people who require an advanced level of understanding about the CA Unicenter Service Desk integration capabilities to successfully configure and maintain their service desk environment.

The information for each integration is organized by chapter into the following major categories:

■   The CA Integration Platform and CA Unicenter Service Desk

■   Out-of-the-Box Integration with CA Solutions

■   Network Management

■   Desktop Management

■   Patch Management

■   Financial Asset Management

■   Service Management

- Live Automation Management

- Password Management

- Software Application Change Management

- Portfolio Project Management

- Change Management

- Work flow Management

- Security Management

- Accessibility Management

- Service Quality Management

- Integration with Third-party Products

Chapter 2: The CA Integration Platform and CA Unicenter Service Desk

CA is an industry leader in providing integrated solutions. Enterprise IT Management (EITM) is CA's vision for enabling a new level of management control across the enterprise by integrating and automating the management of IT applications, databases, networks, security, storage, and systems across departments and disciplines to realize the full potential of each integrated solution and the business services it supports. EITM enables companies to unify and simplify IT management across their enterprise for greater business results, and have comprehensive visibility into the quality, costs, and risks for the services they provide. Through CA's comprehensive integrated solutions, customers can *govern*, *manage*, and *secure* IT – the three pillars of EITM.

The Key Elements of EITM

The key elements of EITM are defined as follows:

Govern

■ IT organizations put IT governance processes in place to ensure they make the most effective IT investment decisions to support their business strategy. CA's governance solutions are designed to help the IT organization understand and account for the portfolio of resources they have, and optimize the deployment of that portfolio in the most cost efficient and effective way possible.

■ Important governance metrics, including insight into the quality and costs of providing an end-to-end service rather than its individual parts, help management make informed decisions to deliver superior business value in an optimized way.

Manage

■ CA's management solutions cover all areas of IT management, from service management through systems and network management to workload automation.

■ No longer operating in a technology silo arrangement, effective IT management now has to link the IT infrastructure components to the business services they support. CA solutions provide dynamic insight into these relationships. This helps customers to get the information they need to create and deliver on service level agreements with other areas of the business that rely on the IT infrastructure to run their day to day business processes.

Secure

■ CA security solutions can protect, monitor and actively manage nearly every facet of the enterprise, from business processes down to every infrastructure asset.

■ Increased focus on risk management, information integrity, and compliance has caused security to evolve from a reactive technical discipline into a frontline business enabler. CA solutions help organizations understand, measure, reduce and mitigate operational and business risk. This helps achieve an organization's goals of ensuring on-going service continuity while meeting compliance with regulations.

The Unified Service Model

The Unified Service Model is the centerpiece of CA's vision for delivering EITM, and provides a complete 360 degree or common view into the IT services delivered to the business. The Unified Service Model defines the characteristics of a business service, including component and relationship details, service levels, prices, costs, quality, risks and exposures, identity and entitlement rights, and more.

The CA Integration Platform

The CA Integration Platform is the architectural foundation upon which CA's products are integrated. It leverages a service-oriented architecture to deliver a set of shared, modular services including an integrated workflow, common policy, consistent user experience, and scheduling services.

The following illustration shows the CA Integration Platform, containing a rich set of management and security services that deliver consistent definition and behavior.
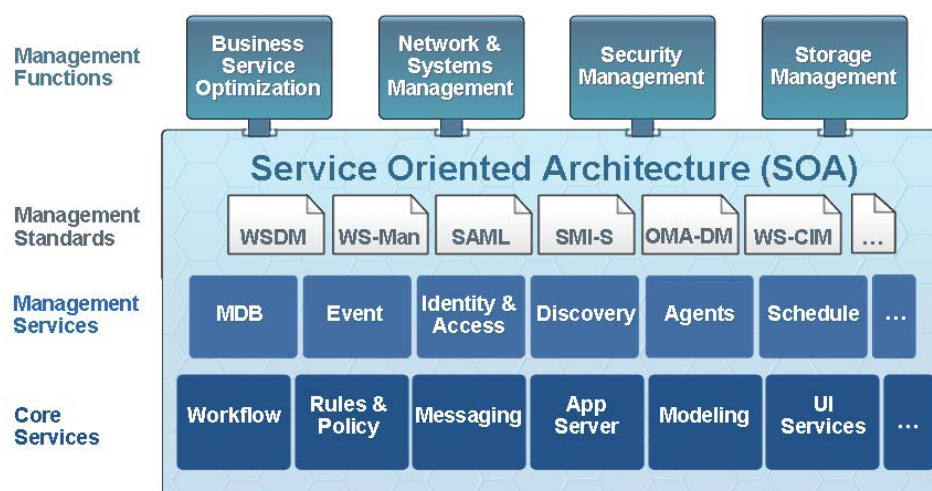
Note: In this Green Book, we will talk about one of the top integration points delivered by CA: A Central Service Hub – Integrated CA Unicenter Service Desk and the value that this integration delivers to our customers.

CA Unicenter Service Desk

CA Unicenter Service Desk is an enterprise-level service desk solution that can meet the most challenging support requirements, and can scale down to meet small and medium service support needs. The technology utilized by the solution can run in heterogeneous UNIX and Windows environments. This comprehensive solution can be integrated with other products to automate and manage both external (customer support) and internal (enterprise service center management) service desks and their associated business processes.

With the increasing adoption of ITIL best practices, the service desk becomes the cornerstone for automating IT processes and provides audit trails for regulatory compliance. CA Unicenter Service Desk provides you with a reliable foundation to improve efficiencies while fostering customer satisfaction and improved productivity.

CA Unicenter Service Desk Features

CA Unicenter Service Desk provides the following features:

■   Provides integrated incident, problem, request, and change management

■   Empowers users with self-service

■   Manages service levels to ensure response time commitments

■   Uses automated customer surveys to ensure high quality support

■   Provides built-in integrations to the CA Unicenter family of products

■   Provides tools for robust integrations with third-party products

CA Unicenter Service Desk Integration Capability

The CA Unicenter Service Desk integration points can be defined in the following terms:

■   Data

■   Common User Interface

■   Interoperability

Data

CA Unicenter Service Desk utilizes the CA Management Database (MDB). The MDB is a common enterprise data repository that supports CA product suites. The MDB provides a unified database schema for the management of data stored by all CA solutions. Customers can extend the MDB to include additional IT management data from non-CA software products and solutions.

The primary business benefits of the MDB approach are the following:

- Integrated IT management data

- Integrated database administration

The following illustration shows how the MDB, in combination with CA solutions, enable full integration for managing your IT infrastructure:
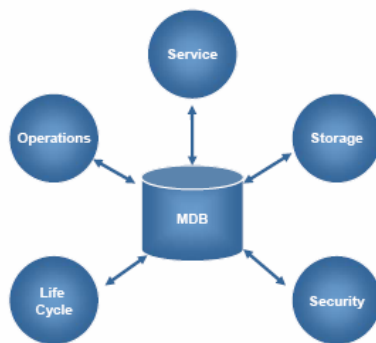
As previously illustrated, a unified database schema for IT management data provides increased visibility into the underlying IT organization, including storage, performance, hardware, and software information. This visibility improves IT decision making and helps to reduce costs and increase utilization of the infrastructure. Using a unified database schema helps you make informed decisions about hardware, software, and storage purchases, as well as provisioning, scheduling, data protection, and more.

The unified MDB schema enables the integration of solutions without additional programming effort. Integration methods that do not include a single database schema require point-to-point programming efforts to access data from disparate sources. Data integration enables rapid development of new product features because you do not have to create interfaces to make disparate data available. Without an MDB, data is stored in multiple locations and schemas, making it difficult to integrate and create new features that take advantage of data relationships.

Note: For information about how the MDB manages and shares data across CA products, see the *Incident and Problem Management Green Book* using the Technical Support link at http://ca.com/support.

Common Asset Viewer

The CA Unicenter Common Asset Viewer (formerly the Asset Maintenance System, or AMS) provides a common interface for viewing purchased and discovered assets, asset types, model definitions, asset families, asset classes, asset statuses, and General Ledger (GL) codes among CA Unicenter solutions involved in some aspect of asset management. The CA solutions that use this common interface include CA Unicenter Service Desk, CA Desktop Management Suite (CA DMS), CA Unicenter Asset Portfolio Management (UAPM), and CA Unicenter Network and Systems Management (CA Unicenter NSM).

Configuring the Common Asset Viewer for the organization provides the following benefits:

■   A combination of at-a-glance data from multiple asset applications

■   Navigation between managed and associated discovered assets

■   Navigation to asset applications in context of the asset

The following illustration shows a sample CA Unicenter Common Asset View window:

Note: For information about the Common Asset Viewer, see the *Incident and Problem Management Green Book* using the Technical Support link at http://ca.com/support.

Application Interoperability

CA Unicenter Service Desk can interact with CA solutions and third-party products through Application Programming Interfaces (APIs) that allow the interoperability and exchange of information based on users' interaction with the solutions. In CA Unicenter Service Desk r11, the following methods provide integration capabilities with other products and solutions:

1. Web Services
2. bop_cmd
3. text_api
4. Database_Level
5. pdm_load
6. pdm_extract
7. pdm_deref
8. eMail Interface
9. Incoming Event Integration (AHD.dll)
10. Outgoing notification integration
11. Remote Reference
12. CTI Integration
13. URL
14. External Authentication
15. External Table Display
16. Majic Triggers
17. Service Desk Events and Action Macro

Note: For more information about each method, see the chapter "Integrating with Third-Party Products."

Integration Value

The following illustration shows a high-level overview of an integrated CA Unicenter Service Desk:

As previously illustrated, the CA service desk solution is designed not only to manage the infrastructure, but also to support business initiatives. For many organizations, it is the central management point that translates business policies into work processes. To accomplish these goals, the service desk needs to seamlessly integrate with numerous CA solutions and third-party products.

The business benefits include the following:

■   Service desk staff members are more productive, and have more time to spend addressing more complex challenges.

■   Users are more productive and can solve their own simple incidents using knowledge capabilities.

■   Service desk managers see a decrease in call volumes, talk times, and resolution times, and user satisfaction increases.

■   Operations staff productivity increases through automation, and service levels improve. The administrative staff has more complete information, which increases the mean time to repair and facilitates more proactive intervention. Potential problems with the IT infrastructure are identified, resolved, and documented. And, in many cases, before users are aware of the problem.

■   Overall productivity increases, due to tighter integration between teams and reduced duplication and manual data entry.

■   An IT support director can reallocate resources because fewer technicians can now effectively support more users. The challenge to provide higher levels of services while reducing costs becomes easier.

- The CIO sees IT systems supported and maintained more effectively, resulting in a better return on the corporate IT investment.

- Reducing user downtime from invalid password authentication with CA Identity and Access Management, which enables users to reset their own passwords securely through self-service capabilities.

- Improving customer satisfaction and first call resolution rates by providing users with answers to common questions and service desk analysts with resolutions to known errors.

- Speeding problem discovery and the resolution process with automatic issue detection and incident creations with CA network and systems management.

- Expediting issue diagnosis and resolution by providing service desk analysts with comprehensive asset details of the affected user with the auto-discovery capabilities in CA asset management.

- Minimizing risk and business disruptions with an enterprise change management solution that manages and controls the change process across the organization.

- Reducing costs by providing service desk analysts with instant access to financial asset information such as warranties, maintenance agreements, or leases, for better decision support.

- Simplifying the provisioning and administration of users to ensure they are able to access information they are authorized to access, and denied access to information they are not authorized to access.

The service desk is empowered by knowledge of the IT and customer environments. Without integration points, the service desk staff works only with user information provided by the affected caller. There is little opportunity for automation and little to leverage for call volume reduction. In contrast, some sites reduce their call volume by as much as 30 percent simply by enabling seamless integration with a self-service password reset technology. The most time-consuming task when resolving issues is related to analysts gathering configuration data. When this data is gathered electronically, the *talk time* decreases and the *first call close rate* increases.

Chapter 3: Out-of-the-Box Integration with CA Solutions

Overview

Many CA solutions include built-in integration to CA Unicenter Service Desk r11. The Application Programming Interface (API) used for these integrations is the Web Services API. The CA Unicenter Service Desk Web Services API is built on the standards set by the World Wide Web Consortium (W3C), including Simple Object Access Protocol (SOAP) and Web Service Description Language (WSDL). These XML-based standards, combined with the well-known HTTP protocol, provide robust integration capabilities. Using a Web Services API to develop integrations provides the following advantages:

- Enables communication between products and solutions, even if they exist on different platforms.

- Permits access to the CA Unicenter Service Desk business logic from virtually anywhere.

The increased use of web services has transformed it into an easily understood and ubiquitous technology, making it an ideal choice for use as an API.

Note: For information about Service Oriented Architecture (SOA) and web services, including background information and examples, see the *Incident and Problem Management Green Book* using the Technical Support link at http://ca.com/support.

High-Level Architectural Components

The following diagram illustrates, at a high level, the components of the CA Unicenter Service Desk Web Services API along with how the components communicate with each other:
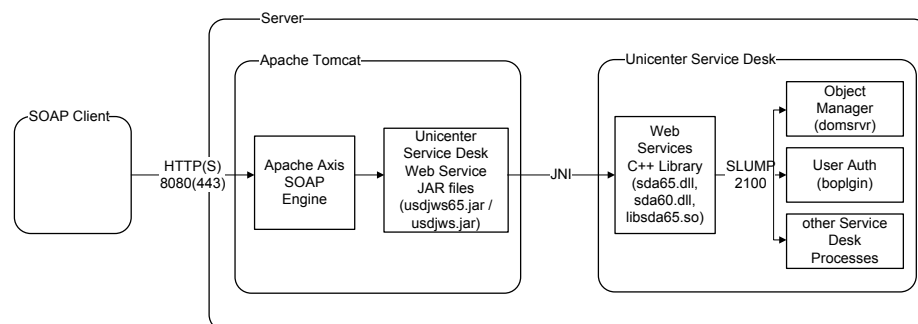
The following information explains each component in the architecture:

- **SOAP Client**. Initiates the communication with the CA Unicenter Service Desk Web Services API. This component can reside virtually anywhere, including directly on the CA Unicenter Service Desk server. By default, communication will be over HTTP on port Transmission Control Protocol (TCP) 8080. The TCP port number is configurable, in addition to the use of Secure Sockets Layer (SSL), which generally use TCP port 443.

- **Apache Tomcat**. The stand-alone Open Source JSP and Servlet Container from the Apache Foundation.

- **Apache Axis**. A SOAP engine that plugs into the Tomcat servlet engine and allows you to deploy CA Unicenter Service Desk Web Services API as a Tomcat web application.

- **CA Unicenter Service Desk JAR files**. The JAR files contain the Java method interface code to support the code that is generated by the Apache Axis products. They must be included in the Axis WEB-INF/lib directory, and they implement handlers for each web service method. Most of the work is then passed to the SDA module using Java Native Interface (JNI). The r11 version of the JAR file is usdjws65.jar, while the r6.0 version of the JAR file is usdjws.jar.

- **CA Unicenter Service Desk Web Services C++ library**. A multi-threaded component that manages the web service sessions, application of CA Unicenter Service Desk Web Service policies, and Public Key Infrastructure (PKI) functionality. This component passes web service method requests to the key internal CA Unicenter Service Desk components over SLUMP (default TCP port 2100). The Windows r11 version of the file is sda65.dll, the Windows r6.0 version of the file is sda60.dll, and on UNIX and Linux, there is only a single file named libsda65.so.

  Note: The web services PKI features provide an additional layer of security when authenticating to CA Unicenter Service Desk. For information about PKI authentication and the loginServiceManaged method, see the *CA Unicenter Service Desk Web Services User Guide* and the *Incident and Problem Management Green Book* using the Technical Support link at http://ca.com/support.

- **Internal CA Unicenter Service Desk Components**

  - > **Object Manager (domsrvr).** The *heart* of the CA Unicenter Service Desk architecture, defining and managing the business objects used by other processes in a product. Key tasks include managing the sharing of objects and lists of objects to minimize database I/O, controlling object locking, and enforcing data-based security restrictions by user.

  - > **User Authentication module (boplogin).** Handles user authentication for a CA Unicenter Service Desk deployment. Processes requiring user authentication send a userid and an encrypted password, and this component provides for the validation of the user.

  - > Other CA Unicenter Service Desk processes

Debugging CA Unicenter Service Desk Web Services

Here are some simple tips to help you confirm the web services functionality and debug issues that you may encounter:

Ensure Tomcat is Running and Axis is "Happy"

Ensure Tomcat is running and Axis is "Happy" by confirming that the following URL produces the Axis Happiness Window:

```
http://localhost:8080/axis/happyaxis.jsp
```

Monitor SOAP Messages

Use the provided Apache Axis utility named TCPmon to monitor SOAP messages moving between your browser and server so that you can see what is being sent along with the response from Tomcat. You must change the port in the client code so that TCPmon can intercept the SOAP traffic, and then relay your client's SOAP messages onto the proper port.

Start TCPmon using the following command:

```
"C:\Program Files\CA\SharedComponents\JRE\1.4.2_06\bin\java.exe" -cp axis.jar
org.apache.axis.utils.tcpmon 8081 localhost 8080
```

In this command, the following is assumed:

■   You are running on the same server as Tomcat

■   Your client code is using Port 8081

■   You are in the <CA Unicenter Service Desk installation Directory>\bopcfg\www\CATALINA_BASE\webapps\axis\WEB-INF\lib directory when running this command

■   The SharedComponents directory is in the default location and you have not changed it

Examine Log Files

If you encounter an error or problem, examine the following log files:

■   In the *<CA Unicenter Service Desk Installation Directory>\log* directory you have the following files:

> stdlog.X files, where *X* is a number between 0 and 9 and the logs increases to a configurable size (default is 3MB) before moving to the next number in the sequence.

> pdm_tomcat.log

> j*.log – look for all logs that start with the letter "j"

■ In the *<CA Unicenter Service Desk Installation Directory>\bopcfg\www\CATALINA_BASE\logs* directory there are a number of log files. Examine any with the appropriate timestamp.

Best Practices

■ It is a good idea to use a separate user ID (do not use the ServiceDesk user) for each different set of web services activity being built. This will allow you to more easily determine which messages in the log file can be attributed to which portion of your web services code.

■ When dealing with a high-volume implementation, the Tomcat web server that is leveraged benefits from configuring it to use additional memory. By default, CA Unicenter Service Desk starts the Tomcat instance with a 512 MB memory pool. This example allows the memory change to be permanent within a CA Unicenter Service Desk r11.2 environment and allows you to increase the Tomcat memory to 1024 MB.

Preparing for the Memory Increase

To help prepare for the memory increase, complete the following steps on your test system:

■ Make sure your system can handle the recommended memory increase.

Note: To verify that the Java Virtual Machine (JVM) utilized by Tomcat can handle the increased memory, enter the following command at the operating system prompt:

```
Java –Xmx<value>|more
```

In this command, *<value>* is the memory value. That is, Java –Xmx1024M|more

If you see an error similar to the following, the setting will not work.

```
Error occurred during initialization of VM
Could not create Java virtual machine
Could not reserve enough space for object heap
```

However, if you see information explaining the use of Java.exe, the test was successful.

■ Make a backup of every file that you have modified. This will allow for a back out recovery.

Increasing the Memory

The default memory value is specified in the CA Unicenter Service Desk pdm_tomcat startup parameter list in the CA Unicenter Service Desk stdlog file. The default parameter is –Xmx**512M**. CA Unicenter Service Desk stdlog logs are located under the $NX_ROOT/log directory, where $NX_ROOT corresponds with the CA Unicenter Service Desk installation directory.

1. Add a new variable named @NX_TOMCAT_CMD to the end of the $NX_ROOT\pdmconf\NX.env_nt.tpl file with a new memory value, beside the –Xmx parameter.

   Important! The following is an example of the variable to add to the file. Do not copy or use this example in your installation. The file directory and drive locations in this example will most likely be different from your implementation.

   ```
   X_TOMCAT_CMD="D:/PROGRA~1/CA/Shared~1/JRE/1.4.2_06/bin/javaw -
   Djava.net.preferIPv4Stack=true -Xrs –Xmx1024M -
   Djaas.config=D:/PROGRA~1/CA/SERVIC~1/add-ons/caflow/jaas.config -
   Djava.security.auth.login.config=D:/PROGRA~1/CA/SERVIC~1/add-
   ons/caflow/jaas.config -
   Djdbc.baseDriver=com.microsoft.sqlserver.jdbc.SQLServerDriver -
   Dsun.io.useCanonCaches=false -Djava.awt.headless=true -
   Djava.endorsed.dirs=D:/PROGRA~1/CA/Shared~1/tomcat/4.1.31/common/endorsed -
   classpath
   D:/PROGRA~1/CA/Shared~1/tomcat/4.1.31/common/lib/tools.jar;D:/PROGRA~1/CA/Shared
   ~1/JRE/1.4.2_06/java/lib/tools.jar;D:/PROGRA~1/CA/SERVIC~1/java/lib/tools.jar;D:
   /PROGRA~1/CA/Shared~1/tomcat/4.1.31/bin/bootstrap.jar;D:/PROGRA~1/CA/SERVIC~1/ja
   va/lib/sqljdbc.jar -
   Dcatalina.base=D:/PROGRA~1/CA/SERVIC~1/bopcfg/www/CATALINA_BASE -
   Dcatalina.home=D:/PROGRA~1/CA/Shared~1/tomcat/4.1.31 -
   Djava.io.tmpdir=D:/PROGRA~1/CA/SERVIC~1/bopcfg/www/CATALINA_BASE/temp
   org.apache.catalina.startup.Bootstrap"
   ```

2. Press the Enter key after adding the new text to the file.

   Looking closely at this example, there are three *main attributes* that you will need to implement:

   > The use of forward slashes (UNIX style) instead of the back slashes.

   > The use of eight character path names instead of the Windows long file names.

   > The removal of double quotes, because long file names are not used. Double quotes are only specified at the beginning and end of the large string.

   Important! The previous line after **@NX_TOMCAT_CMD=** must come from the CA Unicenter Service Desk stdlog files of the computer where this will be implemented. Search for "startup_tomcat" within your stdlog file located under $NX_ROOT\log. When you start CA Unicenter Service Desk, you will see a line similar to the following in the stdlog files. This example illustrates the default value of 512 MB.

   ```
   02/06 09:24:11.58 yunle01A pdm_tomcat        3304 SIGNIFICANT  pdm_tomcat.c
   479 startup_tomcat: C:\Program Files\CA\SharedComponents\JRE\1.4.2_06\bin\javaw
   -Djava.net.preferIPv4Stack=true –Xrs -Xmx512M -
   Djaas.config=C:\PROGRA~1\CA\SERVIC~1\add-ons\caflow\jaas.config -
   ```

```
Djava.security.auth.login.config=C:\PROGRA~1\CA\SERVIC~1\add-
ons\caflow\jaas.config -
Djdbc.baseDriver=com.microsoft.sqlserver.jdbc.SQLServerDriver -
Dsun.io.useCanonCaches=false -Djava.awt.headless=true -
Djava.endorsed.dirs="C:\Program
Files\CA\SharedComponents\tomcat\4.1.31\common\endorsed" -classpath "C:\Program
Files\CA\SharedComponents\tomcat\4.1.31\common\lib\tools.jar;C:\Program
Files\CA\SharedComponents\JRE\1.4.2_06\java\lib\tools.jar;C:\PROGRA~1\CA\SERVIC~
1\java\lib\tools.jar;C:\Program
Files\CA\SharedComponents\tomcat\4.1.31\bin\bootstrap.jar;C:\PROGRA~1\CA\SERVIC~
1\java\lib\sqljdbc.jar" -
Dcatalina.base="C:\PROGRA~1\CA\SERVIC~1\bopcfg\www\CATALINA_BASE" -
Dcatalina.home="C:\Program Files\CA\SharedComponents\tomcat\4.1.31" -
Djava.io.tmpdir="C:\PROGRA~1\CA\SERVIC~1\bopcfg\www\CATALINA_BASE\temp"
org.apache.catalina.startup.Bootstrap start
```

The following example illustrates in simpler terms:

```
02/22 15:03:46.80 valre03- pdm_tomcat        3100 SIGNIFICANT  pdm_tomcat.c
479 startup_tomcat: <HUGE COMMAND LIKE THE ONE ABOVE>
```

This is the one that you need to use. The last word, "start", on the HUGE COMMAND is not used. Remember to change the stdlog value that you will be specifying with the new @NX_TOMCAT_CMD variable that you will be adding to match the three previously mentioned attributes.

3. Open the $NX_ROOT\pdmconf\pdm_startup.i.tpl file.

   At the end of this file, you will see the following:

   command = "$NX_ROOT/bin/pdm_tomcat_nxd -s"

4. Change the command to the following:

   command = "$NX_ROOT/bin/pdm_tomcat_nxd -s **-C $NX_TOMCAT_CMD**"

5. Run pdm_configure.

   Your Tomcat server's maximum memory size will now be 1024 MB.

   You will typically need to restart the Tomcat server to reboot the system. The JVM must be able to access a contiguous block of memory on startup.

Verifying the Memory Increase

Use the following steps to verify that the memory increase was successful:

1. Check stdlog.x, located under the $NX_ROOT/log directory, for "startup_tomcat" during the timeframe of the current startup of CA Unicenter Service Desk after completing all changes. Within the stdlog line where "startup_tomcat" is specified, you should see –Xmx**1024M**, containing the new value of **1024M** that the previous example used.

   Important! If you do not see this, there has been an editing error and you should repeat the steps to increase the memory.

2. Verify that you can log in to the CA Unicenter Service Desk Web Interface successfully.

3. If you are using CA WorkFlow, verify that you can log in to CA WorkFlow from the CA Unicenter Service Desk primary computer successfully and that you do not see any "cannot connect to PM" errors.

   Important! If you receive a "cannot connect to PM" error, Tomcat did not start successfully. This is caused by an editing error and you should repeat the steps to increase the memory.

Backing Out the Memory Increase

If you still have problems verifying the memory increase after repeating the steps to increase the memory multiple times, replace all the edited files with the backups you created, and then rerun pdm_configure to recover.

What's New in CA Unicenter Service Desk r11?

With CA Unicenter Service Desk r11, the CA Unicenter Service Desk Web Services API brought the Knowledge Tools and CA Unicenter Service Desk WSDL functionality together while providing a backward-compatible WSDL for the CA Unicenter Service Desk 6.0 functionality (*excluding* Knowledge Tools). Providing all business policy through one WSDL simplifies the process of using this functionality. In addition, r11 introduced a number of enhancements around policy-based support in the Web Services API. Specifically, it is now possible to configure Web Services policy records in CA Unicenter Service Desk, and then by referring to these policy records it is possible to do the following:

■ Control the security rules and authentication mechanisms.

■ Control whether a secure protocol such as HTTPS should be used for login, all communication, or both.

■ Prevent ill effects of ticket flooding and denial of service attacks.

■ Provide simplified access into the configuration of business policies and work flow so that developers do not need to design this type of logic into their web services code.

   Note: This allows you to leverage your existing configured processes, potentially covering a wide variety of areas such as Incident, Problem, Change, Configuration, Release, Availability and Service Level Management, without your development staff having to be concerned with the specifics of these processes. This can be configured by the CA Unicenter Service Desk administrator.

When to Use the CA Unicenter Service Desk Web Services API

The CA Unicenter Service Desk Web Services API is invaluable when you need to integrate with the following:

■ An external product

■   A new custom coded feature

Integration with numerous external products is available out-of-the-box with companion solutions from CA. In addition, some partner products offer bridging solutions for simplifying the integration between products. If your requirements exceed the available list of supported products, then a custom integration can be designed and implemented.

When to Avoid the CA Unicenter Service Desk Web Services API

Since the CA Unicenter Service Desk Web Services API is fundamentally a programming interface, it may not be the best option in the following circumstances:

■   You need a solution that is addressed by an out-of-the-box or configurable component.

■   Your organization has little or no development support staff, or is unwilling to implement and maintain custom code.

Summary

The CA Unicenter Service Desk Web Services API is a robust and full-featured interface into the business logic that has been configured to support your IT processes. It is heavily used in the implementation of CA's out-of-the-box inter-product integrations and by partner solutions. This interface can also be leveraged by custom-written code, but this option should be carefully analyzed to ensure it fits with your organizational policy and goals.

CA Solutions and Out-of-the-Box Integration with CA Unicenter Service Desk r11

The following table illustrates CA solutions and versions that can integrate out-of-the-box with CA Unicenter Service Desk versions r11.0, r11.1, and r11.2. These integrations are described in this guide. Integrations with other products will be addressed in future updates to this guide.

CA Solution	CA Unicenter Service Desk r11.0	CA Unicenter Service Desk r11.1	CA Unicenter Service Desk r11.2	Integration Method
CA Unicenter Network and Systems Management (NSM)	r11.0	r11.1	r11.1	Web services ahd.dll
CA Desktop Management Suite (DMS)	r11.0	r11.1	r11.1, r11.2	Web services
CA Unicenter Asset Portfolio Management (UAPM)	r11.0	r11.1	r11.1, r11.2.1, r11.3	Web services
CA Unicenter Patch Management	r11.0	r11.1	r11.1, r11.2	Web services
CA CMDB	N/A	N/A	r11.0, r11.1	N/A
CA Identity Manager	8.0	8.0	8.0	URL
CA SiteMinder®	r6	R6	r6	URL
CA SupportBridge™ Live Automation	N/A	N/A	6.0	Web services
CA Clarity™	N/A	N/A	7.5.3	Web services
CA SCM for Distributed (formerly CA Harvest)	N/A	7.1 Patch 1	7.1 Patch 1	Web services JHSDK (Java API) and Remote References
CA Unicenter Service Catalog	r11.0	r11.1	r11.1, r11.2	Work flow and Web services
SQM (Service Quality Management)	N/A	N/A	4.0	Web services

Chapter 4: Network Management – Integrating with CA Unicenter NSM

What Is CA Unicenter NSM?

CA Unicenter Network and Systems Management (CA Unicenter NSM) is a comprehensive management solution that you can use to do the following:

■ Discover, monitor, and analyze infrastructure elements

■ Gather reports on system and network status to help avoid potential bottlenecks

■ Set up automated actions to improve performance

■ Analyze the health, status, and performance of your infrastructure elements to improve management and facilitating optimization

■ Display and report system status, problems, and history in a number of secure and personalized facilities including the Management Command Center (CA Unicenter MCC), CA Unicenter Management Portal (CA Unicenter MP), Event Console, and visual maps (such as Association Browser and 2D Map)

■ Display overview and detailed views of infrastructure elements with a single interface, the CA Unicenter MCC. These visualizations let you identify and understand information relevant to particular operational roles

Integration Value

The CA Unicenter NSM integration is available through the following CA Unicenter NSM components:

■ Alert Management (AMS)

■ NSM Event Console

■ Management Command Center (MCC)

■ WorldView – 2D Map

■ CA Unicenter Management Portal (UMP)

The Alert Management System (AMS)

The Alert Management System (AMS) is a CA Unicenter NSM component that can be used to organize and track the most important events in an enterprise, or a logical segment of an enterprise, allowing you to focus on and manage the highest severity IT events. The AMS provides tools for defining alert policy and multiple panes in the Management Command Center for viewing alerts.

Integration Points from the CA Unicenter NSM Alert Management System (AMS)

The AMS integration functionality is installed automatically with CA Unicenter NSM. Alert Management integration is new in CA Unicenter NSM r11; however it can be configured to work with CA Unicenter Service Desk 6.0 and r11. It was developed using Web Services, which makes CA Unicenter NSM a service-aware application. This means that CA Unicenter NSM can trigger the creation, update, and close of incidents, problems, or requests into CA Unicenter Service Desk 6.0 and r11, based upon the occurrence of certain events or alerts in CA Unicenter NSM. Through the Alert Management integration, there is additional functionality that can be used to start a search of CA Unicenter Service Desk Knowledge Tools in context, by using the Alert Text as a search argument. The result is a display of knowledge base articles that are potentially related to the alert.

The CA Unicenter NSM Event Console

The Event Console is a visual window into event activity that lets you view and immediately respond to events, as they occur. The console provides two areas to view event messages written to the console log.

■   The *held messages* area displays messages that require a response. These messages are often critical and require immediate attention or require an operator reply. Held messages that require an operator reply are Write To Operator with Reply (WTOR) messages. When no held or WTOR messages exist, the area reserved for messages of that type is not displayed. If a reply pending message (WTOR) has been sent, and either the Event Manager or the entire system goes down while the message is still pending, the message is queued and activated automatically (appears to be still active) when the Event Manager is brought back up.

■   The *log messages* area displays all logged messages (including held messages). Through message records and actions, you can further highlight these messages with a variety of colors and attributes to make them more noticeable on the console display.

Integration Points from the CA Unicenter NSM - Event Console

The integration capability through the CA Unicenter Event Management console is enabled by making a CA Unicenter Service Desk AHD_Call process from CA Unicenter NSM Messages Records and Actions, allowing CA Unicenter NSM to do the following:

- Send generic event data to CA Unicenter Service Desk

- Automatically generate requests and incidents

- Automatically post announcements to the CA Unicenter Service Desk scoreboard

- Add asset records to CA Unicenter Service Desk from CA Unicenter NSM

The Management Command Center (MCC)

The Management Command Center (MCC) integrates all CA Unicenter enterprise and network monitoring functionality into a single console. The MCC provides dynamic multi-viewer content relevant to any asset in the MDB.

Using the MCC, you can do the following:

- Create customized views of data

- Apply filters to data

- Create class-based queries

- View alerts and Event consoles

- Create reports

Integration Points from Management Command Center (MCC)

The following are integration points from the Management Command Center (MCC):

- Interactive start, in context, of the CA Unicenter Service Desk web interface to view or create a request in CA Unicenter Service Desk from the CA Unicenter NSM Alert Management window.

- View or create a request in CA Unicenter Service Desk for a particular node from the MCC topology functionality.

- Look up and access CA Unicenter Service Desk knowledge documents based on the alert from within MCC.

The Worldview - 2D Map

The 2D Map is a two-dimensional, geographical representation of the logical structure of your enterprise. The map simplifies managing your business resources with tools you can

use to customize visual representations of your business enterprise. This display is primarily based on objects that represent entities in the MDB.

Integration Points from the Worldview - 2D Map

The CA Unicenter NSM 2D Map lets you access and create CA Unicenter Service Desk requests by right-clicking a managed object to display two menu options: Create Request and Request List.

The WorldView Interface Integration Points (Business Process Views on the 2D Map or MCC Topology Interface)

In addition, as part of the integration capabilities, CA Unicenter Service Desk r11.x offers Change Impact Analyzer (CIA) functionality to support the Change Management Process. CIA can be integrated with CA Unicenter NSM Worldview, allowing customers to implement change management controls over the organization's CIs and visualize the changes through Business Process Views (BPVs) displayed on the 2D Map or Management Command Center (MCC).

Communication between the CA Unicenter Service Desk and WorldView is accomplished utilizing a Windows service routine. This communication allows the transfer and synchronization of asset data between CA Unicenter Service Desk and WorldView, which is critical for keeping WorldView refreshed with the most up-to-date asset information from the CA Unicenter Service Desk database. Asset information includes asset status and relationship data. This data, when viewed graphically in WorldView, allows you to see the real-time statuses and relationships of assets, and to identify problems and formulate solutions.

You can also introduce changes to the current WorldView map to see the impact the changes will have on your organization. Through menu items provided in the interface, CA Unicenter Service Desk makes accessible the facilities to export asset information to WorldView, synchronize status information between the two systems, and view and maintain relationship information. In addition, Change Impact Analyzer provides command line utilities that can be run on-demand. Most of the utilities are run automatically during initial setup, or can be run directly from CA Unicenter Service Desk. They are provided as stand-alone components for flexibility in managing the Change Impact Analyzer integration.

Note: Change Impact Analyzer (CIA) is also available in CA Unicenter Service Desk 6.0, and can run with CA Unicenter NSM r11.x.

The CA Unicenter Management Portal (UMP)

The CA Unicenter Management Portal (UMP) is your personalized online workplace, providing interactive web access to the information that is important to you. The portal includes the following features:

- **Multiple customizable workplaces**. You can create one or more personalized workplaces to organize your information and use a single sign-on to access the

information. After creating your workplaces, you can add content using an intuitive *select and act* process. You can also resize the display areas for maximum effect.

■ **Dynamic personalization**. The technology behind the portal lets the administrator customize content for your preferences and needs. Automatic scheduled updates mean the information to which you subscribe is always up-to-date.

■ **Cross-application content**. The portal seamlessly integrates most web-enabled information sources, such as Microsoft Office documents, email, web pages, and reporting and analysis tools. You can share information with other users by publishing and categorizing content and subscribing for automatic delivery of relevant updates.

■ **Discussion Boards**. You can create discussion boards in the portal to share information.

■ **Advanced search features**. You can use the built-in search features to perform precise searches and accurately find the information you need in the portal server database, on your intranet, or on the World Wide Web (WWW). You can even publish the information you find directly to the portal server for easy retrieval later.

■ **Content Management**. The new content management feature of CleverPath Portal supports enhanced information delivery. It provides both standard content management abilities and work flow management. These capabilities are presented through a series of new portlets.

Integration Points from the CA Unicenter Management Portal

The CA Unicenter Management Portal allows users to access the CA Unicenter Service Desk interface in context directly from the portal interface. In addition, when CA Unicenter Service Desk is defined as a CA Unicenter Management Portal portlet, the CA Unicenter Service Desk user interface can be accessed through the CA Unicenter Management Portal along side the CA Unicenter NSM user interfaces, thereby providing a "single pane of glass" interface starting point.

Command Prompt Integration Points

Assets discovered using CA Unicenter NSM or CA DMS can be imported into the CA Unicenter Service Desk MDB using pdm_nsmimp.

Integration Value

By integrating CA Unicenter NSM and CA Unicenter Service Desk, organizations can do the following:

■ Automatically control network management issues detected by CA Unicenter NSM Event and Alert Management

- Automatically coordinate critical management events detected by CA Unicenter NSM with incident management

- Network administrators and service desk staff can easily determine impact

- Avoid request or incident storms

- Auto-update request or incidents

- Implement business rules as a best practices for network and service support management

- Auto-dispatch new request or incident occurrences

- Automate the interaction with CA Unicenter Service Desk to reduce the workload of the customer support staff by eliminating some manual tasks

- Reduce the number of request and incidents open by users when a problem in the enterprise occurs because the request or incident has been created automatically as soon as it occurs

How the Integration Works

CA Unicenter NSM r11.x can be integrated with CA Unicenter Service Desk 6.0 and r11.x using two different methods:

- Using Web Services

- Using a call to the AHD.dll functionality

How to Integrate Using Web Services

CA Unicenter NSM creates CA Unicenter Service Desk requests based on policies defined in Event Management System (EMS), Advanced Event Correlation (AEC), and Alert Management System (AMS), as illustrated in the following diagram:
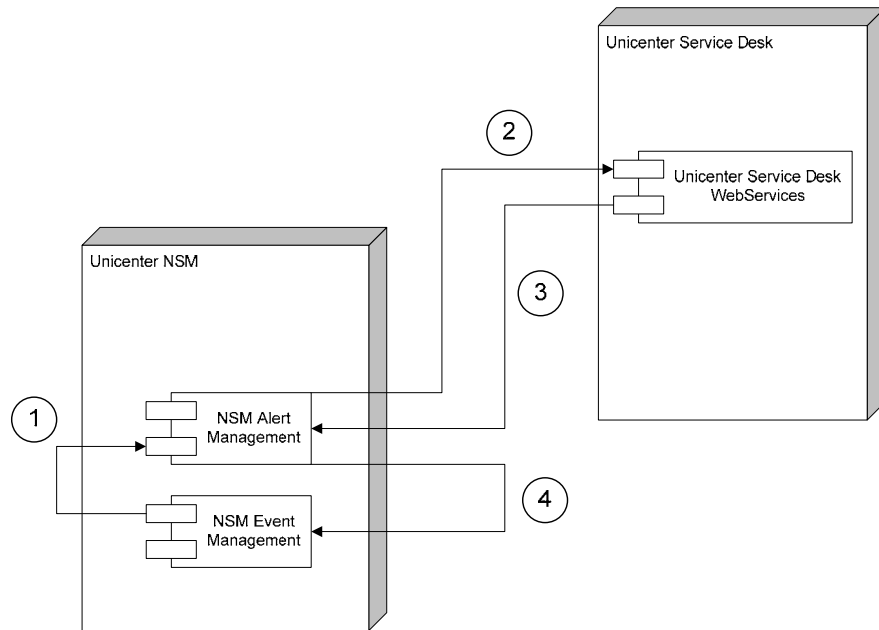
The following information applies to the previous diagram:

1. A situation is evaluated by either Event Management System (EMS) message records and actions, or by Advanced Event Correlation (AEC) rules. If the event has been categorized on CA Unicenter NSM as serious, an alert is generated.

2. Alert Management System (AMS) class or escalation policy determines that a service desk request is appropriate, and creates one.

3. CA Unicenter NSM comes with Event Management System (EMS), Advanced Event Correlation (AEC), and Alert Management System (AMS) policy, and through the use of the Web Services API, can automatically create and close service desk requests.

4. You can also write your own policy using message records and actions, correlation rules, and alert classes and escalation policies, to continue managing the network event in CA Unicenter NSM using Event Management System (EMS) and Alert Management System (AMS).

How the CA Unicenter NSM – Alert Management System Interacts with CA Unicenter Service Desk

The following information explains how the CA Unicenter NSM – Alert Management System interacts with CA Unicenter Service Desk:

■   Alert policy definitions specify that CA Unicenter Service Desk requests be opened and closed during the life cycle of an alert.

■   To open a CA Unicenter Service Desk request when an alert is created, use the Actions tab on the Alert Class window.

Note: AMS does not open a request if an existing request has identical summary, description, and asset properties. This prevents multiple reports for the same root problem.

■   To open a CA Unicenter Service Desk request when an alert is escalated, use the Escalation Policy Editor.

■   To close a CA Unicenter Service Desk request when the alert that opened it is closed or made inactive, use the context menu in the CA Unicenter Management Command Center (MCC). To make an alert inactive, use the Alert Class window or Alert Properties dialog.

■   Alerts that are associated with CA Unicenter Service Desk requests include the request reference number. CA Unicenter Service Desk requests created by alerts indicate that an outside product opened the request.

■   The activity log of a CA Unicenter Service Desk request is updated automatically with additional information from the Alert Management System (AMS) when duplicate alerts are created.

■   The context menu in the CA Unicenter Management Command Center (MCC) lets you interact manually with CA Unicenter Service Desk. You can view and open requests, and search CA Unicenter Service Desk Knowledge Tools. For example, when you right-click an alert, you can see requests associated with that alert. When you right-click a managed object in the 2D Map or Topology view, you can see requests for the selected node.

■   To execute a Knowledge Document search based on the alert text, AMS uses the text in the Message Record Action, overrides tab (field labeled as the desired Alert Text).

   Note: For more information about the Alert Management System and AMS alert policies, see the CA Unicenter NSM online help.

How to Integrate by Calling AHD.dll Functionality as an Action Message in CA Unicenter NSM Event Management

You can integrate by using a call to the AHD.dll functionality, as illustrated in the following diagram:
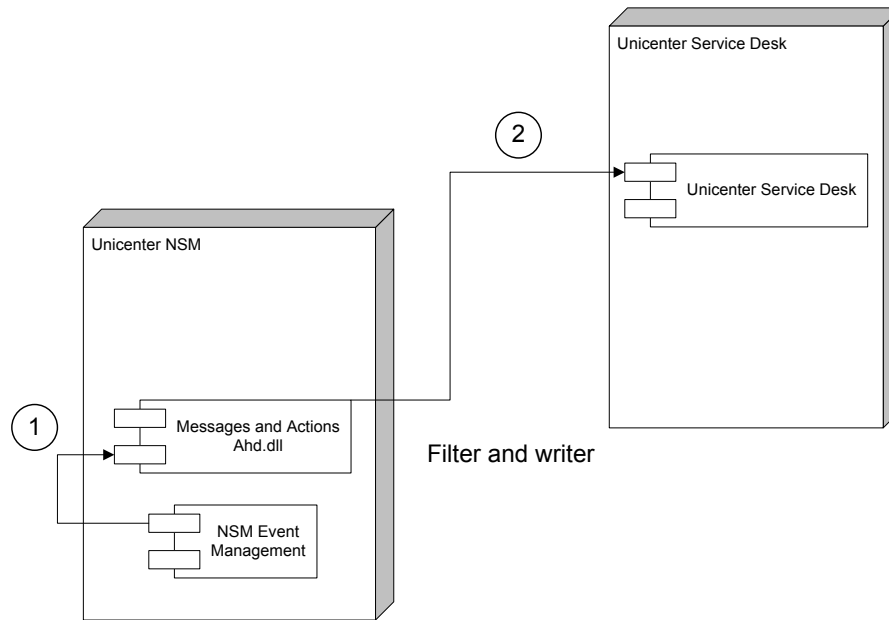
The following information applies to the previous diagram:

1. A situation is evaluated by the Event Management System (EMS). If the event has been categorized in CA Unicenter NSM as serious, a call to the AHD.dll is made.

2. The event is sent to CA Unicenter Service Desk through the AHD.dll. The event is filtered by the CA Unicenter Service Desk Filter to avoid request storms created from unwanted events. The filter functionality selects the appropriate events based on predefined rules, and then sends them to the writer functionality, which utilizes a set of rules and parameters. The user_parms parameter in the writer rule definitions is passed to the Text API. The CA Unicenter Service Desk writer process (tngwriter) defines its own replacement parameters for changing the text before sending it to the Text API, which manages create and update requests for an object (parameter defined in text_api) in CA Unicenter Service Desk.

Note: For information about defining event writer rules, see the *CA Unicenter Service Desk Implementation Guide*. For information about text_API parameters, see the *CA Unicenter Service Desk Administration Guide*.

Example of the CA Unicenter NSM Integration

Business Challenge

Forward, Inc.'s IT infrastructure is composed of thousands of individual elements from the major domains of networks, systems, applications and databases. The number of potential points of failure may easily exceed 100,000. Andrea Myers, CTO of the organization, has requested that Anita Hirsch, VP of IT Services, coordinate all the necessary components to guarantee that when an incident occurs in the enterprise, it is solved in the least time and cost possible, guaranteeing all IT resources operations 24 hours a day, 7 days a week, and keeping up the users' productivity.

CA Approach

The Forward, Inc. organization is facing many challenges, including the need to control costs, the ability to manage growing IT complexity, and the increasing demand to achieve service level objectives to support the business needs. To meet these challenges, Forward, Inc. must ensure the availability and performance of the underlying IT elements that are the foundation for critical business processes. To maximize service availability, Forward, Inc. must rely on intelligence and automation, based on user-defined policy, to resolve incidents in real-time.

In addition, to effectively manage the delivery of service support and meet critical service level agreements (SLAs), IT components must be aligned with and managed as part of the business service. When an incident occurs, it must be resolved in the least time possible. In addition to resolving the incident, information about the incident must be captured for both real-time tracking and historical analysis, and reports must be generated to help IT take proactive actions, therefore guaranteeing IT resources productivity, increasing efficiency, and reducing operating costs. CA advised Forward, Inc. to take advantage of the integration between CA Unicenter Service Desk and CA Unicenter NSM currently in production in the organization.

Configuring a Solution

Anita Hirsch has spoken to Paul Min, Director of the IT Services department. In turn, Paul has spoken with his team to discuss what steps have to happen to guarantee that all IT resources are available 24 hours a day, 7 days a week to support operations and staff productivity. Based on Paul and his team's analysis of the requirements and CA's advice, Forward, Inc. will utilize the CA Unicenter Service Desk and CA Unicenter NSM integration capabilities.

Paul and his team have defined specific incident management policies. For each type of incident, these policies define the seriousness of the problem (severity level), the person or group to which the incident should be assigned, and so forth. These definitions are used to establish automatic escalation policies. By utilizing this functionality to automate incident management, Paul can guarantee that problems do not get lost and the appropriate people are notified about the progress of an incident.

Knowing that Incident Management policies have been defined, Paul's team will utilize the CA Unicenter NSM integration configuration, taking advantage of the integration points available in CA Unicenter NSM r11 through Alert Management System (AMS), along with the Management Command Center (MCC) and the CA Unicenter Management Portal functionality.

Best Practices

Reference the following best practices when using the CA Unicenter NSM integration:

- ■ Establish equipment configuration to identify the system components that are candidates for incident tracking.

- ■ Establish incident policies to identify how incidents will be classified and assigned. For example, incident categories, status codes, and responsibility areas.

- ■ Establish incident escalations policies to define how priorities and responsibilities will change based upon how long a problem has been unresolved.

Configure Alert Management to Work with CA Unicenter Service Desk

George Hun, the IT Administrator at Forward, Inc., has received information about all critical event messages that may impact enterprise operations. Based on this information, George will create alerts for the event messages and configure the CA Unicenter NSM integration so the required alerts will create incidents in CA Unicenter Service Desk when they occur. In this example, George will create an alert named Tier1.

Note: You can configure Alert Management System (AMS) alert classes and escalation policies to automatically create CA Unicenter Service Desk incidents and record the incident number as an attribute within the alert (in the request field). This attribute is then referenced by the Management Command Center for subsequent start, in context, into CA Unicenter Service Desk.

To configure Alert Management, George must successfully complete the following steps:

1. Start and log in to the CA Unicenter NSM server.

2. Begin configuring AMS by first starting the Alert Global Definition interface, either through the classic EM interface or through the MCC.

    > **MCC**. Select Enterprise Management from the left pane. Expand the tree under your server and select Alert Management. Select Alert Global Definitions and click the Launch Alert Global Definitions item in the right pane.

    > **EM Interface**. Open EM Classic and select your server. Click to open AMS, and then click on Alert Global Definitions.

3. Click Alert Global Definitions and complete the information for the URI, User ID and Password as follows:

    > **URI** (Uniform Resource Identifier). Identifies the address of the web server on your CA Unicenter Service Desk primary server.

    For CA Unicenter Service Desk 6.0, the default URI is
    http://server[:port]/usd_ws/usd_ws.asmx

    For CA Unicenter Service Desk r11.x, the default is
    http://servername[:port]/axis/services/USD_WebServiceSoap

    Note 1: The default Tomcat port is 8080, but you should check first that this port is not already used by another application. You can use the following command line to check whether this port is already in use: netstat –a |more

Note 2: If CA Unicenter Service Desk is hosted on a secure web server (indicated by *https* in the URI) you must import the server's SSL certificate to the server on which AMS is hosted. For information, see the *CA Unicenter NSM Inside Event Management and Alert Management Guide*.

4. Add the server as a node by clicking the New button in the Worldview Repositories panel and selecting the user ID nsmadmin.

    >   **User ID**. Identifies the user ID needed to access CA Unicenter Service Desk.

    Note: As a best practice, the ServiceDesk user should be replaced by a different user (for example, System_AHD_generated). This user must exist in CA Unicenter Service Desk and the Access Type must be Analyst.

    >   **Password**. The corresponding password for the previous user.

    Note: For more information about the URI format, see the How the Integration with CA Unicenter Service Desk Works topic in the *CA Unicenter NSM Inside Event Management and Alert Management Guide*.

The following illustrates a sample Alert Global Definition window:

5. Click OK.

6. From Alert Management System (AMS) click Alert Classes and create a new class with the following information in the Main tab.

    Class ID: Tier1

    Class Name: Tier1 Alert

    Description: Tier1 Critical Alert

    Note: For more information, see the *Service Availability Management Green Book* using the Technical Support link at http://ca.com/support.

7. Click the Actions tab and enter the following in the URL field.

http://forwardinc:8092/CAisd/pdmweb.exe

Note 1: In this example we are using port 8092. The default Tomcat port is 8080, but you should check first that this port is not already used by another application. You can use the following command line to check whether this port is already in use: netstat –a |more

Note 2: Forwardinc is the CA Unicenter Service Desk server name for the Forward, Inc. organization.

8. Select the Create request when alert is opened and Synchronize closure of request and alerts check boxes.

9. Click the Messages Policies tab, click New, create a new message record with *TIER1** as the Message id, and save the record.

10. Open the record message you created.

11. On the Message Record window, click either the Actions button (EM interface) or Actions tab (MCC), as illustrated in the following sample windows:

12. Create a new action with the following information:

    Sequence Number: 10

Alert Class id: Tier1

Text: &TEXT

Action: ALERT

Active and Evaluate check boxes: Selected

Attribute: DEFAULT

Color: DEFAULT

13. Click OK.

14. Create another new action with the following information:
    Sequence Number: 20

    Text: &TEXT

    Action: SENDKEEP

    Active and Evaluate check boxes: Selected

    Attribute: DEFAULT

    Color: RED

15. Click OK.

    The Message Record Action window closes.

16. Click OK.

    The Message Record window closes.

17. From the AMS window, click Alert Queues and create a new alert queue with the following information:
    Alert queue name: Tier1

    Description: Tier1 Alerts

18. Click OK and return to the Tier1 queue window recently created.

19. From the queue window, Alert Classes panel, click New, Add a new Alert Class.

20. Select the queue named Tier1 class from the *Initial queue* drop-down list.

21. Select the active check box, click OK to exit from the Alert class window, and click OK to exit the queue window.

Customizing CA Unicenter Service Desk Tickets Created By Alert Management

Key attributes from the alert are automatically passed to CA Unicenter Service Desk when the ticket is created, including the name of the affected object which will be specified to CA Unicenter Service Desk as the affected configuration item (CI). In addition to these attributes, AMS includes support for the following special tags. These tags pass additional

information in the Alert Detail fields to CA Unicenter Service Desk where it can be used to automatically populate the appropriate fields within the ticket.

CA Unicenter Service Desk Tag	Description
SDTemplate	Specifies a CA Unicenter Service Desk template to use for creating the request. Templates provide values for common situations and allow requests to be created quickly. If other CA Unicenter Service Desk tags are entered, their values override the default template properties.
SDAssignee	Specifies the CA Unicenter Service Desk contact to assign to the request. Use the format "lastname, firstname" and include the comma even if a name is blank.
SDGroup	Specifies the CA Unicenter Service Desk group that is responsible for the request. Use the Group Name property.
SDPriority	Indicates the priority of the request. Values range from 1 to 5.
SDImpact	Indicates the impact of the request. Values range from 1 to 5.
SDUrgency	Indicates the urgency of the request. Values range from 1 to 5.
SDConfigItem	Specifies the affected item or resource. If not included, the alert node is used.
SDRequestArea	Specifies the CA Unicenter Service Desk request area or category.
SDRootCause	Specifies the CA Unicenter Service Desk root cause.
SDSummary	Specifies the CA Unicenter Service Desk summary for the request. If not included, the alert text is used.
SDDescription	Specifies the CA Unicenter Service Desk description for the request. If not included, the alert detail is used.
SDTicketType	Specifies the CA Unicenter Service Desk ticket type. Valid values include: I (Incident), P (Problem), or R (Request). If not included, a request is opened.
SDServiceType	Specifies the CA Unicenter Service Desk service type for the request.

For example, as illustrated in the following sample window, the following event text creates a CA Unicenter Service Desk incident for the user June Arnold, with a priority 1:
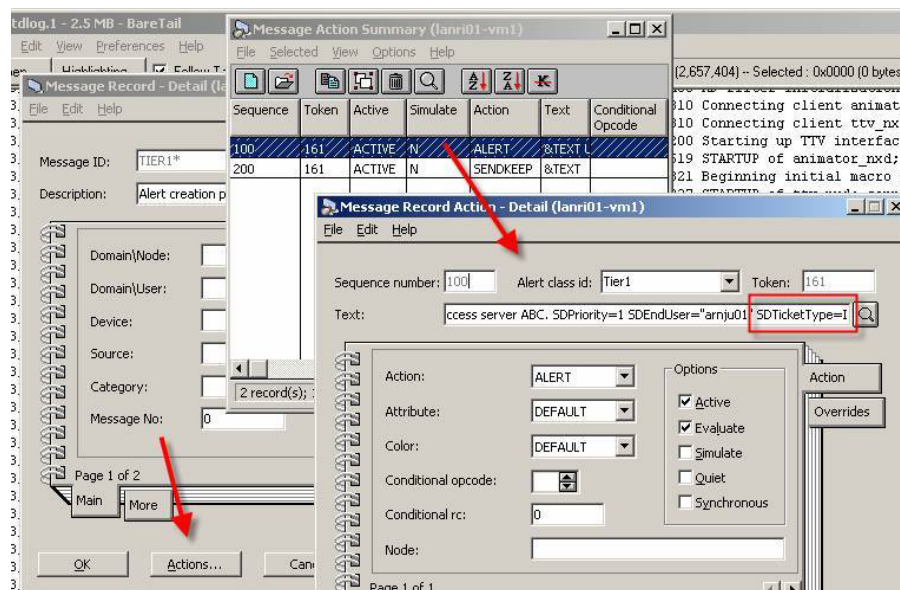
Configure the Management Command Center (MCC) to Integrate with CA Unicenter Service Desk

CA Unicenter Service Desk integration with the Management Command Center (MCC) is configured from the View, Options menu available from within the MCC. This configuration allows CA Unicenter Service Desk to start, in context, functionalities available from the Management Command Center (Create Request, View Alert Request, View Request, and Knowledge Tools).

To configure the MCC, select View, Options from the main MCC console, and select the Connections tab to view or modify the CA Unicenter Service Desk and Knowledge Tools invocation URLs. The typical settings for the CA Unicenter Service Desk URL will be different, based on which CA Unicenter Service Desk release is being used. The following is an example for CA Unicenter Service Desk 6.0 and r11 URLs.

CA Unicenter Service Desk 6.0

CA Unicenter Service Desk

http://Forwardinc/CAisd/pdmweb.exe

Knowledge Tools

http://Forwardinc/causp/KT/KTmain.asp
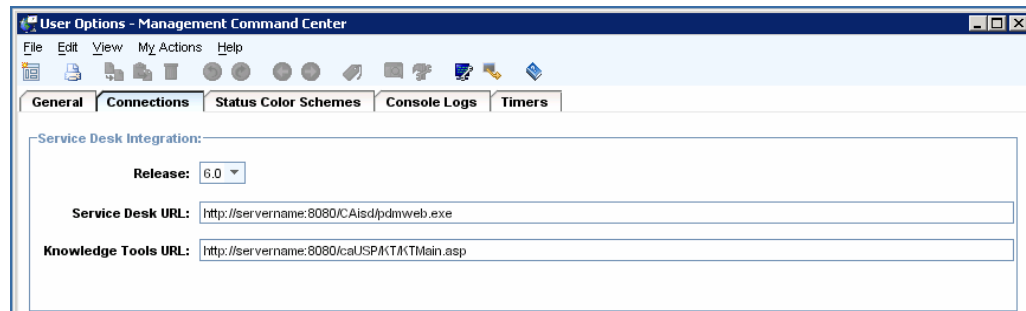
CA Unicenter Service Desk r11

http://Forwardinc:8080/CAisd/pdmweb.exe

Note 1: The default Tomcat port (used in this example) is 8080, but you should check first that this port is not already used by another application. You can use the following command line to check whether this port is already in use: netstat –a |more

Note 2: When running on IIS the default port is 80. There is no need to specify it after the machine name as when the web server runs on Tomcat.
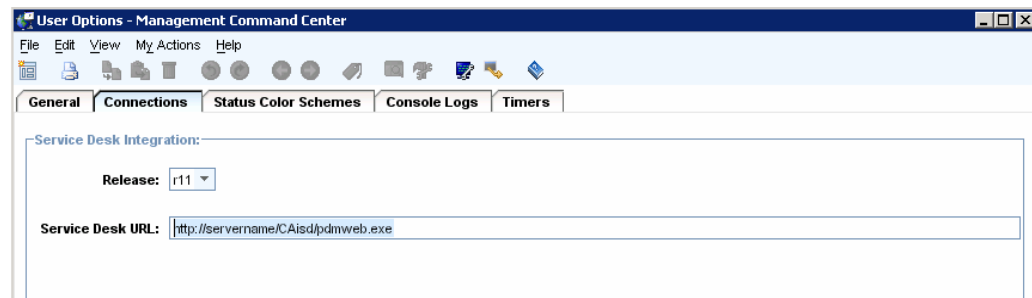
Note 3: The CA Unicenter Service Desk - Knowledge Document search can be executed using the alert text. This is the text in the Message Record Action, overrides tab, in the field labeled Alert Text.

Configure the Integration with CA Unicenter Management Portal

You must complete the following configuration steps in CA Unicenter Service Desk:

1. Log in to CA Unicenter Service Desk as an Administrator.
2. Click the Administrator tab, Options Manager, Security.
3. Click the option manager named Portal Safe List, and click Edit.
4. Click Install.
5. On the Portal Safe List, complete the Option Value field with information about the CA Unicenter Management Portal server name and Tomcat port information (for example, forwardinc:8080).

Note: When running on Tomcat the default port is 8080, but you should check first that this port is not already used by another application. You can use the following command line to check whether this port is already in use: netstat –a |more

6. Click the Safe button.

7. Recycle CA Unicenter Service Desk Services.

You must complete the following configuration steps in CA Unicenter Management Portal (UMP):

1. Log in to CA Unicenter Portal as an Administrator.

2. Click the Administrator workplace tab.

3. Select Task 1 to configure the management portal.

4. Click the connection… link that relates to the CA Unicenter Service Desk Management component.

    A window opens that lists the currently configured CA Unicenter Service Desk servers.

5. Click New to add a new CA Unicenter Service Desk server.

6. Select the appropriate CA Unicenter Service Desk version that you are configuring:

    CA Unicenter Service Desk r11.x Integration

    a. Enter the CA Unicenter Service Desk server name.

    b. Enter the Tomcat port required to connect to the CA Unicenter Service Desk server (that is, the default 8080).

    c. Enter the full CA Unicenter Service Desk URL. That is, *http://forwardinc:8080/CAisd/pdmweb.exe*

    d. If the CA Unicenter Service Desk server has been configured to use SSL, make sure the SSL check box is selected, as illustrated in the following sample window:

    CA Unicenter Service Desk V6.0 Integration

    a. Enter the CA Unicenter Service Desk server name.

    b. Enter the CA Unicenter Service Desk Knowledge base URL. That is, *http://Knowledgetool server/causp/kt/KTMain.asp*

    c. Select the CA Unicenter Service Desk Server Operating system (Windows / Linux).

> *d.* Enter the full CA Unicenter Service Desk URL. That is,
> *http://Servicedesk server/CAisd/pdmweb.exe*
>
> e. If the CA Unicenter Service Desk server has been configured to use SSL, make sure the SSL check box is selected, as illustrated in the following sample window:

7. If this is the first CA Unicenter Service Desk server being configured, select the Set Default Data Source check box.

   If the configuration is valid, the Manage Component window will contain a Green check box, as illustrated in the following sample window:

8. To publish the CA Unicenter Service Desk portlet, click the UMP Administration workplace tab and select Task 2, Create or Modify UMP Portlets, as illustrated in the following sample window:

9. To create a new CA Unicenter Service Desk portlet, click the New icon associated with the CA Unicenter Service Desk Portlets components.

10. Select the CA Unicenter Service Desk server previously defined from the drop-down list. For example, forwardinc.

11. To publish this server, click Publish.

    A confirmation window opens, indicating where the newly published content can be found within the knowledge tree (Library, Enterprise Management, ServiceDesk, Server (forwardInc).

12. Click Open/Update/Close CA Unicenter Service Desk Request.
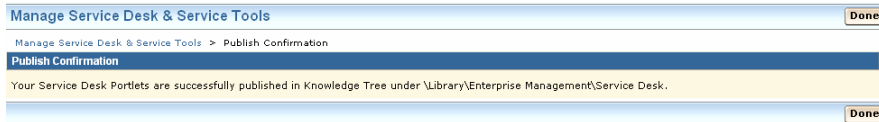
The following is a sample window:
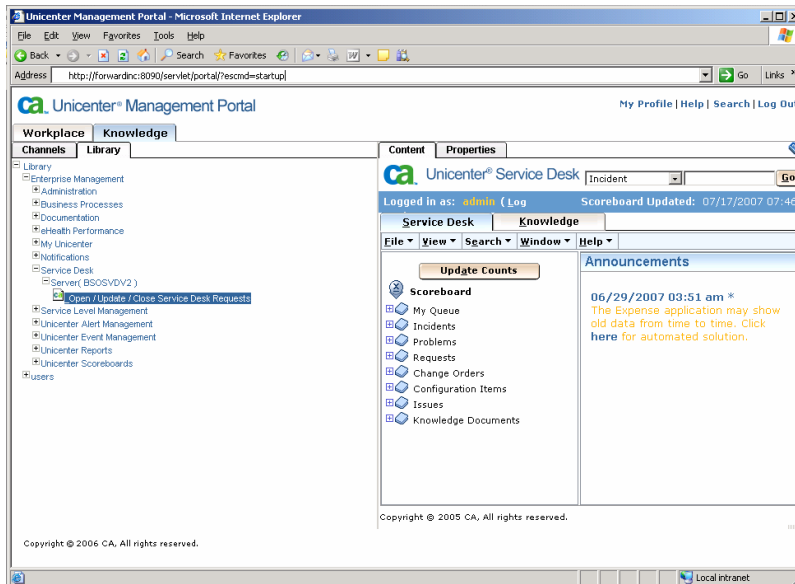
Note: For this integration to work, the user must be defined within the CA Unicenter Management Portal and must be a valid user in CA Unicenter Service Desk.

Configure the CA Unicenter Service Desk Integration with the Event Console Messages and Actions Using AHD.DLL and Filter and Writer

Note: This is a typical CA Unicenter Service Desk and CA Unicenter NSM integration. These integration steps are valid for the following release combinations:

■ CA Unicenter Service Desk 6.0 and CA Unicenter NSM 3.1

■ CA Unicenter Service Desk 6.0 and CA Unicenter NSM r11.x

■ CA Unicenter Service Desk r11.x and CA Unicenter NSM 3.1

■ CA Unicenter Service Desk r11.x and CA Unicenter NSM r11.x

You must complete the following configuration steps in CA Unicenter Service Desk:

1. Verify that 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. Verify that the CA Unicenter NSM Integration option is installed.

   You can select the CA Unicenter NSM Integration option during the initial installation or later by modifying an existing installation.

   Note: When the CA Unicenter Service Desk server and CA Unicenter NSM are installed on the same computer, you must install the CA Unicenter NSM Integration option. You can install this option on either a UNIX or Windows server. You can select the CA Unicenter NSM Integration option during the initial installation or later by modifying an existing installation.

   (**Windows**) When the CA Unicenter Service Desk server and CA Unicenter NSM are installed on different computers, you must install the CA Unicenter NSM Integration option on the CA Unicenter Service Desk primary server. You must also install and configure a CA Unicenter Service Desk client or secondary server on the CA Unicenter NSM server, or on a CA Unicenter NSM administrative client.

   (**UNIX**) When the CA Unicenter Service Desk server and CA Unicenter NSM are installed on different computers, you must install the CA Unicenter NSM Integration option on both the CA Unicenter Service Desk primary server and the CA Unicenter NSM server. After installing the CA Unicenter NSM Integration option on the CA Unicenter NSM server, you must run pdm_configure to configure CA Unicenter Service Desk as a secondary server on that installation.

3. Use the following steps to configure the converter using the pdm_edit utility:

   a. On the CA Unicenter Service Desk primary server installation, browse to the $NX_ROOT/samples/pdmconf directory.

   b. Enter the following command to start the pdm_edit utility.

      pdm_perl pdm_edit.pl

   c. Answer the prompts based on your requirements, and when you get to the main menu, enter N to select the Edit UNI Converters (UNIX_ONLY) option.

   d. Select A to add, then enter the name or IP address of the CA Unicenter NSM server computer (configured as the CA Unicenter Service Desk secondary server) when prompted for a host name.

   e. The script prompts you for an IP address. Enter the IP address of the CA Unicenter Service Desk primary server computer.

   f. Press Enter to return to the main menu.

   g. Select X to save and exit.

      A file named pdm_startup.rmt is created that stores your new configuration values.

h.   Create a backup of the pdm_startup.tpl file in the $NX_ROOT/pdmconf directory of the CA Unicenter Service Desk primary server installation.

i.   Replace the backup with the newly created pdm_startup.rmt file.

j.   Run pdm_configure on the CA Unicenter Service Desk primary server without making any changes. The new configuration settings take effect the next time you start the CA Unicenter Service Desk server.

    Important! Do not reinitialize your database when you reconfigure.

k.   As the privileged user, run pdm_proctor_init on the CA Unicenter NSM server to start the CA Unicenter Service Desk proctor.

l.   As the privileged user, restart the CA Unicenter Service Desk services to start the CA Unicenter Service Desk daemons. Enter pdm_status to display the status of the daemons.

Consider the following information as you make your choices during the integration process:

>   Before the process of "receiving a CA Unicenter NSM event and creating a request in CA Unicenter Service Desk" 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 Unicenter NSM events when CA Unicenter Service Desk is down (meaning it has been paused instead of stopped from the Microsoft Windows Services Panel). When CA Unicenter Service Desk is restarted, it processes the queued events.

>   The CA Unicenter NSM event converter service queues events up to a maximum specified by the NX_TNGCNV_QUEUE_SIZE environment variable.

>   When the CA Unicenter NSM repository is rebuilt after integrating with CA Unicenter Service Desk, CA Unicenter Service Desk menu entries are not saved. To restore them, you must rerun the integration on the CA Unicenter NSM Windows computer. To do this, run integAHD.exe, located in the installation-directory\bin directory.

>   If the CA Unicenter NSM event converter starts during CA Unicenter NSM events generation, events that occur before the event converter is fully initialized are not saved.

4.   Make the necessary entries in the topology file, topology.cfg, and define filter and writer rules to filter unwanted events.

5.   Modify the filter and writer rules to fit into Forward, Inc. business processes.

Note: For information about Filter and Writer, see the *CA Unicenter Service Desk Implementation Guide*.

Configure CA Unicenter NSM

To utilize CA Unicenter NSM Event Management to create or update requests or incidents in CA Unicenter Service Desk, CA Unicenter NSM must be configured to monitor console messages by using AHD.DLL.

To successfully configure AHD.DLL, complete the following steps:

1. From the CA Unicenter NSM Enterprise Management, select Event, Messages, and create a new message with the following information:

    Message ID: NSM

    Description: OOTB Classic CA Unicenter Service Desk integration

2. On the same message record window, click the action button and create a new action with the following information:

    Sequence Number: 10

    Text: ahd.dll AHD_Call

    Action: External

    Attribute: Default

    Color: Default

3. Browse to the event console and enter the command : opreload

    Note: You must have a parameters definition, as follows, in CA Unicenter Service Desk in the tngwriter_rule.dat file.

    NSM*:::.*:::*:::tng:::*:::CR_CREATE:::::::::NONE::::::

Testing the Integration

Once George has configured the settings for the integration, he needs to test whether the integration is successfully working. If the integration has been correctly configured, the following should work:

■   Automatic Request Creation

■   Interactive Options in the MCC (Create Request, View Alert Request, View Request, and Searching in Knowledge Tools)

■   User Interactive actions from 2D Map

Testing Automatic Request Creation

Based on the alert created by George in the previous example (Tier1), when a message is sent to CA Unicenter NSM Event and the event's message contains the word TIER1 at the beginning (this is the way George configured the alert Tier1*), a new incident is created in CA Unicenter Service Desk, as illustrated in the following sample window:

In addition, in the MCC - Alert Management, you can see the queue of alerts and the alerts that have generated incidents associated with the alert class Tier1. The numbers are displayed in the appropriate columns, as illustrated in the following sample window:

User Interactive Actions from the Management Command Center (MCC)

Complete the following steps to access the CA Unicenter Service Desk Integration start points from the Management Command Center:

1. Select either Topology or Business Process Views from the left hand pane menu drop-down list, as illustrated in the following sample window:
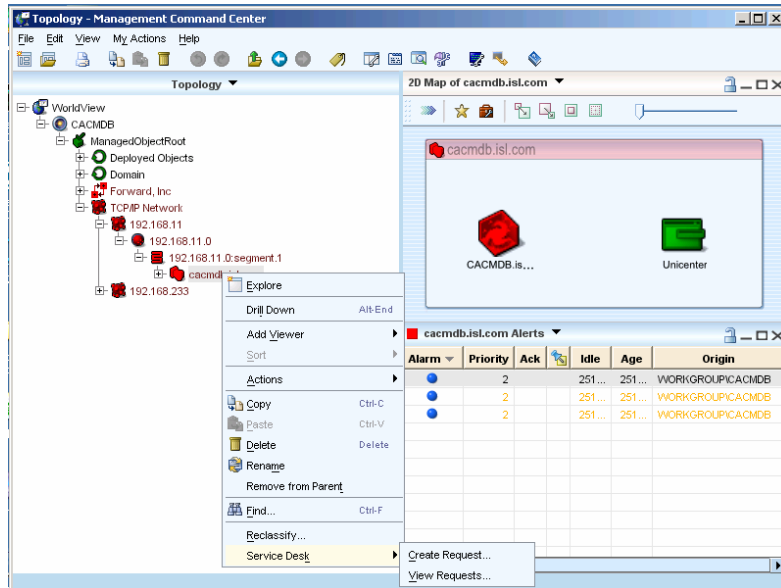
2. Right-click any Host object in the left-hand tree and select Service Desk from the menu. The following two submenu options are available:

> **Create Request:** Select to create a new CA Unicenter Service Desk request using the selected object as the Configuration Item (CI) that is affected and which should be the subject of the issue being created.

> **View Requests**: Select to view all requests created within CA Unicenter Service Desk for the selected object (all requests related to this CI).

Note: A Service Desk menu is also available within the Alerts view of the MCC. To access the menu and its submenus, right-click the alert in the right-hand pane of the MCC and select Service Desk, as illustrated in the following sample window:
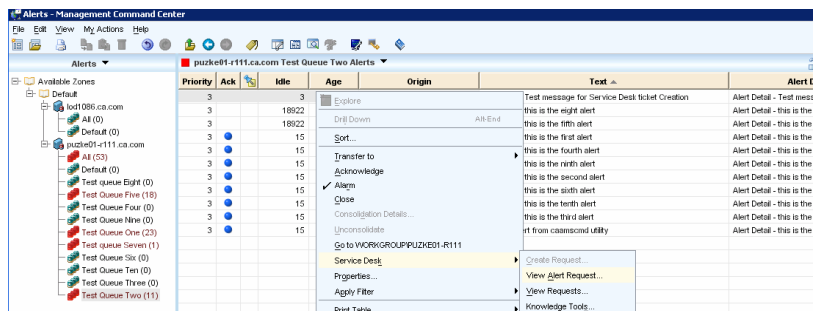
The following sub-menu options are available:

> **Create Requests** - This option is only available if a CA Unicenter Service Desk ticket has not already been created by the Alert Management System (AMS). If selected, a CA Unicenter Service Desk ticket will be created using information from the currently selected alert. The CA Unicenter Service Desk summary field will be populated from the alert text field and the CA Unicenter Service Desk description field will be populated by the alert detail field.

> **View Alert Request** - This option is only available if a CA Unicenter Service Desk ticket has already been created by AMS. AMS alert classes and escalation policies can be configured to automatically create a CA Unicenter Service Desk ticket. That ticket number is stored as an attribute within the alert (request), as discussed in the previous section. If a ticket has been created, this option will automatically start the CA Unicenter Service Desk user interface directly in context to the relevant ticket.

> **View Request** – Select this option to view all CA Unicenter Service Desk requests for the node where the alert originated.

> **Knowledge Tools** – Select this option to start a search of the CA Unicenter Service Desk Knowledge Tools in context, using the Alert Text as a search argument. The result will be a display of all Knowledge Base articles that are potentially related to the alert, as illustrated in the following sample windows:
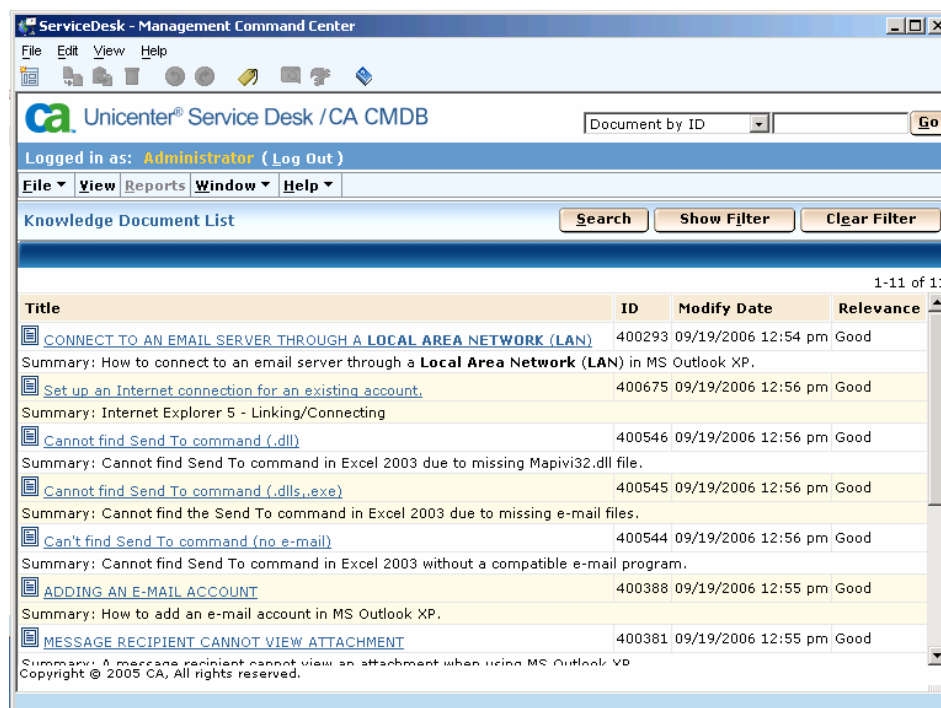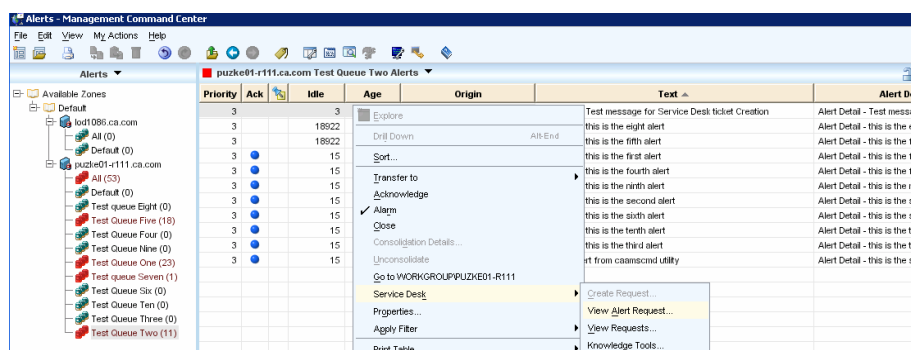
User Interactive Actions from the 2D Map

Right-click any Host object on the 2D Map to display a menu with the following options for the selected object:

■   **Create Incident.** Select this option to create an incident using the selected object as the Configuration Item (CI) that is affected and which should be the subject of the incident being created.

■   **Incident List.** Select this option to view all incidents created within CA Unicenter Service Desk for the selected object (all incidents related to this CI).
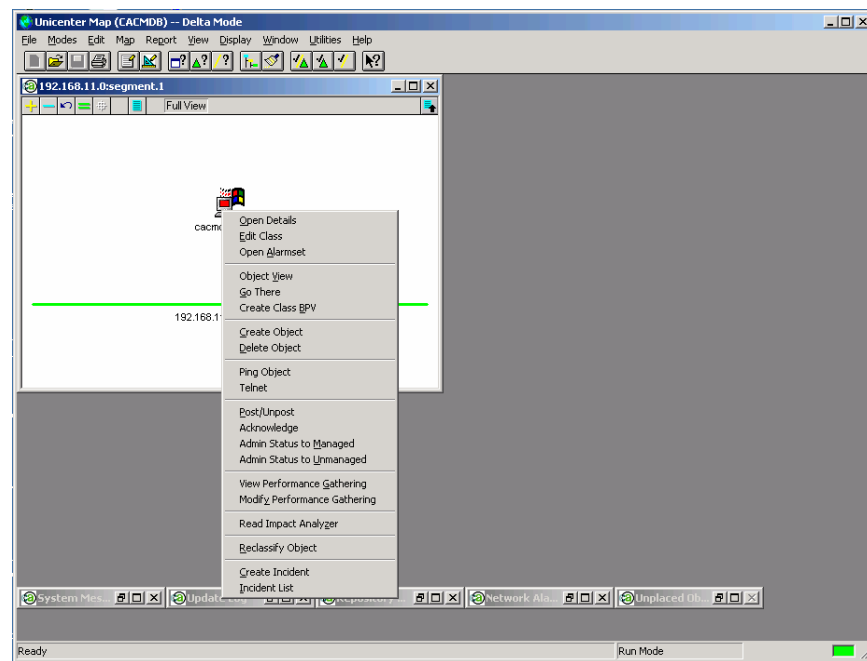
Test Automatic Request Creation from the Event Console

To test the automatic request creation, start the Event Console, enter the following command, and press Enter.

```
cawto NSM Mission Critical Server is down
```

A new CA Unicenter Service Desk request is created, as illustrated in the following sample window:
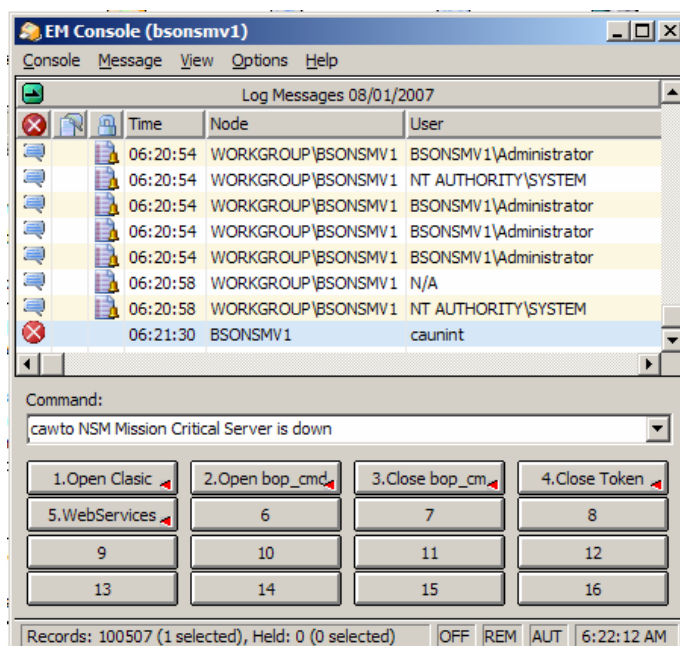
Integration Summary

CA Unicenter NSM creates CA Unicenter Service Desk incidents based on policies defined in the Event Management System (EMS), Advanced Event Correlation (AEC), and Alert Management System (AMS). The following is a summary of events:

- A situation is evaluated by either EMS message records and actions, or by AEC rules. If the event is serious, an alert is generated.

- AMS class or escalation policy determines that a CA Unicenter Service Desk request is appropriate, and creates a request.

  Note: CA Unicenter NSM is delivered with EMS, AEC, and AMS policy that can automatically create and close CA Unicenter Service Desk requests. You can also write your own policy using message records and actions, correlation rules, and alert classes and escalation policies.

CA Unicenter NSM interacts with CA Unicenter Service Desk in the following way:

Alert policy definitions specify that CA Unicenter Service Desk incidents be opened and closed during the life cycle of an alert:

- Open a CA Unicenter Service Desk request when an alert is created. This is specified by using the Alert Class Window.

Note: AMS does *not* open an incident if an existing incident has identical summary, description, and asset properties. This prevents multiple trouble tickets being opened that describe the same root problem.

■   Open a CA Unicenter Service Desk request when an alert is escalated. Use the Escalation Policy Editor.

■   Close a request when the alert that opened it is closed or made inactive. Use the context menu in the CA Unicenter MCC to close an alert. Use the Alert Class window Main page or Alert Properties dialog Status page to make an alert inactive.

Alerts that are associated with CA Unicenter Service Desk requests include the request reference number. Likewise, CA Unicenter Service Desk requests created by alerts indicate that an outside product opened the request.

The activity log of a CA Unicenter Service Desk request is updated automatically with additional information from AMS when duplicate alerts are created.

The context menu in the CA Unicenter MCC lets you interact manually with the CA Unicenter Service Desk. You can view requests, open a request, and search the CA Unicenter Service Desk Knowledge Tools. For example, when you right-click an alert, you can see requests associated with the alert. When you right-click a managed object in the 2D Map or Topology view, you can see requests for the selected node.

Scenarios

This section contains examples of situations that may trigger the creation of an alert and a CA Unicenter Service Desk Incident.

Scenario 1: System Agent on a Critical Server

■   An agent metric exceeds a threshold.

■   An Event Management System (EMS) event is generated.

■   EMS message record policy creates an alert for the event.

■   AMS escalation policy opens a CA Unicenter Service Desk request, because the alert is open more than 30 minutes.

■   A technician resolves the issue and closes the alert, and the CA Unicenter Service Desk incident is closed automatically.

Scenario 2: Third-party Product

■   Third-party products produce a series of events in the system log indicating a failure.

- EMS captures the events.

- AEC policy evaluates the events and creates an alert.

- AMS class policy opens a CA Unicenter Service Desk request immediately.

- Operations staff resolves the problem and closes the alert. The Incident is closed automatically.

Note: For additional NSM-centric information about the CA Unicenter Service Desk integration with CA Unicenter NSM, see the *Service Availability Management Green Book* using the Technical Support link at http://ca.com/support.

Chapter 5: Desktop Management – Integrating with CA DMS

What is CA DMS?

CA Desktop Management Suite (CA DMS) is an IT resource management solution that includes functionality from CA Unicenter Asset Management, CA Unicenter Software Delivery, CA Unicenter Patch Management, and CA Unicenter Remote Control.

Asset Management Functionality

The asset management functionality available within CA DMS provides a powerful solution for proactively managing IT assets in a business environment. It provides full-featured asset tracking capabilities including hardware, software, and network inventory, configuration management, and software usage monitoring.

Some key asset management features include the following:

- **Hardware Inventory**. Provides detailed inventory information such as serial numbers, CPU model and speed, amount of system RAM, available disc space, and more.

- **Software Inventory**. Detects application software installed on inventoried host systems. CA provides over 15,000 predefined software definitions and periodic online updates.

- **Software Usage Monitoring**. Lets you have complete control of the software running in the enterprise and align your licenses to your needs.

- **Built-in Script Language**. Enables you to create and run jobs on the agent systems.

- **Policies**. Lets you set alarms or automate the management of your IT environment.

- **Reports**. Predefined and customized reports deliver all the information you need about your enterprise.

Software Delivery Functionality

The software delivery features available within CA DMS provide a flexible and comprehensive mechanism for building, distributing, installing, and managing software on target systems across the enterprise. Policy-based automation makes it simple to maintain the state of software on all the computers within the enterprise.

Some key software delivery features include the following:

- An agent plug-in resides on all target computers and is responsible for installing, updating, and removing software packages in response to orders from the manager.

- The ability to remotely install, update, and remove software packages on-demand or scheduled, without having to grant system administrative privileges to users.

- A powerful data transport infrastructure optimizes network use.

- The ability to deploy existing software packages in many packaging formats.

- A powerful scripting language to perform tasks on target computers.

- A software packaging tool (Packager) can be used to create software packages for deployment. The Packager captures all files and configuration changes that occur after a product has been installed. This information is collected and automatically transformed into a software package that is ready to be included in the software package library.

- Operating system installation management features allow for the remote deployment of operating systems without user input (*unattended* mode).

- The ability to track software installation, and identify who installed the software, when and where it was installed, and how it was installed.

Remote Control Functionality

Remote control functions are used in many situations, including remote support and training. The remote control feature lets you concurrently access multiple computers with different operating systems, transfer files, and chat. But more importantly, you can control and manage this functionality from a central location.

Some key remote control features include the following:

- **Viewer**. The Viewer is the user interface used to connect to remote systems. The Viewer is fully integrated with the CA DMS administration interface (CA DMS Explorer), and can also be installed and run separately.

- **Host**. The host function runs on the computer that you want to control. The host is a CA DMS agent plug-in. The Viewer connects to the host and establishes a session. The host responds to keyboard and mouse input sent from the viewer and, in return, sends a video image of its desktop to the Viewer.

- **Replayer**. The Replayer lets you play back previously recorded host sessions and manage recently recorded or replayed sessions.

■ **Central Management**. Central management is accomplished using the CA DMS Explorer including access permissions, global address books management, active session monitoring, and session audits.

The CA DMS Integration

The integration of CA DMS with CA Unicenter Service Desk makes CA DMS a service-aware solution. This means that CA DMS, based on certain events of its managed assets, can trigger an event to create tickets in CA Unicenter Service Desk using web services technology. For this integration, ticket creation in CA Unicenter Service Desk is controlled by the Service Aware policy, which enumerates a list of problem types. CA DMS uses problem types to categorize the problem and address the kind of ticket to be created.

The CA DMS integration provides a graphical user interface that allows context-sensitive launches of each solution in terms of the integration; that interface is named the Common Asset Viewer.

Note: The configuration considerations in this section apply to CA DMS domain and enterprise managers. Tickets can be incidents, requests, problems, change orders, and issues.

Integration Points and Functionality

The CA DMS integration allows CA DMS to automatically open CA Unicenter Service Desk tickets when a CA DMS asset management policy is being violated, and when a Software Delivery job fails. In addition, the integration allows a user to start the CA Unicenter Service Desk Ticket Detail window in context to create an incident manually associated to a particular Configuration Item (CI). The CA Unicenter Service Desk web interface can be started from the CA DMS Explorer on the Quick Launch window. Finally, the Asset Management discovered assets information is stored in a common asset data model in the MDB for use by multiple CA products, which is used by CA Unicenter Service Desk.

Integration Points from CA DMS

The following are CA Unicenter Service Desk integration points from CA DMS:

■ Automatic ticket creation from CA DMS. Asset Management for query or event policies violation.

■ Automatic ticket creation from CA DMS. Software delivery for job failure.

■ Interactive start, in context, menu option for ticket creation associated with a discovered Configuration Item (CI) from CA DMS.

■ In-context start of CA Unicenter Service Desk from the CA DMS Explorer interface.

- The MDB's support of common asset properties strengthens the identification of assets between CA Unicenter Asset Management and CA Unicenter Service Desk.

- Common Asset Viewer, provided with CA Unicenter Service Desk and CA Unicenter Asset Management, displays an integrated view of an asset. It also enables navigation between asset handling applications.

Integration Points from CA Unicenter Service Desk

The following are CA Unicenter Service Desk integration points that allow CA Unicenter Service Desk analysts to do the following:

- Interactively view both the managed and discovered asset spaces, using the Asset Viewer or Common Asset View option available in the CA Unicenter Service Desk Configuration Item Detail window.

- Analyst can also selectively choose discovered assets for inclusion into the CA Unicenter Service Desk managed space, from the CI Search List window.

Integration Value

The CA DMS integration provides the following value:

- Instant Configuration Item (hardware and software) control at all levels in the infrastructure.

- Control of automatic incident creation through flexible event policies, problem category, and priority setting definitions.

- Simplified impact assessment when policy violation occurs.

- No extra maintenance required.

- Improved user support.

- Improved infrastructure management and increased efficiency.

- Flexible role and process management.

- Enhanced incident and problem solving at the organization level.

- Visual interaction with the user to monitor user activity and help improve how users perform certain tasks.

■ Proactively anticipate incident creation and advise the CA Unicenter Service Desk by knowing what policy or application changes are being made using Software Delivery reporting.

How the Integration Works

The following diagram illustrates how the CA DMS integration works:
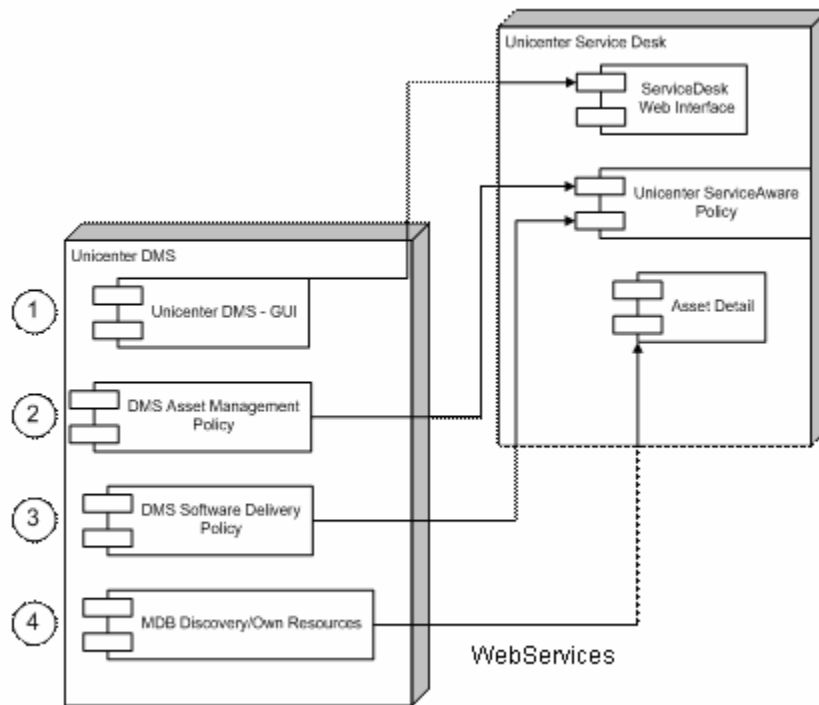
The following information applies to the previous diagram:

1. The integration allows a user to start the CA Unicenter Service Desk Ticket Detail window in context to create an incident manually associated to a particular Configuration Item (CI). In addition, the CA Unicenter Service Desk web interface can be started from the CA DMS Explorer, Quick Launch window.

2. The integration allows CA DMS to automatically open CA Unicenter Service Desk tickets when a CA DMS asset management policy is being violated.

3. The integration allows CA DMS to automatically open CA Unicenter Service Desk tickets when a Software Delivery job fails.

4. CA DMS discovered asset information is stored in a common asset data model in the MDB for use by multiple CA solutions, including CA Unicenter Service Desk, by certifying the Asset and Configuration Item (CI) from CA DMS into CA Unicenter Service Desk from Asset Search List, Discovered Asset functionality.

Example of the CA DMS Integration

Business Challenge

Anita Hirsch, VP of IT services at Forward, Inc., has requested that Don Hailey, the IT manager, coordinate all activities necessary to upgrade all computers in the enterprise that currently have less than 1 GB of RAM. Anita makes this request after analyzing some of the information reported by the CA Unicenter Service Desk dashboard with Paul Kim, the IT Director. Both have seen a long list of incidents related to slow response time from the recently-deployed accounting software. Paul has said that most users run Microsoft Word and Excel simultaneously with the accounting software. The recommended solution from the help desk has been to close some of the open applications, but this recommendation is having a negative impact on productivity in some of the busiest Forward, Inc. branches. In fact, some customers have already complained about the poor services when requiring quotes or paying bills.

CA Approach

Forward, Inc. IT is responsible for managing complex desktop and server environments during this time of rapid change, creating an enormous management burden. The result is an inconsistent desktop environment that is difficult to maintain and does not align with business goals. CA advised Forward, Inc. to implement CA DMS to have a comprehensive and accurate hardware inventory, so they are able to make informed decisions based on accurate information for the resources available for the day-to-day operation. Knowing that Forward, Inc. is currently running CA Unicenter Service Desk r11 in production, CA also advised them to implement the CA DMS integration as part of the CA DMS implementation.

Configuring a Solution

Susan Jobin, the IT Asset Manager, has done a good job working as a team with other departments to successfully implement the CA DMS-Asset Management integration in Forward, Inc. After the implementation, the IT Asset Management department can accurately provide other IT departments with information about available hardware resources. Susan plans to start the second phase of the project, which is to implement the integration between CA Unicenter Service Desk and the CA DMS-Asset Management component. The goal of the integration is to automatically create incidents in CA Unicenter Service Desk when one of the software resources does not fit into the standards established by IT management at Forward, Inc.

Configure the Integration from CA Unicenter Service Desk r11

Complete the following steps in CA Unicenter Service Desk to configure the integration:

1. Log in to CA Unicenter Service Desk as an Administrator.
2. On the Service Desk tab, select Search, Contacts.
   The Contact Search window opens.

3. In the Last Name field, type *System_MA_User* and click search.

The Last Name System_MA_User appears in the last name column.

4. Make sure that *Analyst* appears in the Contact Type column, and *Administrator* appears in the Access Type column.

5. Browse to the CA Unicenter Service Desk main window and click the Administrator tab.

6. Review the information on the left and expand the option named Web Services Policy.

7. Select the policies of the Web Services Policy node.

8. On the right, click the Managed Asset Events symbol.

   The Managed Asset Events window opens.

9. Click Edit.

10. On the edited window, click the Problem Types tab.

    Note: There are 11 different problem types that can be associated with the Managed Asset Events policy.

11. Click Asset Event-based Policy high.

    The Asset event-based policy window opens.

12. Click Edit.

13. Select the default check box.

14. In the Ticket template type drop-down list, choose Incident.

15. In the Ticket template name search field, select Asset Event ITIL Policy High.

    Note: Information in the templates can be added and modified, and additional templates can be created.

16. Click the Duplicate handling tab and select Add Activity Log (do not create ticket).

17. Click Save.

Configure the Integration from CA DMS r11

Complete the following steps in CA DMS to configure the integration:

1. Click the DMS Explorer shortcut on the desktop of the CA DMS Server.

   The left side of the window displays a domain named FORWARDINC.

2. Click the plus symbol to expand the domain folders.

3. Expand Control Panel to Configuration, Configuration Policy.

4. Right-click Default Computer Policy and click Un-Seal.

5. Expand the Configuration Policy and Service Desk Integration options and click Default.

6. Verify that *Default Options* (right side of the window) has the following information:

   Enable: True

   Identifier Field: summary

   Logon Password: servicedesk

   Logon Service Aware Policy: MANAGED_ASSET_EVENTS

Logon User Name: System_MA_User

ServiceDeskendpoint: *http://forwardinc:8080/axis/services/USD_r11_WebService*

Note: When running on Tomcat the default port is 8080, but you should check first that this port is not already used by another application. You can use the following command line to check whether this port is already in use: netstat –a |more

Throttling: False

Timeout: 120

Type of Logon to ServiceDesk: notmanaged

Note: notmanaged uses the local windows account for the user System_MA_User.

7. (Optional) If you use *notmanaged*, create on Windows the user System_MA_User with the following definitions:
    User Name: System_MA_User

    Password: servicedesk

    User cannot change the password: Select this check box.

    Password never expires: Select this check box.

8. (Optional) If you use *managed*, complete the following steps:
    Note: managed is controlled by PKCS#12 certificate.

    a. Start CA Unicenter Service Desk services.

    b. Using a command prompt window, run pdm_pki -pMANAGED_ASSET_EVENTS.

        This command creates the MANAGED_ASSET_EVENTS.p12 file in the current directory. In this policy, the password is equal to the policy name (in this case, MANAGED_ASSET_EVENTS). This command will also add the certificate's public key to the pub_key field (public_key attribute) in the sapolicy table/object.

    c. Start the CA Unicenter Service Desk web interface and choose Administration, Web Services Policies, Policies, Managed Asset Events.

    d. In the policy MANAGED_ASSET_EVENTS insert the Proxy Contact (in this case, ServiceDesk) and confirm that the MANAGED_ASSET_EVENTS policy record has Key = YES.

    e. In CA DMS, enter the following information:

        Enable: True

        Identifier Field: summary

        Logon Password: password for the logon for CA Unicenter Service Desk web services

Logon Service Aware Policy: MANAGED_ASSET_EVENTS

Logon User Name: ServiceDesk

Service Desk endpoint:
http://forwardinc:8080/axis/services/USD_r11_WebService

Note: When running on Tomcat the default port is 8080, but you should check first that this port is not already used by another application. You can use the following command line to check whether this port is already in use:
netstat –a |more

Throttling: False

Timeout: 120

Type of Logon to Service Desk: managed

f. Copy the MANAGED_ASSET_EVENTS.p12 file to the <CA Unicenter Service Desk install>\bopcfg\www\CATALINA_BASE\webapps\axis directory.

g. On the CA DMS computer, use a command prompt window to execute the following command:

   cacertutil import -i:MANAGED_ASSET_EVENTS.p12 -
   ip:MANAGED_ASSET_EVENTS -t:MANAGED_ASSET_EVENTS

h. Return to the Configuration Policy option and click Seal.

i. Close and open the CA DMS interface.

j. Verify that the CA Unicenter Service Desk – Web services functionality is running by entering the following URL in the Internet Explorer browser:

   http://forwardinc:8080/axis/services/USD_r11_WebService

   Note: When running on Tomcat the default port is 8080, but you should check first that this port is not already used by another application. You can use the following command line to check whether this port is already in use:
   netstat –a |more

   After entering this URL, a window similar to the following sample window should open:
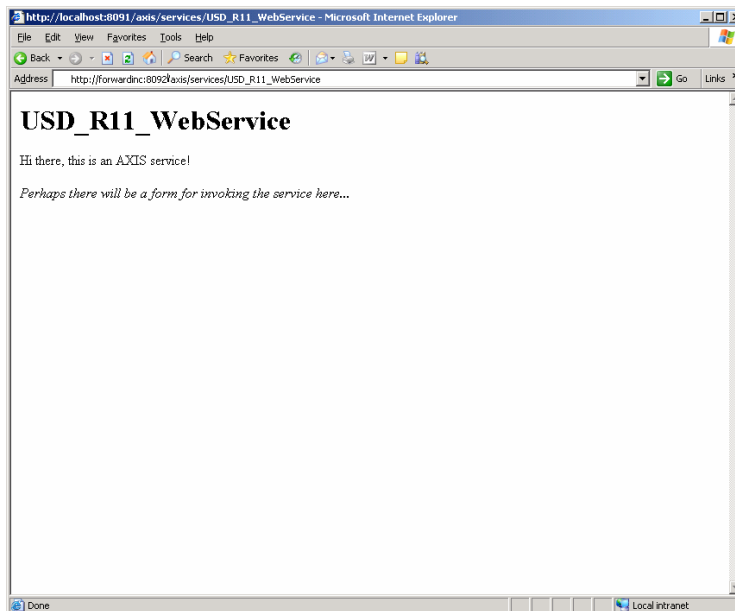
Test the Integration using Interactive Mode

Complete the following steps to test the integration using Interactive mode.

1. Right-click a computer from the computers list in the DMS Domain.
2. From the context menu, select the Create Service Desk ticket option.

   A window similar to the following sample should open:
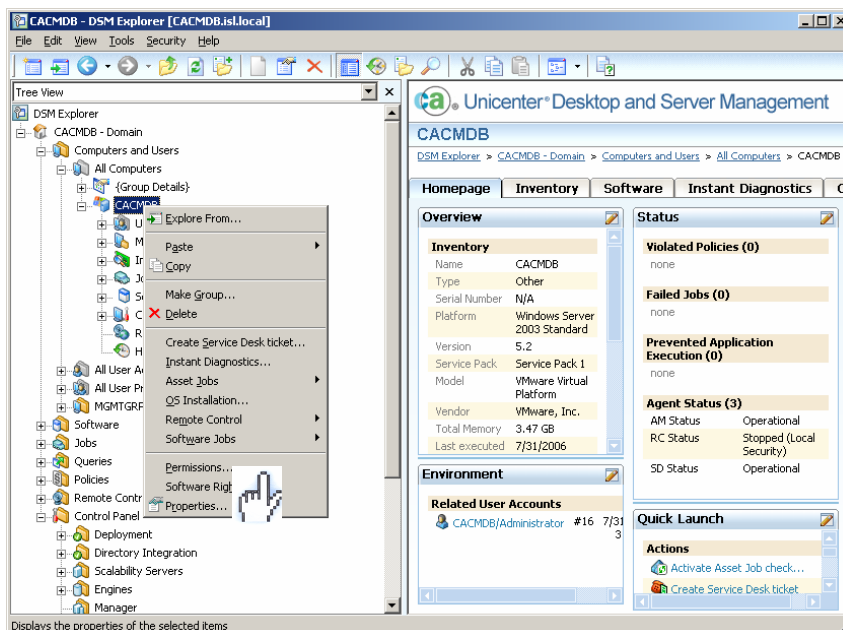
   A new CA Unicenter Service Desk incident window opens so the Asset Management analyst can manually create a new incident. There is some information already populated (that is, Configuration Item) on the incident detail window. That information

is part of the template used. You can add and modify the template information, if necessary, as illustrated in the following sample window:

Test the Integration using Non-Interactive Mode

Using CA DMS event policies, the CA Unicenter Service Desk and CA DMS integration can open and update tickets (Incident, Change Order, Problem, and Request) in CA Unicenter Service Desk. When an event policy is violated, a new ticket is created, or an activity log can be added to an existing ticket.

Complete the following steps to configure a new query policy that will be violated when it finds a computer that is using less than 1GB of RAM:

1. From the CA DMS explorer, browse to policies.
2. Click the plus symbol to expand the policies folder, select Query Based, right-click the Query Based option and click New.
3. In the Policy Name field, enter *Machines running < = 1 GB*.
4. In the Query that policy depends on field, select the query named Assets with Memory Size between 512 and 1024 MB RAM.
5. Click alert Policy Severity (first icon from left to right).
6. From the left side of the Policy designer, click Service Desk.
7. Select the Enable Service Desk Integration check box.
8. In the problem type drop-down list, select ASSET_EVENT_POLICY_HIGH and click OK.
9. Expand Policies, Query Based, select the policy named Servers Running on Machines running < = 1 GB, right-click the policy, and choose Evaluate Now.
10. Review the information on the right associated with the policy named Machines running < = 1 GB.

11. In the Information, Open related CA Unicenter Service Desk Ticket window, click it and wait a moment.

    The Service Desk Request List window opens, listing the incidents created for the computers that violated the policy. As a result, CA Unicenter Service Desk users can now request memory for the computers that are part of the incident and are being recognized as those computers not following Forward, Inc. standards.

Integration Summary

The integration of CA DMS with CA Unicenter Service Desk makes CA DMS a service-aware solution. This means that CA DMS can trigger based on certain events occurring with its managed assets and create tickets in CA Unicenter Service Desk.

Ticket creation in CA Unicenter Service Desk is controlled by the Service Aware policy with the name ManagedAssetEvents. The Service Aware policy is automatically installed when CA Unicenter Service Desk is installed. CA DMS uses a set of problem types available with the Service Aware policy to categorize the problem and to address the type of ticket to be created.

CA DMS and CA Unicenter Service Desk provide an interface that allows each product to be started in-context from the other product. The following table provides an overview of the supported in-context starts between the CA DMS Explorer or the Web Console and the CA Unicenter Service Desk Web Interface:

From	To	In Context	Context
Explorer/Web Console	CA Unicenter Service Desk Web Interface	Software job	Ticket being created on job failure
Explorer/Web Console	CA Unicenter Service Desk Web Interface	Asset policy	Ticket being created on policy violation
CA Unicenter Service Desk Web Interface	Explorer/Web Console	Ticket detail	Software job
CA Unicenter Service Desk Web Interface	Explorer/Web Console	Ticket detail	Asset policy

■   CA DMS can automatically create tickets in CA Unicenter Service Desk from a query or event policy, when those polices are violated.

■   CA DMS- Software Delivery integrates with CA Unicenter Service Desk only through policies. A ticket can automatically be raised for a failed Software Delivery job.

- CA DMS creates tickets in the context of discovered assets, for example, computers or users. When a ticket is created, a discovered asset is mapped to an owned asset, which is known in CA Unicenter Service Desk. This allows CA Unicenter Service Desk administrators to browse and report on relationships between tickets and owned assets.

- Assets that have been discovered by CA DMS, that CA Unicenter Service Desk is not aware of, can be certified as a CA Unicenter Service Desk owned resource from the search Asset List window, Discovered Asset functionality.

Chapter 6: Patch Management – Integrating with CA Unicenter Patch Management

What is CA Unicenter Patch Management?

CA Unicenter Patch Management is a dedicated solution for managing software patches in heterogeneous environments. It uses the capabilities of your existing CA Unicenter Software Delivery and CA Unicenter Asset Management, or CA Desktop Management Suite (DMS) installation to automate the identification, gathering, packaging, deployment, and ongoing validation of patches and related software configuration changes throughout your enterprise. There are two primary components to CA Unicenter Patch Management:

■ **CA Unicenter Patch Management**, which resides on your computer and provides a wizard-driven user interface to simplify the patch management process (package creation, testing, enterprise deployment, patch-level assurance).

■ **CA Online Content Service**, an online service available through SupportConnect to manage the collection of metadata for available or applicable patches. This service is provided as part of your CA Unicenter Patch Management subscription.

Using CA Unicenter Patch Management, you can do the following:

■ Know which software and patches are installed on your computer.

■ Establish patch level policies and ensure automatic compliance with the defined policies.

■ Perform impact and compliance analysis. For example, if patch ABC is required for product XYZ, you can quickly determine the computers that use that software, whether the patch has been applied, the departments where the computers are located, and the users who might be affected if the patch is not installed. This information helps you determine the appropriate plan for distributing the patch without disrupting day-to-day business.

■ Enforce the best practices for patch testing and distribution.

By using CA Unicenter Patch Management, you can recognize the following benefits:

■ A dedicated online patch research service that monitors for available patches, gathers the available patch data, and validates and identifies dependencies before publishing and pushing the patch information to the CA Unicenter Patch Management server.

- A formal patch management test phase. Packages can be deployed to test resources, allowing you to assess the impact of a patch before it is deployed enterprise-wide.

- Newly deployed, found, or rebuilt computers are automatically brought up to patch level compliance.

- Patch distribution is tracked real-time.

- State compliance can be determined and patch level assurance enforced.

- A flexible, complete web-based portal reporting system with automatic report scheduling.

- Integration with CA Unicenter Service Desk r11 and CA Unicenter Desktop and Server Management r11.

The CA Unicenter Patch Management Integration

The CA Unicenter Service Desk installation enables a web services interface allowing CA Unicenter Patch Management administrators to configure CA Unicenter Patch Management to automatically create service desk change orders for managing patch life cycles. When the integration is enabled, all patch processes that fall into the CA Unicenter Patch Management work flow attempt to create a change order with the configured CA Unicenter Service Desk server based on the specified template. The change order created contains the patch's summary and description.

Integration Points and Functionality

CA Unicenter Patch Management is designed to allow patch processing work flow to be invoked to drive patch promotion. There is built-in functionality that provides the ability to create a change order in CA Unicenter Service Desk when work flow is enabled and a CA Unicenter Service Desk server exists and is accessible from the CA Unicenter Patch Management server. If a valid CA Unicenter Service Desk server, username/password, and issue template are set in the CA Unicenter Patch Management settings, the CA Unicenter Patch Management work flow processing will attempt to create a change order in CA Unicenter Service Desk every time a patch enters the enabled work flow state.

Note: CA Unicenter Patch Management work flow is a component unique to CA Unicenter Patch Management, and is not related to the CA Unicenter Service Desk work flow IDE.

Integration Points from CA Unicenter Patch Management

The following is an integration point from CA Unicenter Patch Management:

- Automatic Change Order creation when a user selects a patch in any of the following states from CA Unicenter Patch Management:

> Acceptance

> Deferral

> Approval

> Deployment

CA Unicenter Service Desk Integration Points

The following is an integration point from CA Unicenter Service Desk:

■ Change the status of the patch in CA Unicenter Patch Management once a work flow task activity has been accepted or approved.

Integration Value

The CA Unicenter Patch Management integration provides the following value:

■ Automatic change order creation for patches as they become available.

■ Consistent approval process for the changes.

■ Complete management of the patch process from approval through testing, and finally deployment.

■ Automatic notification to the service desk staff that a patch has already been, or will be deployed, to allow service desk management to correctly anticipate analysts' workloads.

■ The automatic updating of Configuration Items from the patch's point of view.

How the Integration Works

The following diagram illustrates how the CA Unicenter Patch Management integration works:
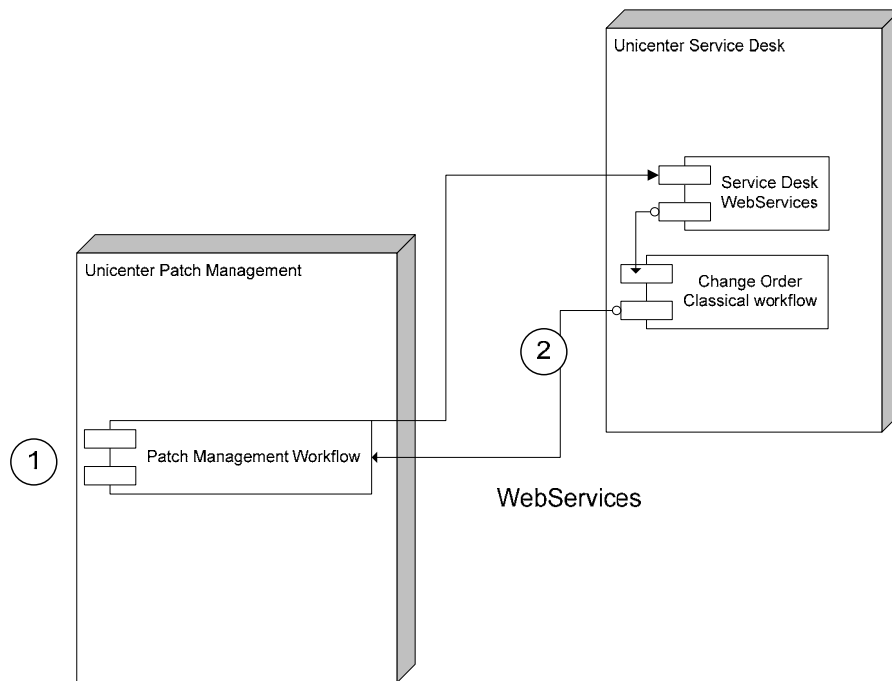
The following information applies to the previous diagram:

1. A change order is created in CA Unicenter Service Desk when the CA Unicenter Patch Management work flow is enabled and a CA Unicenter Service Desk server exists and is accessible from the CA Unicenter Patch Management server. The CA Unicenter Patch Management work flow processing will attempt to create a change order in CA Unicenter Service Desk whenever a patch enters the enabled work flow state. This requires a valid CA Unicenter Service Desk server, username/password, and issue template to be set in the CA Unicenter Patch Management settings.

2. The CA Unicenter Service Desk change order has work flow capabilities that, for example, enable CA Unicenter Service Desk to interact with CA Unicenter Patch Management work flow (using remote references functionality) to manage the status for a CA Unicenter Patch Management patch status in the promotion life cycle.

Example of the CA Unicenter Patch Management Integration

Business Challenge

The Forward, Inc. HR representative, June Arnold, is attempting to submit her expense report in the new Expenses product developed in-house. June has encountered an error in the software, and an incident has automatically been opened in CA Unicenter Service Desk using the integration between the Expenses product and CA Unicenter Service Desk. The integration was implemented by Forward, Inc. with the objective of making CA Unicenter Service Desk aware of potential errors that users may encounter in the most commonly-used products in the organization.

Donald Bell, first-level support for Forward, Inc's help desk, researched the incident in the CA Unicenter Service Desk Knowledge Tools documents, and concluded that a required patch is needed on the Expenses product running on June's computer. Donald knows that

to apply fixes, the Change Management department must provide an authorization. However, Donald is not familiar with the official policy and procedure, and he also does not know who else may be running the Expenses product. Donald only knows that the product was installed on more than 300 computers across the organization three weeks ago.

CA Approach

Keeping all software in an organization controlled and updated can be difficult and expensive. CA offers CA DMS-Software Delivery and CA Unicenter Patch Management. Working together, these products can automate the identification, gathering, packaging, deployment, and ongoing validation of patches and related software configuration changes throughout your enterprise.

Best Practices

For best practices to implement CA Unicenter Patch Management, see the following location:

http://supportconnectw.ca.com/public/unipatchmgt/infodocs/TEC395266.pdf

Configuring a Solution

Susan Jobin, the IT Asset Manager, is responsible for coordinating all necessary resources and activities to meet the requirements to automate the procedure to update the software in Forward, Inc's Configuration Items (CI). Part of the solution is that an incident must trigger the software update and change procedure. To implement the solution, an integration between CA Unicenter Service Desk and CA Unicenter Patch Management must be set up.

Configure the Integration from CA Unicenter Patch Management

Complete the following steps in CA Unicenter Patch Management to configure the integration:

1. From the Start menu, choose Programs, Computer Associates, CA Unicenter, CA Unicenter Patch Management, Deploy workflow integration.

    A command prompt window opens. The CA Unicenter Patch Management Workflow is being activated, and the process runs in the background. When the process is complete, the command prompt window closes.

    Note: CA Unicenter Patch Management work flow is a component unique to CA Unicenter Patch Management. It is not a component of the CA Unicenter Service Desk workflow IDE.

2. Log in to CA Unicenter Patch Management as an Administrator.

3. Click the Administration tab.

4. On the left side of the window, click events.

    On the right side of the window, the configuration window opens.

5. Enter the following information for the configuration fields in the window named Service Desk:

   Web Services URL: *http://forwardinc:8080/axis/services/USD_WebServiceSoap*

   Note: When running on Tomcat the default port is 8080, but you should check first that this port is not already used by another application. You can use the following command line to check whether this port is already in use: netstat –a |more

   User: ServiceDesk

   Password: Enter the CA Unicenter Service Desk password.

   Note: As a best practice, the CA Unicenter Service Desk user should be replaced by a different user (that is, System_UPM_generated). This name is used when configuring Patch Management. This user must exist in CA Unicenter Service Desk and the Access type must be Analyst. CA Unicenter Patch Management is compatible with USPSD 6.0 and r11 web services. However, the versions offer different web service URL strings. CA Unicenter Service Desk 6.0 implements a .NET web service, with the following default URL:

   http://<USPSDServerAddress>:80/USD_WS/usd_ws.asmx

   CA Unicenter Service Desk r11 offers two web services. CA Unicenter Patch Management will *only* integrate with the r11 web service that is backward-compatible with USPSD 6.0 web service clients. The following is the default, backward-compatible web service URL for r11:

   http://<USPSDServerAddress>:8080/axis/services/USD_WebServiceSoap

6. On the External Event Confirmation window, select the Patch Acceptance, Patch Deferral, Patch Approval, Patch Deployment, and Policy Update check boxes.

   Note: For each option you select in this step, a Change Order Template in CA Unicenter Service Desk must already have been created. In addition, the name of the template must be populated into the respective field.

7. (Optional) A CA Unicenter Service Desk Change Order template is a Change Order model that can be used when creating new CA Unicenter Service Desk Change Orders. To create the template, complete these steps:

   a. Log in to CA Unicenter Service Desk as an Administrator or Analyst.

   b. Click the Service Desk tab and choose File, New Change Order.

   c. Enter the information for the new change order.

   d. Click the Template tab in the Change Order Detail window.

   e. In the Template Name field, enter a name for the template.

   f. Click Save.

Configure the Integration from CA Unicenter Service Desk

Complete the following steps in CA Unicenter Service Desk to configure the integration:

Note: As previously mentioned, Change Order Templates must be created for each status selected in the UPM Administration tab, External Event Confirmation in the following steps. It is assumed that the templates have already been created.

1. Log in to CA Unicenter Service Desk as an Administrator.

2. Create a upminteg.bat file in any directory. In this example, the file will be created in the c:\directory, containing the following command line:

```
java -Djava.class.path="C:\Program Files\CA\CA Unicenter Patch
Management\bin\event.jar"

com.ca.CA Unicenter.upm.eventmanager.UPMEMSAck
```

"http://bsodsmv1:8090/upm/services/UPMEMS" %1 true

3. Create a remote reference using the following steps:

   a. From the main CA Unicenter Service Desk window, click the Administration tab and choose Servicedesk, Application Data, Remote References.

   b. Click Create New in the left side of the window.

   c. Complete the fields in the Create New Remote Reference window, using the following sample window as reference.

4. From the main CA Unicenter Service Desk window, click the Administration tab, Events and Macros, Macros.

5. Create a new macro to call the remote reference previously created.

   Object Type: workflow task

   Macro Type: execute remote reference

6. Create a new change order category, for example *Change.IT.Workstation.Config*.

7. Add a work flow task named *approval*, and in the task behavior for the approve status, add the macro previously created in step 5 as the action on true.

Test the Integration

Complete these steps to verify that the integration is working correctly:

1. Log in to CA Unicenter Patch Management.
2. From the Dashboard tab, click one of the patches in the left side window named Patches Pending Acceptance.

   A new change order should be created in CA Unicenter Service Desk, as illustrated in the following sample window. The description contains all the information defined in CA Unicenter Patch Management for the patch, as well as the Status based on the Work flow functionality active in CA Unicenter Patch Management.

After the Change order approval task has been approved, the status for the patch will change from Pending Acceptance to Patches Pending Testing in the CA Unicenter Patch Management Dashboard.

Note: The CA Unicenter Service Desk administrator can complete the steps described in Configuring the CA Unicenter Service Desk and CA Unicenter Patch Management Integration from CA Unicenter Patch Management. For other statuses in the CA Unicenter Patch Management work flow, more CA Unicenter Service Desk work flow tasks can be added to start remote references with the proper line to update CA Unicenter Patch Management work flow status.
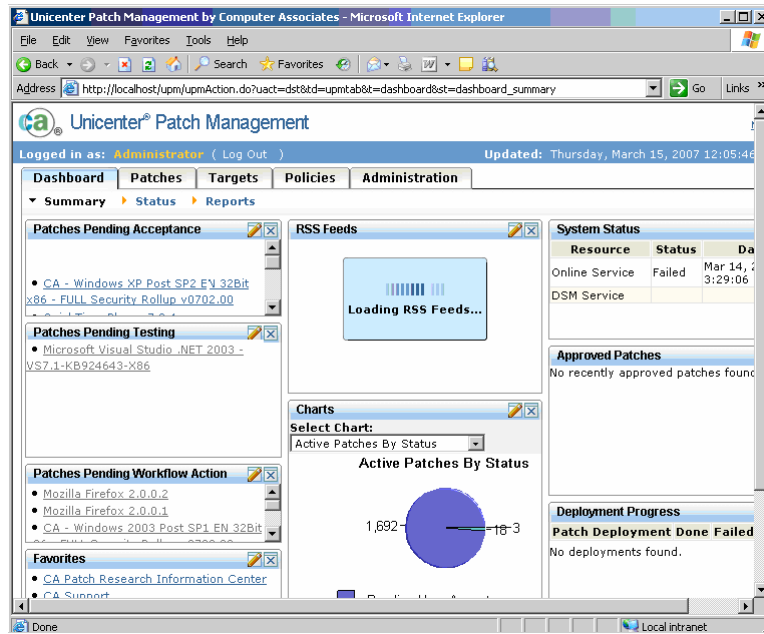
Integration Summary

CA Unicenter Service Desk provides the ability to enable a web services interface for remote administration. CA Unicenter Patch Management takes advantage of these web services, giving CA Unicenter Patch Management Administrators the ability to create CA Unicenter Service Desk change orders for patch work flow management. Additionally, CA Unicenter Service Desk can execute a set of remote references to complete the patch life cycle in both CA Unicenter Service Desk and CA Unicenter Patch Management, allowing Change Management departments to control and track all software updates inside the organization.

Chapter 7: Financial Asset Management – Integrating with UAPM

What Is UAPM?

CA Unicenter Asset Portfolio Management (UAPM) r11.x is a comprehensive IT ownership management solution that helps to manage the financial and ownership information of an organization's IT-related Configuration Items (CI). UAPM supports the entire life cycle of a CI from procurement through acquisition, allocation, use, and disposition, including legal and cost information.

The UAPM Integration

The integration of UAPM r11.x with CA Unicenter Service Desk makes UAPM a service-aware solution. This means that UAPM can trigger the creation of incidents in CA Unicenter Service Desk, using web services technology, based on certain events in its management data. For this integration, incident creation in CA Unicenter Service Desk is controlled by the service-aware policy that enumerates a list of problem types. UAPM uses the type to categorize the problem and create the appropriate type of incident in CA Unicenter Service Desk.

Integration Points and Functionality

UAPM manages a component named *Integration Services*. This component provides an outbound command using web services between UAPM and CA Unicenter Service Desk. When a change that should create a CA Unicenter Service Desk incident report takes place within the UAPM data, the Integration Services component performs the following actions to generate incidents in CA Unicenter Service Desk and communicate the incident to your IT support organization:

- The UAPM Notification Server sends notifications to users when changes occur in pertinent data within the UAPM repository.

- CA Unicenter Service Desk user acts as the user who receives the notification. These notifications are in the form of commands to CA Unicenter Service Desk.

- UAPM Notification Events in the Notification Server invoke the pertinent Integration Services command. You specify the events that are to trigger Integration Services in the server.

- Integration Services inserts the appropriate information into the notification or command parameters using three XML files (ServiceDescriptor.xml, a template file, and

ObjectDescriptor.xml) and instructs CA Unicenter Service Desk to open the appropriate service desk incident, based on the template selected when selecting the web services policy when configuring the integration.

How Integration Services Uses XML Files

Integration Services uses three XML files to build a command for CA Unicenter Service Desk:

■ ServiceDescriptor.xml provides the CA Unicenter Service Desk command information.

■ Template files control the UAPM data that is sent to CA Unicenter Service Desk. The UAPM data is inserted into the template, and is then passed to Integration Services for execution.

■ ObjectDescriptor.xml describes the UAPM object types and the possible variable substitutions, as well as the text of the substitution variable. This file should *not* be modified.

Integration Services merges data automatically from the three files to create the command information for the CA Unicenter Service Desk ticket. The command is transmitted to CA Unicenter Service Desk using the CA Unicenter Service Desk web services interface.

Integration Points from UAPM

The following are integration points from UAPM:

■ UAPM and CA Unicenter Service Desk share data when running on top of the same MDB for Configuration Items (CI), Contacts, Models, Locations, and Company/Companies information. For example, a CI created in UAPM is immediately available to be managed in CA Unicenter Service Desk. The same situation applies to all shared data.

■ Automatic incident creation triggered by a UAPM event notification.

■ The ability to generate list of incidents by CI in UAPM.

■ The ability to start the Common Asset Viewer (CAV) from UAPM and then have access to all information for the Managed CI, as well as a service desk link to initiate a start, in context, of CA Unicenter Service Desk.

■ The ability to take ownership of a CI when it has been created in UAPM first. This means that although the CI is being shared with CA Unicenter Service Desk, the relevant UAPM information is read-only for the CA Unicenter Service Desk staff.

Integration Points from CA Unicenter Service Desk

The following are integration points from CA Unicenter Service Desk:

- CA Unicenter Service Desk analysts can interactively view the Managed CI information being shared between UAPM and CA Unicenter Service Desk, using the Common Asset Viewer (accessible from the CA Unicenter Service Desk Configuration Item detail window). The UAPM link can start a launch, in context, to UAPM.

- CA Unicenter Service Desk and UAPM share data when running on top of the same MDB for CIs, Contacts, Models, Locations, and Company/Companies information. For example, a CI created in CA Unicenter Service Desk is immediately available to be managed in UAPM. The same situation applies for all shared data.

Integration Value

The CA Unicenter Asset Portfolio Management integration provides the following value:

- Control, track, and improve life cycle management of CIs

- Make owned CI data available to first-line support through the Common Asset Viewer

- Provide easy access to service contract information and other legal information

- Control stock

- Provide better user support

- Help lower the total cost of ownership (TCO)

- Control, track, and improve procurement procedures

- Allow service desk analysts to determine whether an asset having problems is under maintenance, and can therefore be fixed at no cost

How the Integration Works

The following diagram illustrates how the CA Unicenter Asset Portfolio Management integration works:
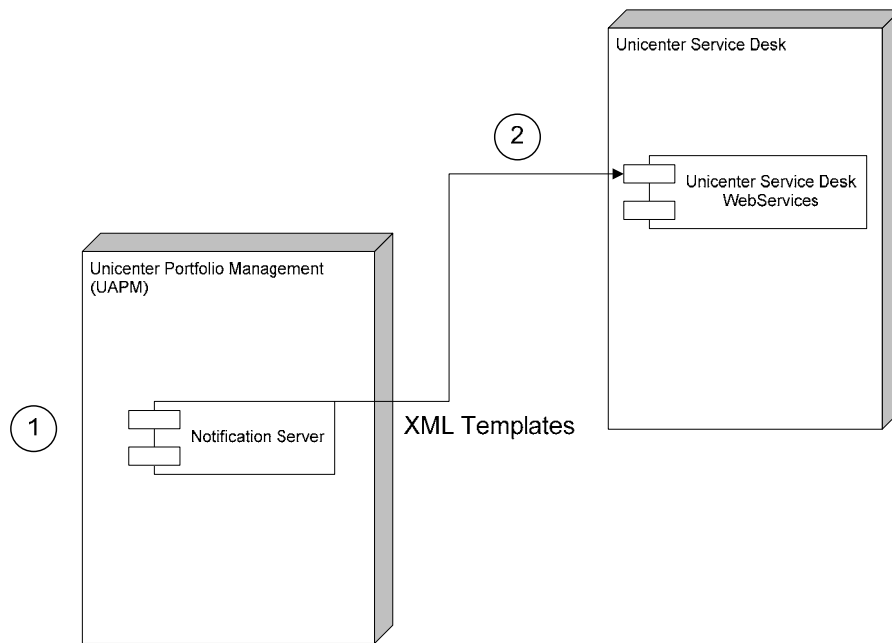
The following information applies to the previous diagram:

1. UAPM Notification Events in the Notification Manager invoke the pertinent Integration Services command. You specify the events to trigger Integration Services in the server. Integration Services inserts the appropriate information into the notification or command parameters using three XML files (ServiceDescriptor.xml, a template file, and ObjectDescriptor.xml).

2. Integration Services instructs CA Unicenter Service Desk to open the appropriate service desk ticket, based on the template selected when selecting the web services policy when configuring the integration.

Example of the UAPM Integration

Business Challenge

Marge Walton, Procurement Manager at Forward, Inc., has raised a concern to Paul Kim, Director of IT Services, about the procurement department's procedures regarding orders from their providers (servers, printers, laptops, and so forth) that are not matching user requirements. Her team has not been very effective in delivering the correct resources to the requesters, because the devices are being delivered directly from the provider to the user. This procedure has caused a large number of incidents to be opened at the service desk due to incorrect hardware and software configurations. Michael Reed, the Service Desk Manager, has suggested that Marge implement a procedure that will let the service desk know when the hardware is being delivered, so that his staff can intercept the devices and allow IT Services to configure them according to the pre-established IT Services standard configurations.

CA Approach

To manage and control Configuration Items (CI), other Forward, Inc. departments (in addition to procurement) need to be involved in all processes that affect the life cycle of the assets/CIs. For the particular problem that Marge has reported to Paul, the integration of UAPM with CA Unicenter Service Desk will allow procurement to notify the service desk when hardware devices are delivered. Then, IT Services technicians can properly configure and test the devices before delivering them to the users. When new technologies arrive in the organization, this information will be made available through the dashboard to allow proactive decisions to be made. One example is the creation of instructional training to help prevent users opening unnecessary incidents within the service desk.

Configuring a Solution

Paul scheduled a meeting with Marge and Michael Reed to coordinate the details about the official procedure to order, receive, and deliver new hardware. Paul wants the hardware to be properly configured and tested before it is delivered to the user. He also wants to receive information about the delivery in a report so that Forward, Inc. management will know what is happening with both the provider orders and the user's hardware requests.

Configure the Integration from CA Unicenter Service Desk R11

Complete the following steps in CA Unicenter Service Desk to configure the integration:

1. Log in to CA Unicenter Service Desk as the ServiceDesk or Administrator user.
2. On the Service Desk tab, select Search, Contacts.

   The Contact Search window opens.

3. In the Last Name field, type *System_Argis_User* and click search.

   The name *System_Argis_User* appears in the search results.

4. Make sure that *Analyst* appears for the Contact Type, and *Administrator* appears for the Access Type.
5. Browse to the CA Unicenter Service Desk main window and click the Administration tab.
6. Review the information on the left and expand the option named Web Services Policy.
7. Click Policies.
8. On the right, click the symbol UAPM Policy.

   The Web Services Access Policy Detail window opens.

9. Click edit.
10. On the editing window, click the Problem Types tab.

    Note: There are 13 different problem types that can be associated with the UAPM Policy.

11. Click the name UAPM Event.

    The Web Services Problem Type Detail window opens.

12. Click edit.

13. Select the Default check box.

14. In the Ticket Template Type drop-down list, choose Incident.

15. In the Ticket Template Name search field, select UAPM ITIL Asset Event.

    Note: Information in the templates can be added and modified, and additional templates can be created.

16. Click the Duplicate Handling tab and select Add Activity Log (do not create ticket).

17. Click Save and exit from both windows that were previously opened.

Configure the Integration from UAPM R11

Complete the following steps in UAPM to configure the integration:

1. Verify that the optional Integration Services for UAPM components have been previously installed.

2. From SQL Query Analyzer, connect to MDB DB with the user uapmadmin and password uapmadmin.

3. Execute the script named ING_ARGDEFLT.sql, located in the C:\Program Files\CA\Unicenter Asset Portfolio Management\Manager\IntServices directory.

    Note: If the script generates an error within MSSQL or Oracle, run a manual query instead. Use the following syntax within the query: UPDATE dbo.arg_argdeflt SET dfvalue='0' WHERE dfkey='Safe Subset Used';

4. Choose Start, All Programs, CA, Unicenter Asset Portfolio Management, Configurator to open the UAPM Configurator.

5. Click Integration Services.

6. Specify the following information:

    > Select the database being used.

    > Specify the appropriate server and database name information. The table owner is usually dbo.

    > Make sure you select the Service Desk Security option, and enter the privileged CA Unicenter Service Desk Login ID and CA Unicenter Service Desk Password.

    > Modify the Service Desk URL field that contains the entry: *http://servername:8080/axis/services/USD_R11_WebService?wsdl*.

      This URL contains the default Tomcat port 8080. You should check first that this port is not already used by another application. You can use the following command line to check whether this port is already in use: netstat –a |more

      Change the server name and port to the values currently being used to access your CA Unicenter Service Desk installation.

7. Click Update, and then close the window.

8. From the configuration window, click Asset Portfolio Management.

9. Make sure the Enable Scripts and Enable Events check boxes are selected, and click update.

10. Exit the UAPM Configurator.

11. Choose Start, All Programs, CA, Unicenter Asset Portfolio Management, Notification Manager to start the Notification Server.

12. Expand the tree for local (traffic light), click INI Settings, and modify the values to match the following sample window:
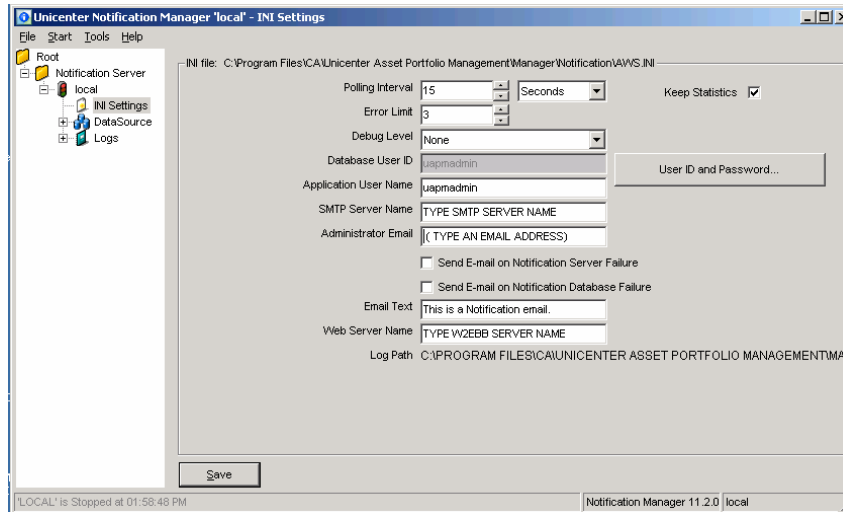
13. Click Save to stop the Notification Server.

14. Start the Notification Server again by choosing the Start option on the main menu at the top of the Unicenter Notification Manager window.

15. Choose Start, All Programs, CA, Unicenter Asset Portfolio Management, APM Utility to update the NotifyConfirm Script.

16. Log in to the APM Utility as uapmadmin.

17. From the main menu, choose Utility, Edit Script.

18. Expand Event Notification in the tree and click Confirm.

19. In the script field, highlight the current information and delete it.

20. Make sure the selected language is VBScript, and that both the Disabled and Show End of Line Marks check boxes are not selected.

21. Click Import.

22. Browse to import the script file *NotifyConfirm.txt* from the C:\Program Files\CA\Unicenter Asset Portfolio Management\Manager\IntServices directory.

23. Click Check.

    You should receive a message from the script editor displaying the following text: *There were no syntax errors found*.

24. Click Save.

25. Exit the UAPM Portfolio Management Utility.

Check the List of Open Tickets for an Asset

Use the following steps to add a new option to the Asset Detail window in UAPM to check the list of tickets open for a particular Asset:

1. From the UAPM Home page (http://servername/apm), click Assets.

2. Click the Tree Node Shortcuts tab.

3. Select the Display all tree node shortcuts for Asset check box.

4. Select Service Desk request for Asset (Sample).

5. Click Edit.

6. Select Save as.

7. Change the Title to *Service Desk Request for Asset*.

8. Select the assignment and role for the person who will be able to view this shortcut.

9. Click Save.

10. Navigate to any asset detail window.

    At the end of the folders at the left, you should see an option named *Service Desk Request for Asset*.

11. Expand the node and click View Data.

    Tickets created in CA Unicenter Service Desk for this particular asset display.

Test the Integration from UAPM

To test the integration and create a notification to trigger an incident in CA Unicenter Service Desk, complete the following steps:

1. From the UAPM Home page (*http://servername/apm*), click the Events and Notifications link on the right side of the window.

2. Expand Change Events, By Field.

3. Expand Asset Version, Portfolio Status and click <New>.

4. On the right side of the window, enter the following information:

    Event Name: Creating Test Incident (You can enter anything you want).

    Event Cause/Data Changed: Field Changed

    Value Change from: Any value

    Enabled: Select this check box.

    Event Notification Override: Keep pending event notifications

    Value Change to: Any value

5. Click Insert Row on the right side of the window.

6. Enter the following information in the columns:

    Enable: Select this option

    Type: Initial Event

    Recipient: command - (type this in the field)

    Days After: 0

Text: icmdsvc /cmdtemplate AssetEvent /objectid {ObjectID} /new_value "{new_value}" /old_value "{old_value}" /eventid {EventID}

7. Click Save.

8. Return to the UAPM Home Page.

9. Click the Asset (Manage Assets) link and perform a basic search to find any available asset.

10. Click an asset.

    The Asset Detail window opens.

11. On the tree menu, expand Status history for asset and click the current asset status link.

12. Change the status drop-down list to a different value and click Save.

    A notification will be generated, and a new incident will appear in CA Unicenter Service Desk.

13. To verify that the incident has been created, log in to CA Unicenter Service Desk and search the assigned Incidents for an UAPM Event and with a contact System_Argis_User.

Integration Summary

The UAPM integration is accomplished by the Integration Services, which is an interface between both solutions. Using Integration Services you can generate CA Unicenter Service Desk tickets automatically when you make change in your UAPM repository. Integration Services ensures that changes in your IT asset management environment are accurately communicated to your IT support organization in one step. For example, you could use Integration Services to inform your IT support organization about new desktops. As your IT asset management group creates asset records for the new desktops, your IT support group could receive a service ticket with deployment instructions for each new desktop.

This interface is triggered by the Notification Server. Integration Services provides a predefined service template, which you can use to open CA Unicenter Service Desk Requests based upon status changes to UAPM assets. You can use this template without modification, or you can modify it to create tickets based on other UAPM event types. The type of ticket created by this template is controlled by a CA Unicenter Service Desk web services policy named UAPM_POLICY, which can be modified to provide support for all ticket types, including Information Technology Infrastructure Library (ITIL) versions of requests and change orders.

Chapter 8: Service Management – Integrating with CA Unicenter Service Delivery and CA Unicenter Service Catalog

What are CA Unicenter Service Delivery and CA Unicenter Service Catalog?

CA Unicenter Service Delivery is a fully automated and integrated, multi-vendor service management solution that lets you closely link IT operations with business requirements. With CA Unicenter Service Delivery, you can track resource usage by user, business unit, or line of business, set and manage service levels, and gain visibility into the costs of the services that you provide. CA Unicenter Service Delivery includes the following products:

■ CA Unicenter Service Catalog

■ CA Unicenter Service Accounting

■ CA Unicenter Service Assure

■ CA Unicenter Service Metric Analysis

The CA Unicenter Service Delivery component that integrates with CA Unicenter Service Desk is CA Unicenter Service Catalog. The CA Unicenter Service Catalog provides a container that consists of offerings published by a business unit or enterprise. Offerings are built with one or more rate plans to describe IT services and how to charge for them. The service catalog also allows an organization to model its business units and manage the user accounts contained within those units. Offerings in the service catalog may be organized into folders and may contain detailed information about the price of a service. Catalog offerings may include one or more metrics, and are governed by service level agreements.

In addition, CA Unicenter Service Delivery provides a set of common components. This integration utilizes Unicenter Service View. This required common component provides the user interface for all CA Unicenter Service Delivery products. Unicenter Service View contains a portal that is used to view the product windows and reports. This dashboard serves as the gateway through which information is made available anywhere using a web browser. It is also the access gateway to other components in the CA Unicenter Service Delivery group of products. All user management is performed using this component.

The CA Unicenter Service Catalog Integration

CA Unicenter Service Catalog and CA Unicenter Service Desk integrate out-of-the-box using CA Workflow through the use of their respective web services. CA Workflow definitions are included with both products out-of-the-box to provide sample process flows that enable the communication between CA Unicenter Service Catalog and CA Unicenter Service Desk. This communication allows the integration to accommodate a full service life cycle, taking user requests from submission, to approvals, to generation of CA Unicenter Service Desk tickets, with complete fulfillment processes.

Integration Points and Functionality

CA Unicenter Service Catalog comes with a full set of rules, conditions, actions and associated CA Workflow process definitions to take user requests from submission to fulfillment. All of the rules are initially set to Disabled. For a rule to be used, it must be set to Active. In addition, for some of the rules there are two mutually-exclusive actions, one to be used when interfacing with CA Unicenter Service Desk and one to be used when CA Unicenter Service Desk is not installed.

The CA Unicenter Service Catalog integration with CA Unicenter Service Desk is able to create a ticket based on the status of a request performed by a user in the CA Unicenter Service Catalog. Service Delivery Work flow validates the predefined status and creates a CA Unicenter Service Desk incident or change order using web services.

Integration Points from the Service Catalog

The following are integration points from the Service Catalog:

■   Automatic incident and change order creation from a CA Unicenter Service Catalog request.

■   A powerful work flow component that is able to send incident information to CA Unicenter Service Desk.

■   A powerful work flow component that is able to interact with CA Unicenter Service Desk web services to import and export information between CA Unicenter Service Desk and CA Unicenter Service Catalog to fit into an organization's business needs.

Integration Points from CA Unicenter Service Desk

The following integration point is from CA Unicenter Service Desk:

■   A powerful work flow IDE component that is able to interact with CA Unicenter Service Desk web services to import and export information between CA Unicenter Service Desk and CA Unicenter Service Catalog to fit into an organization's business needs.

Integration Value

The CA Unicenter Service Catalog integration provides the following value:

■   CA Unicenter Service Catalog can be used as the organization's services front end for user requirements. This helps to ensure that only requests that require service desk analyst involvement go to CA Unicenter Service Desk. In turn, this helps reduce the service desk staff workload, allowing service desk analysts to increase productivity as well as improve customer service.

■   Service Catalog requests can go through a work flow process in which a sub-process can be invoked to open a CA Unicenter Service Desk ticket. The ticket can also initiate a separate process that is strictly tied to the type of CA Unicenter Service Desk request opened. This process does not interfere with the higher-level process that the Service Catalog request is taking. The benefit is that the integration can allow for the streamlining of complementary processes.

■   In the long term, if a customer wants to be able to understand costs, financials, and Service Level Management through Service Contracts, then CA Unicenter Service Desk and Service Catalog requests that are related (regardless of which request initiated the other) will be able to leverage all the other components of Unicenter Service Management such as Service Accounting, Assure, and SMA. Without the integration, CA Unicenter Service Desk cannot extend to a Service Delivery model.

How the Integration Works

The following diagram illustrates how the CA Unicenter Service Catalog integration works:
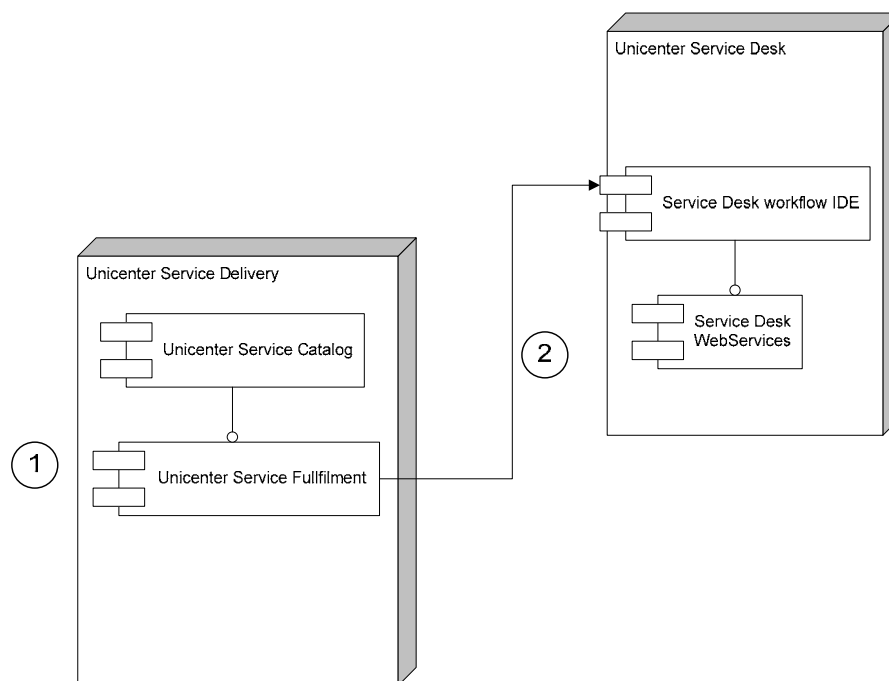
The following information applies to the previous diagram:

1. CA Unicenter Service Catalog interacts with Unicenter Service Fulfillment to create a request in CA Unicenter Service Desk based on an action by a user in the Service Delivery Catalog.

2. Service Delivery fulfillment interacts with the CA Unicenter Service Desk Workflow IDE to make a web service call, and then creates a CA Unicenter Service Desk Request or Change Order, depending on how the work flow process has been configured.

Example of the CA Unicenter Service Catalog Integration

Business Challenge

Paul Kim, IT Director at Forward, Inc. has detected that the service areas in the organization have multiple entry points. There are multiple service providers (such as HR, Finance, Procurement, Accounting, and the service desk) with unstructured and undocumented manual processes. This causes errors, an inability to execute tasks in a timely manner, and reduces the quality of service delivery. In addition, Paul has found that multiple requirements and questions directed to the service desk are not really appropriate for the service desk. This reduces the productivity of the service desk staff. Paul needs to find out how to offer a unique front end to all Forward, Inc. users which will allow them to make their requests, regardless of which department is the main provider, and to ensure that the request is sent to the correct department for fulfillment.

CA Approach

Typically, users have separate products and processes they may use to submit requests for IT-related services. To complicate the situation, users may have to know the appropriate product and process they must use for the specific request they are submitting, and the product used to make the request may not be intuitive or user-friendly.

When a user requests IT services or goods, they should not have to know detailed and technical specifications for the goods or service. Delivery of the IT services should be performed in a consistent manner so that contracted service levels are met. CA advises Forward, Inc. to implement Service Catalog, and to integrate it with the most critical service applications such as CA Unicenter Service Desk, Procurement, and HR.

Configuring a Solution

Paul Kim has requested that Don Hailey, the IT Manager, coordinate all pertinent activities to implement CA Unicenter Service Catalog as a front end for the users. The first phase of the project should give priority to the service providers in the service desk, procurement, and HR departments.

Configure the Integration from CA Unicenter Service Desk R11

Complete the following steps in CA Unicenter Service Desk to configure the integration:

Note: All windows referenced in this example are part of the *Service Delivery Implementation Guide*. USM refers to Unicenter Service Management - Service Delivery. USD refers to CA Unicenter Service Desk. Finally, CA Unicenter Service Desk r11.x has been configured to run on ITIL and use eIAM authentication.

1. Use the USD Web Screen Painter to add the following attributes to chg table and commit to schema:

    > zusmrequestid (make attribute an integer)

    > zusmrequestitemid (make attribute an integer)
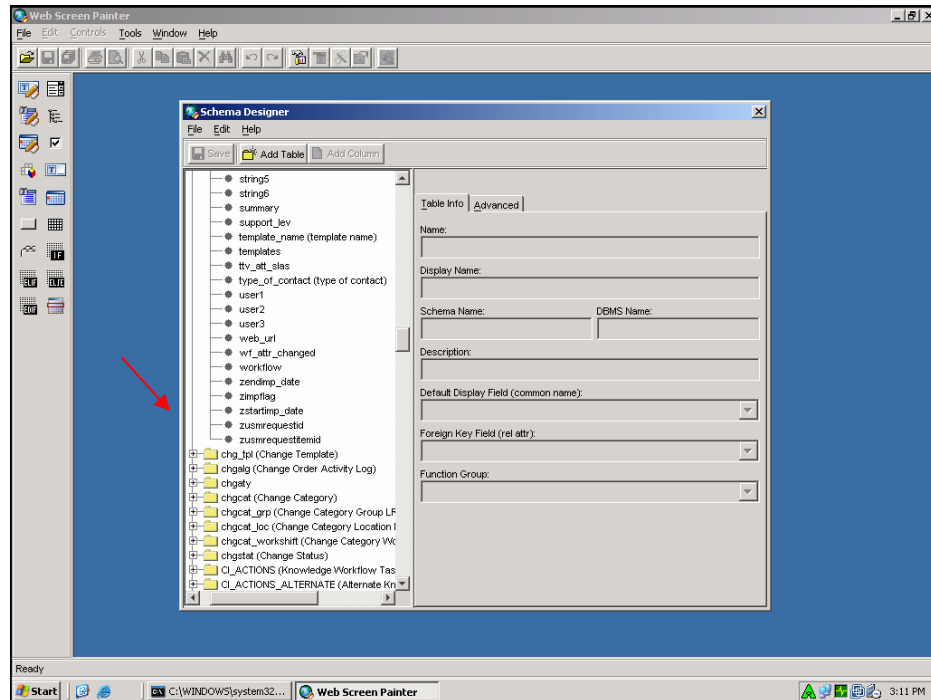
2. Import the web service actor named *USM_RequestService_en.xml* into the USD work flow. You can find this file in the %USM_HOME%\fulfillment\scripts\service_desk\actors directory on the USM Server.
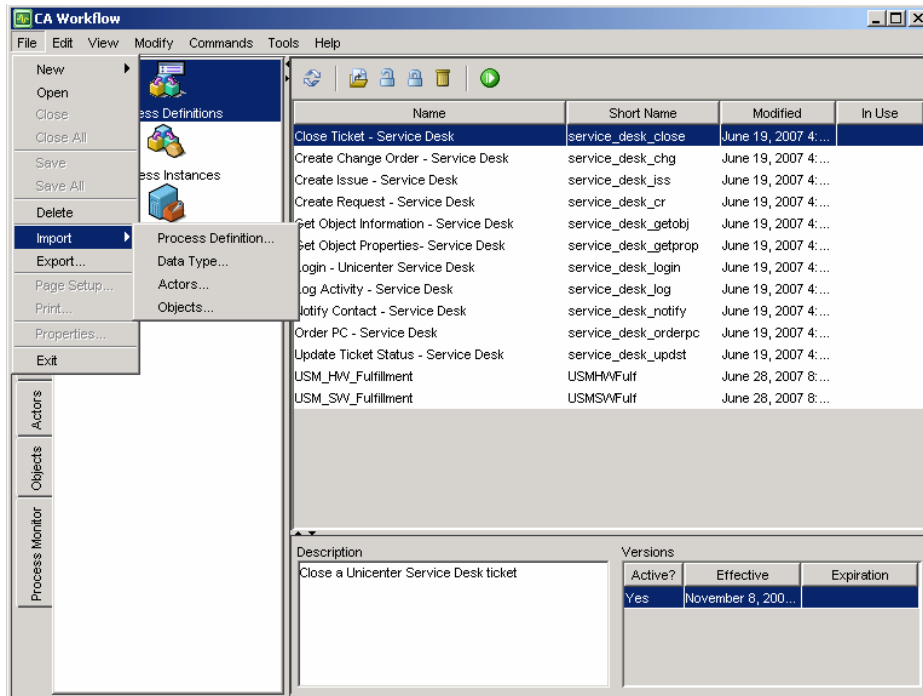
Note: This is the web actor that is used by USD to communicate with USM. If Unicenter USM is in a separate computer from the USD, the import may not correctly reference CA Unicenter Service Delivery. Verify that the WSDL URL correctly references the CA Unicenter Service Delivery location by clicking the actors tab in the Service Desk Workflow IDE Window, WebService. Then, right-click the web service named USM_RequestService, Select Properties. If the CA Unicenter Service Delivery port has been changed after the actor has been imported, delete the USM_Request Service actor in USD Workflow. Then, add it back and point it to the WSDL URL (*http://<CA Unicenter Service Delivery host>:<CA Unicenter Service Delivery port>/axis/services/RequestService?wsdl*).

3. Import the following work flow definitions into the USD workflow IDE from USM. You can find these definitions in the %USM_HOME%\fulfillment\scripts\service_desk\processdefinitions directory on the USM server.

   > USM_HW_Fulfillment_en.xml

   > USM_SW_Fulfillment_en.xml

4. Log in to CA Unicenter Service Desk as an Administrator or a CA Unicenter Service Desk user and add the following change categories to the Change Order component:

   > USM.Hardware (Set this to use CA-Workflow and reference the Unicenter Service Delivery_HW_Fulfillment workflow)

   > USM.Software (Set this to use CA-Workflow and reference the Unicenter Service Delivery_SW_Fulfillment workflow)

5. Import the USD web service actor into USM work flow. USM needs this to log into USD.

Note: For information about WSDL, see the *Unicenter Software Delivery Administrator Guide* and *Unicenter Software Delivery Implementation Guide*.

6. Verify the USD Workflow using the following steps to synchronize the hostname, ports, and URLs:

   a. From the USD Workflow ID window, click Process Manager, and double-click the process named USM_HW_Fullfillment process.

   b. Click the Configure HW node (open sub-process in tab).

   c. Double-click Email Configuration User.

   d. Click the Event tab.

   e. Select the Value Expression field and make sure the URL references *http://<USM host>:<USM port>*. Repeat this step to make sure the Document URL row in the parameter table is referencing USM. Finally, repeat this step to make sure the correct URL is in the Email Delivery User activity node and the Document URL is in the parameter table.

7. Verify the USD Workflow using the following steps to synchronize passwords:

   Note: This step is important if the CA Unicenter Service Desk default password (servicedesk) has been changed.

   a. From the USD Workflow IDE window, click Process Manager, and double-click the process named Login-Service Desk work flow definition.

   b. Double-click Call Login and find username/password in the work item expression.

8. Verify the USM Workflow using the following steps to synchronize passwords:

   a. From the USM Workflow window, click Process Manager, and double-click the process named USD_Login.

   b. Double-click USD Login and find username/password in the work item expression. This must contain the valid CA Unicenter Service Desk user and password to log in to the USD Workflow IDE product.

      Important! If USD is not using EIAM, Asset Manager must be created as a local user on the USD box (with admin privileges). Otherwise, the USD technician role will not be resolved on the local box for USM_HW_Fulfillment.

Configure the Integration from CA Unicenter Service Delivery r11

Complete the following steps in Service Delivery to configure the integration:

Note: All windows referenced in this example are part of the *Service Delivery Implementation Guide*. USM refers to Unicenter Service Management - Service Delivery. USD refers to CA Unicenter Service Desk. Finally, CA Unicenter Service Desk r11.x has been configured to run on ITIL and use eIAM authentication.

1. Install View.

2. Install Service Catalog.

3. Install Best Practices Foundation.

4. Install Workflow.

5. Create the organizations Business Unit structure (if any).

6. Log in to the CA Unicenter Service Delivery View as the user spadmin with the password spadmin.

7. (Optional). To integrate with UAPM, complete the following steps:

   a. Click the Service Builder tab, IT Support Services, Hardware, Procure New Hardware.

   b. Click the icon named define procure laptop.

   c. Click show all, select the Procure Laptop field, and click Procure Laptop.

   d. Click the column1, row 1.

   e. Click the Options tab, and select as an Asset field.

8. As the user spadmin on Service Delivery, click the Administration tab, users, and create the following users:

   Note: Set the role of catalogenduser for each user.

   > Jim Manager (jmanager)

   > Joe User (juser) assign manager to jmanager

   > Jon AssetMgr (jassetmgr)

   > Jay ITServices (jitservices)

     Note: These users are only examples for the illustrated business scenario. Every organization must create the appropriate users to meet their unique business process requirements.

9. In eIAM, assign a permanent password for each user, or log into USM with the password and set the permanent as part of the first-time login process.

10. Set up USM triggers, events, rules, and actions using the following steps:

    a. Select Administration, Tools.

    b. In the Event Type column, click Request/Subscription Item Change.

    c. Select the check box for all rules except WFAllSimpleFulfillment and WFAllUSDRequestFulfillment, and click Enable.

       Note: These rules provide a fulfillment process that is mutually exclusive from that provided by the other rules, and therefore should not be selected.

11. Click the rule WFHWCheckAvailability and edit the action WFCheckAvailability. Assign this task to the asset manager, jassetmgr, by changing the FulfillmentUser1 and FulfillmentUserList process parameter values.

12. For the integration with USD, we want to enable hardware and software fulfillment rule actions that will open a USD change order and disable the ones that notify IT services that a fulfillment task must be completed. To accomplish this, complete the following steps:

   a. Disable the WFHWFullment action by selecting the Action column check box and clicking Disable.

   b. Enable the WFHWFulfillment_USD action by clicking the Edit icon.

   c. Select Workflow from the Type drop-down list. The window will resolve itself to show the input parameters for the USM_HW_Fullment_USD process definition. We want to assign this task to the IT Services fulfiller, jitservices, by changing the FulfillmentUser1 and FulfillmentUserList process parameter values.

USM Troubleshooting

If the USM PORTS/MACHINENAME was changed, complete these steps:

1. Delete and add all of the USM actors/data schemas in USM workflow.
2. Change the row in usm_system_install.
3. Change the server.xml (there is one place to change the port).
4. Recycle View.

Testing the Integration

Based on the predefined work flow, log in to CA Unicenter Service Catalog as a user and make a request. Once the request is filled from the inventory, it is then ready for configuration by an internal technician in preparation for delivering the unit ready-for-use.

A CA Unicenter Service Desk ticket (incident, request or change, depending what you have defined in the work flow definition) should be automatically opened. If the ticket is open, verify that the integration is working from CA Unicenter Service Desk by clicking the hyperlink in the summary/description fields in the Ticket Detail window to see if CA Unicenter Service Catalog – Request can be started from CA Unicenter Service Desk.

If these testing activity work correctly, the integration is working properly. If not, you will have to troubleshoot the work flow definitions.

Integration Summary

CA Unicenter Service Management integrates with CA Unicenter Service Desk in several areas. Unicenter Service Management events can be used to automatically open CA Unicenter Service Desk tickets. Catalog request fulfillment can be configured to automatically open CA Unicenter Service Desk requests or change orders. If the Unicenter Asset Portfolio Management integration is being used, the associated assets become configuration items on the related change order. The CA Unicenter Service Desk requests and change orders can be viewed in the CA Unicenter Service Management request's Related Tickets column.

For the integration between CA Unicenter Service Catalog and CA Unicenter Service Desk to work properly, CA Unicenter Service Desk and CA Unicenter Service Management must share the MDB and eIAM. Using the integration with CA Unicenter Service Desk, when a user requests a service from the catalog, such as a standard desktop computer, and it is approved, the fulfillment process could identify the correct existing asset. Then, a CA Unicenter Service Desk change order could be opened to assign a configuration and delivery task to a technician for the asset. In addition, after a primary CA Unicenter Service Desk is established, you can use a rule action to open a CA Unicenter Service Desk ticket. You can use any of the existing event types, or you can add your own event type and use it from an external source, and then use a rule action to open a CA Unicenter Service Desk ticket.

Chapter 9: Live Automation Management – Integrating with CA SupportBridge

What Is CA SupportBridge Live Automation?

CA SupportBridge™ Live Automation is a support solution designed to prevent, detect, and repair computer problems before they cause significant disruption to operations. The technology is standards-based, scalable, and integrates effectively with existing service desk systems out-of-the-box using web services technology.

Note: For information about CA SupportBridge, see the support automation information in the *Incident and Problem Management Green Book* using the Technical Support link at http://ca.com/support.

CA SupportBridge Live Automation optimizes the technicians' contact time by automating the detection of problems, thereby reducing costs in the expensive live support process.

CA SupportBridge Live Automation assists the service desk in the following areas:

■   *Call optimization* by automating diagnosis and repair processes using Automation

■   *Call deflection* through CA SupportBridge Self Service Automation

The CA SupportBridge server is the integrated platform that enables sharing of automated tasks and support automation analytics for the CA SupportBridge products.

Integration Points and Functionality

When the CA SupportBridge Live Automation integration is enabled, the following events occur:

■   The CA SupportBridge server ensures that all incidents handled in Live Automation have an associated ticket in CA Unicenter Service Desk. If an incident does not have an associated CA Unicenter Service Desk ticket ID (when the incident originated in CA SupportBridge), one is created automatically.

■   At the conclusion of all Live Automation incidents, a link to the Live Log is created in CA Unicenter Service Desk.

■   The Live Log referenced from CA Unicenter Service Desk is stored in the CA SupportBridge server.

- The CA SupportBridge server automatically allows single sign-on by CA Unicenter Service Desk technicians and customers. If a CA Unicenter Service Desk technician is redirected to the CA SupportBridge server through the integration, the server automatically generates a CA SupportBridge user, if one does not already exist.

- Authentication is achieved using the existing CA Unicenter Service Desk BOPSID method. When users are directed from the CA Unicenter Service Desk web interface to the CA SupportBridge web interface, a BOPSID parameter is specified. This is an encoded value that the CA SupportBridge server uses to identify and authenticate the user. If a user previously unknown to CA SupportBridge is authenticated using this method, a user is automatically created in CA SupportBridge. Customers are created using the standard guest template; technicians are created based on a configured template technician.

  Note: BOPSID is a numeric token that allows CA Unicenter Service Desk users to log in seamlessly to the CA Unicenter Service Desk web interface.

- The CA Unicenter Service Desk incident shows the RefNum, not the incident number (for example, CR:576, not 576).

Integration Points from CA SupportBridge

The following are integration points from CA SupportBridge:

- Automatic incident creation in CA Unicenter Service Desk for all events managed by CA SupportBridge.

- Creation of a link to the live log in CA Unicenter Service Desk at the conclusion of all CA SupportBridge incidents.

- All incidents handled in CA SupportBridge will have an associated incident in CA Unicenter Service Desk. If an incident does not have an associated CA Unicenter Service Desk Incident ID (when the incident originated in CA SupportBridge), an incident is created automatically in CA Unicenter Service Desk.

Integration Points from CA Unicenter Service Desk

The following are integration points from CA Unicenter Service Desk:

- The ability to start, in context, CA SupportBridge Live Automation from the CA Unicenter Service Desk analyst interface. Specifically, from the following CA Unicenter Service Desk points:

  > Incident Detail window, Live Automation button in the Activities tab

  > Incident Search List results window, when right-clicking an incident number

> Incident Detail window, Activities menu option

> Incident Detail window, Activities tab, Type Column

■ The ability to start, in context, CA SupportBridge Live Automation from the CA Unicenter Service Desk user interface. Specifically, from the Chat link located in the Request Support panel on the self-service main interface.

Integration Value

The CA SupportBridge Live Automation integration provides the following value:

■ The integration helps to eliminate service desk costs through problem prevention, and reduces the time required to solve support incidents through deflection and optimization.

■ CA SupportBridge Live Automation optimizes CA Unicenter Service Desk analysts' contact time by facilitating a cost-effective channel of interaction for live support and automating detection, diagnosis, and repair in the live support process. This reduces operating costs, improves customer satisfaction, and reduces downtime. The following are the key pieces in the automation:

> Collaboration tools such as Live Chat and Desktop Sharing provide the platform for enhanced remote interaction between CA Unicenter Service Desk analysts and customers. Collaboration can occur between multiple analysts and customers. Within the collaborative mode, automated predefined responses speed the communication with the customer.

> Detailed information about a customer's computer can be gathered instantly and accurately through the use of automated tasks. Telemetry data is gathered from the customer's computer and can be seen by the service desk analyst, who can then be prompted to provide input that controls additional actions to be taken on the customer's computer.

> The Advanced Repair Tools allow the support technician to configure the customer's computer remotely. Critical operating system properties, file systems, and registry keys can be accessed and changed. Using the file explorer, remote registry, shut down, and reconnect features, CA Unicenter Service Desk analysts can copy files to and from the customer's computer, compare their registry to the customer's, change system files, and reboot the customer's computer.

■ CA SupportBridge Self Service Automation allows customers to support themselves without having to complete complicated technical steps. This automation optimizes the self-service process to ensure effective call deflection, including the following:

> The ability to link to any self-service automated task from any web-based environment using simple HTML.

> Self-service automated tasks can interact with the user and process their input.

> Self-service automated tasks can make changes to the file system, the registry, download software and install it, work with the Windows Management Instrumentation (WMI), and so on.

Note: WMI is the infrastructure for management data and operations on Windows-based operating systems. Administrators can write WMI scripts or applications to automate administrative tasks on remote computers. In addition, WMI supplies management data to other parts of the operating system and products.

How the Integration Works

The following diagram illustrates how the CA SupportBridge Live Automation integration works:
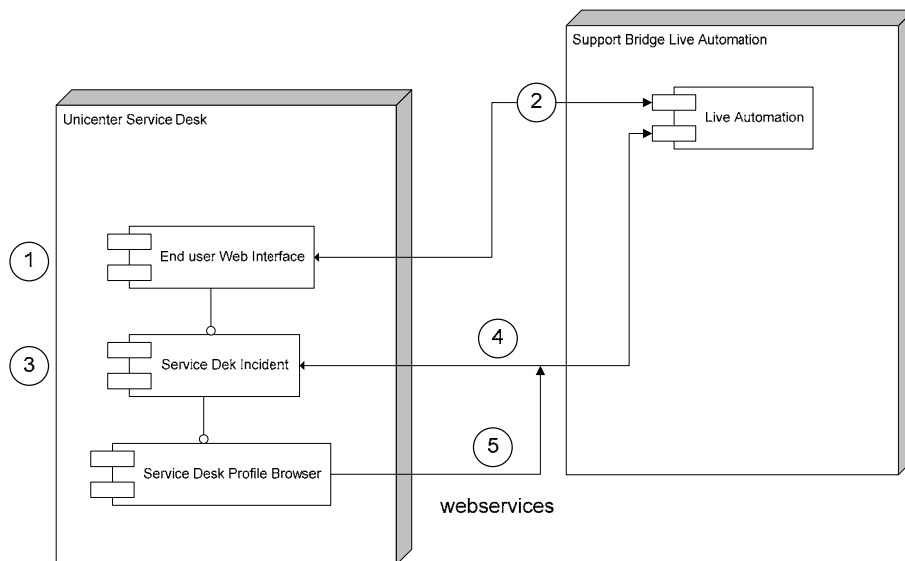
The following information applies to the previous diagram:

1. The user and technician use Live Automation as their primary interface for support.

2. The technician handles the customer in Live Automation, using established procedures. When the technician handles the customer, a CA Unicenter Service Desk ticket is created automatically and the incident number is assigned. At the conclusion of the incident, the details, including the Live Log, are sent to CA Unicenter Service Desk and are formulated as a ticket.

3. In this scenario, the customer is initially in the CA Unicenter Service Desk self-service interface and is presented with a Live Chat link (before or after creating a ticket). When the customer clicks this link, a browser window is opened, targeting a CA SupportBridge URL. The customer is authenticated and automatically put on hold in the Live Automation queue.

> **Registered User**. If the customer is logged in to CA Unicenter Service Desk as a Registered User, the customer does not have to log in to Live Automation or reenter the initial support question.

> **Anonymous User**. If the customer logged in to CA Unicenter Service Desk as an Anonymous User, the customer will have to log in through the CA SupportBridge login window, providing a username and typing the support question. The question is always required if there is no CA Unicenter Service Desk ticket.

4. Once the incident is opened, a CA Unicenter Service Desk ticket is created and assigned. If the customer has not already created a ticket, one is created automatically when handled by a technician. When the incident is closed, the Live Automation Live Log is created and appears in the CA Unicenter Service Desk interface.

5. The technician is addressing an open CA Unicenter Service Desk incident. The technician selects the Live Automation option from the Activities drop-down list. The technician clicks the Live Automation button on either the normal ticket interface or the contact browser in CA Unicenter Service Desk. CA SupportBridge authenticates the technician against CA Unicenter Service Desk and then logs the technician into Live Automation automatically. The existing Live Automation Direct Session creates a CA SupportBridge incident automatically. The technician then directs the customer to a CA SupportBridge URL and provides the Direct Session code that the customer enters to join the incident. The Direct Session code is the Live Automation incident number. When the incident is closed, the Live Automation Live Log is created and appears in the CA Unicenter Service Desk interface.

Example of the CA SupportBridge Live Automation Integration

Business Challenge

Michael Reed, Service Desk Manager at Forward, Inc., is concerned about the high cost and low customer satisfaction for the service that his staff provides to users who contact the service desk by phone. He has found the following relevant points that he would like to improve in a short timeframe:

■ The telephone is the typical way a user enters the service desk, and the user may encounter long wait times.

■ Information collected from users over the telephone is gathered in a non-standard way, so the assigned support technician normally has to call the user back to get additional information.

■ The user has to call the service desk periodically to find out the status of their request.

CA Approach

CA advises Forward, Inc. to provide web interface access to all users that use CA Unicenter Service Desk. Using the CA Unicenter Service Desk self-service web interface will allow users to solve simple issues using Knowledge Tools documents, so their level of service will improve. In addition, Forward, Inc. can integrate CA Unicenter Service Desk and CA

SupportBridge Live automation. This will allow Forward, Inc. to use Self Service Automation to optimize the self-service process and ensure effective call deflection.

Configuring a Solution

Michael Reed has coordinated the activities required to grant access to the CA Unicenter Service Desk web interface to all users. Now, users can search for solutions to their problems, and if necessary, open incidents from the CA Unicenter Service Desk web interface with more detailed information. In addition, he has asked Paul Kim, the IT Director, to provide him with technical resources to implement the CA SupportBridge Live Automation integration with CA Unicenter Service Desk. This will lay the foundation for a complete solution to provide and deliver an excellent service support experience to all users of the service desk at Forward, Inc.

Configure the Integration from CA Unicenter Service Desk R11

Complete the following steps in CA Unicenter Service Desk to configure the integration:

1. Download the following CA Unicenter ServiceDesk r11.2 patches for the CA SupportBridge integration from SupportConnect using the Technical Support link from www.ca.com/support:

    > Patch QO81159

    > Patch QO81160

2. Browse to the SupportBridge page on SupportConnect at the following location:

    http://supportconnectw.ca.com/public/supportbridge/supportbridge_supp.asp

3. Download the home.htmpl package from the SupportBridge page on SupportConnect to the following location:

    $NX_ROOT\site\mods\www\htmpl\web\employee\

    Note: The home.htmpl package contains the modified employee home page that will make the links for Self-Service Automation available.

4. Log in to CA Unicenter Service Desk as either a ServiceDesk user or an Administrator.

5. Click the Administration tab, Options Managers, Request-Change-Issue.

6. Modify the following values:

    supportbridge_division: option value = 1

    This value is the numerical representation of the Division Name in the SupportBridge database. When a division is created in SupportBridge, it is assigned a DivisionID. This value instructs CA Unicenter Service Desk to create a Live Automation session that is associated with the defined division in SupportBridge.

supportbridge_url: option value = http://ForwardInc:8082/SupportBridge

> In this sample URL, the syntax indicates to enter the web address for SupportBridge. In the Service Desk Option Manager, you can get help on the form for syntax as well. By default, the option is set as http[s]://servername:port/{web application name}

7. Install the two options managers.
8. Recycle the CA Unicenter Service Desk services.

Configure the Integration from CA SupportBridge

Complete the following steps in CA SupportBridge to configure the integration:

1. Log in to the CA SupportBridge administration interface.
2. Choose Workflow Processing Options, and click Service Desk Integration.
   The Service Desk Integration window opens.

3. Enter the following information:

   > Select the check box to enable the CA Unicenter Service Desk Integration.

   > **Service Desk Installation location**. Specify the URL to call CA Unicenter Service Desk Web Services:
   > *http://ForwardInc:8080/axis/services/USD_R11_WebService?wsdl*

   > **Note**: The default Tomcat port is 8080, but you should check first that this port is not already used by another application. You can use the following command line to check whether this port is already in use: netstat –a |more

   > **Service Desk Username**. Specify a username that CA SupportBridge will use when making web services calls into CA Unicenter Service Desk. This user will be granted permissions in CA Unicenter Service Desk to create tickets using web services.

   > **Service Desk Password**. Specify the password for the user in the Service Desk Username field.

   > **Anonymous User SD Userid**. Specify a CA Unicenter Service Desk Analyst ID for CA SupportBridge technicians who do not have a CA Unicenter Service Desk ID. You can use *anonymous*.

4. Confirm the connection to the CA Unicenter Service Desk web services by clicking Test Web Services Connectivity.
5. The CA SupportBridge server uses CA Unicenter Service Desk web services to update information in CA Unicenter Service Desk. Enter information in the following fields to determine how the user will be logged into CA SupportBridge:

   > **Use web chat for incoming customers**. Indicates that administrators can specify whether customers forwarded to CA SupportBridge from CA Unicenter Service Desk enter using Web Chat or log in directly to the full customer EXE.

Since CA Unicenter Service Desk supports customers on various browser and operating system combinations, the default for this option is No. If set to No, the customer receives the EXE download. Unless the administrator is confident all users are on Windows, we recommend that this option be left to the default Web Chat setting.

> **Service Desk Technician Template User**. Specifies a profile for CA Unicenter Service Desk analysts who start CA SupportBridge. Technicians created using this template inherit all role information from the template technician. By default, the Guest Technician in the SupportBridge database is identified by User ID "3". Change the value from "-1" to "3".

> **Default Technician password**. Specifies the password for the technician template user, which should be the Guest Technician user (user ID "3"). Change the password for the Guest Technician User using the SB Administration interface (Use Password for an analyst in CA Unicenter Service Desk).

6. Log in to CA Unicenter Service Desk as an Analyst.

7. Test the integration by opening a request or incident and clicking Live Automation on the Activities tab in the Incident/Request window.

   A window similar to the following sample should appear:
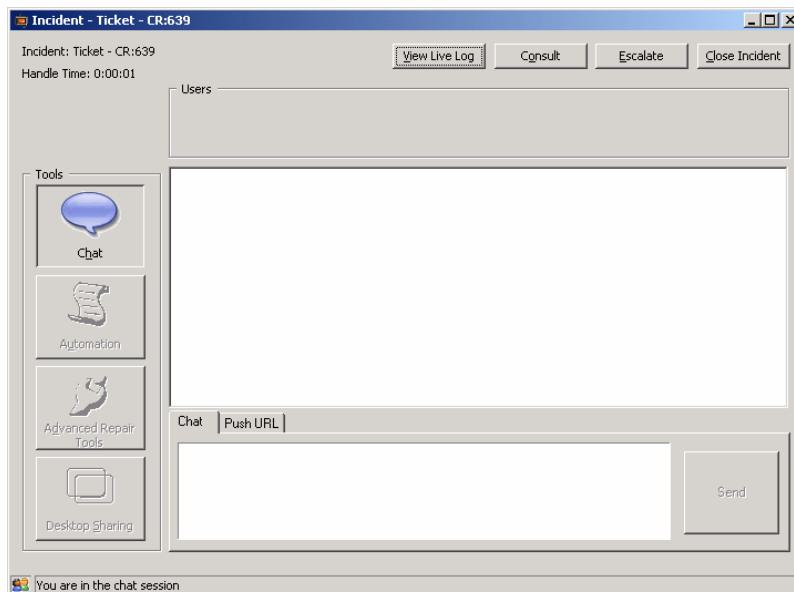
8. Log in to CA SupportBridge as a user and click the chat option in the Request Support window.

A window similar to the following sample should appear:
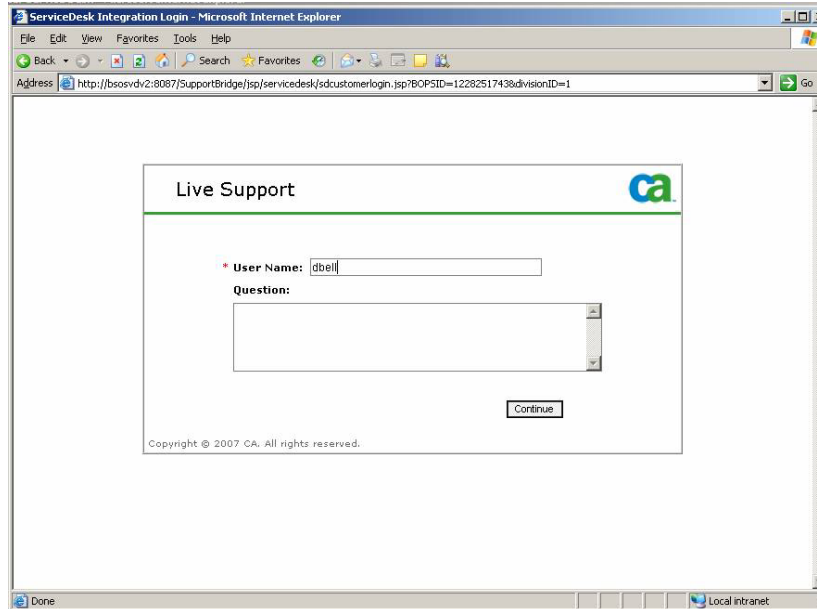
Integration Summary

The CA Unicenter Live Automation integration supports the following requirements:

1. Starting Live Automation from CA Unicenter Service Desk (from a request or the profile browser)
2. Associating Live Automation back into CA Unicenter Service Desk as an activity
3. Supporting user synch/BOP Security Identifier (BOPSID) authentication

The CA SupportBridge server ensures that all incidents handled in CA SupportBridge have an associated ticket in CA Unicenter Service Desk. If an incident does not have an associated CA Unicenter Service Desk ticket ID (when the incident originated in CA SupportBridge), an incident is created automatically. At the conclusion of all CA SupportBridge incidents, a link to the Live Log is created in CA Unicenter Service Desk.

The Live Log referenced from CA Unicenter Service Desk is stored in the CA SupportBridge server. The CA SupportBridge server automatically allows single sign-on by CA Unicenter Service Desk technicians and customers. If a CA Unicenter Service Desk technician is redirected to the CA SupportBridge server through the integration, the server automatically generates a CA SupportBridge user, if one does not already exist.

Authentication is achieved using the existing CA Unicenter Service Desk BOPSID method. When users are directed from the CA Unicenter Service Desk web interface to the CA SupportBridge web interface, a BOPSID parameter is specified. This is an encoded value that the CA SupportBridge server uses to identify and authenticate the user.

If a user previously unknown to CA SupportBridge is authenticated using this method, a user is automatically created in CA SupportBridge. Customers are created using the standard guest template; technicians are created based on a configured template technician.

The CA Unicenter Service Desk incident shows the RefNum (not the ticket number).

The integration has been developed using web services and calls to CA SupportBridge Live Automation URLs.

Chapter 10: Password Management – Integrating with CA Identity Manager

What is CA Identity Manager?

CA Identity Manager is the successor product to eTrust Admin. CA Identity Manager integrates and automates identity management services (the creation, modification and deletion of user accounts and entitlements) for enterprise systems, based on the users' relationships with the organization and their respective roles and responsibilities. The solution handles a comprehensive array of users, including employees, contractors, customers, and business partners. CA Identity Manager provides the following main features:

- **Delegated User Administration.** Delegation of user administration with centralized control lets IT organizations selectively distribute user administration tasks to those people and organizations that are best equipped to provide the services, whether they are inside or outside the enterprise. Controlled delegation of user administration can dramatically improve the scalability of IT and other organizations that are responsible for supporting users.

- **User Self Service.** CA Identity Manager provides a highly flexible browser-based administrative interface that lets users manage their own profiles, passwords, and entitlements.

- **Integrated Work flow.** Integrating work flow allows automation and enforcement of user administration processes. It provides a customizable platform to support the unique way each organization conducts its user administration for each of its user communities.

- **Password Management**. CA Identity Manager provides extensive password management services, including self-service, forgotten password support, bidirectional password synchronization, centralized password composition rules, flexible application of password policies, Graphical identification and authentication (GINA) support, and automated enforcement of periodic password changes.

  Note: GINA is the pluggable part of WinLogon that third parties may replace to customize the functionality or the user interface of the logon experience in Windows.

- **Structured Management Model.** CA Identity Manager provides a customizable yet structured model for the management of user and administrator roles and responsibilities. This model leverages key constructs such as roles, rules, groups, organizations, tasks and role-based access control (RBAC).

- **Integrated Compliance Support.** CA Identity Manager supports organizations' regulatory compliance initiatives in a number of ways. By encoding entitlement policies (including such things as segregation of duties and entitlement certification) into an enterprise solution and enforcing the associated business processes with work flow, management can have greater confidence that application and resource entitlement policies are under control. In addition, integrated auditing and reporting allows organizations to have better tracking and reporting on current user entitlements and past activities.

- **Comprehensive Support of Target Systems.** Many organizations have a diverse and complex array of client/server, mainframe, and web applications, all requiring user and entitlement management. CA Identity Manager provides out-of-the-box integration with many key systems.

- **Open Interfaces.** Identity management processes need to interoperate with many third-party systems within an organization for full value to be realized. To address this, in addition to the support of target systems previously discussed, CA Identity Manager provides a number of open interfaces such as support of SPML, a Web Service interface, and Java APIs.

- **Connector Xpress.** CA Identity Manager provides this generic utility to help reduce the cost and complexity of creating connections to custom applications. Using standard interfaces such as RDBMS and featuring a wizard, Connector Xpress is easy to use and requires no programming skills. It lets you create virtual views of accounts in your product, generate a connector metadata, and deploy the metadata.

The CA Identity Manager Integration

CA Unicenter Service Desk integrates with CA Identity Manager to support the self-service functionality that CA Unicenter Service Desk provides in its web interface. From this interface, users can submit requests, check status, and browse the knowledge base. In addition, self-service password reset capabilities are available through integration with CA Identity Manager - Password Management/Password Reset Functionality.

Integration Points and Functionality

From the CA Unicenter Service Desk user interface, users can click Reset My Password Using eTrust to start the CA Identity Manager - Reset Password Interface. The Forgotten Password Reset Admin Task is public, and requires no authentication to execute. Users are challenged with three self-authentication questions that must first be answered, either during self-registration or by modifying their user profile. After answering the questions correctly, they must change their password. The new password must meet complex requirements defined in the Password Policy. When the password is changed, it is synchronized to their correlated user accounts.

Integration Value

The CA Identity Manager integration provides the following value:

■ Gain significant cost savings by allowing users to reset and configure their passwords. CA Identity Manager provides facilities for users to choose question and answer pairs that will be used for self-authentication and subsequent password change and synchronization. Organizational password policies are enforced when the user selects a password.

■ Reduce the volume of low complexity calls, freeing up analysts for more important activities.

■ Improve security, enhance regulatory compliance and governance, and increase user satisfaction.

How the Integration Works

The following diagram illustrates how the CA Identity Manager integration works:
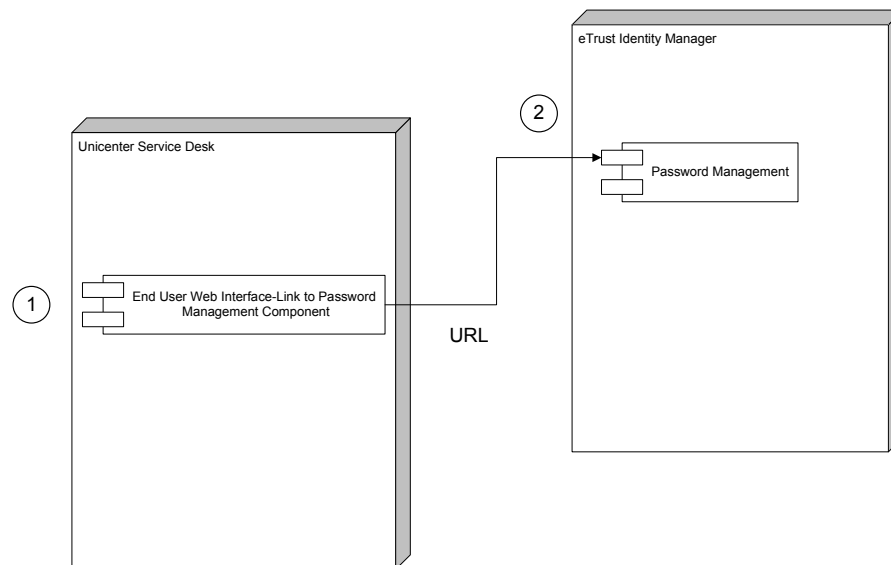
The following information applies to the previous diagram:

1. The CA Unicenter Service Desk user web interface displays a link to the CA Identity Manager-Password Management interface.

2. Clicking the URL, the CA Unicenter Service Desk user has access to password management functionality to reset their password.

Example of the CA Identity Manager Integration

Business Problem

After analyzing the Incident Reports generated by the CA Unicenter Service Desk Dashboard, Michael Reed, the Service Desk Manager for Forward, Inc., has found that 20-35 percent of all incidents fall into the Password Reset category. It appears that the most frequent occurrences are on Monday mornings, and when users return from vacation. Michael thinks this issue is affecting the productivity of his team, as resetting one password typically takes an analyst an average of 15 minutes. And, it can sometimes take more time because the operating system administrators, who have to physically reset the passwords, are involved in other critical activities, so they push password resets into a pending status. Another concern is that some offices are staffed to work 24 hours a day, 7 days a week, but the Forward, Inc. service desk only operates from 8 a.m. until midnight. Employees may not know how to proceed if they have forgotten their password and are unable to access important information and systems during the off hours. Michael decides that this problem needs to be taken care of immediately.

CA Approach

Forgotten passwords can increase the total operational cost in any organization, as users cannot do their job for extended periods of time. CA Identity Manager provides facilities for users to choose question and answer pairs that will be used for self authentication and subsequent password change and synchronization. Organizational password policies are enforced when the user selects a password. CA advised Forward, Inc. to implement the password reset functionality that is available in CA Identity Manager, and then integrate it with the CA Unicenter Service Desk self-service interface. This will allow Forward, Inc. users to reset their passwords and will provide tremendous business value for all users, particularly those whose work hours are outside of the schedule hours of the service desk.

Configuring a Solution

Michael Reed has requested the security department to provide him with the URL that allows CA Identity Manager – Self Service Password functionality to be started in context. After receiving this information, he can go through the reset password functionality configuration task.

Configure the Integration from CA Identity Manager

To configure the integration, the CA Identity Manager Administrator must configure the self-service, password reset functionality. The administrator must then provide the URL that starts the self-service password reset user interface. In this example, the URL appears as follows:

```
http://ForwardInc.domain.com:8080/idm/public/ca/index.jsp?task.tag=ForgottenP
asswordReset
```

Configure the Integration from CA Unicenter Service Desk

Complete the following steps in CA Unicenter Service Desk to configure the integration:

1. Log in to CA Unicenter Service Desk as a ServiceDesk user or an Administrator.
2. Click the Administration tab and choose Options Managers, Security, url_etrust_password_reset.
3. In the Option Value field, enter the URL from Configure the Integration from CA Identity Manager, above.
4. Click Install.
5. Click Save.
6. Recycle CA Unicenter Service Desk Services.

Test the Integration

Complete these steps to verify that the integration is working correctly:

1. Log in to the CA Unicenter Service Desk web interface as a user.
2. Click the Use eTrust Admin to reset my password link.

   The following window opens to reset the password.
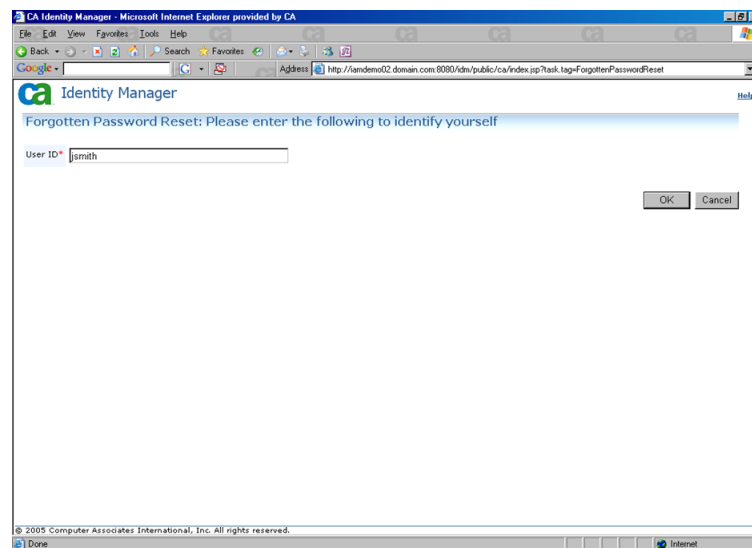
Integration Summary

CA Unicenter Service Desk's main forms for the Employee, Customer, and Guest web interfaces include an eTrust Password Reset button. When clicked, the button starts a CA Identity Manager web session so the user can answer questions to verify their identity (for example, password reset hints) and interactively reset their password.

Chapter 11: Application Change Management – Integrating with CA Software Change Manager for Distributed (formerly CA Harvest)

What is CA Software Change Manager for Distributed?

Today's development teams build large, distributed application systems. Teams work from heterogeneous platforms at remote locations and make simultaneous changes to a multitude of interrelated software modules and system documentation. The only way to effectively track this complex, enterprise-wide development activity is with a comprehensive, repository-based, web-enabled change and configuration management (CCM) solution with an open architecture.

Manual methods and simple version control systems are not robust enough to help you improve the development process and bolster service levels. CA Software Change Manager for Distributed (formerly CA Harvest) helps organizations to synchronize development activities throughout the application development life cycle on multiple platforms and across your enterprise. The product also scales to serve project teams working on your largest client/server enterprise systems, and scales down to meet the needs of individual developers.

Features

CA Software Change Manager for Distributed (CA SCM for Distributed) provides the following features:

■ Process-driven, integrated change and configuration management

■ Powerful change and defect tracking

■ Straight-forward inventory management

■ Easy version control and release management

■ Automated build management

The CA Software Change Manager for Distributed Integration

A typical scenario would involve an incident, reported to CA Unicenter Service Desk by a user. The service desk analyst determines that resolution of the incident will require some work to be performed on a software product by the development team. The software product has been defined as an application configuration item (CI) or asset within CA Unicenter Service Desk. The software product has also been defined in CA SCM for Distributed as a project. The service desk analyst who reviews the incoming incidents creates a change order that is managed in CA SCM for Distributed as a package.

Each time a service desk analyst creates a new change order with a category that invokes the application change management work flow, the approval of the work flow will automatically create a package in CA SCM for Distributed.

When a new package is created in CA SCM for Distributed, an Event Occurred comment is entered in the CA Unicenter Service Desk Change Order Activity Log. This log is visible to service desk analysts.

The integration process also provides for CA Unicenter Service Desk to send additional information about the change order to the CA SCM for Distributed Change form. This form is maintained at the package level in CA SCM for Distributed, and it contains Origination information as well as the CA Unicenter Service Desk change order number that created the package. The default Package Name assigned in CA SCM for Distributed is the CA Unicenter Service Desk change order number that created it.

This integration has been developed using the CA SCM for Distributed JHSDK (Java API) along with the CA Unicenter Service Desk Remote Reference macro and web services.

Integrations Points and Functionality

CA Unicenter Service Desk can create a new package in CA SCM for Distributed when creating a change order categorized as Project.Maint Harvest Only. The other required fields for the change order are the Order Description, Need by Date, Project, Requester, Affected End User, and Priority. Once the package is created in CA SCM for Distributed, the service desk analyst can monitor its CA SCM for Distributed life cycle using CA Unicenter Service Desk until the work is completed.

A CA SCM for Distributed package, which is a basic unit of work, is stored in CA SCM for Distributed under a *project*. There can be multiple packages associated with a project, each representing a work request for the project. The packages created from CA Unicenter Service Desk allow CA SCM for Distributed users to become aware of issues originating in CA Unicenter Service Desk related to their projects. The package is managed in CA SCM for Distributed as work is being performed, and is promoted through the planning, development, testing, and production life cycle states by developers, testers, and managers.

The CA SCM for Distributed integration also updates the associated CA Unicenter Service Desk change order's Activity Log with the work activities performed in CA SCM for Distributed, such as promoting the package to another life cycle state. This two-way

communication keeps the CA Unicenter Service Desk analyst up-to-date on any package events occurring in CA SCM for Distributed which affect the associated change order in CA Unicenter Service Desk.

Integration Points from CA Unicenter Service Desk

The following is the integration point from CA Unicenter Service Desk:

■ Automatic creation of a package in CA SCM for Distributed when a change order is approved for a Software Application CI/Asset that is also defined in CA SCM for Distributed as a project.

Integration Points from CA SCM for Distributed

The following are integration points from CA SCM for Distributed:

■ Automatic updates into the associated CA Unicenter Service Desk change order's Change Order Activity Log with the work activities performed in CA SCM for Distributed, such as promoting the package to another life cycle state.

Integration Value

The CA SCM for Distributed integration provides the following value:

■ Reduce costs and risk of errors by unifying and automating the change management life cycle.

■ Increase efficiencies and accelerate deployment by employing process-driven integration and best practices.

■ Increase productivity, due to tighter integration between teams and reduced duplication and manual data entry.

■ Faster response time due to increased operations visibility and transparency of the end-to-end change process for audit and compliance purposes.

How the Integration Works

The following diagram illustrates how the CA SCM for Distributed integration works:
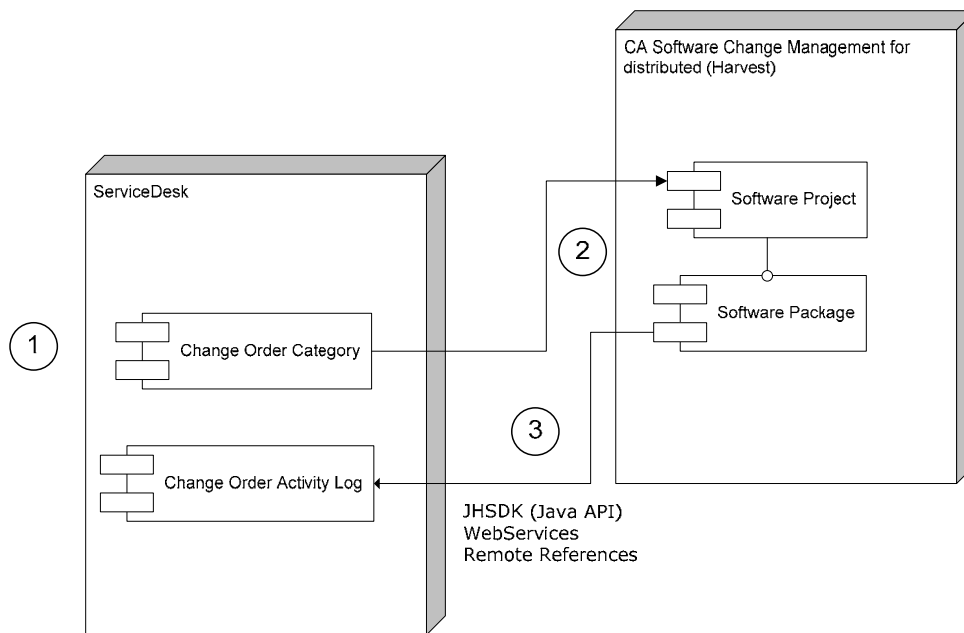
The following information applies to the previous diagram:

1. CA Unicenter Service Desk can create a new package in CA SCM for Distributed when creating a change order categorized as Project.Maint Harvest Only. The other required fields for the change order are the Order Description, Need by Date, Project, Requester, Affected End User, and Priority. This change order represents the work required to fix an incident reported by a CA Unicenter Service Desk user.

2. A package is created in CA SCM for Distributed under a CA Change Management Software Project. The service desk analyst can monitor the CA SCM for Distributed life cycle using CA Unicenter Service Desk until the work is completed.

3. The package is managed in CA SCM for Distributed as work is being performed, and is promoted through the planning, development, testing, and production life cycle states by developers, testers, and managers. The integration updates the associated CA Unicenter Service Desk change order's Activity Log with the work activities performed in CA SCM for Distributed, such as promoting the package to another life cycle state. This two-way communication keeps the CA Unicenter Service Desk analyst up-to-date on any package events occurring in CA SCM for Distributed which affect the associated change order in CA Unicenter Service Desk.

The CA Unicenter Service Desk and CA SCM for Distributed Connection

You can create a CA Unicenter Service Desk change order for application software work that needs to be managed in CA SCM for Distributed as a new package. Each change order represents the work required to fix an issue reported by a CA Unicenter Service Desk user. Once the package is created in CA SCM for Distributed, you can monitor the associated life cycle using CA Unicenter Service Desk until the work is completed.

The following steps outline this process:

1. Create a new configuration item (CI) in CA Unicenter Service Desk.

   First, you must establish an association between CA Unicenter Service Desk and the existing CA SCM for Distributed project. The service desk analyst must define the CA SCM for Distributed project in CA Unicenter Service Desk by creating a project CI.

2. Create a change order to initiate a CA SCM for Distributed package.

   The service desk analyst then creates a CA Unicenter Service Desk change order for the work that is to be performed, and associates it with the CA SCM for Distributed project that they created.

3. Initiate the CA SCM for Distributed integration process.

   The service desk analyst initiates the integration process by setting the "create Harvest package" work flow task to "Complete". Once the integration has been initiated and the package has been created for CA SCM for Distributed, developers, testers, and managers begin working on the package.

4. Monitor the CA SCM for Distributed package life cycle.

   The service desk analyst is kept informed of the work performed in CA SCM for Distributed as the package is moved through the CA SCM for Distributed life cycle states.

5. Close the change order.

   The service desk analyst is notified when the package is promoted to the production state in the life cycle. This life cycle state indicates that the package work is complete. The analyst closes the change order in CA Unicenter Service Desk.

Create a Project in CA Unicenter Service Desk

Depending on how your CA Unicenter Service Desk interface is configured, the project you create will either be a configuration item (CI) or an asset. If your CA Unicenter Service Desk has been configured with an Information Technology Infrastructure Library (ITIL) interface, the term *CI* is used. For non-ITIL users, the term *asset* is used to define components in families such as projects, hardware, software, and so on. The examples in this document use the ITIL term CI when defining the CA SCM for Distributed project in CA Unicenter Service Desk.

Before you can create a CA Unicenter Service Desk change order which will initiate and create a package in CA SCM for Distributed, the service desk analyst must create a CA Unicenter Service Desk project CI. This identifies the CA SCM for Distributed project to CA Unicenter Service Desk. This project CI must contain the CA SCM for Distributed project name. The CA Unicenter Service Desk project CI object that you create here will be used later in the Project field when you create the change order. You only need to create this project CI once for each unique CA SCM for Distributed project you want to identify to CA Unicenter Service Desk.

To create a CA Unicenter Service Desk Project CI

1. Start CA Unicenter Service Desk.
2. Select File, New Configuration Item.

   The Create New Configuration Item window opens.

3. Enter the following information:

   > **Name**. Enter the name of the CA Unicenter Service Desk project CI.

   > **Note:** As best practice, use the CA SCM for Distributed project name as the name of the CA Unicenter Service Desk project CI so that CA Unicenter Service Desk users can easily identify the relationship between the CA Unicenter Service Desk project CI and the project in CA SCM for Distributed.

   > **Class**. Enter *ECM Project*. The Enterprise Change Management (ECM) Project class is a unique class that identifies the CA Unicenter Service Desk project CIs that can be used for integration.

   > Enter any additional data fields that you require. Make sure to enter some descriptive notes about the project.

4. Click Continue.

   The following detail window opens so you can finish creating the CA Unicenter Service Desk project CI:

5. At the bottom of the window, on the Project tab, enter the name of the CA SCM for Distributed project and the name of the CA SCM for Distributed server.

   Important! The project name *must* match the CA SCM for Distributed project as defined in CA SCM for Distributed using the life cycle template. If you do not know the name of the CA SCM for Distributed project, contact your CA SCM for Distributed manager. In addition, for the integration to work you *must* have the broker name. The CA SCM for Distributed server name should be the broker name value. On occasion, it is possible to assign the broker name a different value than the name of the server where CA SCM for Distributed is installed, but this is not recommended.

6. Click Save.

   The CA Unicenter Service Desk project CI is saved, and a message displays at the top of the window indicating it was saved.

Create a Change Order to Initiate a CA SCM for Distributed Package

The CA Unicenter Service Desk analyst creates a change order to identify the work to be performed. This information is transferred to CA SCM for Distributed in the package created under the associated CA SCM for Distributed project.

To create a change order

1. Start CA Unicenter Service Desk.

2. Select File, New Change Order.

   The Create New Change window opens.

3. Enter the following required fields:

> **Order Description**. Enter a detailed description of the change order or request.

   Note: On search windows that display this field, enter the first characters in the field and click Search.

> **Need By Date**. Enter the date and time specified in the CA Unicenter Service Desk change order by which you want the change order closed and resolved.

> **Project**. CA Unicenter Service Desk's project CI or asset that was set up to reference the project in the external system, such as CA SCM for Distributed or Clarity.

   Note: You must define the external project to CA Unicenter Service Desk before you can initialize integration with the external project. This field is then used in the change order that you create to link it to the project in the external system.

> **Requester**. The name of the person initiating the change order. This person must be a defined contact in CA Unicenter Service Desk.

> **Affected End User**. The name of the person affected by the change order. This may be the same person as the Requester. This person must be a defined contact in CA Unicenter Service Desk.

> **Priority**. Shows how much attention a ticket should receive. Your system administrator can modify the default priority codes, so they can vary from one installation to another.

   Note: In search areas and on windows where you can edit this field, choose the value you want to use from the associated drop-down list.

4. Enter any other data fields that you need, such as Order Summary.

5. Use the link above the field to enter *Project.Maint Harvest Only* as the required category, which you can select from the Category Selection window under the Project category. This is the predefined category you must use for the CA SCM for Distributed integration. It has three work flow tasks associated with it.

   Note: Customers have the option to change the names for these categories, or even create new ones, that can invoke the same remote reference macros.

6. Enter the required Project field, which should be the CA Unicenter Service Desk project CI that you previously created. This field identifies the CA SCM for Distributed Project to CA Unicenter Service Desk and associates the change order you create with the CA SCM for Distributed project. Use the link above the field to search for the project.

7. Click Save.

   The change order is created.

Initiate the CA SCM for Distributed Integration Process

By default, CA Unicenter Service Desk includes categories that have been defined and associated with the appropriate work flow tasks for the CA SCM for Distributed and Clarity integration. The change order you created has three predefined work flow tasks attached. The following work flow tasks are associated with the Project.Maint Harvest Only category used in the change order:

■ 100 Create Harvest Package

■ 200 Monitor Harvest Package

■ 300 Notify Assignee

Note: You can also create your own categories and work flows if necessary. CA Unicenter Service Desk analysts can view work flow tasks on the Workflow Tasks tab in the Change Order Detail window:

To initiate the CA SCM for Distributed integration process

1. Start CA Unicenter Service Desk.
2. Open the change order record, and click the Workflow Tasks tab.
3. Select the *100 Create Harvest Package* work flow by clicking the work flow number.

   The Change Workflow Detail window opens.

4. Click Edit.

   The Update Change Workflow window opens.

5. Select Complete from the Status field drop-down list.
6. Click Save.

The Create Harvest Package integration macro is started. The work flow is saved, the status is changed to Complete, and a CA SCM for Distributed package is created.

When the package is successfully created, CA SCM for Distributed changes the status of the 200 Monitor Harvest Package work flow to In Progress and logs an Event Occurred comment in the CA Unicenter Service Desk Change Order Activity Log List.

Note: If you did not enter all the required fields in the change order, the status of the Create Harvest Package work flow is set to Failed, and a log comment is entered in the CA Unicenter Service Desk Change Order Activity Log List detailing the integration failure. To reinitiate the integration process, enter the missing required fields and save the change order. Then, change the status of the Create Harvest Package work flow task back to Complete.

Monitor the CA SCM for Distributed Package Life Cycle

Once you have initiated the CA SCM for Distributed integration and the package has been created, the typical package life cycle involves the following tasks:

■ The CA SCM for Distributed manager is responsible for manually assigning a resource to the package. The manager can review the package and promote the package to the development life cycle state. A log comment is entered in the associated CA Unicenter Service Desk Change Order Activity Log List.

■ A developer works on the package fix and promotes the package to the test life cycle state. A log comment is entered in the associated CA Unicenter Service Desk Change Order Activity Log List.

■ A QA tester tests the package fix and passes the fix. An Operations Manager promotes the fix to the production life cycle state. A log comment is entered in the associated CA Unicenter Service Desk Change Order Activity Log List, and the status of the 200 Monitor Harvest Package Workflow is set to Complete.

■ Information from the CA Unicenter Service Desk change order that created the package is maintained in CA SCM for Distributed at the package level. For example, you can view the change order number from the Service Desk Clarity Connector Request for Change form, as illustrated in the following sample window:
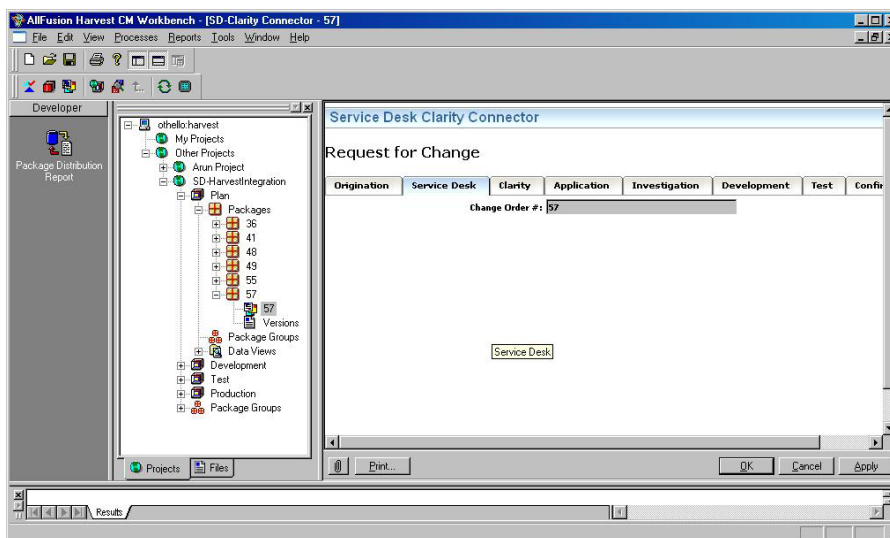
The Origination tab on the Service Desk Clarity Connector Request for Change form contains data that was transferred from the CA Unicenter Service Desk change order that created the package, as illustrated in the following sample window:
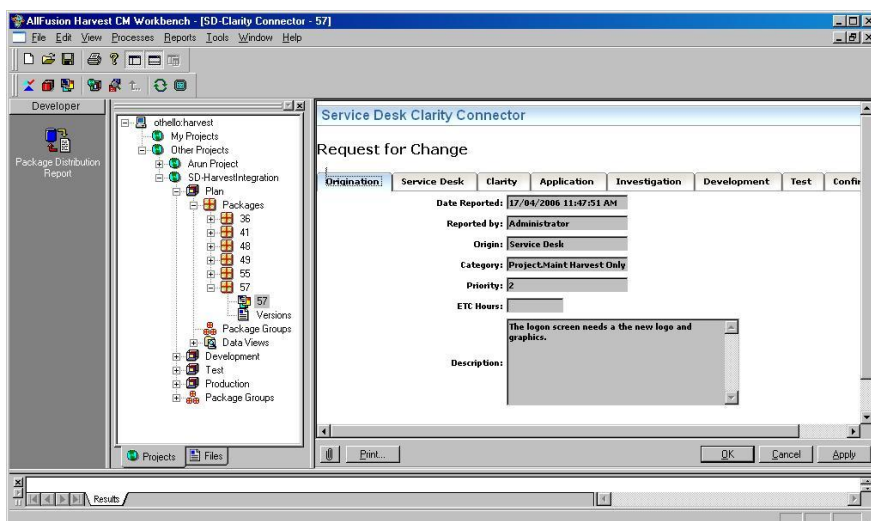
Close the Change Order

In CA SCM for Distributed, when the package is promoted to the production life cycle state, CA Unicenter Service Desk automatically sets the status of the 300 Notify Assignee work

flow task to Pending. After the notification is sent back to the CA Unicenter Service Desk change order's Activity Log List, the status of the work flow is automatically changed to Complete. The service desk assignee will then update and close the change order and notify the service desk user.

User Actions and Service Desk's Change Order Activity Log List

The following table lists the user actions that trigger comments to be entered in CA Unicenter Service Desk's Change Order Activity Log List:

User Action	Product Action	Activity Log Type
In CA Unicenter Service Desk, set the status of the Create Harvest Package work flow task to Complete on a CA Unicenter Service Desk change order.	A new package is created in CA SCM for Distributed under the designated project. The default package name is the change order number that initiated it.	An Event Occurred comment is logged in the Change Order Activity Log List.
A CA Unicenter Service Desk Change Order is entered with missing required field information, and the status of the Create Harvest Package work flow task is set to Complete.	A new package is *not* created in CA SCM for Distributed. The work flow task fails.	An Event Occurred comment is logged in the Change Order Activity Log List indicating that the create package failed.
A CA SCM for Distributed package is promoted or demoted by the CA SCM for Distributed manager.	In CA SCM for Distributed, the package is moved between the plan, development, test, and production life cycle states.	A log comment is entered in the Change Order Activity Log List. When a package is promoted to the production life cycle state, a comment is logged, and the status of the Notify Assignee work flow task is set to Complete.

Note: If a CA SCM for Distributed package is deleted, you must manually update CA Unicenter Service Desk to change the work flow task. You must also manually close the associated change order. In addition, if the status of the Create Harvest Package status changes to Failed, for example, due to missing data, you can manually restart the work flow task by changing its status back to Complete.

The CA SCM for Distributed package status is tracked in CA Unicenter Service Desk and is logged to the originating change order's Change Order Activity Log List.

In the following sample window, the list displays the initial change order creation, the Create Harvest Package work flow event, and promotion of the package in CA SCM for Distributed.
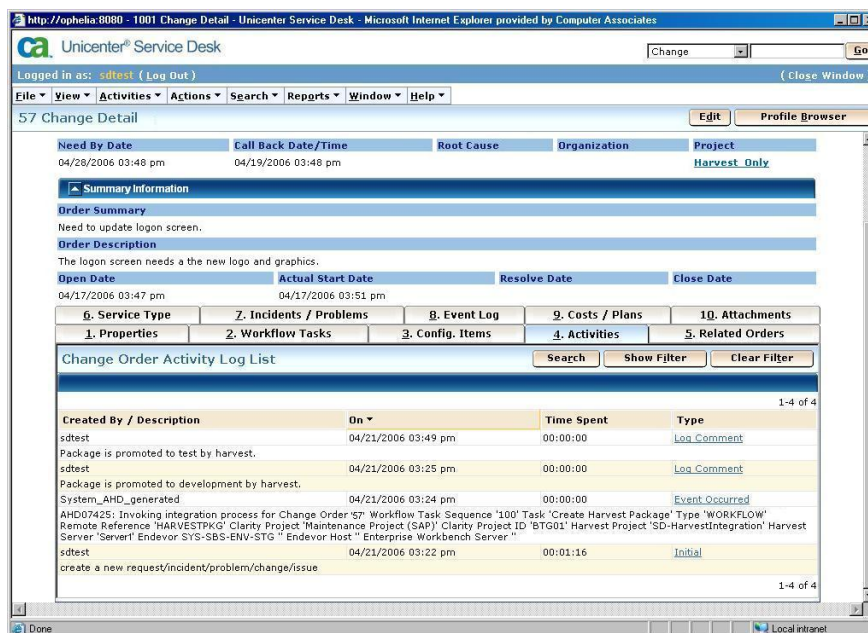
Integration Summary

CA Unicenter Service Desk can create a new package in CA SCM for Distributed when creating a change order categorized as Project.Maint Harvest Only. Once the package is created, the service desk analyst can monitor the life cycle in CA SCM for Distributed using CA Unicenter Service Desk until the work is completed.

A package, which is a basic unit of work, is stored in CA SCM for Distributed under a CA SCM for Distributed project. There can be multiple packages associated with a project, each representing a work request for the project. The packages created from CA Unicenter Service Desk allow CA SCM for Distributed users to become aware of issues originating in CA Unicenter Service Desk related to their projects. The package is managed in CA SCM for Distributed as work is being performed, and is promoted through the planning, development, testing, and production life cycle states by developers, testers, and managers.

The CA SCM for Distributed integration also updates the associated CA Unicenter Service Desk change order's Activity Log with the work activities performed in CA SCM for Distributed, such as promoting the package to another life cycle state. This two-way communication keeps the CA Unicenter Service Desk analyst up-to-date on any package events occurring in CA SCM for Distributed which affect the associated change order in CA Unicenter Service Desk.

Chapter 12: Portfolio Project Management – Integrating with CA Clarity

What is CA Clarity?

CA Clarity, the industry-leading IT Governance solution, enables IT organizations to achieve world-class performance by improving the quality of their engagements with the business and enhancing their ability to run at peak efficiency. The CA Clarity features integrate portfolio planning, demand management, project management, resource planning, and time and cost management.

Features

CA Clarity provides the following features:

■ Clarity Portfolio Manager provides a structured environment for deciding which projects, programs, or initiatives to fund, which to sustain, and which to cancel.

■ Single, integrated system provides easy access to any portfolio type.

■ Out-of-the box metrics offer flexibility and accuracy when measuring investment evaluations.

■ Identify best business alternatives with multiple "what if" scenarios.

■ Real-time investment status allows faster response to obstacles.

■ Optimize the portfolio with efficient frontier graphical analysis.

The CA Clarity Integration

Service desk analysts use CA Unicenter Service Desk to manage and evaluate change orders. When CA Unicenter Service Desk is integrated with CA Clarity, an additional Project field is available on the CA Unicenter Service Desk Change Order Detail window. In addition, new change order categories that are associated to predefined work flows have been added as part of the integration. Your CA Unicenter Service Desk administrator can define additional categories and work flows using these out-of-the-box components.

If a change order in CA Unicenter Service Desk represents work to be tracked and managed on an existing CA Clarity project, the CA Unicenter Service Desk and CA Clarity integration can create CA Clarity Change Order Tasks, CA Clarity Incidents, or CA Clarity Ideas by selecting the appropriate category and completing the required information on the CA

Unicenter Service Desk change order (Order Description, Description, Project, Need By Date, Requester, and Affected End User). When CA Clarity creates and updates a work flow task for the project, it updates CA Unicenter Service Desk by logging a comment in the CA Unicenter Service Desk Change Order Activity Log.

In this integration, CA Unicenter Service Desk uses a CA Clarity Remote Reference call to send data to CA Clarity using its XOG API. CA Clarity, in turn, updates CA Unicenter Service Desk using its web services.

Integration Points and Functionality

CA Unicenter Service Desk can create the following CA Clarity elements from a CA Unicenter Service Desk Change Order:

■   CA Clarity Change Order Task

    Service desk analysts manage and evaluate CA Unicenter Service Desk change orders. When integrated with CA Clarity, an additional Project field is available on the CA Unicenter Service Desk Change Order Detail window. For the CA Clarity integration, CA Unicenter Service Desk includes predefined change order categories that are associated with predefined work flows able to trigger the integration.

    If a CA Unicenter Service Desk change order represents work to be tracked and managed on an existing CA Clarity project, selecting the category field Project.Maint Clarity Only on the Change Order Detail window creates a new change order task for the corresponding CA Clarity project. When that task is created, CA Clarity sends an update back to CA Unicenter Service Desk that adds a log comment to the CA Unicenter Service Desk change order Activity Log.

■   Clarity Incidents

    A service desk analyst typically initiates a CA Clarity incident once the change order representing the demand has been analyzed and it has been determined that a CA Clarity incident report should be created. This process is invoked by selecting Project.Other Maint Work in the category field of the Change Order Detail window.

    Once created, the CA Clarity incident can remain as a CA Clarity incident, or the CA Clarity change manager can convert the incident to a CA Clarity project.

    Note: For information about CA Clarity incidents and how to convert them to projects, see the *CA Clarity Using Demand Management Guide*.

    When a CA Clarity Incident is created using this integration, CA Clarity in turn sends an update back to CA Unicenter Service Desk. This update is entered as a log comment in the CA Unicenter Service Desk Change Order Activity Log List.

■   Create CA Clarity Ideas from CA Unicenter Service Desk

A service desk analyst typically initiates a CA Clarity idea once the change order representing the demand has been analyzed and it has been determined that a CA Clarity idea should be created. This process is invoked by selecting Project.New System Development in the category field of the Change Order Detail window.

The CA Clarity idea can remain as an idea, or the CA Clarity demand manager can convert the idea to a project.

Note: For information about CA Clarity ideas and how to convert them to projects, see the *CA Clarity Using Demand Management Guide*.

When a CA Clarity Idea is created using this integration, CA Clarity in turn sends an update back to CA Unicenter Service Desk. This update is entered as a log comment in the CA Unicenter Service Desk Change Order Activity Log List.

Integration Points from CA Unicenter Service Desk

The following are integration points from CA Unicenter Service Desk:

- Automatic creation of CA Clarity Change Order Tasks, CA Clarity Incidents, and CA Clarity Ideas from the CA Unicenter Service Desk Change Order Detail window, by selecting the correct CA Unicenter Service Desk change order category.

- Status updates to the correct work flow tasks associated with the selected category.

Integration Points from CA Clarity

When CA Unicenter Service Desk is integrated with CA Clarity, certain CA Clarity actions trigger an update to be sent from CA Clarity to CA Unicenter Service Desk. These updates are logged as comments on the CA Unicenter Service Desk change order Activity Log.

The following CA Clarity actions trigger comments to be logged on the CA Unicenter Service Desk Change Order Activity Log:

- Converting CA Clarity incidents to projects

- Converting CA Clarity ideas to projects

- Marking CA Clarity change order tasks as Scheduled

- Creating CA SCM for Distributed packages from CA Clarity change order tasks

- Canceling CA Clarity change order tasks

- Marking CA Clarity change order tasks as Complete

- Marking CA Clarity projects as Complete

Integration Value

The CA Clarity integration provides the following value:

- Ensure the most efficient and cost effective use of IT resources by providing complete cost and resource visibility for the change process.

- Gain greater alignment between your business units and the teams who are responsible for application maintenance and operations.

- Integrate change management processes throughout the change life cycle, from initiation through closure.

- Eliminate manual and error-prone double entry of information.

How the Integration Works

The following diagram illustrates how the CA Clarity integration works:
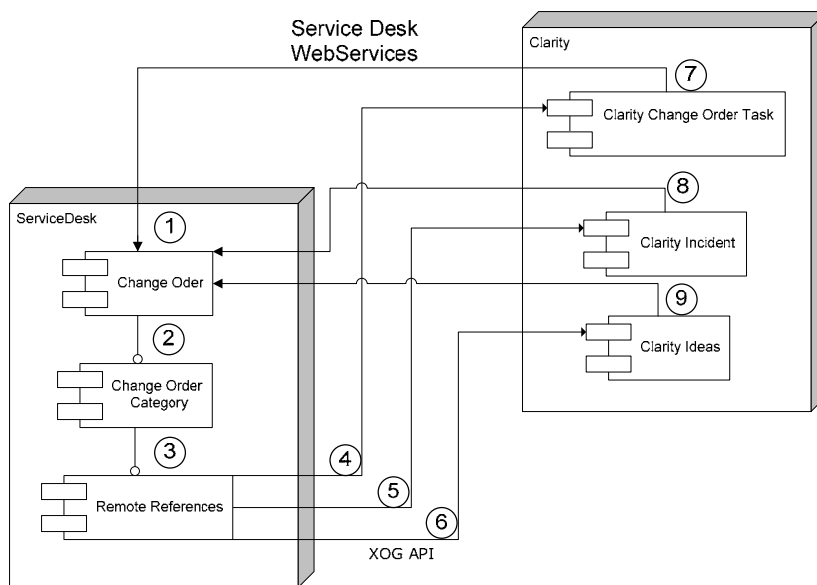
The following information applies to the previous diagram:

1. A change order is created in CA Unicenter Service Desk. When CA Unicenter Service Desk is integrated with CA Clarity, an additional Project field is available on the CA Unicenter Service Desk Change Order form.

2. CA Unicenter Service Desk includes predefined change order categories that are associated with predefined work flows. Your service desk administrator can define additional categories and work flows using these out-of-the-box components. If a CA Unicenter Service Desk change order represents work to be tracked and managed on an existing CA Clarity project, select the appropriate category.

3. Based on the Change Order selected category, CA Unicenter Service Desk creates one of the following:

    > A CA Clarity Change Order

> A CA Clarity Incident

> CA Clarity Ideas by using Remote Reference functionality to call and send data to CA Clarity using its XOG Clarity API.

4. When CA Unicenter Service Desk is integrated with CA Clarity, certain CA Clarity actions trigger an update to be sent from CA Clarity to CA Unicenter Service Desk using CA Unicenter Service Desk's web services. These updates are logged as comments on CA Unicenter Service Desk's Change Order Activity Log.

The following CA Clarity actions trigger comments to be logged on the CA Unicenter Service Desk Change Order Activity Log:

> Converting CA Clarity incidents to projects

> Converting CA Clarity ideas to projects

> Marking CA Clarity change order tasks as Scheduled

> Creating CA SCM for Distributed packages from CA Clarity change order tasks

> Canceling CA Clarity change order tasks

> Marking CA Clarity change order tasks as Complete

> Marking CA Clarity projects as Complete

Note: In CA Unicenter Service Desk, you can view the Change Order Activity Log List from the Activities tab of the Change Order Detail window. For information, see the CA Unicenter Service Desk online help.

Example of the CA Clarity Integration

This example describes the integration between CA Unicenter Service Desk and CA Clarity. The example describes how to create CA Clarity change order tasks using CA Unicenter Service Desk, how to create CA Clarity incidents and ideas, and how to monitor the integration progress using CA Unicenter Service Desk's Change Order Activity Log.

Create CA Clarity Change Orders

Complete the following steps to create CA Clarity change orders:

1. Open a CA Unicenter Service Desk change order.

2. At the Category field, choose Project.Maint Clarity Only from the drop-down list.

3. Complete the following required fields:
   > **Description**. The Order Description from the original CA Unicenter Service Desk change order that created the CA Clarity task.

   > **Need By Date**. The date and time by which you want the CA Unicenter Service Desk change order resolved and closed.

> **Project**. CA Unicenter Service Desk's asset or project CI that was set up to reference the project in the external system, such as CA SCM for Distributed or CA Clarity.

You must define the external project to CA Unicenter Service Desk before you can initialize integration with the external project. This field is then used in the change order that you create to link it to the project in the external system.

> **Requester**. The name of the person initiating the change order. This person must be a defined contact in CA Unicenter Service Desk.

> **Affected End User**. The name of the person affected by the change order. This may be the same person as the Requester. This person must be a defined contact in CA Unicenter Service Desk.

4. Click Save.

5. Change the CA Unicenter Service Desk change order's Evaluate Clarity Task work flow task status from Pending to Evaluate.

   The work flow invokes the integration process between CA Unicenter Service Desk and CA Clarity, and the CA Clarity change order task is created on an existing CA Clarity project.

   Note: Changes to work flow tasks will not display until the ticket is saved.

6. Click Save.

Create CA Clarity Incidents from CA Unicenter Service Desk

Service desk analysts typically initiate CA Clarity incidents once the change order representing the demand has been analyzed and it has been determined that the change order needs to be managed in CA Clarity. The CA Clarity incident can remain as an incident, or the CA Clarity change manager can convert the incident to a project.

Note: For information about incidents and how to convert them to projects, see the *CA Clarity Using Demand Management Guide*. For information about CA Unicenter Service Desk work flows, see the *CA Unicenter Service Desk Administrator Guide* or CA Unicenter Service Desk online help.

To create a CA Clarity incident from CA Unicenter Service Desk

1. In CA Unicenter Service Desk, open a change order and click Category.

2. Select the Project.Other Main Work change order category.

3. Complete the following required fields:

   > **Order Description**. A detailed description of this change order or request.

   Note: On search windows that display this field, enter the first characters in the field and click Search.

   > **Category**. Displays the type of change order to which this task is associated.

> **Description**. The Order Description from the original CA Unicenter Service Desk change order that created the CA Clarity task.

> **Need By Date**. The date and time specified in the CA Unicenter Service Desk change order by which you want the ticket closed and resolved.

> **Requester**. The name of the person initiating the change order. This person must be a defined contact in CA Unicenter Service Desk.

> **Affected End User**. The name of the person affected by the change order. This may be the same person as the Requester. This person must be a defined contact in CA Unicenter Service Desk.

Create CA Clarity Ideas from CA Unicenter Service Desk

Service desk analysts typically initiate CA Clarity ideas once the change order representing the new application or product idea request has been analyzed and it has been determined that the change order will be managed in CA Clarity. The CA Clarity idea can remain as an idea, or the CA Clarity demand manager can convert the idea to a project.

Note: For information about CA Clarity ideas and how to convert them to projects, see the *CA Clarity Using Demand Management Guide*. For information about CA Unicenter Service Desk work flows, see *the CA Unicenter Service Desk Administrator Guide* or CA Unicenter Service Desk online help.

To create a CA Clarity idea from CA Unicenter Service Desk

1. In CA Unicenter Service Desk, open a change order and click Category.
2. Select the Project.New System Development change order category.
3. Complete the following required fields:

> **Order Description**. A detailed description of the change order or request.

> **Note:** On search windows that display this field, enter the first characters in the field and click Search.

> **Category**. Displays the type of change order to which this task is associated.

> **Description**. The Order Description from the original CA Unicenter Service Desk change order that created the CA Clarity task.

> **Need By Date**. The date and time specified in the CA Unicenter Service Desk change order by which you want the ticket closed and resolved.

> **Requester**. The name of the person initiating the change order. This person must be a defined contact in CA Unicenter Service Desk.

> **Affected End User**. The name of the person affected by the change order. This may be the same person as the Requester. This person must be a defined contact in CA Unicenter Service Desk.

4. Click Save.

5. From the Workflow Tasks tab, click the sequence number associated with the Create Clarity Idea work flow.

   The Change Workflow Detail window opens.

6. Change the CA Unicenter Service Desk change order's Create Clarity Idea work flow task status from Pending to Complete.

   Note: Changes to work flow tasks will not display until the ticket is saved.

7. Click Save.

Integration Summary

Service desk analysts use basic CA Unicenter Service Desk functionality to manage and evaluate change orders. When integrated with CA Clarity, an additional Project field is available on the CA Unicenter Service Desk Change Order form. CA Unicenter Service Desk includes predefined change order categories that are associated with predefined work flows. Your service desk administrator can define additional categories and work flows using these out-of-the-box components.

If a CA Unicenter Service Desk change order represents work to be tracked and managed on an existing CA Clarity project, select the appropriate category. Based on the selected category, a CA Clarity Change Order, a CA Clarity Incident, or a CA Clarity Idea is created from CA Unicenter Service Desk using Remote Reference calls using the XOG Clarity API.

Some activities in CA Clarity can update a CA Unicenter Service Desk Change order in the Change Order Activity log. This update is automatically performed using CA Unicenter Service Desk's web services API.

All the integration from CA Unicenter Service Desk to CA Clarity happens using the XOG Clarity API. The order of the integration is as follows:

■ A change order is approved in CA Unicenter Service Desk.

■ CA Unicenter Service Desk initiates the remote reference macro passing all the change order information.

■ The remote reference macro invokes a CA Clarity process using XOG.

■ The CA Clarity process analyzes the information sent by CA Unicenter Service Desk and creates an idea, incident, or task as appropriate.

Chapter 13: Application Change Management – CA Change Management Integration (CMI)

What Is CMI?

CA Change Management Integration (CMI) is the integration of CA Unicenter Service Desk with CA Clarity and CA Software Change Manager for Distributed (formerly CA Harvest). It comprises a composite system for end-to-end change management that automates the entire change life cycle - from change request creation, prioritization, impact assessment, review and approval, through project management, resource allocation, software release and deployment. By automating manual activities and integrating formerly disparate processes, this system helps IT organizations increase service levels and ensure the most efficient and cost effective use of IT resources. It also helps to enforce quality-based standards, application development methodologies, and best practices such as ITIL and CMMI.

The CMI Integration

The primary objective of the CMI integration is to allow the three CA solutions, CA Unicenter Service Desk, CA Clarity, and CA Software Change Manager for Distributed (CA SCM for Distributed), to be process-aware and to interact with each other. With this integration, you can transfer CA Unicenter Service Desk change orders to CA Clarity as tasks on an existing project, incident, or idea, where they can either be costed with effort and charged back to the customer, or transformed into project tasks and projects, if necessary. Once transferred to CA Clarity, the work can be tracked in CA SCM for Distributed packages, and the status is communicated to the CA Unicenter Service Desk managers and users.

Integration Points and Functionality

CA Unicenter Service Desk provides a centralized source for change orders generated from incident or problem management. Service desk analysts initiate, categorize, and approve change order work flow tasks, converting them into a task on an existing CA Clarity project. CA Clarity provides a centralized source for the management of changes to existing solutions, while CA SCM for Distributed provides the mechanism for implementing changes related to software development. CA Clarity project managers can create CA SCM for Distributed packages on an existing project in CA SCM for Distributed from CA Clarity change order tasks.

You can also manage change orders that result in software fixes outside of CA Clarity using only CA Unicenter Service Desk and CA SCM for Distributed.

The following scenarios illustrate how to manage change orders using a combination of CA solutions (CA Unicenter Service Desk, CA Clarity, and CA SCM for Distributed) as well as how to manage other maintenance work.

Integration Value

By using the CMI integration, you will gain greater alignment between teams who are responsible for application maintenance and operations, and your business units. Additional benefits include the following:

■ Increased productivity, due to tighter integration between teams and reduced duplication and manual data entry.

■ Faster response time, due to increased operations visibility.

■ Transparency of the end-to-end change process for audit and compliance purposes.

Managing Change Orders Using CA Unicenter Service Desk, CA Clarity, and CA SCM for Distributed

The following scenario incorporates incident and request management in CA Unicenter Service Desk, project (resource and work management) in CA Clarity, and software change management in CA SCM for Distributed. When a CA Unicenter Service Desk change order is transitioned to CA Clarity, it is managed as a CA Clarity change order task on an existing project used to model maintenance activities. CA Unicenter Service Desk analysts can also transition change orders to CA SCM for Distributed as a new package in a CA SCM for Distributed project where the CA SCM for Distributed manager can manage the new product development.

How a Change Order Moves through CA Solutions

The following information describes the flow of a change order from CA Unicenter Service Desk through the default life cycle in CA Clarity and CA SCM for Distributed.

1. In CA Unicenter Service Desk, the user logs and submits a software request.
2. The CA Unicenter Service Desk analyst completes the following steps:
   > Reviews the request and determines whether it should be managed as a possible software problem in CA Clarity.

   > Converts the request to a change order representing the demand for a software fix to be managed in CA Clarity.

   > Creates a CA Clarity change order task on an existing CA Clarity project.

3. In CA Clarity, the CA Clarity project manager completes the following steps:
   > Considers the CA Clarity change order task's resource capacity, and then schedules the task.

> Creates the CA SCM for Distributed package.

> Monitors task activity and progress using CA Clarity.

4. In CA SCM for Distributed, the CA SCM for Distributed manager completes the following steps:

> The manager reviews the resource assignment for the CA SCM for Distributed package.

> The assigned resource makes the requested software change and promotes or demotes the software fix through the life cycle states.

> When the software change is approved for production, the package is promoted to the production (PROD) life cycle state.

5. In CA Clarity, the CA Clarity project manager marks the task as 100 percent complete.

6. In CA Unicenter Service Desk, the assignee completes the following steps:

> Updates and closes the CA Unicenter Service Desk change order.

> Notifies the service desk end-user.

Managing Change Orders Using CA Unicenter Service Desk and CA Clarity

The following scenario incorporates request and change order management in CA Unicenter Service Desk with resource and work management in CA Clarity. In the following scenario, a CA Unicenter Service Desk change order is converted into a CA Clarity change order task on an existing CA Clarity project.

How a Change Order Moves through CA Solutions

The following information describes the flow of a change order from CA Unicenter Service Desk through the default life cycle in CA Clarity.

1. In CA Unicenter Service Desk, the user enters a software request.

2. The CA Unicenter Service Desk analyst completes the following steps:

> Reviews the request and determines its disposition.

> Converts the request to a change order representing the demand for a software fix to be managed in CA Clarity.

> Creates a CA Clarity change order task that will be tracked on an existing CA Clarity project.

3. In CA Clarity, the CA Clarity project manager completes the following steps:

> Evaluates the CA Clarity change order task's resource capacity, and then schedules the task.

> Monitors task activity and progress using CA Clarity.

> Marks the task as 100 percent complete.

4. In CA Unicenter Service Desk, the assignee completes the following steps:

> Updates and closes the CA Unicenter Service Desk change order.

> Notifies the CA Unicenter Service Desk user.

New Product Management Flow

CA Unicenter Service Desk change orders representing the demand for a new application or product are sent to CA Clarity as ideas. The CA Clarity demand manager then reviews and converts the CA Clarity idea into a new project.

Note: For more information about CA Clarity incidents and ideas, see the *CA Clarity Using Demand Management Guide*.

Project tasks are linked to CA SCM for Distributed projects and packages through their association to the CA SCM for Distributed project in CA Clarity. You can manage the new product management process using CA Clarity, and manage the software development activity using CA SCM for Distributed. Real-time progress, percentage complete values and life cycle state changes, is reported in CA Clarity.

In the following scenarios, you will learn how to manage change orders using a combination of these CA solutions: CA Unicenter Service Desk, CA Clarity, and CA SCM for Distributed.

Managing New Products Using CA Unicenter Service Desk, CA Clarity, and CA SCM for Distributed (Ideas)

The following scenario incorporates request and change order management in CA Unicenter Service Desk, resource and work management in CA Clarity, and software change management in CA SCM for Distributed. In this scenario, a CA Unicenter Service Desk change order is sent to CA Clarity as a CA Clarity idea, which is later converted into a project, and the software work is managed as a package in a CA SCM for Distributed project.

How a New Product Moves through CA Solutions

The following information describes the flow of a new product from CA Unicenter Service Desk through the default life cycle in CA Clarity and CA SCM for Distributed.

1. In CA Unicenter Service Desk, the user enters and submits a request for a new application or product idea.
2. The CA Unicenter Service Desk analyst completes the following steps:

> Reviews the request and determines its disposition.

> Converts the request to a change order representing the demand for a new application or product idea to be managed in CA Clarity.

> Initiates the creation of a CA Clarity idea.

3. The CA Clarity demand manager evaluates the CA Clarity idea and converts it into a project.

4. In CA SCM for Distributed, the manager completes the following steps:

   > Creates a new CA SCM for Distributed project.

   > Communicates the CA SCM for Distributed project name to the CA Clarity project manager.

5. In CA Clarity, the CA Clarity project manager completes the following steps:

   > Creates the CA SCM for Distributed project in CA Clarity, and uses the CA SCM for Distributed project's name that was communicated from the CA SCM for Distributed manager as the name of the CA SCM for Distributed project in CA Clarity.

   > Defines all tasks and WBS, and identifies the summary level tasks as the CA Clarity feature tasks.

   > Associates the CA Clarity feature tasks to the CA SCM for Distributed project in CA Clarity.

   > Submits the CA Clarity feature tasks for approval.

   > Creates the CA SCM for Distributed packages from the CA Clarity feature tasks.

   > Monitors task activity and progress using CA Clarity.

6. In CA SCM for Distributed, the manager completes the following steps:

   > Creates additional packages associated with the CA Clarity feature task, if needed.

   > Reviews and promotes the packages.

7. In CA Clarity, the project manager completes the following steps:

   > Marks the tasks as 100 percent complete when all of the CA SCM for Distributed packages have been promoted to the production life cycle state.

   > Marks the project as complete.

8. In CA Unicenter Service Desk, the assignee completes the following steps:

   > Updates and closes the CA Unicenter Service Desk change order.

   > Notifies the CA Unicenter Service Desk user.

Managing New Products Using CA Clarity and CA SCM for Distributed

The following scenario incorporates resource and work management in CA Clarity with software change management in CA SCM for Distributed. In this scenario, the demand for a new product results in a new CA Clarity project. The CA Clarity project manager associates

the project's feature tasks with the CA SCM for Distributed project in CA Clarity, which provides the link to the CA SCM for Distributed project.

How a New Product Moves through CA Solutions

The following information describes the flow of a new product through the default life cycle in CA Clarity and CA SCM for Distributed.

1. In CA Clarity, the project manager completes the following steps:

    > Creates a new project representing the demand for a new product.

    > Notifies the CA SCM for Distributed manager about the new project and communicates the CA Clarity project ID.

2. In CA SCM for Distributed, the manager completes the following steps:

    > Creates a new CA SCM for Distributed project using the Service Desk Clarity Connector life cycle template, and uses the CA Clarity project ID that was communicated from the CA Clarity project manager.

    > Communicates the CA SCM for Distributed project's name to the CA Clarity project manager.

3. In CA Clarity, the project manager completes the following steps:

    > Creates the CA SCM for Distributed project in CA Clarity, and uses the CA SCM for Distributed project's name that was communicated from the CA SCM for Distributed manager as the name of the CA SCM for Distributed project in CA Clarity.

    > Defines all tasks and WBS, and identifies the summary level tasks as the CA Clarity feature tasks.

    > Associates the CA Clarity feature tasks to the CA SCM for Distributed project in CA Clarity.

    > Submits the feature tasks for approval.

    > Creates the CA SCM for Distributed packages.

4. In CA SCM for Distributed, the manager completes the following steps:

    > Creates additional packages associated with the CA Clarity feature task, if needed.

    > Reviews and promotes or demotes the packages.

5. In CA Clarity, the project manager completes the following steps:

    > Monitors task activity and progress using CA Clarity.

    > Schedules (or manually executes) the process to send the status of CA SCM for Distributed packages to CA Unicenter Service Desk.

> Marks the tasks as 100 percent complete when all of the CA SCM for Distributed packages have been promoted to the production life cycle state.

> Marks the project as complete when all of the project's tasks are complete.

Additional Maintenance Work Flow

In the following scenario, you will see how the need for additional maintenance work is managed using CA Unicenter Service Desk, CA Clarity, and CA SCM for Distributed. This scenario incorporates call and change order management in CA Unicenter Service Desk, resource and work management in CA Clarity, and software change management in CA SCM for Distributed.

The request for a change begins in CA Unicenter Service Desk. The request results in a CA Unicenter Service Desk change order for additional maintenance work that is sent to CA Clarity as an incident. The CA Clarity change manager manages the CA Clarity incident as a standalone issue, converts it to a task on an existing project, or converts it to a new project.

The CA Clarity change manager determines the scope of the work and determines if the work will be managed as a standalone incident, a task on an existing project, or a new project. If the work is managed as a standalone incident or converted to a task on an existing project, there is no automatic association back to the CA Unicenter Service Desk change order. If the work is converted to a project, the association to the CA Unicenter Service Desk change order is maintained.

Example of the CMI Integration

Business Problem

According to Paul Kim (IT Director, Forward, Inc.), service desk statistics show that an average of 587 incidents and 102 change orders are being opened in the service desk every day. Approximately 50 percent of the change orders belong to the Software_Applications_Fix category and are assigned to Forward, Inc.'s development manager John McCarthy. Although the service desk has analysts to address the incidents and change orders, the work processes to close the tickets have a high level of complexity. This is due to the number of software products that Forward, Inc. has implemented to support its business processes. Currently, there are over 200 software products, developed by both internal end external resources, running on different operating and database platforms and using several programming languages. At the present time, it is almost impossible to create all the requested software fixes and keep track of the changes.

CA Approach

Forward, Inc. requires an integrated change management solution that can help them effectively manage complex, enterprise-wide development activities throughout the entire product development life cycle. CA advised Forward, Inc. to deploy CA CMI which will allow

the organization to control the full change management process from incident, problem and change request creation, prioritization, impact assessment, review and approval, through project management, resource allocation, software release and deployment. This will effectively track their enterprise-wide product development activities, reduce operational costs, and increase efficiency and productivity of their staff.

In addition, Forward, Inc. should define development and operation process to increase success. This process automation will reduce the risk of lost control in the product development change process, and help eliminate manual and error-prone double entry of information.

Configuring a Solution

Paul Kim has coordinated the implementation and deployment of the CA Change Management Integration application for Forward, Inc., composed of CA Unicenter Service Desk, CA SCM for Distributed, and CA Clarity. He has assigned project leads who have created the proper project work plan.

The following activities describe how to configure CMI components to be able to integrate the products and meet Forward, Inc.'s requirements. For information about software versions and requirements, use the following link.

http://supportconnectw.ca.com/premium/endev_ccc_lib_panv/cccharvest/infodocs/ccchar v-integ_readme.asp

Note: This link requires a SupportConnect login.

Install and Configure the CMI Integration on CA Unicenter Service Desk

Note: For this integration, CA Unicenter Service Desk must be configured to utilize the Information Technology Infrastructure Library (ITIL)-based forms and processes. Using the configuration utility (pdm_configure.exe), CA Unicenter Service Desk can be configured with an ITIL interface. By selecting the Use ITIL Methodology check box during the database initialization, an ITIL interface will be produced for your CA Unicenter Service Desk installation.

The ITIL Service Desk interface supports additional data objects not used in the standard out-of-the-box CA Unicenter Service Desk. For example, Problem and Incident objects are supported in ITIL, and the term *Asset* is replaced with the term *Configuration Item (CI)* in ITIL.

The examples in this document use the ITIL terminology for objects - configuration items (CIs), problems, and incidents.

Note: Service desk incidents are not supported in the CMI integration with CA Clarity and CA SCM for Distributed.

Install and Integrate CA Unicenter Service Desk with CA Clarity and CA SCM for Distributed

To install and integrate CA Unicenter Service Desk with CA Clarity and CA SCM for Distributed, you must first install the r11.2 version of CA Unicenter Service Desk, and then integrate the solution with CA Clarity and CA SCM for Distributed.

Note: For instructions to install CA Clarity and CA SCM for Distributed, see the individual product documentation.

To install CA Unicenter Service Desk for integration with CA Clarity and CA SCM for Distributed

1. Install the r11.2 version of CA Unicenter Service Desk.
2. Apply r11.2 software patches, if applicable.
3. After the installation successfully completes, use a command prompt window and specify the location of the \data directory within the CA Unicenter Service Desk installation files. For example:

    C:\Program Files\CA\Service Desk\data

4. Issue the following command:

    pdm_load –f projex.dat

Verify the CA Unicenter Service Desk Installation

After you have installed CA Unicenter Service Desk for integration with CA Clarity and CA SCM for Distributed, you need to verify the CA Unicenter Service Desk installation.

To verify the CA Unicenter Service Desk integration installation

1. Log in to the CA Unicenter Service Desk web client as a user with Administrative Access rights, using the same credentials that you used to run the configuration.
2. From the Administration tab, select Administration, Service Desk, Application Data, Configuration Items, Configuration Item Families.

The Configuration Item Family List appears in the right pane, as illustrated in the following sample window:
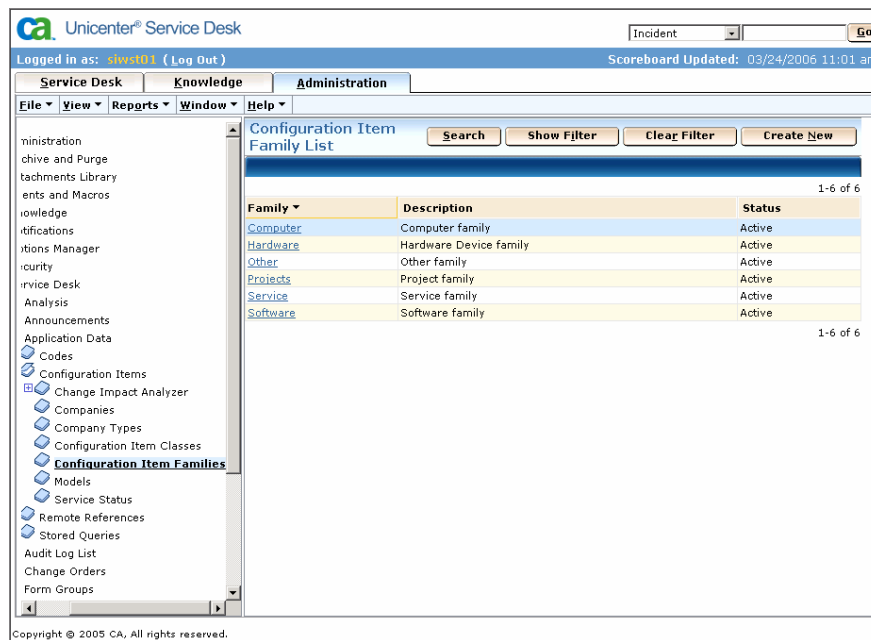
A new CI family appears representing a project. A CA Unicenter Service Desk project CI is used to store information about projects that exist in CA Clarity and CA SCM for Distributed. This information is used during the various integration processes.

3. From the Administration tab, select Administration, Service Desk, Configuration Items, Configuration Item Classes.

    The Configuration Item Class List appears in the right pane, as illustrated in the following sample window:
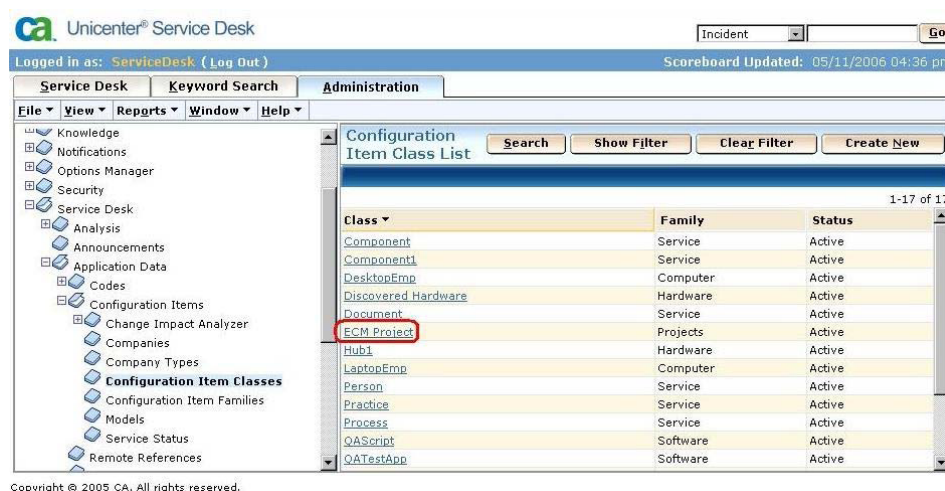
A class named ECM Project appears in the list. This class is a member of the project's CI family. You can use this class to create the project CIs for the CMI integration.

4. From the Administration tab, select Administration, Events and Macros, Macros.

   The Macro List appears in the right pane. Service Desk macros are used to start the integration processes, and are launched during specific status changes in work flow tasks.
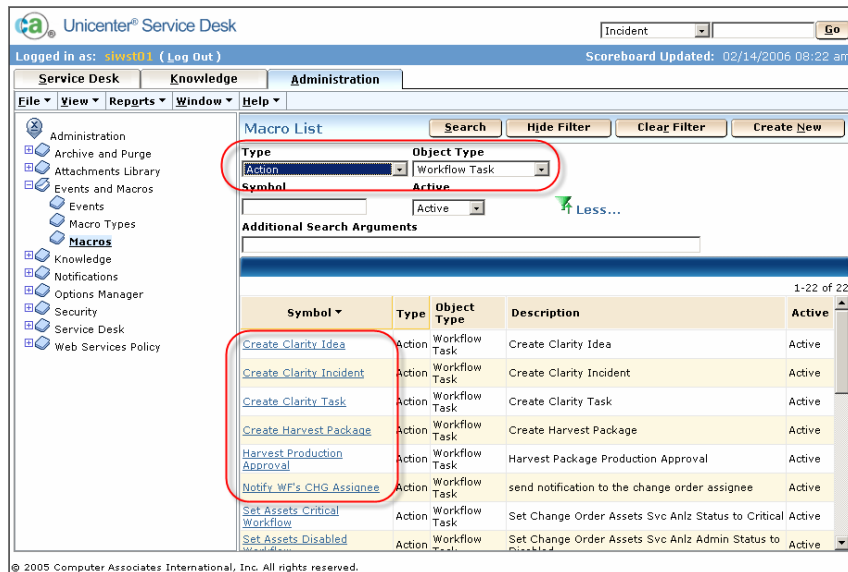
5. Verify that the previous macros exist by performing a search using the following criteria:

   > In the Type field, choose Action.

   > In the Object Type field, choose Workflow Task.

   > Click Search.

6. From the Administration tab, select Administration, Service Desk, Application Data, Remote References.

   The Remote Reference List appears in the right pane. Remote references store the command line specification that is used to execute the integration processes. The values of the command line specifications that are store here are used during the processing of the Action Macros explained in the previous step.

7. Perform this search using the following criteria and verify that the same search results are returned:
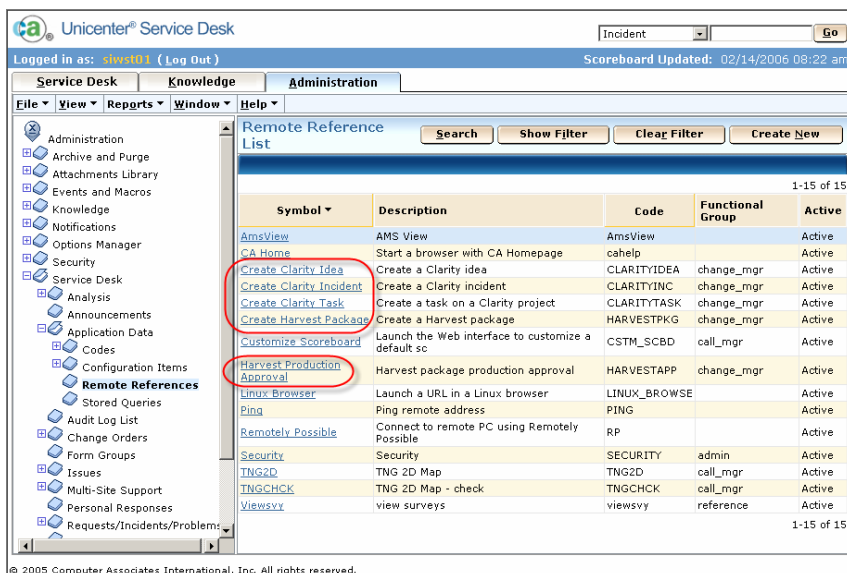
Several new work flow task statuses have been added for the integration processes.

8. From the Administration tab, select Administration, Service Desk, Change Orders, Workflow Task Status Code.

The Task Status List appears in the right pane.

9. Verify that the Evaluate and In Progress task statuses are listed, as illustrated in the following sample window:
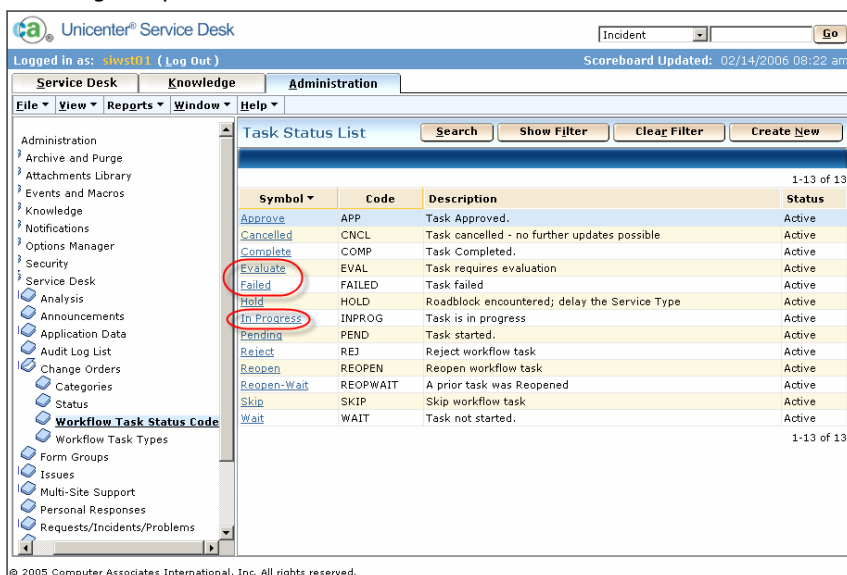
10. From the Administration tab, select Service Desk, Change Orders, Workflow Task Types.

The Task Type List appears in the right pane. Work flow task types are templates that are used to create the work flow tasks that are used in a change order. The task types represent a work flow task and associate the various statuses that will be used.

11. Verify that the following work flow task types are listed, as illustrated in the following sample:

> Create Clarity Idea

> Create Clarity Work Request

> Create Harvest Package

> Evaluate Clarity Task

> Monitor Clarity Project

> Monitor Clarity Work Request

> Monitor Harvest Project

> Notify Assignee
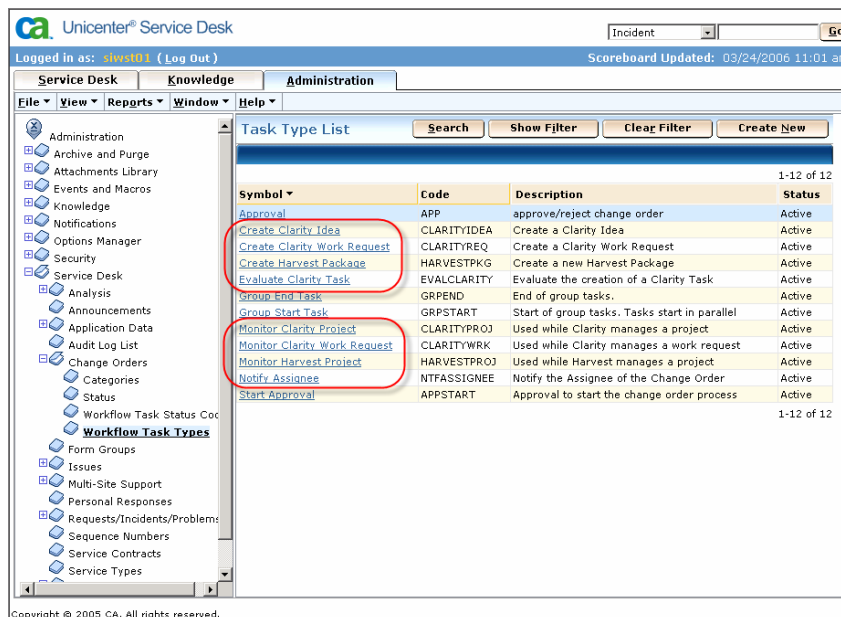
12. From the Administration tab, select Service Desk, Change Orders, Categories.

The Change Category List appears in the right pane. These sample change categories define the work flow tasks, valid task statuses, and action macros associations needed for the various types of change scenarios that can be performed with the integration.

13. Perform a search for all change categories that begin with "project" by entering *project%* in the Symbol field, and then click Search.

14. Verify that the following change categories are listed, as illustrated in the following sample:

> Project.Maint Clarity Harvest

> Project.Maint Clarity Only

> Project.Maint Harvest Only

> Project.New System Development

> Project.Other Maint Work
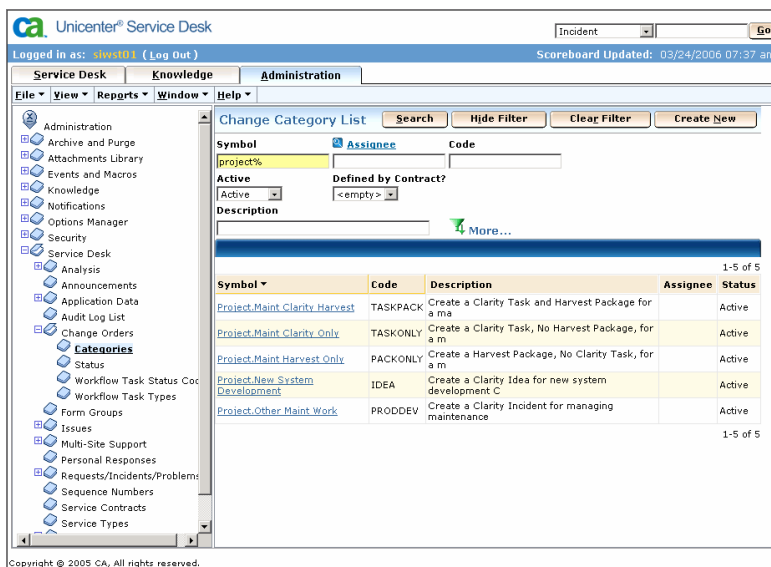
15. Using the following table, access the detail window for each category and verify the Change Category, Workflow Task, Valid Status, and Status = Action Macro:

Detail Window Properties

	Change Category	Work Flow Tasks	Valid Status	Status and Actions Macro
1	Project. New System Development	100 Create Clarity Idea	■ Cancelled ■ Complete ■ Failed ■ Pending ■ Skip ■ Wait	Complete=Create Clarity Idea

	Change Category	Work Flow Tasks	Valid Status	Status and Actions Macro
2		200 Monitor Clarity Project	■ Cancelled ■ Complete ■ Failed ■ In Progress ■ Pending ■ Skip ■ Wait	
3		300 Notify Assignee	■ Cancelled ■ Complete ■ Failed ■ Pending ■ Skip ■ Wait	Pending=Notify WF's CHG assignee
4	Project.Maint Harvest Only	100 Create Harvest Package	■ Cancelled ■ Complete ■ Failed ■ Pending ■ Skip ■ Wait	Complete=Create Harvest Package
5		200 Monitor Harvest project	■ Cancelled ■ Complete ■ Failed ■ In Progress ■ Pending ■ Skip ■ Wait	

	Change Category	Work Flow Tasks	Valid Status	Status and Actions Macro
6		300 Notify Assignee	■ Cancelled ■ Complete ■ Failed ■ Pending ■ Skip ■ Wait	Pending=Notify WF's CHG assignee
7	Project.Other Maint work	100 Create Clarity Work Request	■ Cancelled ■ Complete ■ Failed ■ Pending ■ Skip ■ Wait	Complete=Create Clarity Incident
8		200 Monitor Clarity Work Request	■ Cancelled ■ Complete ■ Failed ■ In Progress ■ Pending ■ Skip ■ Wait	
9		300 Notify Assignee	■ Cancelled ■ Complete ■ Failed ■ Pending ■ Skip ■ Wait	Pending=Notify WF's CHG Assignee

	Change Category	Work Flow Tasks	Valid Status	Status and Actions Macro
10	Project.Maint Clarity Harvest	100 Evaluate Clarity Task	■ Cancelled ■ Complete ■ Evaluate ■ Failed ■ Pending ■ Skip ■ Wait	Evaluate=Create Clarity Task
11		200 Monitor Harvest Project	■ Cancelled ■ Complete ■ Failed ■ In Progress ■ Pending ■ Skip ■ Wait	
12		300 Notify Assignee	■ Cancelled ■ Complete ■ Failed ■ Pending ■ Skip ■ Wait	Pending=Notify WF's CHG Assignee

	Change Category	Work Flow Tasks	Valid Status	Status and Actions Macro
13	Project.Maint Clarity Only	100 Evaluate Clarity Task	■ Cancelled ■ Complete ■ Evaluate ■ Failed ■ Pending ■ Skip ■ Wait	Evaluate=Create Clarity Task
14		200 Notify Assignee	■ Cancelled ■ Complete ■ Failed ■ Pending ■ Skip ■ Wait	Pending=Notify WF's CHG Assignee

Notes

■ If the system is restarted after installation, you must manually reboot the CA Unicenter Service Desk server.

■ Some versions of Windows truncate the PATH value if it exceeds a total length of 1023 characters. If the value of the PATH is too long, you can either move the CA SCM for Distributed-related PATH entries to the front of the PATH variable, or you can use 8.3 file name notation to shorten the overall length of the PATH variable.

■ If you modify Windows environment variables, you must reboot the CA Unicenter Service Desk server for the CA Unicenter Service Desk service to recognize your changes.

■ The command lines of the new Remote References located on the Remote Reference Detail window contain the command that will be executed to call the integration. The command lines that are stored in the Remote References appear as follows:

${XOG_HOME}\bin\cai (Clarity Processes)

${SD_CL_CNCTR_HOME}\bin\run_i (Harvest Processes)

■ CA Unicenter Service Desk changes the command lines based on the platform from which the server is running. These command lines include the following:

Windows

"%XOG_HOME%\bin\cai.bat"

"%SD_CL_CNCTR_HOME%\bin\run_i.**bat**"

Linux/UNIX

$XOG_HOME/bin/cai.sh

$SD_CL_CNCTR_HOME/bin/run_i.**sh**

The environment variables are then expanded, the command line parameters are added, and the command is executed. The following sample Create Clarity Task Remote Reference Detail window illustrates the command line specified in the Remote Reference:

Note: For information about how to configure CA Clarity and CA SCM for Distributed to integrate with CA Unicenter Service Desk using CMI, use the following link:

http://supportconnectw.ca.com/premium/endev_ccc_lib_panv/cccharvest/infodocs/cccharv-integ_readme.asp

Look for a PDF document named CaChangeManagementIntegrationGuide_enu.pdf

Integration Summary

The primary objective of CA Change Management Integration (the integration between CA Unicenter Service Desk, CA Clarity, and CA SCM for Distributed) is to allow the three CA solutions to be process-aware and to interact with each other.

With the CA Change Management Integration, you can transfer CA Unicenter Service Desk change orders to CA Clarity as tasks on an existing project, incidents, or ideas where they can be costed with effort and charged back to the customer, or transformed into project tasks and projects if necessary. Once transferred to CA Clarity, the work can be tracked in CA SCM for Distributed packages and the status is communicated back to the CA Unicenter Service Desk managers and users.

CA Unicenter Service Desk provides a centralized source for change order issues. Service Desk analysts initiate and approve change orders, and can then convert them to a task on an existing CA Clarity project. CA Clarity provides a centralized source for the management of changes to existing applications, while CA SCM for Distributed provides the mechanism for implementing changes related to software development. CA Clarity project managers can create CA SCM for Distributed packages on an existing project in CA SCM for Distributed from CA Clarity change order tasks.

Chapter 14: Change and Configuration Management – Integrating with CA CMDB

What is CA CMDB?

CA CMDB is a functional solution that unifies and simplifies the management of configuration information. CA CMDB consolidates and reconciles disparate sources of IT-related data in the context of business services and provides full visibility to configuration item (CI) information such as resource attributes, relationships, and dependencies. These capabilities help organizations to manage IT services that support critical business services or functions. In addition, by capturing and managing IT relationships, CA CMDB provides the basis for performing impact analysis, an important function for monitoring the adverse affects of change within an organization. Overall, CA CMDB provides organizations with the ability to enable and support ITIL Service Management processes as part of an over-arching IT strategy.

The CA CMDB Integration

CA CMDB integrates with CA Unicenter Service Desk out-of-the-box. By installing and configuring CA CMDB in conjunction with CA Unicenter Service Desk, the integration will immediately take advantage of the complementary functionality. CA CMDB provides a single source of truth about configuration items and graphically depicts the relationships between them. It also streamlines the incident and problem management processes by providing the service desk analyst with the following:

- Federated views of CIs and their attributes from multiple data sources

- Full visualization of an impacted business service and all of its relationships/dependencies, which assists with the following functions:

    > Impact analysis

    > Assisting with accurate prioritization of incidents and problems

    > Root cause analysis

Integration Points and Functionality

CA CMDB, when integrated with CA Unicenter Service Desk, adds a Visualizer menu option to the Configuration Item Detail window in the CA Unicenter Service Desk Analyst and Administrator interfaces, along with new tabs for Relationships and extended Attributes. It also adds new MDB content for Families, Classes, Manufacturers, Model Definitions, Relationship Types, Stored Queries, and Access Types.

Integration Points from CA Unicenter Service Desk

The following are integration points from CA Unicenter Service Desk:

■ Launch-in-context capabilities of the CA CMDB Visualizer from the CI Detail window, providing a graphical display of CIs and their relationships supporting the service desk analyst activities.

■ Access to Advantage Data Transformer and the Universal Federation Adapter, which enables the import of data from third-party sources (named Management Data Repositories, or MDRs) using either the CA Unicenter Service Desk or CA CMDB interface.

■ Initiation of the MDR Launcher from the CI Detail window. The MDR Launcher is configurable with no programming required, and provides access to the management data repositories containing federated information about a displayed CI.

Integration Points from CA CMDB

The following are integration points from CA CMDB:

■ Use of the full set of service support functionality available in CA Unicenter Service Desk, such as access to incidents, problems, change requests, and other process artifacts for a particular CI.

Integration Value

The CA CMDB integration provides the following value:

■ CA CMDB provides the service desk with a single source of the truth for detailed information about IT-managed components, including CI attributes and relationships.

■ When integrated with CA CMDB, CA Unicenter Service Desk has access to a visual representation of all CIs and their relationships/dependencies. This is a tremendous time-saver for incident and problem resolution activities, leading to the following benefits:

> Decreased Mean Time to Resolution (MTTR), resulting in decreased cost of outages

> Enhanced troubleshooting capabilities, including streamlined root cause analysis

> Increased analyst productivity

> Higher levels of customer satisfaction

■ Using CA CMDB capabilities allows the incident and problem management team to have instant access to the product changes that are being discovered and tracked within CA CMDB. Therefore, this integration also provides the facility in which rules can automatically create an incident based upon a rule violation. The CA CMDB integration provides a robust understanding of the potential impact of current CIs with outages. It

enables problem management to ascertain potential risk mitigation for all of the affected CIs impacted by a problem.

How the Integration Works

The following diagram illustrates how the CA CMDB integration works:
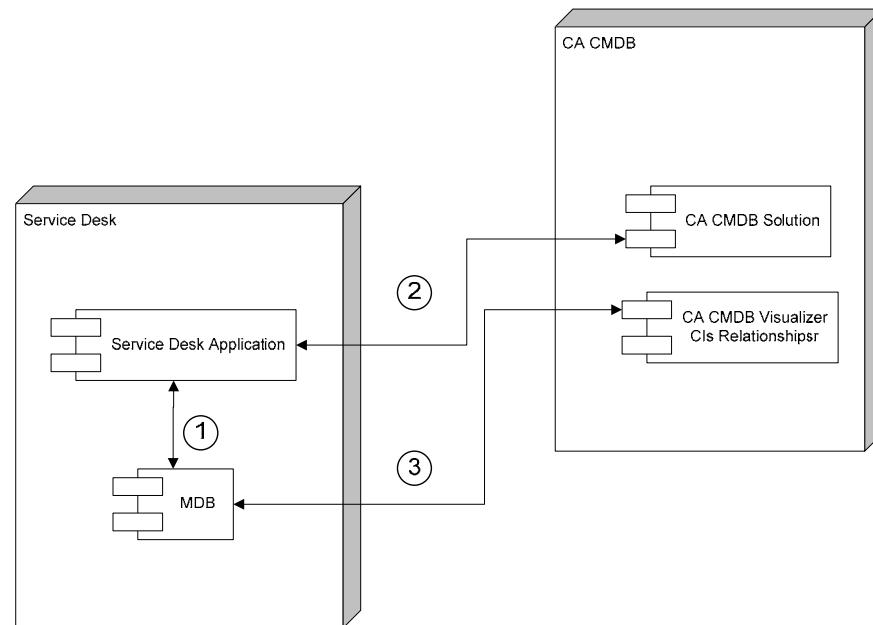
The following information applies to the previous diagram:

1. CA CMDB integrates with CA Unicenter Service Desk out-of-the-box.

2. By installing and configuring CA CMDB in conjunction with CA Unicenter Service Desk, the integration will immediately take advantage of the complementary functionality.

3. CA CMDB provides a single source of truth about configuration items and graphically depicts the relationships between them using the CA CMDB Visualizer.

Example of the CA CMBD Integration

Business Challenge

Michael Reed, the Service Desk Manager, has been contacted by Sam Higgins, the Problem Management Manager, to report his concerns about isolating the root cause of incidents and problems. When the problem is reported at either the configuration item or business service level, it is difficult to determine the extent of the impact and the root cause of the failure with current processes and tools.

Traversing the gap between the business services that are affected and the CIs that make up that service can be difficult for the service desk. Determining the types of relationships

that exist between CIs and the business services with which they are involved can also be challenging.

CA Approach

Effective root cause analysis speeds up incident resolution. Start at the failed component CI and drill down to see related CIs and the dependent services that are affected, or start at the business service CI that is experiencing a problem and follow relationships to see what CIs make up the service. In either case, it is easier to pinpoint the cause of a failure. Proactive root cause analysis avoids incidents and leads to the following:

■   Fewer outages, resulting in increased Mean Time Between Failures (MTBF).

■   Less outage time, resulting in decreased Mean Time To Repair (MTTR).

■   Fewer calls, resulting in reduced cost of providing service.

■   Reduced costs and improved availability, resulting in return on investment (ROI).

CA advises Forward, Inc. to integrate CA Unicenter Service Desk with CA CMDB. This integration will allow Forward, Inc. to take advantage of the ability to track and view configuration items and their relationships, both to other configuration items and to business services. This will lay the foundation required to support root cause analysis activities and streamlined incident and problem management capabilities.

Configuring the Solution

Michael Reed, the Service Desk Manager, has received a "High Priority" notification about a problem opened in CA Unicenter Service Desk for Forward, Inc.'s email service. Knowing that different elements can be part of the email service, Michael goes to the Scoreboard, selects Configuration Items, Service Query. He can see that there is a problem open resulting from incidents reported against this service. So, Michael drills down into the email service, beginning his analysis to determine the root cause for this outage, as illustrated in the following sample window:
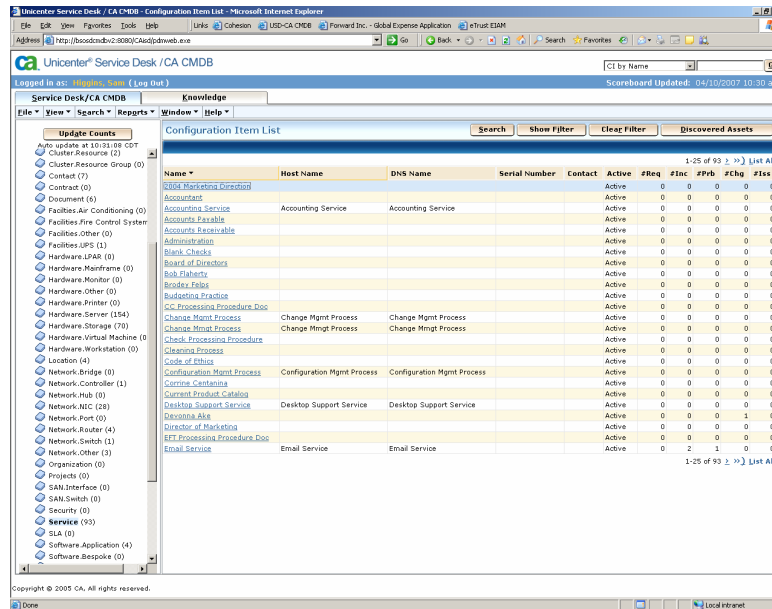
Michael clicks the Email Service name in the Configuration Items List.

The email service CI Detail window opens, and he clicks the CMDB Relationships tab. On this tab, he can see that two of the three Exchange Servers appear to have a Warning state, which could be the source of the problem. So, he drills in further to see if he can find the true root cause of the problem.
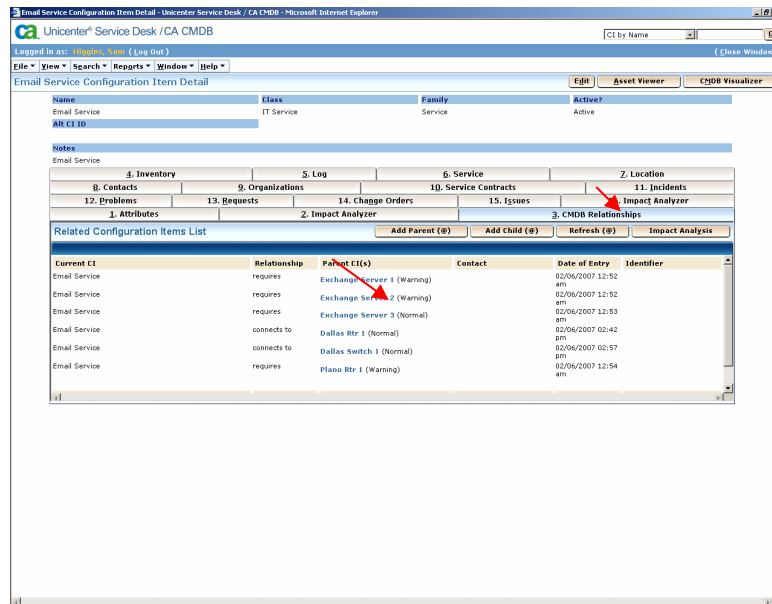
When Michael clicks the Exchange Server 1 link in the Related Configuration Items list, the Exchange Server 1 Detail window appears:
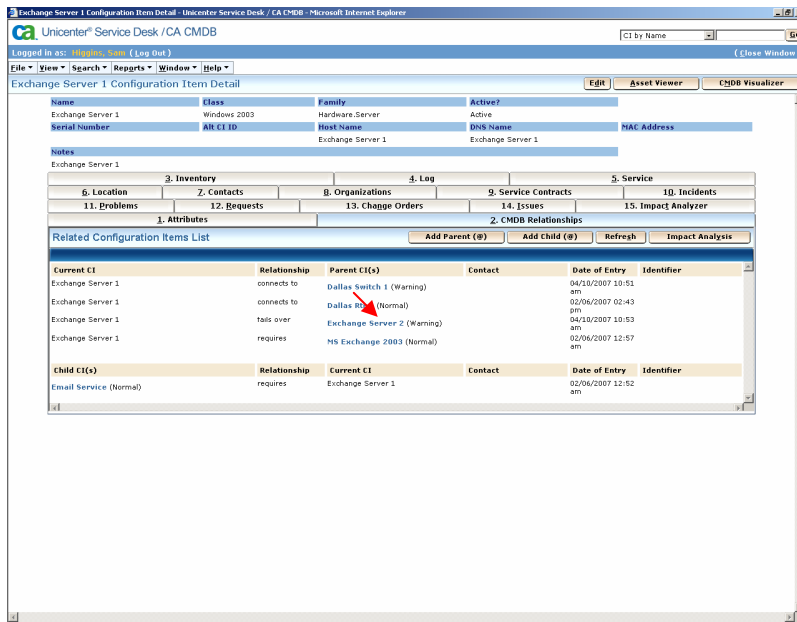
The Exchange Server CI Detail window tells Michael that Dallas Switch 1 is also in a Warning state, which may be the true cause of the problem. So, Michael reviews the Relationships using the CMDB Visualizer to analyze the situation in further detail, as illustrated in the following sample window:
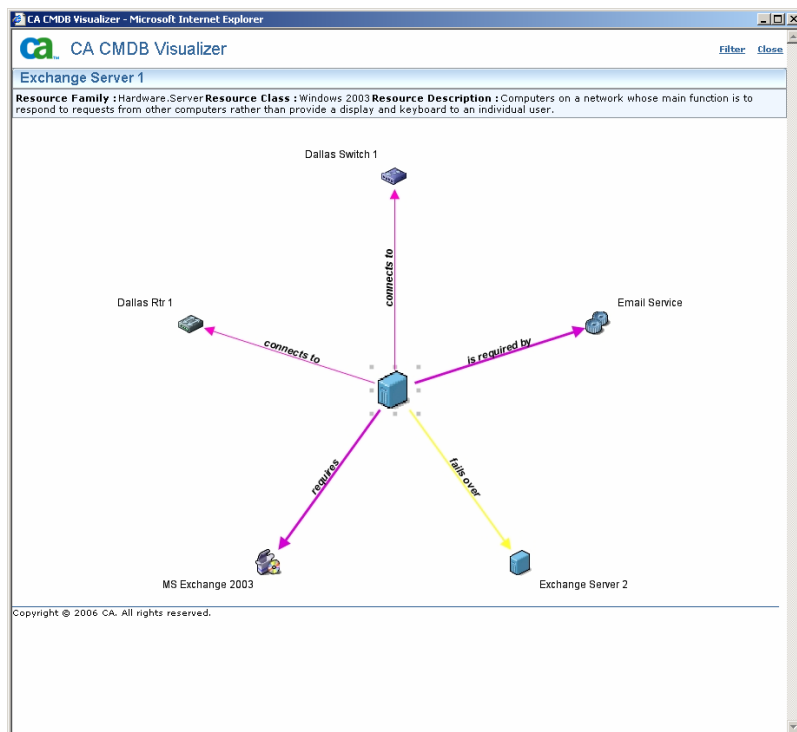
Michael can now see all of the related CI's that are part of the Exchange Server 1 CI. He can see that the Exchange Server 2 is a failover server for Exchange Server 1. In addition,

he can see that there is a Dallas Switch 1 associated to this CI. He clicks Exchange Server 2 CI to see if there is anything that matches, as illustrated in the following sample window:
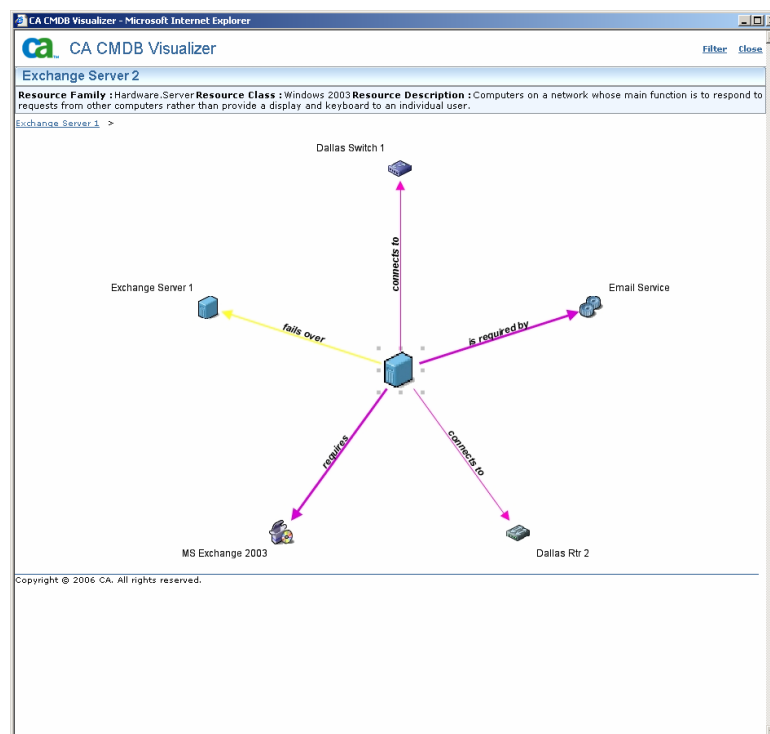
Since Michael can see that this switch affects both the Exchange Server 1 and the Exchange Server 2, and he knows that the status of the Dallas Switch 1 is in a Warning status, he can quickly determine that the root cause of the incident is a result of the switch problem. So, he goes back to the CI and updates the problem with his findings, helping the service desk staff for level 1 and level 2 to work more efficiently on solving the problem.

Note: The status of the CI can be automatically changed in the CMDB through integration to federated sources such as Unicenter NSM, or by implementing CA Unicenter Service Desk integrations with third-party management systems (that is, CA Unicenter Service Desk's web services API).

Configuring the CA CMDB Data (CIs and Relationships)

The following activities describe the manual steps required to configure the CA CMDB entities (CIs and relationships) to support the integration scenario described in the previous example. These steps will enable the service desk team to carry out root cause analysis for a disruption in Forward, Inc.'s email service.

Note: The same configuration can be executed mechanically for multiple CIs by using the CA CMDB Federation Adapter.

Create the Email Service CI

You can create a new CI in two ways:

■ From the ServiceDesk/CA CMDB Scoreboard Tab by selecting File, New Configuration Item.

■ From the Administration Tab by selecting CMDB Administration, Configuration Items, Create New.

1. Click New Configuration Item.

   The Create New Configuration Item window opens.

2. Enter the following information:

   > **Name**. Enter *Email Service*

   > **Class**. Enter *IT Service*

   Note: Class indicates the class of the asset or model. Asset classes are general categories for your assets.

3. Click Continue.

   The Email Service Configuration Item Detail window opens.

4. Click Save.

Create the Email Service CI Relationships with Other CIs

Complete the following steps to create the email service CI relationships with other CIs:

1. From the Email Service Configuration Item window, click the CA CMDB Relationship tab.

   Note: The CMDB Relationships notebook page provides two buttons, Add Parent and Add Child, which allow you to add related assets or configuration items.

2. Add the following CIs as parents, and create relationships using the information in the following table.

CI	Relationship
Exchange Server 1	Requires
Exchange Server 2	Requires
Exchange Server 3	Requires
Dallas Rtr 1	Connects to
Dallas Switch 1	Connects to
Plano Rtr 1	Requires

To create a relationship between configuration items

1. After Clicking Add Parent, and if you are adding a provider relationship, the Create New Configuration Item Relationship window opens.

2. Complete the following required fields:

   > **Symbol**. A unique identifier for the relationship record.

   > **Relationship Type**. Select a relationship type from the drop-down list. The available choices in the list are determined by the configuration item for which the relationship is being created.

3. Click Save.

To create a new relationship type

1. On the CA CMDB Administration tab, expand the System Configuration node in the navigation tree on the left side of the window.

2. Expand the CA CMDB Application Data node and click CI Relationship Types.

   The Relationship Type List appears on the right side of the window.

3. Click Create New.

   The Create New Relationship Type window opens.

4. Complete the following fields:

   > **Dependent to Provider Label**. This label is the name that appears in the list of relationship types and describes the relationship from the dependent (child) to the provider (parent). For example, "is serviced by" or "is managed by."

   > **Provider to Dependent Label**. This label is the name that appears in the list of relationship types and describes the relationship from the provider (parent) to the dependent (child). For example, "services" or "manages."

   > **Peer-to-Peer**. Set this field to Active if the relationship has two equals instead of a provider and dependent. Examples of peer-to-peer include "connects to" and "fails over."

   > **Active**. Set this field to Active to have the relationship type appear in lists and be available for selection in configuration item relationships.

5. Click Save.

6. Make sure the Record Status field is set to Active.

Visualize the Email Service CI Relationships

To visualize the email service CI relationships, click the CMDB Visualizer button from the Email Service Configuration Item window. This opens the CA CMDB Visualizer, which displays a graphical representation of the Email Service CI and shows its relationship to other CIs.

Link CA Unicenter Service Desk with the Email Service CI

Complete these steps to link CA Unicenter Service Desk with the email service CI:

1. From the CA Unicenter Service Desk/CA CMDB main window, click the Service Desk/CA CMDB tab, Scoreboard, and expand the Incidents folder.

2. On the Incident Search window, click Show Filter.

3. In the Configuration Item field, type *Email Service*.

4. Perform the Search, which will display a list of the incidents created against the Email Service CI.

5. From the search results list, double-click any of the incidents.

   The Incident Detail window opens.

6. Click the Create Problem button.

   Fields such as affected end user, assignee, configuration item, priority, summary and description have been populated on the problem record. Service desk staff can now go to the Email Service CI Detail window and, from the CMDB Visualizer or from the CA CMDB Relationships tab, drill down to the email service and see the root cause for the outage.

Integration Summary

The CA CMDB is a functional solution that unifies and simplifies the management of configuration information. The CA CMDB consolidates and reconciles disparate sources of IT-related data in the context of business services, and provides full visibility to configuration item (CI) information such as resource attributes, relationships, and dependencies.

CA CMDB integrates with CA Unicenter Service Desk out-of-the-box. By installing and configuring CA CMDB in conjunction with CA Unicenter Service Desk, the integration will immediately take advantage of the complementary functionality. CA CMDB provides a single source of truth about configuration items and graphically displays the relationships between them using the CA CMDB Visualizer.

Additional Integrations

Two more integrations that involve CA Unicenter Service Desk, CA CMDB, and CA Unicenter NSM are included later in this guide. The first integration supports change management and the second supports root cause analysis. For more information, see the External Table Display Method section in the chapter Integrating with Third-Party Products.

Chapter 15: Work Flow Management – Integrating with CA Workflow

What is CA Workflow?

CA Workflow is a sophisticated graphical common work flow engine for the automation and orchestration of a business process, in whole or in part, during which documents, information, and tasks are moved from one person to another for action, according to a predefined flow.

The following are some examples of the entities that can be orchestrated:

- Human Interaction

    > Approval requests

    > Information gathering (forms)

    > System Components

- Applications / Backend Systems

    > Databases

    > Web Services

    > Application Program Interface (API)s

The CA Workflow Integration

CA Unicenter Service Desk provides improved work flow management through the integration with CA Workflow. The work flow system provides a graphical work flow definition environment, sophisticated branching mechanisms, and the ability to interact with people who do not have direct access to CA Unicenter Service Desk to get approvals for change order tasks. The previous change order work flow system is still in place to enable backward compatibility.

Work flow is integrated into CA Unicenter Service Desk using web services.

Integration Points and Functionality

CA Unicenter Service Desk and CA Workflow integrate from the CA Unicenter Service Desk Change Order and Issue components. This integration associates a change order or issue category with a work flow process definition that is a model of all the events that can occur in your business process, and is, in fact, a representation of your business process. This is a configuration process that does not require any customized code to achieve.

Authentication

CA Workflow has no knowledge of the Contact records in CA Unicenter Service Desk. CA Workflow uses eIAM for user authentication. This means that a user must have an eIAM user record to access the CA Workflow IDE or Worklist application. The CA Workflow administrator, specified during CA Unicenter Service Desk configuration, has full access to CA Workflow. By default, this user is used by CA Unicenter Service Desk for the CA Workflow integration. This user account is set by the cawf_username and cawf_password options in CA Unicenter Service Desk Options Manager. Make sure the username and password set in these options is correct and the user has full access to CA Workflow resources within eIAM.

Integration Points from CA Unicenter Service Desk

The following are integration points from CA Unicenter Service Desk:

■ The ability to associate a Workflow Process definition to a change order or issue by associating the process definition to the change order or issue category.

■ The ability to integrate CA Unicenter Service Desk with other CA or third-party products using web services, Java, command line, or API calls.

Integration Points from CA Workflow

The following are integration points from CA Workflow:

■ The ability to retrieve information from a change order or issue to be validated, execute logic over it, and continue the defined process flow using the new information.

■ The ability to exchange and share information between CA Unicenter Service Desk and other CA or third-party products.

Integration Value

The CA Workflow integration provides the following value:

■ Ensure regulatory and corporate policy compliance through automation, such as certification of entitlements as required by SOX.

- Simplify auditing through extensive log-in and reporting capabilities across all entry points.

- Improve agility through integration with business rules, isolating the more volatile components (the policies) from the less volatile (the process definitions).

- Implement separation of duties to prevent unauthorized change activities and corporate security policy compliance.

- Provide resilience to organizational change by using role-based task assignment and integration with the corporate directory (directly, or by using CA Identity and Access Management solutions).

- Restructure and reprioritize workloads to ensure service level agreements are met. CA solutions determine whether the work flows need to be restructured based on information about the current operating environment (such as the duration of steps in the work flow, backlog, completion rates, and so on) stored in the CA Integration Platform.

Without a common work flow to automate processes and ensure consistency and quality, you may encounter errors from inappropriate actions or missed steps. With complex processes (and especially processes where some participants do not perform the required action on a regular basis), there is often inconsistency and uncertainty about what to do next. There can be delays or loss of capability due to missed deadlines. It is difficult to follow best practices. Finally, there is often duplication of effort when ownership of a task is uncertain and not enforced by the system.

How the Integration Works

The following diagram illustrates how the CA Workflow integration works:
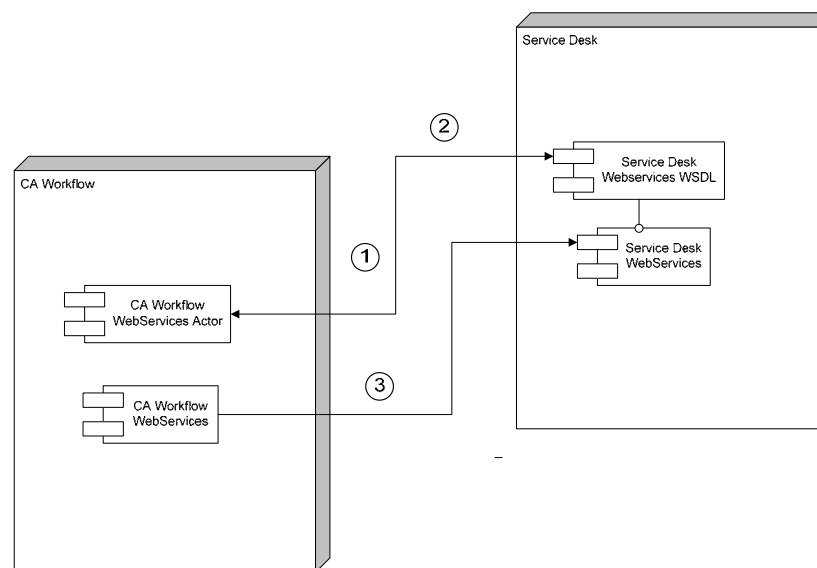
The following information applies to the previous diagram:

1. CA Workflow has a Web Services Actor, which references any CA Unicenter Service Desk Web Services WSDL. This is done during the CA Unicenter Service Desk configuration process and no manual steps are required.

2. When CA Unicenter Service Desk creates a work flow process (such as Login, createChange, UpdateLog, and Logout), it uses the CA Workflow Web Services Actor to make calls to the CA Unicenter Service Desk Web Services.

3. CA Workflow also exposes a set of its own web services, which CA Unicenter Service Desk uses to start processes.

Example of the CA Workflow Integration

Business Problem

After reviewing the service desk survey information, Michael Reed, the service desk manager at Forward, Inc. noticed that users have rated the quality of service associated with the request "Order a New PC" as "poor". In addition, users included comments in the survey to simplify and improve the steps to request a new computer. Users said that the order process is taking a long time, and in some cases when they receive their computer, it has not been configured properly and they therefore cannot do their job properly. As a result, computers not having the required configuration are generating a large number of incidents for the service desk. Michael needs to coordinate the necessary actions with the Procurement and Change Management managers to correct this situation as soon as possible.

CA Approach

CA Unicenter Service Desk r11 provides Forward, Inc. service desk analysts with the tools they need to edit, approve, and complete change orders. In addition, CA Unicenter Service Desk r11 has embedded the CA Workflow solution which allows repetitive business processes to be automated and tracked. This ensures that the change orders will be assigned to the correct people and the order status can be effectively tracked, providing the appropriate information when the requesting, approving, and rejecting activities occur. CA advises Forward, Inc. to use the CA Workflow capabilities to implement all business processes that involve change orders, including activities executed by different departments within the organization. This will help improve the service delivery, and will also help reduce the number of incidents created due to incomplete information and skipping steps in a predefined business procedure.

Configuring the Solution

Paul Kim, IT Director at Forward, Inc., agrees with Michael and wants to use CA Workflow to automate the business process information related to requesting a new computer. Paul has asked Michael to gather and analyze all of the necessary information about the "Order PC" business process and coordinate the process flow, as well as actions and information managed in every step of the process, with all affected parties. Once everyone agrees, Michael can implement the process in CA Unicenter Service Desk using CA Workflow.

Configure the Integration from CA Unicenter Service Desk r11

Michael Reed has defined all of the information related to the "Order PC" business process flow. He has defined the logical steps and actions, as well as the displays of information necessary to automate the process. In addition, he has also has identified the people at Forward, Inc. that will interact with the process. Michael has implemented all definitions in CA Workflow and will now configure CA Unicenter Service Desk to run the "Order PC" process definition.

The "Order PC" order flow needs an approver and requester, so Michael must define, in the product, both the requester of the new computer and an approver. The requester must be a user defined in both CA Unicenter Service Desk and eIAM. The approver can be defined to both, but is only required to exist in eIAM. eIAM is the authentication mechanism and contact store for CA Workflow.

In this example, here are the players:

> **Requester**. Name is *William Requester* and Userid is *requester*

> **Approver**. Name is *Mary Approver* and Userid is *approver*

The following are the steps that Michael must complete to create contacts in CA Unicenter Service Desk and eIAM to implement the solution for the sample work flow:

1. Log in to the eIAM web interface at the following location:

   https://EIAMSERVERNAME:5250/spin/eiam/eiam.csp

   EIAMSERVERNAME is the name of the server where eIAM is installed. This can be localhost, and it can also be the remote server where the Management Database (MDB) is installed. Be sure to log in with the "Service Desk" application instance, which by default is "ServiceDesk-<machine name>". Log in using the Privileged User Name and Privileged User Password that was set when CA Unicenter Service Desk was configured.

   Note. You can also log in as the EiamAdmin user who is the administrator of eIAM. The eIAM Admin password is set when CA Unicenter Service Desk is configured. This user is not an operating system user. If you do not remember the EiamAdmin password, contact CA Support and they can help you recover or reset the password.

2. From the Manage Identities tab, click Go to search for all users.

   By default, only the ServiceDesk and machine Administrator users exist in eIAM. To create a new user, click the New User icon to the left of the Users folder on the lower left part of the window. You must create a user for the Requester and Approver in the work flow. A single admin contact (for example, ServiceDesk) can be used to complete all steps in the work flow, but we recommend that you have both a requester and approver.

3. Create the Requester contact in both eIAM and CA Unicenter Service Desk.

   When created in both solutions, make sure that the userid for both is *requester*. Make sure the Requester contact is assigned an email address in CA Unicenter Service Desk by validating the Contact Detail within CA Unicenter Service Desk. In this example, *William Requester* is used as the requester.

4. Create the Approver in eIAM and CA Unicenter Service Desk.

   Out-of-the-box, the Approver is set to the assignee of the Change Category that launches the Workflow Process Definition, so assign this user to your Change Category. Assign an email address within eIAM for the Approver contact. In this example, the user is *Mary Approver*. Be sure that the userids, both in eIAM and CA Unicenter Service Desk, are set to *approver*.

   Note: This is just an example. As previously mentioned, the approver can easily be changed to any user, or attribute. This user does not necessarily need to be an analyst or even exist in CA Unicenter Service Desk. The only requirement is that the Approver exists in eIAM.

5. In eIAM, verify that the ServiceDesk user, or the appropriate CA Workflow Administrator, is defined in CA Unicenter Service Desk and is a member of the Workflow Process Initiator Group. By default, CA Unicenter Service Desk launches all work flow processes using this actor.

Configure the Integration from CA Workflow

Complete the following steps to configure the integration from CA Workflow:

1. From the Start menu, choose Programs, Computer Associates, Unicenter, Service Desk, Workflow IDE.
2. Log in to the CA Workflow Design Environment as the ServiceDesk user with the password that was entered for the ServiceDesk user during configuration.
3. Double-click the Login – Unicenter Service Desk workflow process to verify the login parameters are correct.
4. Double-click the Call Login Activity Node.

   The login to CA Unicenter Service Desk opens. This preloaded "Call Login" work flow activity node is built with a user ID and password from "ServiceDesk". This work flow activity is used to invoke a login process to the CA Unicenter Service Desk web services. If you have changed this password, you must update this parameter.

   This is the user that will log in to CA Unicenter Service Desk using the web services. This user's rights and security are designated by their Access Type in CA Unicenter Service Desk. You can change the username and password, but make sure the user can perform basic tasks such as creating and updating tickets in CA Unicenter Service Desk.

   Note: When moving work flow processes into a production environment, you should strongly consider *not* using the Service Desk user. Using a specific work flow user will make it much easier to attribute the activity at the Service Desk when debugging problems. In fact, for more complex implementations, separate work flow users should be specified for each major work flow process that is built.

   If the ServiceDesk user password is incorrect and conflicts with what was specified during configuration, which is the case in this example, you must change it here and then save the updated work flow process. In this example, do *not* change the ServiceDesk user, but *change* the password to servicedesk (all lowercase letters). After making this change, you must click the save button to update the process definition. Clicking OK within the activity does not save the entire work flow process.

Out-of-the-box, the approver of the "Order PC" Workflow is assigned to the Assignee of the Change Category launching the Workflow. The corresponding attribute is $ApproverUserid. You can reassign the approver role to a different user, or different attribute, by going to the Order PC Process Definition and navigating to the Roles tab at the bottom of the window.

5. Complete the following steps to verify that CA Workflow is configured to send emails:
    a. Navigate to the Process Manager tab, Server Configuration option.

    b. Enter a valid SMTP server name in the SMTPHost field.

    c. Click save.

Configure the Change Order in CA Unicenter Service Desk

Complete the following steps to configure the change order in CA Unicenter Service Desk:

1. In CA Unicenter Service Desk, select Administration, ServiceDesk, Change Orders, Categories.
2. Click Create New to create a new category PC.Order.
3. Complete the information for the Change Category, including the Assignee, which should be set to *Mary Approver*.
4. Navigate to the Workflow tab, select Use CA Workflow, and select the Order PC – Service Desk Workflow definition from the list of work flows.

    Note: This list only shows definitions relate to CA Unicenter Service Desk. To view your own custom process definition within CA Unicenter Service Desk you must have an external string variable named *usd_persid* defined in the definition. Otherwise, the definition is not known to CA Unicenter Service Desk.

5. Click save.

    The new category is displayed with the id of the CA Workflow listed on the Workflow tab.

Test the Integration

Complete the following steps to verify that the CA Workflow integration has been set up correctly:

1. Create a new change order in CA Unicenter Service Desk, and assign it to the PC.Order category.
2. Save the change order, and then review the progress of the work flow using the Workflow Tasks tab.
3. Scroll to the bottom of the task list and click the RequestForm task. If you do not see this task, refresh the list by selecting View, Refresh.

    Note: The requester will also receive a notification to log in to their worklist and complete an online request form. This notification will be based on the notification method specified in their contact record in CA Unicenter Service Desk. Assuming the notification method is email, the email is delivered to the account defined in CA Unicenter Service Desk. The requester can either log in to the Worklist using the RequestForm Link in the change order, or click the link in the email.

4. Log in to the CA Workflow Worklist interface as the Requester using the following URL:

    http://forwardinc:8080/wl/login.jsp

    Note: In this URL, you need to change forwardinc: 8080 to your server information. Make sure you use the same case for the User Name as defined in eIAM. If you use a different case, you can log in to the Worklist, but your Worklist will be empty.

    In the task list, the Requester should see a single RequestForm task.

5. Click the Perform link on the right side of the window to fill out the form and continue the work flow process.

6. Complete and submit the form.

    After you submit the form, you return to the Worklist window and you will receive confirmation that the task is complete.

7. Once you have submitted the request, log out of the Worklist.

8. Return to the change order in CA Unicenter Service Desk to see how the process has advanced in the work flow.

    After the form was completed, an email from CA Workflow was sent to the approver and requested that they perform the next approval task.

    Note: The approver will also receive an email notification to log in to their Worklist and complete an online approval form. This notification will be based on an event within CA Workflow. The email comes from the account defined within CA Workflow.

9. The approver can either log in to the Worklist using the Approver Form Link in the Change Order, or click the link within the email.

10. Log in to the CA Workflow Worklist interface as the Approver contact and click Perform to view the approval form for the requested PC configuration.

    The form you see is a simple text form that requires no design within the CA Workflow Design Environment. This is an example of the options you have when designing forms.

11. Once you see the form, click Yes to approve the request.

    You will then see confirmation back on the main Worklist screen.

12. Review the change order in CA Unicenter Service Desk to see further updates to the Workflow Tasks tab.

    Each activity in the work flow process is listed on this tab.

13. Send an email notification to the user, letting them know that their request was approved.

14. Close the ticket.

    In a production environment, this may trigger a procurement system to place an order or notify the correct person to continue processing the order.

    Note: For the CA Workflow to close the ticket, all the work flow tasks must first be complete. Because of this, you may occasionally see a delay in the status update within the change order. To see the change, refresh the ticket by selecting View, Refresh.

15. Review the Activities tab to see all updates CA Workflow performed within CA Unicenter Service Desk, such as manual notifications and closing the ticket.

Note: For more information about workflow, see the following locations:

http://supportconnectw.ca.com/public/caworkflow/caworkflow_supp.asp

http://supportconnectw.ca.com/public/caworkflow/infodocs/caworkflowr10-mandtryinfo.asp

http://supportconnectw.ca.com/public/caworkflow/infodocs/caworkflow_tecdoc.asp

Best Practices

When dealing with a high-volume implementation of CA Workflow, the Tomcat web server that is leveraged by both CA Unicenter Service Desk web services and CA Workflow benefits from configuring it to use additional memory. By default, CA Unicenter Service Desk starts the Tomcat instance with a 512 MB memory pool. This example allows the memory change to be permanent within a CA Unicenter Service Desk r11.2 environment and allows you to increase the Tomcat memory to 1024 MB.

Preparing for the Memory Increase

To help prepare for the memory increase, complete the following steps on your test system:

- Make sure your system can handle the recommended memory increase.

  Note: To verify that the Java Virtual Machine (JVM) utilized by Tomcat can handle the increased memory, enter the following command at the operating system prompt:

  ```
  Java –Xmx<value>|more
  ```

  In this command, *<value>* is the memory value. That is, Java –Xmx1024M|more

  If you see an error similar to the following, the setting will not work.

  ```
  Error occurred during initialization of VM
  Could not create Java virtual machine
  Could not reserve enough space for object heap
  ```

  However, if you see information explaining the use of Java.exe, the test was successful.

- Make a backup of every file that you have modified. This will allow for a back out recovery.

Increasing the Memory

The default memory value is specified in the CA Unicenter Service Desk pdm_tomcat startup parameter list in the CA Unicenter Service Desk stdlog file. The default parameter is –Xmx**512M**. CA Unicenter Service Desk stdlog logs are located under the $NX_ROOT/log directory, where $NX_ROOT corresponds with the CA Unicenter Service Desk installation directory.

1. Add a new variable named @NX_TOMCAT_CMD to the end of the $NX_ROOT\pdmconf\NX.env_nt.tpl file with a new memory value, beside the –Xmx parameter.

   Important! The following is an example of the variable to add to the file. Do not copy or use this example in your installation. The file directory and drive locations will most likely be different from your implementation.

```
NX_TOMCAT_CMD="D:/PROGRA~1/CA/Shared~1/JRE/1.4.2_06/bin/javaw -
Djava.net.preferIPv4Stack=true -Xrs –Xmx1024M -
Djaas.config=D:/PROGRA~1/CA/SERVIC~1/add-ons/caflow/jaas.config -
Djava.security.auth.login.config=D:/PROGRA~1/CA/SERVIC~1/add-
ons/caflow/jaas.config -
Djdbc.baseDriver=com.microsoft.sqlserver.jdbc.SQLServerDriver -
Dsun.io.useCanonCaches=false -Djava.awt.headless=true -
Djava.endorsed.dirs=D:/PROGRA~1/CA/Shared~1/tomcat/4.1.31/common/endorsed -
classpath
D:/PROGRA~1/CA/Shared~1/tomcat/4.1.31/common/lib/tools.jar;D:/PROGRA~1/CA/Shared
~1/JRE/1.4.2_06/java/lib/tools.jar;D:/PROGRA~1/CA/SERVIC~1/java/lib/tools.jar;D:
/PROGRA~1/CA/Shared~1/tomcat/4.1.31/bin/bootstrap.jar;D:/PROGRA~1/CA/SERVIC~1/ja
va/lib/sqljdbc.jar -
Dcatalina.base=D:/PROGRA~1/CA/SERVIC~1/bopcfg/www/CATALINA_BASE -
Dcatalina.home=D:/PROGRA~1/CA/Shared~1/tomcat/4.1.31 -
Djava.io.tmpdir=D:/PROGRA~1/CA/SERVIC~1/bopcfg/www/CATALINA_BASE/temp
org.apache.catalina.startup.Bootstrap"
```

2. Press the Enter key after adding the new text to the file.

   Looking closely at this example, there are three *main attributes* that you will need to implement:

   > The use of forward slashes (UNIX style) instead of the back slashes.

   > The use of eight character path names instead of the Windows long file names.

   > The removal of double quotes, because long file names are not used. Double quotes are only specified at the beginning and end of the large string.

   Important! The previous line after **@NX_TOMCAT_CMD=** must come from the CA Unicenter Service Desk stdlog files of the computer where this will be implemented. Search for "startup_tomcat" within your stdlog file located under $NX_ROOT\log. When you start CA Unicenter Service Desk, you will see a line similar to the following in the stdlog files. This example illustrates the default value of 512 MB.

```
02/06 09:24:11.58 yunle01A pdm_tomcat        3304 SIGNIFICANT  pdm_tomcat.c
479 startup_tomcat: C:\Program Files\CA\SharedComponents\JRE\1.4.2_06\bin\javaw
-Djava.net.preferIPv4Stack=true -Xrs -Xmx512M -
Djaas.config=C:\PROGRA~1\CA\SERVIC~1\add-ons\caflow\jaas.config -
Djava.security.auth.login.config=C:\PROGRA~1\CA\SERVIC~1\add-
ons\caflow\jaas.config -
Djdbc.baseDriver=com.microsoft.sqlserver.jdbc.SQLServerDriver -
Dsun.io.useCanonCaches=false -Djava.awt.headless=true -
Djava.endorsed.dirs="C:\Program
Files\CA\SharedComponents\tomcat\4.1.31\common\endorsed" -classpath "C:\Program
Files\CA\SharedComponents\tomcat\4.1.31\common\lib\tools.jar;C:\Program
Files\CA\SharedComponents\JRE\1.4.2_06\java\lib\tools.jar;C:\PROGRA~1\CA\SERVIC~
1\java\lib\tools.jar;C:\Program
```

```
Files\CA\SharedComponents\tomcat\4.1.31\bin\bootstrap.jar;C:\PROGRA~1\CA\SERVIC~
1\java\lib\sqljdbc.jar" -
Dcatalina.base="C:\PROGRA~1\CA\SERVIC~1\bopcfg\www\CATALINA_BASE" -
Dcatalina.home="C:\Program Files\CA\SharedComponents\tomcat\4.1.31" -
Djava.io.tmpdir="C:\PROGRA~1\CA\SERVIC~1\bopcfg\www\CATALINA_BASE\temp"
org.apache.catalina.startup.Bootstrap start
```

The following example illustrates in simpler terms:

```
02/22 15:03:46.80 valre03- pdm_tomcat       3100 SIGNIFICANT  pdm_tomcat.c
479 startup_tomcat: <HUGE COMMAND LIKE THE ONE ABOVE>
```

This is the one that you need to use. The last word, "start", on the HUGE COMMAND is not used. Remember to change the stdlog value that you will be specifying with the new @NX_TOMCAT_CMD variable that you will be adding to match the three previously mentioned attributes.

3. Open the $NX_ROOT\pdmconf\pdm_startup.i.tpl file.

    At the end of this file, you will see the following:

    command = "$NX_ROOT/bin/pdm_tomcat_nxd -s"

4. Change the command to the following:

    command = "$NX_ROOT/bin/pdm_tomcat_nxd -s **-C $NX_TOMCAT_CMD**"

5. Run pdm_configure.

    Your Tomcat server's maximum memory size will now be 1024 MB.

    You will typically need to restart the Tomcat server to reboot the system. The JVM must be able to access a contiguous block of memory on startup.

Verifying the Memory Increase

Use the following steps to verify that the memory increase was successful:

1. Check stdlog.x, located under the $NX_ROOT/log directory, for "startup_tomcat" during the timeframe of the current startup of CA Unicenter Service Desk after completing all changes. Within the stdlog line where "startup_tomcat" is specified, you should see – Xmx**1024M**, containing the new value of **1024M** that the previous example used.

    Important! If you do not see this, there has been an editing error and you should repeat the steps to increase the memory.

2. Verify that you can log in to the CA Unicenter Service Desk Web Interface successfully.

3. If you are using CA WorkFlow, verify that you can log in to CA WorkFlow from the CA Unicenter Service Desk primary computer successfully and that you do not see any "cannot connect to PM" errors.

    Important! If you receive a "cannot connect to PM" error, Tomcat did not start successfully. This is caused by an editing error and you should repeat the steps to increase the memory.

Backing Out the Memory Increase

If you still have problems verifying the memory increase after repeating the steps to increase the memory multiple times, replace all the edited files with the backups you created, and then rerun pdm_configure to recover.

Integration Summary

CA Unicenter Service Desk and CA Workflow integrate from the CA Unicenter Service Desk change order and issue components. This integration associates a change order and issue category to a work flow process definition that is a model of all the events that can occur in your business process and is, in fact, a representation of your business process. This is simply a configuration process and requires no customized code to achieve.

CA Workflow uses the CA Unicenter Service Desk Web Services functionality to retrieve information from change orders and issues, or set values on change orders and issues. In fact, it can do anything that the CA Unicenter Service Desk Web Services can do.

Note: For a list of all web services functionality, see the *CA Unicenter Service Desk Web Services User Guide*.

CA Unicenter Service Desk uses the Workflow Web Services to start the specified process for a given change order or issue when that change order or issue is saved.

CA Workflow has no knowledge of the Contact records in CA Unicenter Service Desk. CA Workflow uses eIAM for user authentication; a user must have an eIAM user record to access the CA Workflow IDE or Worklist application. The CA Workflow administrator, specified during CA Unicenter Service Desk configuration, has full access to CA Workflow.

Finally, remember that when moving work flow processes into a production environment, you should strongly consider not using the "ServiceDesk" user. Using a specific work flow user will make it much easier to attribute the activity at the Service Desk when debugging problems. In fact, for more complex implementations, we recommend that you use separate work flow users for each major work flow process that is built.

Chapter 16: Security Management – Integrating with eIAM

What is eIAM?

eTrust® Embedded Identity and Access Management (eIAM) allows applications to share common access policy management, authentication, and authorization services. eIAM provides a set of security services. The following security services are available:

Configuration Services

■   Registering and removing the registration for application instances

■   Administrative scoping of application administrators

■   Delegating administrative rights

■   Managing users and groups

Administration Security Services

■   Managing access, event, and obligation policies

■   Managing calendars

Run-time Security Services

■   Authenticating users

■   Authorizing access

■   Logging security events

The eIAM Integration

CA Unicenter Service Desk and eTrust Identity and Access Management (eIAM) can be integrated when running the CA Unicenter Service Desk configuration process. This is an optional integration. When integrated, CA Unicenter Service Desk uses eIAM to validate a

user's login. This replaces the default validation performed by the host operating system. Additionally, eIAM is used by the CA Workflow integration for authentication.

eIAM can be configured to either reference an external Lightweight Directory Access Protocol (LDAP) directory, or use the MDB to store user information. eIAM has an LDAP interface for use when it is configured to use the MDB. CA Unicenter Service Desk can use eIAM as an authentication and authorization mechanism, as well as an LDAP repository, if eIAM has been previously integrated with LDAP.

Integration Points and Functionality

eTrust Identity and Access Management is CA's solution to manage user access. It is designed to be a central repository of user information (identities), and defines the users' authentication and access to other products. CA Unicenter Service Desk uses a special version of this product, named Embedded Identity and Access Management. If you have several CA solutions installed, some of them may be using eIAM to store identities and access policies.

The eIAM repository of user records is either one of two sources:

■   An external LDAP directory

■   Its own internal tables in the MDB

eIAM has an LDAP interface for use when it is configured to use the MDB. If your organization makes use of a directory server, such as Active Directory or eTrust Directory, consider configuring eIAM to use that directory for its user base. This makes the users in your directory accessible by any other application that uses eIAM.

CA Unicenter Service Desk stores contact information in the MDB. These tables, such as ca_contact, have no relationship to eIAM. CA Unicenter Service Desk does not use eIAM for access or identity management. As in past releases, CA Unicenter Service Desk manages its own access and security with Access Types and Data Partitions. CA Unicenter Service Desk's only use of eIAM is for authentication. This is a configuration option to replace CA Unicenter Service Desk's operating system authentication with the authentication scheme in eIAM. For more information, see the online help.

Integration Points from CA Unicenter Service Desk

CA Unicenter Service Desk only uses eIAM for authentication. This is a configuration option to replace the CA Unicenter Service Desk operating system authentication with the authentication scheme in eIAM.

Note: For more information about authentication, see the *Incident and Problem Management Green Book* using the Technical Support link at http://ca.com/support.

Integration Points from eIAM

eTrust Identity and Access Management is CA's solution to centralized user management. It is integrated with CA Unicenter Service Desk when eIAM has been selected to manage CA Unicenter Service Desk authentication during the CA Unicenter Service Desk configuration process.

Note: CA Unicenter Service Desk and eIAM have an indirect integration point when CA Unicenter Service Desk is integrated with CA Workflow. In this situation, CA Workflow uses eIAM to manage the users that have interaction with the flows. In turn, CA Unicenter Service Desk uses the input from those users in the change order management process.

Integration Value

This seamless integration provides CA Unicenter Service Desk with the ability to support an organization's cohesive user authentication strategy.

How the Integration Works

The Default Installation (Configuration 1)

The following diagram represents where CA Unicenter Service Desk and CA Workflow store user information after a default installation:
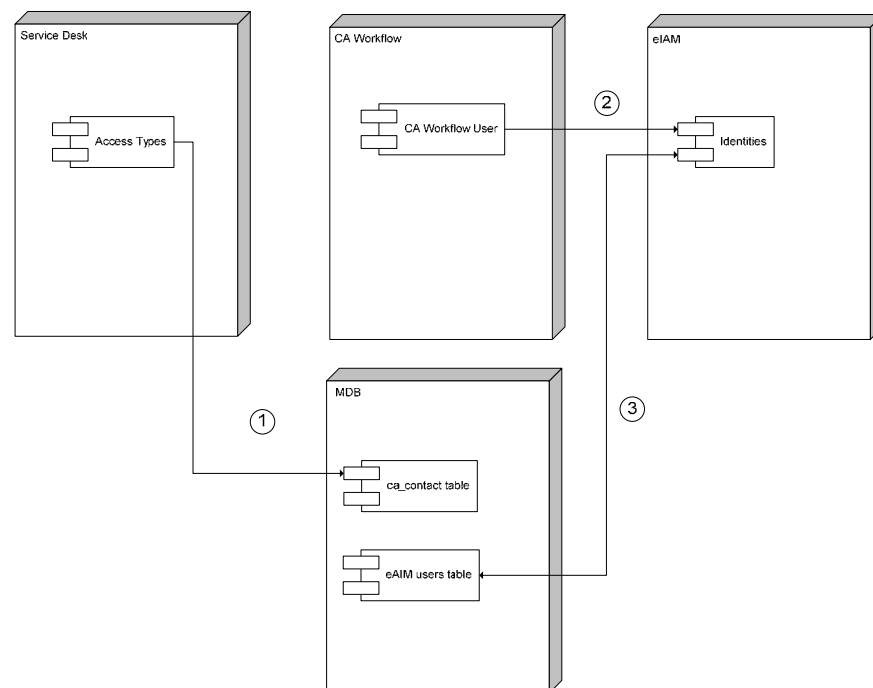
You should consider using this configuration if your site makes little or no use of CA Workflow or a directory (LDAP) server.

The out-of-the-box configuration presents some challenges. The administrator must ensure that an eIAM record exists for any CA Unicenter Service Desk analyst who will use CA Workflow. In fact, any user of CA Workflow requires an eIAM user record. For the CA Unicenter Service Desk analyst, this extra identity means an extra password to track.

The LDAP Server (Configuration 2)

As illustrated in the following diagram, contact administration is much easier when both CA Unicenter Service Desk and CA Workflow are configured to reference the same LDAP server, such as eTrust Directory or Active Server.
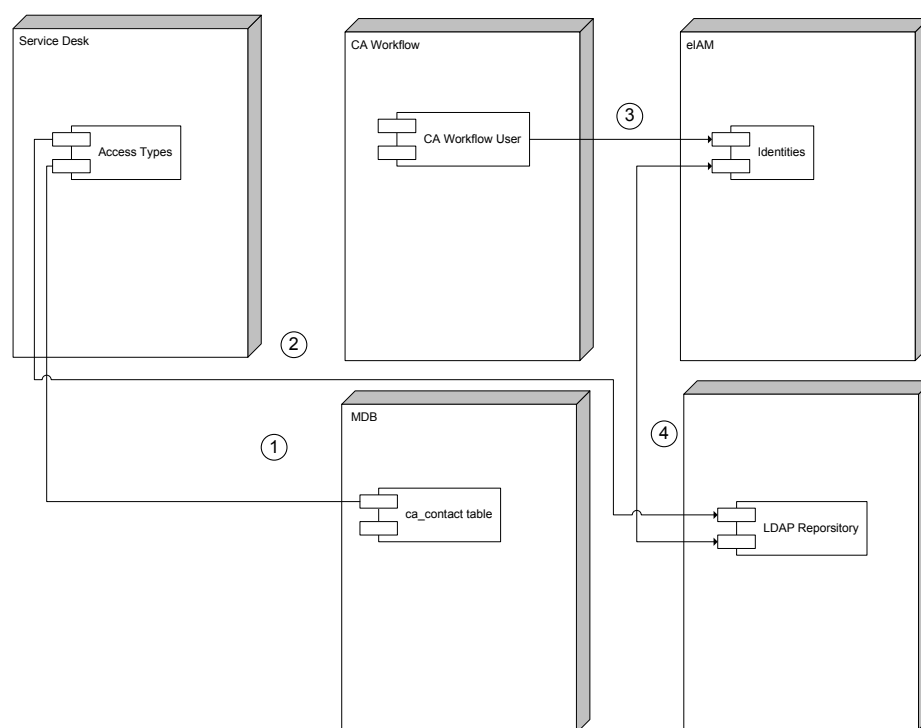
In this scenario, all users in the LDAP server automatically have access to the CA Workflow Worklist web application. CA Unicenter Service Desk still uses the contact tables in the MDB for its user records and normal operation, but these may be created from and periodically synchronized with the LDAP server records.

Note: For information about configuring and using CA Unicenter Service Desk with an LDAP server, see the *CA Unicenter Service Desk Administration Guide*.

This configuration is especially useful if your site is already using an LDAP server as the main directory for user identities. With the assistance of CA Unicenter Service Desk's LDAP capabilities, the userids of contact records should match the ones in the directory, allowing for smooth assignment of work items to CA Unicenter Service Desk analysts. The

integrations become even more seamless if CA Unicenter Service Desk uses eIAM for authentication.

EIAM as LDAP (Configuration 3)

When eIAM is configured to use the MDB (not an external directory) to store users, eIAM exposes the user directory using an LDAP interface. If your site does not use an LDAP server, you can get the advantages of the previously mentioned scenario (Configuration 2) by configuring CA Unicenter Service Desk to use eIAM as an LDAP source.

Note: For information about how to configure CA Unicenter Service Desk to use eIAM's LDAP interface, see the CA Unicenter Service Desk documentation.

This configuration is only applicable when eIAM is configured to use the MDB. If eIAM is configured to use an external LDAP server, configure CA Unicenter Service Desk to reference the same LDAP server, *not* eIAM. For information, see The LDAP Server (Configuration 2).
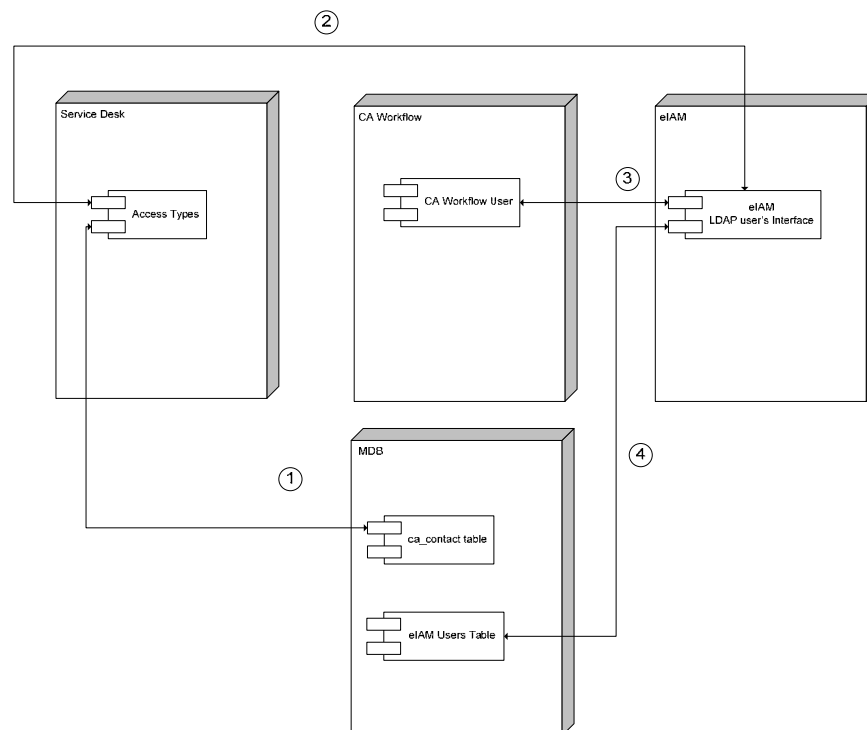
This configuration can be useful if your site does not use an LDAP server, but you want to consolidate user management in eIAM. As previously mentioned, several other CA products also use eIAM, which helps with user management.

Example of the eIAM Integration

Business Challenge

Bill Walker, the VP of Security at Forward, Inc., has informed all IT managers about the new security standard of using Active Directory for all applications running on Windows. All IT managers must coordinate the necessary activities and resources to fit into the newly established standard.

CA Approach

eTrust Identity and Access management is CA's solution for managing user access. It is designed to be a central repository of user information (identities) and defines their authentication and access to other products. CA Unicenter Service Desk uses a special version of this product named Embedded Identity and Access Management, or eIAM (publicly known as IAM Toolkit). If you have several CA solutions installed, some of them may be using eIAM to store identities and access policies (that is, CA Unicenter Service Catalog, UAPM, CA Workflow).

eIAM's repository of user records is either one of two sources:

■   An external LDAP directory

■   Its own internal tables in the MDB.

eIAM has an LDAP interface for use when it's configured to use the MDB. The tables used by eIAM in the MDB are different from the ones used by CA Unicenter Service Desk.

If your organization makes use of a directory server, such as Active Directory or eTrust Directory, consider configuring eIAM to use the directory for its user base. This makes the users in your directory accessible by any other application that uses eIAM.

Configuring the Solution

Michael Reed (the Service Desk Manager) has confirmed with Paul Kim (the IT Director) that he will be configuring eIAM to use Active Directory to bring CA Unicenter Service Desk and Workflow IDE into compliance with the new security standard.

Configure eIAM to use Active Directory

In this task, you will configure eIAM to reference Active Directory as its external directory for users. This will provide the framework for a single user authentication tool for CA Unicenter Service Desk and CA Workflow.

1. Log in to eIAM as EiamAdmin with the password you entered during the configuration process.

2. Click the Configure tab and enter the following information to configure Embedded IAM Server Configuration to reference Active Directory as the External Directory.

> **Type**: Microsoft Active Directory

> **Host**: 192.168.10.5

> **Base DN**: cn=Users,dc=bsolab,dc=forwardinc

> **User DN**: cn=administrator,cn=Users,dc=bsolab,dc=forwardinc

> **Password**: camdb

> **Retrieve Exchange Groups as Global User Groups**: Select this check box
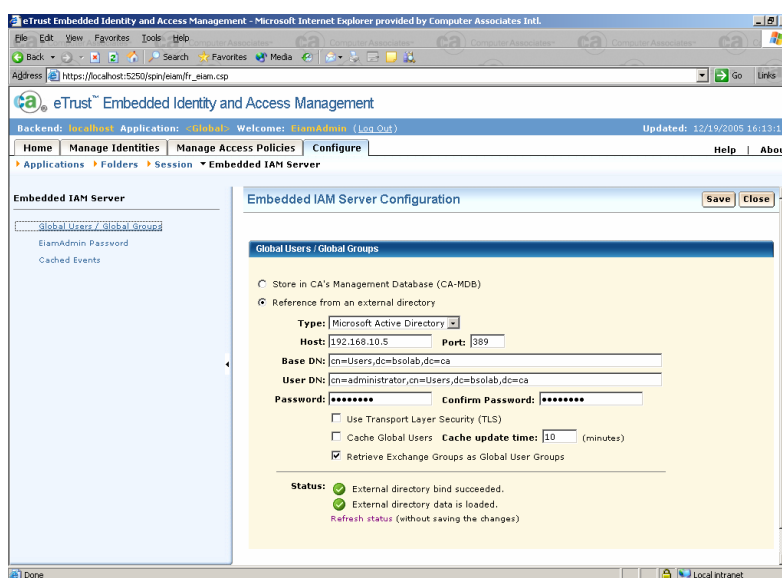
3. Click Save.
4. Refresh the status to ensure that the bind and the data load are successful.
5. Test the integration by searching for users on the Manage Identities tab.

Create Users and Groups in Active Directory

In this task, you will add users and groups to Active Directory for later use.

1. From the Start menu, choose Programs, Administrative Tools, Active Directory Users and Components.
2. Log in to Active Directory.
3. Create an Active Directory group by completing the following steps:

a. Open Active Directory Users and Computers.

b. Expand Domain.

c. Right-click Users and Create a new Group in Active Directory named Employee. This group will be used to link to CA Unicenter Service Desk.

4. Create an Active Directory user by completing the following steps:

a. Right-click Users and select New, User from the menu.

b. Add yourself as a user.

c. Add the user to the new Employee.

5. Validate the Active Directory user in eIAM.

6. Validate the Active Directory group in eIAM.

Configure CA Unicenter Service Desk LDAP

In this task, you will configure the LDAP options in CA Unicenter Service Desk to reference Active Directory for LDAP integration.

1. Log in to the CA Unicenter Service Desk web client as the ServiceDesk user.

2. Click the Administration tab and view the Employee Access Type.

3. Verify that there is no LDAP group option on the Web Authentication window.

4. In the Options Manager, modify the LDAP Options as follows:

> **Ldap_dn.** administrator *(Active Directory user)*

> **Ldap_enable**. Yes

> **Ldap_enable_auto**. Yes

> **Ldap_enable_groups**. Yes

> **Ldap_group_object_class**. group

> **Ldap_host**. COREWIN2K3 *(hostname)*

> **Ldap_port**. 389

> **Ldap_pwd. camdb** *(Active Directory user password)*

> **Ldap_search_base**. cn=Users,dc=bsolab,dc=forwardinc

> **Ldap_service_type**. Active Directory

> **Ldap_user_object_class**. person
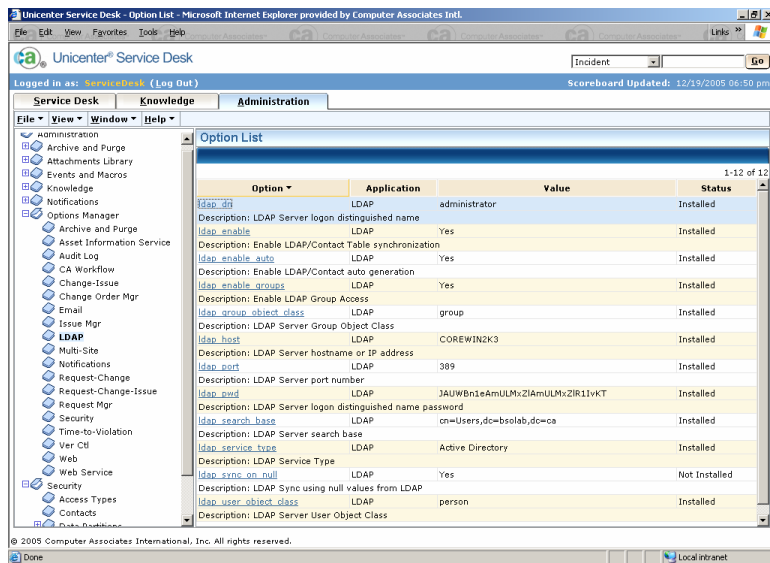
5. Log out of CA Unicenter Service Desk and Recycle Services.

Test the Integration

Complete the following steps to verify that the eIAM integration has been set up correctly:

1. Run ldap_test to verify the LDAP integration is successful.

2. Log in to CA Unicenter Service Desk and make sure the LDAP menu options are available.

Configure Access Types to Link to LDAP Groups

In this task, you will configure the LDAP options in CA Unicenter Service Desk to reference Active Directory for LDAP integration.

1. In Administration, modify the Employee Access Type.

   You should see a new field under the web authentication tab named LDAP Access Group.

2. Modify this to reference the Employee Active Directory group.

3. From the CA Unicenter Service Desk menu, select File, New Contacts From LDAP to search for contacts using LDAP.

4. Search for your user.

5. Select this contact and click save to bring over your user.

   Note: The access type is automatically set.

Integration Summary

CA Unicenter Service Desk uses the MDB to store contact information. CA Unicenter Service Desk also features an LDAP integration, which allows it to create new contacts from an LDAP server and synchronize existing contacts with the directory. eTrust Identity and Access Management is CA's solution to centralized user management. If you have several CA solutions installed, they all may be using eIAM to store identities and access policies.

eIAM may be configured to either reference an external (LDAP) directory or use the MDB to store user information. eIAM has an LDAP interface for use when it is configured to use the MDB. The tables used by eIAM in the MDB are different from the ones used by CA Unicenter Service Desk.

CA Workflow always uses eIAM for its user information and authentication. Work items can be assigned only to users known by eIAM, and only users defined by eIAM may access CA Workflow's and Worklist's web interface.

Chapter 17: Accessibility Management – Integrating with CA SiteMinder

What is CA SiteMinder?

CA SiteMinder® provides a centralized security management foundation that enables user authentication and controlled access to web applications and portals. CA SiteMinder delivers the market's most advanced security management capabilities and enterprise-class site administration, allowing you to reduce IT operational costs and enabling greater IT control and security. CA SiteMinder enables the secure delivery of essential information and solutions to your employees, partners, suppliers and customers, and scales with your growing business needs.

Features

CA SiteMinder® provides the following features:

■ Single Sign-On

■ Strong Authentication Management

■ Centralized Policy-Based Authorization and Audit

■ Identity Federation

■ Enterprise Manageability

CA SiteMinder integrates with industry-leading directory services and user stores, eliminating redundant administration of user information. This integration simplifies administration and provides unique and comprehensive security capabilities. CA SiteMinder fully leverages existing user directories, from leading LDAP directories and relational databases to mainframe security directories.

The CA SiteMinder Integration

CA SiteMinder is the industry's leading directory-enabled web access management system. The CA SiteMinder integration enables security and CA Unicenter Service Desk administrators to assign authentication schemes and define and manage authorization privileges to the pdmweb.exe URL, by creating rules and policies to implement the

authorization permissions against the appropriate user store (that is, LDAP or Administrative Domain (AD)).

Integration Points and Functionality

The integration protects the pdmweb.exe URL in CA SiteMinder by creating a realm, rules, and policy. A rule identifies, and allows or denies access to specific resources that are included in the policy. A CA SiteMinder policy binds rules and responses to users, groups and roles. The responses in a policy enable the solution to customize the delivery of content for each user.

Policies reside in the policy store, which is the data source that contains all the CA SiteMinder entitlement information. When CA Unicenter Service Desk users (analysts, administrators, and users) try to access the /pdmweb.exe solution, CA SiteMinder displays a login window and challenges the user for credentials. The user enters credentials and submits them, and is then authenticated against the CA SiteMinder Policy server. CA SiteMinder sets the REMOTE_USER attribute and redirects the request to the /pdmweb.exe URL, so the CA Unicenter Service Desk web interface is available to users to start their work.

The following steps provide an overview of how the CA Unicenter Service Desk and CA SiteMinder integration works:

1. The user attempts to access a protected resource, which is the CA Unicenter Service Desk web interface.
2. The user is challenged for credentials and presents them to the Web Agent or to the secure Proxy Server.
3. The user's credentials are passed to the policy server.
4. The user is authenticated against the appropriate user store (that is, LDAP or AD).
5. The policy server evaluates the user's entitlements and grants access.
6. User profile and entitlement information is passed to the product.
7. The user receives access to the secured CA Unicenter Service Desk web interface.

Integration Points from CA Unicenter Service Desk

The following are integration points from CA Unicenter Service Desk:

■   CA Unicenter Service Desk can use external authentication definitions based on CA SiteMinder rules and policies for each CA Unicenter Service Desk Access Type.

Integration Points from CA SiteMinder

The following are integration points from CA SiteMinder:

■   User profile and entitlement CA SiteMinder information is passed to CA Unicenter Service Desk to grant users access to the web interface.

Integration Value

The CA SiteMinder integration provides the following value:

■ CA SiteMinder offers the type of solution organizations need to meet the challenge of building and managing secure websites. CA SiteMinder provides the essential security services required to meet this challenge, while also including management features and technical capabilities that can reduce the total cost of ownership.

■ CA SiteMinder integrates with industry-leading directory services and user stores, eliminating redundant administration of user information. This integration simplifies administration and provides unique and comprehensive security capabilities. CA SiteMinder fully leverages existing user directories, from leading LDAP directories and relational databases to mainframe security directories.

■ CA SiteMinder supports a comprehensive set of password services including password composition, dictionary checking, and expiration rules, allowing you to implement robust password management rules. When combined with CA Identity Manager, it provides self-service, forgotten password services, and password synchronization. The integration complements the CA Unicenter Service Desk self-service functionality.

How the Integration Works

The following diagram illustrates how the CA SiteMinder integration works:
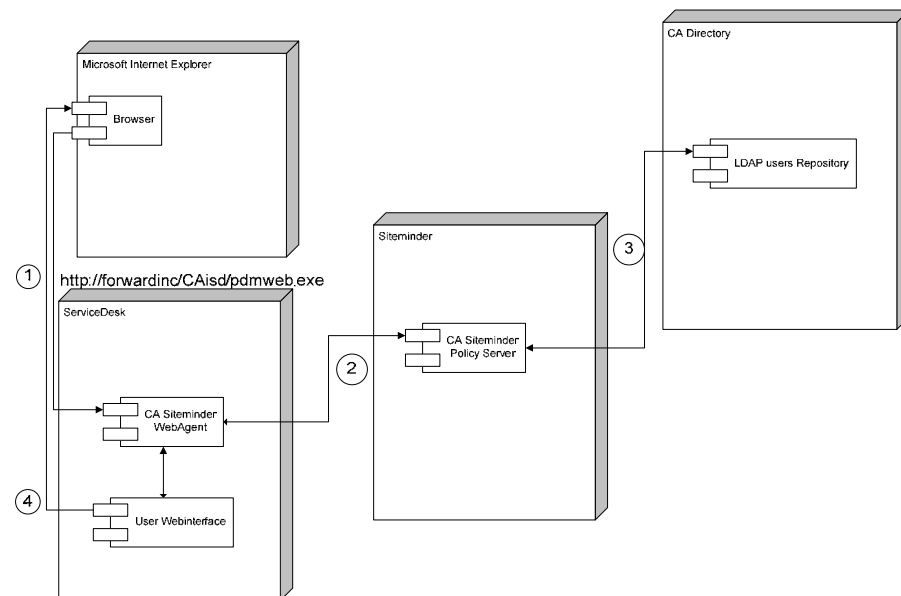
The following information applies to the previous diagram:

1. CA Unicenter Service Desk has been configured to use external authentication as a security setting for its users, and the CA SiteMinder Web Agent has been configured to

protect several web resources for the organization. A user attempts to start the CA Unicenter Service Desk's web interface.

Note: In this example, the Microsoft Internet Explorer browser is being used.

2. The CA SiteMinder Web Agent checks with the CA SiteMinder Policy Server to see if the /CAisd/pdmweb.exe is a protected resource. If so, the user is challenged for credentials.

3. The user enters the credentials in the CA SiteMinder login window. The CA SiteMinder Policy Server validates the credentials against the CA Directory-LDAP repository of users, evaluates the user's entitlements, and grants the appropriate access.

4. The user context is passed to CA Unicenter Service Desk and the user is granted access to the secured CA Unicenter Service Desk web interface.

Example of the CA SiteMinder Integration

Configuring the Solution for the CA Unicenter Service Desk Web Interface

Michael Reed, the Service Desk Manager, is working together with the Forward, Inc. security team to implement the CA SiteMinder integration with CA Unicenter Service Desk to centralize the management of user entitlements for CA Unicenter Service Desk customers, partners, and employees across all web servers, through shared services. This will enforce security policies across the enterprise and eliminate the need for redundant user directories.

In this example, the following information is assumed:

■   CA Unicenter Service Desk is running on IIS as a web server and is using Microsoft Internet Explorer as a web browser.

■   CA SiteMinder and CA Unicenter Service Desk have been successfully installed and configured.

■   LDAP or AD functionality is already running at Forward, Inc., and all users that can potentially connect to CA Unicenter Service Desk belong to the LDAP or Administrative Domain (AD) domain named DOMAIN.

Configure CA Unicenter Service Desk

In CA Unicenter Service Desk, the user authentication must be configured to allow for external authentication. Complete the following steps to configure the user authentication:

Note: You must complete these steps for each Access Type in the system (that is, Administrator, Analyst, and so forth) after you have verified that the integration is working correctly. Forward, Inc. uses IIS as the web server.

1. Start the Administrative interface.
2. Select Administration, Security, Access Type, Employee, Detail.

3. Click the Web Authentication tab.

4. Select both the Allow External Authentication and Open-always allow access check boxes.

5. Click OK.

Set up External Authentication using IIS

Complete the following steps to set up external authentication using IIS:

1. Create a proxy user that can access the CA Unicenter Service Desk virtual directory.

   Note: An example proxy user can be *smproxy*. This user should be created in Active Directory if this is one, or on the local server hosting CA Unicenter Service Desk.

2. Make the proxy user *smproxy* a member of Administrators and Domain Users of the local server, or of the AD Domain.

3. From the Start menu, choose All Programs, Administrative Tools, Domain Security Policy, User Rights Assignment.

4. On the right side of the window, double-click the option named Act as part of the Operating System.

5. Add the user or Group DOMAIN\smproxy using the following steps:

   a. Using the IIS server admin console, select the virtual directory CAisd.

   b. Right-click the virtual directory to select properties of the web site.

   c. Click the Directory security tab.

   d. Make sure the Enable anonymous access check box is selected.

   e. From the browse list, select the user DOMAIN\smproxy and click OK.

6. Stop and start IIS services.

Configure CA SiteMinder

Complete the following steps to configure CA SiteMinder:

1. Log in to the CA SiteMinder Policy Server as the user siteminder.

2. Click the System tab and select Agent Conf Objects (that is, smDemoAgentConfig).

3. Modify the parameters (Parameters Name) for the webagent configuration object as follows:

   > **DefaultUserName**. Uncomment the parameter name by removing the pound sign (#) from the name, if commented, and set the value of this parameter to the name of the proxy user account defined on the IIS web server host (that is, *smproxy*).

   > **DefaultPassword**. Uncomment the parameter name by removing the pound sign (#) from the name, if commented, and set the value of this parameter to the password of the proxy user account defined on the IIS web server host (that is, *smproxy*).

> **UseAnonAccess**. Set this parameter value to yes. The default value is no.

> **SetRemoteUser**. Set this parameter value to yes. The default value is no.

4. From CA SiteMinder Policy Server, click the System tab, System Configuration.

5. Right-click Domains, Create Domain, and enter the following information. The name can be different.

> **Name**. servicedesk

> **Global Policy Apply**. Select this field.

> **User Directories**. Add the Directory name that contains the users who will be logging in to CA Unicenter Service Desk.

6. Click OK.

7. From the CA SiteMinder Policy Server, click the Domains tab.

8. Expand servicedesk.

9. Right-click Realms and create a realm with the name *servicedesk.realm*. The name of the rule can be different.

10. Click the Resource tab and look up the agent. In this example, use the agent *smdemoagent*.

11. Enter the following information:

> **resource filter**. /CAisd

> **Authentication Schema** drop-down list. smDemoFormBased

> **Protected** field. Select on the Default Resource Protection Panel

12. Click OK.

13. Right-click the *servicedesk.realm* previously created and select Realm, Create Rule.

14. Create a rule with the following properties:

   Note: The name of the rule can be different.

> **Name**. servicedesk.protect

> **Realm and Resource**. Select *servicedesk.realm* from the drop-down list.

> **Resource**. /*

> **Action**. Select the Web Agent actions, and select Get and Post

> **Allow Access** and **Enabled** fields. Select both on the AllowDeny and Enable/Disable panel.

15. Click OK.

16. Right-click the *servicedesk.realm* previously created and select Realm, Create Rule.

17. Create a rule with the following properties:

    Note: The name of the rule can be different.

    > **Name**. servicedesk.onAuthAccep.setremoteuser

    > **Realm and Resource**. Select servicedesk.realm from the drop-down list.

    > **Action**. Select the Web Authentication events, and select OnAuthAccep from the drop-down list.

    > **Allow Access** and **Enabled** fields. Select both on the AllowDeny and Enable/Disable panel.

18. Click OK.

19. From CA SiteMinder Policy Server, click the Domains tab, expand servicedesk, right-click Policies, and select Create Policy.

    A new Policy Properties window opens.

20. Enter the following information for the policy:

    > **Name**. servicedesk.policy

    > **Enabled**. Select this field.

    > **Users**. Click Add, and enter All in the entry field.

    Note: This is just for testing. The real entry would map to a user group or be based on a user attribute.

21. Click the Rules tab, click Add/Remove Rules, and add the rules previously created:

    > Servicedesk.protect

    > servicedesk.onAuthAccep.setremoteuser

    Note: The name of the rule can be different.

Test the Integration

Complete the following steps to verify that the CA SiteMinder integration has been set up correctly:

1. From the Start menu, choose All Programs, CA, Service Desk, Service Desk Web Client.

    A SiteMinder log in window opens.

2. Enter the user and password to log in to CA Unicenter Service Desk.

The CA Unicenter Service Desk web interface main window opens.

Integration Summary

Organizations can manage access to CA Unicenter Service Desk by integrating with CA SiteMinder. Real-time transactional security and integrated web services with CA SiteMinder eTelligent rules enables security policies that evaluate dynamic data from a variety of local or external sources, including web services and databases, in real time. Cost and complexity are reduced by eliminating advanced security logic from web applications and centralizing it within CA SiteMinder policies.

Chapter 18: Service Quality Management – Integrating with CA SQM

What is SQM?

CA's Service Quality Management (CA SQM) is a solution that integrates the following products:

■   CA Wily Introscope

■   CA Wily Customer Experience Manager (CA Wily CEM)

■   CA Unicenter Service Desk

■   CA Unicenter Service Metric Analysis (CA Unicenter SMA)

This integration creates an unparalleled solution to ensure superior service delivery for web-based transactions governed by customer experience service level agreements (SLAs). By linking application performance management, service level management, and incident and problem management disciplines, SQM creates a closed-loop system that IT can use to establish and publish SLAs and operational level agreements (OLAs), monitor and report on service level compliance, and streamline problem diagnosis and resolution.

The SQM Integration

The integration between CA Wily Introscope and CA Wily CEM with CA Unicenter Service Desk and CA Unicenter SMA was created to provide advanced incident and problem management and service level management. The code is available for download without additional charge to customers with valid CA/Wily licenses for the applicable product components and current maintenance agreements. Integration between CA Wily CEM and CA Unicenter Service Desk is included with CA Wily CEM 4.0, and there is no requirement to download the additional software.

Effectively managing service quality begins with negotiating and defining measurable SLAs. Using CA Unicenter Service Metric Analysis, service level managers can incorporate user experience metrics and web application infrastructure metrics into service level management processes to track SLAs negotiated with the business. These SLAs are typically expressed in terms of transaction performance and availability, and incident resolution time. CA Wily Introscope and CA Wily Customer Experience Manager monitor service performance from both the customer and web infrastructure viewpoints, and report SLA-governed performance results to CA Unicenter Service Metric Analysis.

Based on thresholds and SLAs set for user transactions, an incident in CA Wily CEM will query CA Wily Introscope for correlated information relating to the incident, and automatically create a service desk ticket in CA Unicenter Service Desk. The ticket is created at the specified request area in CA Unicenter Service Desk, complete with a URL link to CA Wily CEM and CA Wily Introscope triage and diagnostic information in the application infrastructure supporting the transaction. The service desk representative can immediately access critical information about the nature, severity, and business impact of the incident and provide root-cause information to specialist application support teams.

As a result, application support can drill down into CA Wily Introscope and CA Wily CEM and close the ticket within CA Unicenter Service Desk after the issue is resolved, before a customer has even reported the incident. In addition, with key customer transaction and web infrastructure performance data provided by CA Wily Introscope and CA Wily CEM, business managers can regularly run reports and analytics to more accurately measure customer success and better ensure compliance with service level objectives.

Product Requirements

To take advantage of SQM, you must have a minimum of two products as shown in the following table:

Business Value/IT Function	Minimum Product Requirement	Recommended SQM Component Product
Proactive incident, problem, and request management	CA Unicenter Service Desk r11.2 CA Wily CEM 4.0 (SQM software download is *not* required)	CA Wily Introscope 7.1 for evidence collection and root-cause information in trouble ticket.
Service level, management using web application performance metrics	CA Unicenter SMA r11.1 CA Wily Introscope 7.1 (SQM software download *is* required)	CA Wily CEM 4.0 for modeling and monitoring business process SLAs from user viewpoint.
Service level management using user transaction metrics	CA Unicenter SMA r11.1 CA Wily CEM 4.0 (SQM software download *is* required)	CA Wily Introscope 7.1 for modeling and monitoring SLA and OLAs from web application infrastructure metrics.

In this section, we will focus on the integration between CA Wily Customer Experience Manager (CA Wily CEM) and CA Unicenter Service Desk r11.2, which offers customers proactive incident, problem, and request management.

CA Wily Customer Experience Manager (CEM)

CA Wiley CEM is an appliance-like product that measures web application performance, from the business process to the component level, for each customer by user name. This

enables both your business managers and IT staff to understand and resolve performance problems before users call for support.

Web Application Management Challenges

Unlike technical monitoring of applets or SQL statements, CA Wily CEM enables you to monitor user activities, including login, account update, or purchase transactions at the business process level. This provides you with immediate insight into customer experience.

Features

This product offers real-time transaction monitoring, including performance trending, automated reporting, customized alerts, and integration with other CA and multi-vendor solutions.

Platform Support

Any web-based application using HTTP or HTTPS, whether it was built on Java, .NET, mainframe, or other technology, is supported.

The CA Unicenter Service Desk Integration with CEM

Wily CEM provides business management, customer service, application support, and IT operations a new level of real-time visibility into customer (internal and external) transactions. This visibility enables IT organizations to deliver high-quality customer transactions at superior levels of operating efficiency.

Wily CEM monitors customer transactions to isolate the causes of problems in the data center. It measures the performance and quality of customer transactions, identifies defects and variance, and quantifies the impact on customers and the business. By proactively detecting trends in degraded transaction response times and providing a variety of actionable reports, Wily CEM allows you to take action before a problem occurs or service level agreements (SLAs) are out of compliance.

Wily CEM can integrate with several products, and in a variety of scenarios.
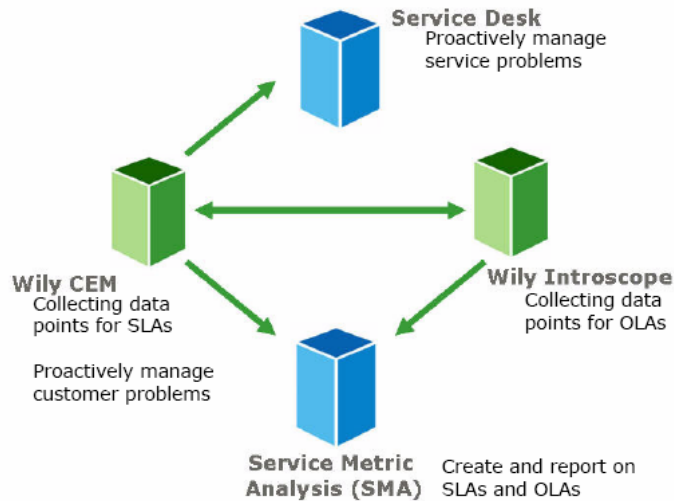
To enable Wily CEM incidents to be created in CA Unicenter Service Desk, you need to configure and enable the CA Unicenter Service Desk plug-in. This establishes communication and defines the way incidents appear in CA Unicenter Service Desk. The integration has been developed using web services which make CEM a service-aware application.

Integration Points from CEM

The following is the integration point from CEM:

■   Automatic creation of request or incidents (if the customer running on ITIL) in CA Unicenter Service Desk based on CEM incident Generation Rules functionality.

Integration Points from CA Unicenter Service Desk

The following is the integration point from CA Unicenter Service Desk:

■   The analyst can launch, in context, the CEM Incident Detail window from the CA Unicenter Service Desk Incident/Request Detail window, Attachment List.

Integration Value

The SQM integration provides the value:

■   Automatic incident detection and problem correlation. User experience issues detected by Wily CEM can automatically trigger the creation of an incident or request in CA Unicenter Service Desk, including diagnostic information about back-end systems from Introscope. This reduces time and resource usage during the incident solving process.

- Allow the service desk staff to isolate the real root cause of the incident based on the information from CEM.

- Integration of real-world metrics for SLA and OLA management. The metrics that reflect user experience and application performance are automatically captured by Unicenter Service Metric Analysis for reporting and analysis against SLAs and OLAs.

- CA Service Quality Management connects service support and service delivery processes to create a robust platform for negotiating and tracking SLAs, and expediting recovery from performance incidents. It provides deep insight into the health and availability of critical business services and brings a new level of discipline and accountability to service level management and incident and problem management. CA Service Quality Management makes it easier to ensure superior service delivery, minimize web application downtime, and improve user experiences.

How the Integration Works

The following diagram illustrates how the SQM integration works:
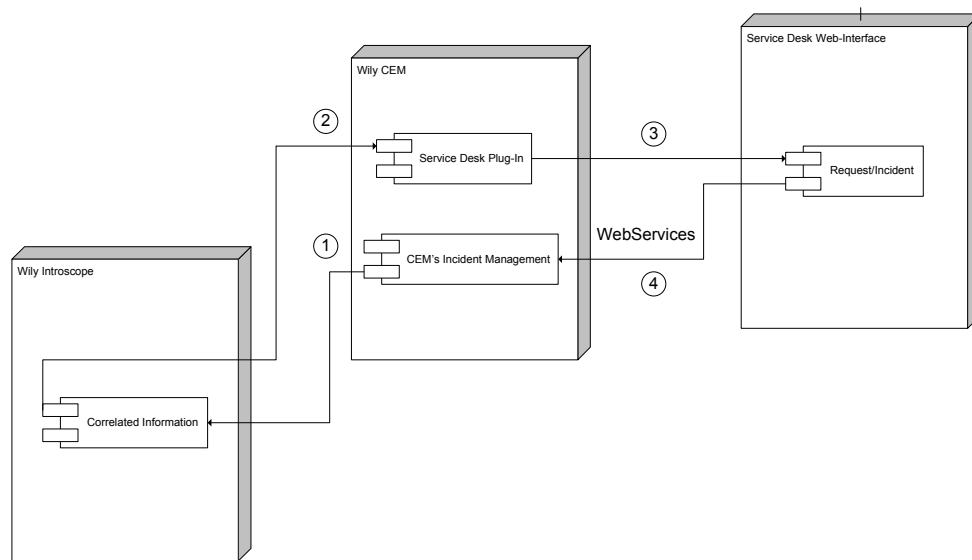
The following information applies to the previous diagram:

1. Based on thresholds and service level agreements set for user transactions, an incident in CA Wily CEM will query CA Wily Introscope for correlated information relating to the CEM incident.

2. Taking into account the correlated information in Wily Introscope, Wily CEM collects all required information defined in the CA Unicenter Service Desk plug-in component.

3. Based on the information in the plug-in, CEM makes a call to CA Unicenter Service Desk Web Services and creates a request or incident (depending on whether the customer has selected the ITIL option when configuring CA Unicenter Service Desk).

4. The request and incident is created with the specified request area or category in CA Unicenter Service Desk, complete with a URL link to CA Wily CEM. It also includes CA Wily Introscope triage and diagnostic information for the application infrastructure

supporting the transaction. The service desk representative can immediately access critical information about the nature, severity, and business impact of the incident, and provide root-cause information to specialist application support teams. Application support can drill down into CA Wily Introscope and CA Wily CEM and close the request and incident within CA Unicenter Service Desk after the issue is resolved, before a customer has even reported the incident.

Example of the SQM Integration

Business Challenge

Denis Markson, the call center director at Forward, Inc., has received survey results that report a low level of customer satisfaction from the Forward, Inc. insurance services customers who submit online transactions. As reported in the survey, the response times of several options on the Forward, Inc. web page are very slow, and sometimes the transactions time out. Customers are unhappy because they have to go back and start the transaction again from the beginning, resulting in wasted time and dissatisfaction with the online service.

CA Approach

CA advises Forward, Inc. to use CA Service Quality Management to solve the business challenge that the Forward, Inc. insurance services department is experiencing. CA SQM allows IT teams and line of business managers to collaboratively and efficiently ensure superior quality of service from their business-critical web solutions. By linking service support and service delivery disciplines, the product creates a closed-loop system that allows organizations to establish and publish service level agreements and SLAs (SLAs) and operational level agreements (OLAs), monitor and report on service-level compliance, and perform problem diagnosis and resolution.

This solution includes CA Wily Customer Experience Manager (CA Wily CEM), a comprehensive product for tracking all customer transactions and business processes. The solution allows IT teams to detect incidents proactively and prioritize their resolution by measuring the business impact. It can also include CA Wily Introscope®, the industry-leading product for managing web application performance, with proven capabilities for detection, triage and diagnosis of web application performance issues. Working together, the SQM solution provides IT teams with unparalleled views into the health and availability of critical business services and the web application infrastructure.

Best Practice

CEM's incidents need to be assigned to the proper request and incident category so the CA Unicenter Service Desk ticket will be transferred to the proper service desk analyst.

Configure the Integration from CEM

Enable the CA Unicenter Service Desk integration by first configuring the Wily CEM plug-in. Optionally, you can create CA Unicenter Service Desk request areas to categorize Wily CEM incidents from other incidents or requests in Unicenter Service Desk.

1. Log in to Wily CEM as an Administrator.
2. On the right part of the window, click Setup.
3. Click the Plug-ins tab.

   The Plug-ins page opens.

4. Select Unicenter Service Desk.

   The Plug-ins window opens.

5. Enter the following information:

   > **Enable Unicenter Service Desk Plug-in**. Select this check box. This allows Wily CEM to send incident information to CA Unicenter Service Desk.

   > **Hostname**. Enter *ForwardInc*. This is the CA Unicenter Service Desk Server that hosts Wily CEM.

   > **Port**. Enter *8080*. This is the Tomcat port selected for CA Unicenter Service Desk.

   > **User Name**. Enter *ServiceDesk*, or a user authorized to log in to CA Unicenter Service Desk.

   > **Password**. Password for the previously selected user.

   > **Create ITIL incidents in Service Desk**. Select this check box (I=Only if customer is running ITIL; otherwise the integration creates a Service Desk request).

   > **Request Area**. CEM1. The Request Area can have any name.

   > **Request Filter**. Enter */* as the filter.

   > **Request Summary**. Enter *for business_process>,<business_transaction)*

   > **Request Description**.

      - Incident ID: <incident_id>

      - Incident Name: <defect_name>

      - Business Process Name: <business_transaction>

      - Affected number of users: <user_count>

      - Defect Count: <defect_count>

- Incident Link: <incident_link>

- Create a Service Desk request if incident Severity is: Select the proper severity value, either low, Moderate, Severe, or Critical.

6. Click Save.

   Note: Future Wily CEM incidents matching your saved configuration will trigger service requests in CA Unicenter Service Desk, effective immediately. If there are any open incidents in Wily CEM, they will not trigger a CA Unicenter Service Desk request.

The Request Filter

You can use the request filter to test the integration with a single business process and business transaction first, rather than allowing all incidents to be forwarded to CA Unicenter Service Desk immediately.

The following table provides some syntax examples:

Syntax Example	Description
Request Filter=BusinessProcess Name/BusinessTransaction Name	Creates a request in CA Unicenter Service Desk only if the incident occurs for the specified business process and business transaction.
Request Filter=BusinessProcess Name/*	Creates a request in CA Unicenter Service Desk for all incidents and any business transaction triggered for the specified business process.
Request Filter=BusinessProcess Name/Tra*	Creates a request in CA Unicenter Service Desk for all incidents and any business transaction that begins with "Tra" triggered for the specified business process.
Request Filter=*/*	Creates a request in CA Unicenter Service Desk for incidents associated with all business processes and all business transactions.

Example: Request Filter Containing a Slash ("/") Character

If the business process or business transaction name contains a slash ("/") character, you must replace it in the filter with a double-slash ("//"). For example, if you have the following business process or business transaction names:

- business process name = Trade

- business transaction name = Sell Big/Medium Stocks

The filter should be the following:

- Request Filter=Trade/Sell Big//Medium Stocks

Example: Request Filter Containing a Comma (",") Character

If the business process or business transaction name contains a comma (",") character, you must replace it in the filter with a slash-comma ("/,"). For example, if you have the following business process or business transaction names:

■   business process name = Trade

■   business transaction name = Sell Big, Medium Stocks

The filter should be the following:

■   Request Filter=Trade/Sell Big/, Medium Stocks

Request Summary for CA Unicenter Service Desk Summary

For the Request Summary that will appear in the CA Unicenter Service Desk Summary, use the following variables to create a customized summary:

■   <defect_name>

■   <business_process>

■   <business_transaction>

You can rearrange the variables to create your own request summary. Each variable will be replaced with actual values when the request summary appears in CA Unicenter Service Desk. For example, <defect_name> for <business_process>,<business_transaction> appears as: Slow Time for Support Desk, KB Lookup. *Business Transaction: <business_transaction> Defect: <defect_name>* appears as: Business Transaction: CheckOut Defect: Slow Time

Request Summary for CA Unicenter Service Desk Description

For the Request Description that will appear in the CA Unicenter Service Desk Description, use the following variables to create a customized description:

■   <incident_id>

■   <defect_name>

■   <business_process>

■   <business_transaction>

■   <user_count>

■   <defect_count>

■   <incident_link>

You can rearrange the variables to create your own request description. Each variable will be replaced with actual values when the request description appears in CA Unicenter Service Desk. The <incident_link> variable will provide a link in CA Unicenter Service Desk. You can use the link to go directly to the incident overview information (CEM, Incident Management, Incident Overview).

Severity

Severity enables the selected CEM incident levels to create requests or incidents in CA Unicenter Service Desk. Select the appropriate check boxes:

■ Low

■ Moderate

■ Severe

■ Critical

Configure the Integration from CA Unicenter Service Desk

If you want a separate request area for the incidents reported by Wily CEM, you must create a request area in CA Unicenter Service Desk. Complete the following steps:

Note: For information about creating and configuring request areas, see the CA Unicenter Service Desk online help.

1. Create a request area (for example, CEM_reported), as illustrated in the following sample window:

2. Attach three properties to the request area:

    a. Business Process

    b. Business Transaction

c.   Incident

Test the Integration

Once an incident matching the information in the filter definition for the plug-in has been recorded in Wily CEM, a CA Unicenter Service Desk request or incident will be created for the Request Area defined in the plug-in. Complete the following steps:

1. Open the incident or request.

    Your window will look similar to the following example.

2. Click the Incident Link in CA Unicenter Service Desk.

    The Wily CEM Incident's related information is launched in context.

3. Navigate within Wily CEM to troubleshoot and discover the root cause of the incident, as illustrated in the following sample window:
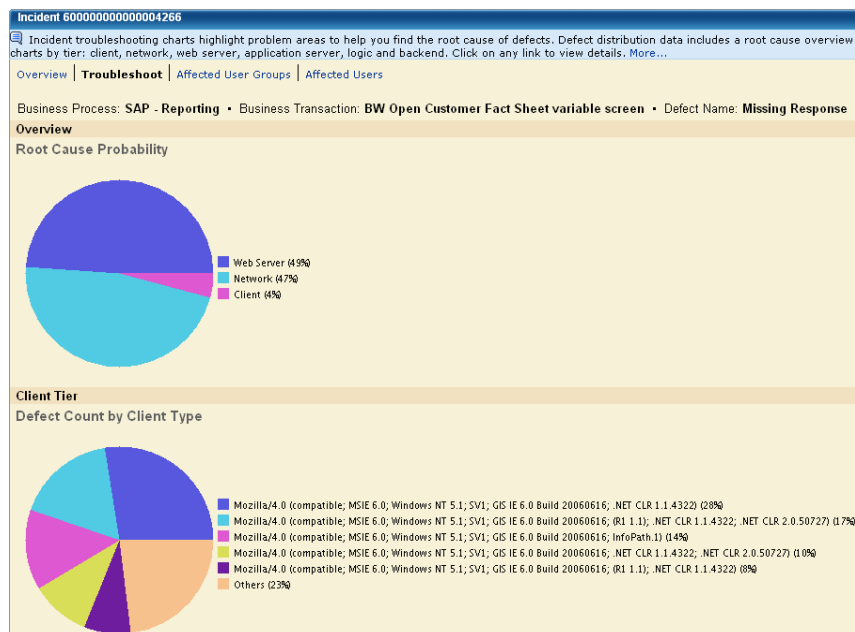
Integration Summary

Wily CEM monitors customer transactions to isolate the causes of problems in the data center. It measures the performance and quality of customer transactions, identifies defects and variance, and quantifies the impact on customers and the business. By proactively detecting trends in degraded transaction response times and providing a variety of actionable reports, Wily CEM enables you to take action before a problem occurs or service level agreements (SLAs) are out of compliance.

Based on thresholds and SLAs set for user transactions, an incident in CA Wily CEM will query CA Wily Introscope for correlated information related to the CEM incident, and automatically creates a service desk request or incident in CA Unicenter Service Desk. Taking the specified information in the CA Wily CEM-Service Desk plug-in, the request or incident is created using CA Unicenter Service Desk Web Services in CA Unicenter Service Desk, complete with a URL link to CA Wily CEM and CA Wily Introscope triage and diagnostic information related to the application infrastructure supporting the transaction. The service desk representative can immediately access critical information about the nature, severity, and business impact of the incident and provide root-cause information to specialist application support teams. Application support can drill down into CA Wily Introscope and CA Wily CEM and close the request and incident within CA Unicenter Service Desk after the issue is resolved, proactively resolving the incident before it impacts the customer.

Chapter 19: Integrating with Third-Party Products

CA Unicenter Service Desk Integration Methods

CA Unicenter Service Desk has been designed to interact with many software solutions, both CA solutions and third-party products. Additionally, CA Unicenter Service Desk is often integrated with products that have been developed in-house by CA customers.

Most of the methods used to integrate CA Unicenter Service Desk with other products are described in this section. Some of these techniques are available only to CA developers, CA Technology Services, and CA certified partners, while others can be utilized directly by customers. This section is *not* intended to provide instructions on how to use the methods, but rather to provide examples and explanations that help provide a basis for deciding about whether an integration method might be suitable for a particular need.

Important! Your site is responsible for testing and maintaining any custom integration, so thorough testing is strongly recommended.

Web Services Application Programming Interface (API)

API Points and Value

This section provides a description of the generic CA Unicenter Service Desk Web Services API, with a focus on what can be accomplished with this toolkit for integration purposes. The two primary areas where the CA Unicenter Service Desk Web Services API can be utilized are in the integration of the following:

1. Separate products (both CA and third-party)
2. New features for CA Unicenter Service Desk

The value of this functionality is that when you want to perform a custom implementation, it can be implemented quickly on a wide variety of platforms using a large number of different programming environments. Alternatively, there are some CA Smart-certified partner products that leverage the Web Services API to provide the desired integration to CA Unicenter Service Desk from a wide variety of third-party products.

This section will illustrate how to use the CA Unicenter Service Desk Web Services API to integrate a new feature into the CA Unicenter Service Desk web interface. In the example, a new feature will be added to assist the user in more efficiently managing the incidents and problems that have been attached to a change order.

How the API Works

The CA Unicenter Service Desk Web Services API is hosted on the Apache Tomcat application server that is installed as part of the CA Unicenter Service Desk installation. As part of the CA Unicenter Service Desk server configuration process, the Apache Axis SOAP engine is applied to the Tomcat instance. Once this process has completed and CA Unicenter Service Desk is started, and assuming you selected port 8080 for your Tomcat web server, the WSDL for the Web Services API is available by navigating to http://<SERVER_NAME>:8080/axis and clicking the View the list of deployed Web services link.

Setting Up, Configuring, and Testing the API

Through the use of the API, you can implement an unlimited number of integrations. As previously mentioned, in this example, we will discuss the integration and addition of a new feature so users can more efficiently manage attached incidents and problems from a parent change order. The functionality to be implemented provides the ability to close, resolve, or reopen attached incidents and problems, in addition to passing a comment to be recorded in the activity history when the status changes. This custom feature allows a change analyst to be able to more easily control and update related tickets from a central interface.

Set Up the Solution

The following diagram illustrates the main components of this custom integrated functionality:
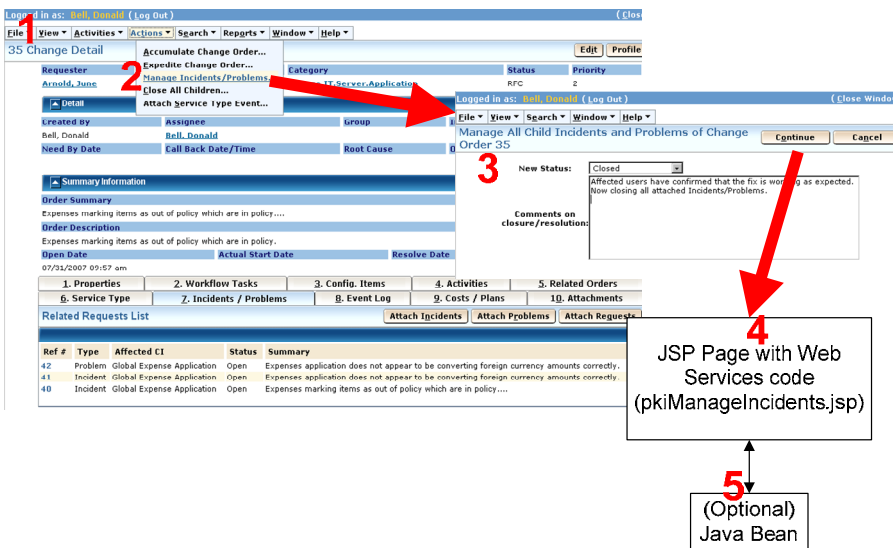
The following information applies to the previous illustration:

1. **Change Order Detail Page**. No modifications were made to this component. However, it is used to launch the newly integrated feature.

2. **menubar_sd.htmpl**. This file is provided as part of the default installation of CA Unicenter Service Desk. It was modified by adding a new option named Manage Incidents/Problems…. This is displayed in the Change Order Detail page under Actions. This new menu option will launch the zManageIncidents.htmpl file with the correct context. The newly created zManageIncidents.htmpl form is described below.

3. **zManageIncidents.htmpl**. Create the zManageIncidents.htmpl web form in the format of the chg_close_all_child.htmpl form. Significant modifications will be necessary to create an HTML form element in the page and then provide a JavaScript function to collect all the necessary data from the form elements, and then submit them to a custom created JSP page which will then run the CA Unicenter Service Desk web services code.

4. **pkiManageIncidents.jsp**. Create a JSP page named pkiManageIncidents.jsp page that uses the PKI login functionality available in CA Unicenter Service Desk web services and then proceeds to collect the information that was sent from the zManageIncidents.htmpl form. The code will then cycle through all of the attached incidents and problems, and set them to the status specified by the user at the zManageIncidents.htmpl file while adding the provided comment to the activity history of each incident and problem.

5. (Optional) **JavaBean**. Optionally, the code logic built into the JSP page could be pulled out of the JSP page and placed into a JavaBean. This would have the effect of making the newly built functionality more reusable, among other benefits.

Business Challenges

Not every IT organization has the type of resources that are able to develop and maintain custom code of this nature. Depending on your requirements, CA Technology Services or a CA partner may be a good fit to help you address your specific requirements.

The best way to minimize the risks associated with your custom code is to ensure detailed documentation is provided with any delivered custom code, and to arrange some form of maintenance contract.

CA Approach

CA Unicenter Service Desk Web Services is provided as a supported mechanism for performing integrations. While any custom created Web Services code is not directly supported by CA Support, the documented Web Services methods used by the custom code would be supported.

Best Practice

As a general rule, CA Best Practice cautions against the use of any custom code. However, if you have a valid business need, then the Web Services API is the *only* supported API. In addition, CA Best Practice also suggests implementing a formal software development methodology to ensure the proper scoping, design, and maintenance of any custom code.

Configuring the Solution

There is no special CA Unicenter Service Desk configuration requirement. Just make sure that CA Unicenter Service Desk web services have been configured. This step is completed when you run pdm_configure on a CA Unicenter Service Desk server.

Enabling the Solution

After configuring CA Unicenter Service Desk and implementing the previously mentioned components, there are no further steps to enable this newly created Web Services-based custom feature.

Testing the Solution

To verify that the solution has been successfully implemented, complete the following typical steps:

1. Create an incident.
2. Create a problem from the incident.
3. Create a change order from the problem and associate the incident from step 1 to this change order.
4. The change order proceeds through the associated work flow.
5. Once the change order is nearing completion, the assigned change analyst would select the Resolved status from the Manage Incidents/Problems page, enter an explanation, and click Continue. This will invoke the web services and modify the status for each attached incident and problem, and attach the provided commentary into an appropriate activity log.
6. Once the change analyst receives confirmation that the change fixed the problem, they can proceed with setting the status of the attached incidents and problems to a Closed status.

The following sample windows illustrate a partial set of the previous steps. Specifically, the parent change order with a child problem containing the newly updated status and the activity log that was accomplished using the custom web services code.

In the previous sample window, the Change Order Detail form shows all three attached problems and incidents in an Open status.

In the previous sample window, this new form captures the comments and the desired new Status value before invoking the web services code in the JSP page when Continue is clicked.

In the previous sample window, the Problem Detail form shows the new Resolved status and the activity history that accompanied the change in status.

Troubleshooting

Use CA Unicenter Service Desk log files to verify any errors in relation to web services call and HTMPL errors.

API Summary

The CA Unicenter Service Desk Web Services API is a powerful and flexible interface that allows for the creation of custom developed code to achieve business requirements through the integration of the following into a CA Unicenter Service Desk instance:

■ A separate product

■ A new feature

While this type of development project should not be undertaken lightly, the available functionality can often be well worth the effort required. However, it is important to ensure that future maintenance of the custom code is accounted for up front in any code planning activities by completing the following steps:

■ Ensure the production of high-quality documentation

■ Establish a maintenance agreement

bop_cmd Command

The bop_cmd command is an object-oriented command-line interface that executes defined methods for CA Unicenter Service Desk objects. It has great flexibility, but requires detailed knowledge of CA Unicenter Service Desk.

How bop_cmd Works

bop_cmd is a command-line utility that can execute fragment files. Fragment files are on demand applications written in "spel" code, which is a CA proprietary language. CA spel closely resembles straight C code, with a large number of built-in functions and features designed around the CA Unicenter Service Desk architecture. The bop_cmd command executes the fragment file which can invoke a set of parameters in the same command line.

The following example illustrates how to execute the fragment file named *cnote.frg* to add an announcement in CA Unicenter Service Desk.

■   From the command prompt on a CA Unicenter Service Desk computer, enter the following command:

```
C:\Program Files\CA\Service Desk\bopcfg\interp>bop_cmd -f cnote_add.frg
"cnote_add (Microsoft Exchange Server is Down Tech Support is aware and working
on solving the problem!)"
```

You can see the announcement that was added about an outage with the Microsoft Exchange server in the following sample page:
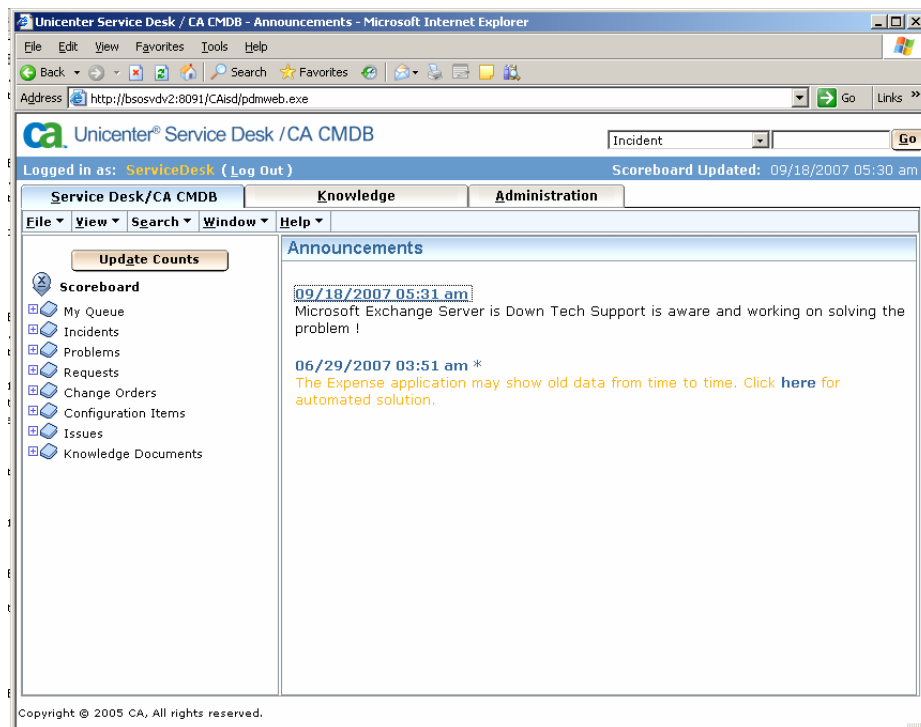
Business Challenge

Michael Reed, the service desk manager at Forward, Inc., is having some difficulties managing accurate information for the incidents and requests that have been assigned to the technicians that travel on-site. The technicians are not able to update the tickets in a timely manner from the customer location because they cannot access CA Unicenter Service Desk. Recently, Forward, Inc. has been successful by providing its staff with handheld devices that are able to send telealert messages to update data in other applications. Knowing that, Michael Reed has requested that those individuals responsible for implementing CA Unicenter Service Desk integrate the message center application with CA Unicenter Service Desk so that service desk technicians will be able to create or update incidents and requests while working at a customer site.

CA Approach

CA Services has been engaged at Forward, Inc. Working together with a developer at the client, they have created two fragment files that will provide a solution to Michael's request.

Best Practices

Reference these best practices when working with the bop_cmd command:

1. Become familiar with the CA Unicenter Service Desk architecture and the way the objects and attributes have been defined and work.
2. Identify the objects and attributes to be updated during the fragment file execution.
3. Identify the attributes required to save the appropriate information in the object.
4. During the testing phase, check the CA Unicenter Service Desk stdlogs for possible messages which can provide notifications about inconsistencies.

Configuring the Solution

Preparing to Generate a New Incident or Request

To prepare for configuring the solution and generate a new incident or request, first read the information about the gencr method (gencr.frg) in Appendix A.

Note: If you are using CA Unicenter Service Desk r11.2, the file named gencr.frg, explained in Appendix A, can be found in the following directory: C:\Program Files\CA\Service Desk\samples\call_mgt.

Configuring the Solution from CA Unicenter Service Desk

To configure the solution from CA Unicenter Service Desk, copy the gencr.frg file into the following directory:

$NX_ROOT\bopcfg\interp

Configuring the Solution from the Alert Message Product

To configure the solution from the Alert Message product, configure the third-party product so that it can to pass the required parameters, and ensure the parameters are in the following format:

bop_cmd -f gencr.frg "gencr("""My first Incident from the command line:-)""", """BSODEMOS1""","""""", """2""", """Work In Progress""", """Applications""","""ServiceDesk""", """""", """back""", """""", """2""", """2""", """Incident""")"

Testing the Solution

To test the solution, open a command prompt window on the computer where bop_cmd is available and execute the following command:

```
bop_cmd -f gencr.frg "gencr("""My first Incident from the command line:-)""",
"""BSODEMOS1""","""""", """2""", """Work In Progress""",
"""Applications""","""ServiceDesk""", """""", """back""", """""", """2""",
"""2""", """Incident""")"
```

The following table describes the input for this command:

Input	Description
Description	My first Incident from the command line:-)
Asset	BSODEMOS1
Request Template Name	blank
Priority	2
Status	Work In Progress
Request Area	Applications
Assignee	ServiceDesk
Group	blank
Charge Back	back
Impact	blank
Urgency	2
Severity	2
Type	Incident

This command returns Request number, messages

Important! BSODEMOS1 is an asset example. You should use a valid asset/CI from your environment.

As illustrated in the following sample page, incident number 58 was created in CA Unicenter Service Desk:

Updating a Request

Preparing to Update the Request Status and Add an Activity Log Comment

To prepare for configuring the solution and update the request status and add an activity log comment, first read the information about the ZUpdateCr method in Appendix A.

Configuring the Solution from CA Unicenter Service Desk

To configure the solution from CA Unicenter Service Desk, copy the ZUpdateCr.frg into the following directory:

$NX_ROOT\bopcfg\interp

Configuring the Solution from the Alert Message Product

To configure the solution from the Alert Message product, configure the third-party product so that it can to pass the required parameters, and execute the following command:

```
bop_cmd -f ZUpdateCr.frg "ZUpdateCr('56', 'belldo01', 'Researching',
'Telalert Message')"
```

The fragment can be used to update the status of a request and add an activity log by executing the command line using the bop_cmd command. The following information provides an example:

Ref_Num

56

Assignee

belld01 (User who sent the message)

Status

Researching

Activity Log detail

Telalert Message

Testing the Solution

Complete the following steps to test the solution:

1. Request number 56 has been created and has been assigned to the analyst Donald Bell. Donald is on-site and is going to update the status of the request by sending a text message to the telealert product from his telephone.
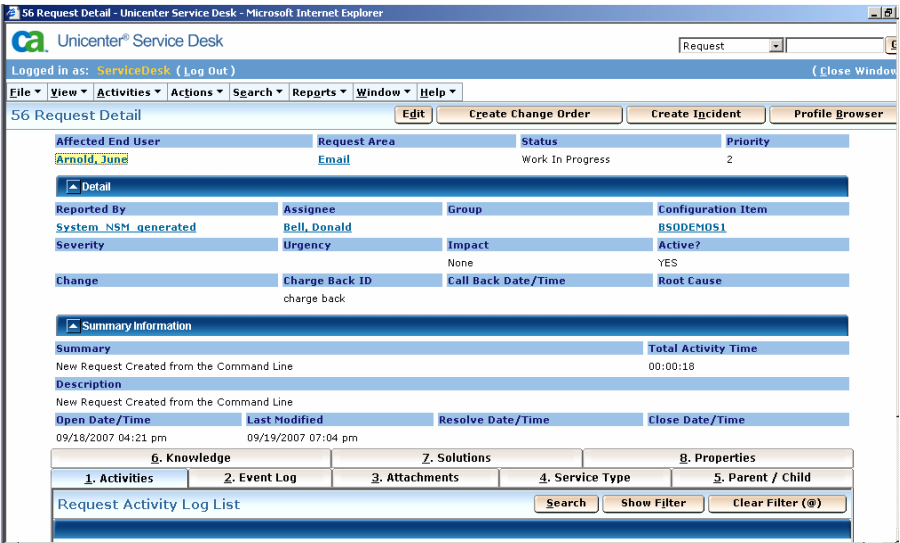
2. The telealert product creates an executable file with the following command:

```
bop_cmd -f ZUpdateCr.frg "ZUpdateCr('56', 'donaldbell', 'Researching', 'Telalert
Message')"
```

3. After the command has been received, the status of the request is updated from Work In Progress to Researching, and a new activity log has been added in the Activities tab. The activity log contains the name of the user who sent the message, as well as the status change.
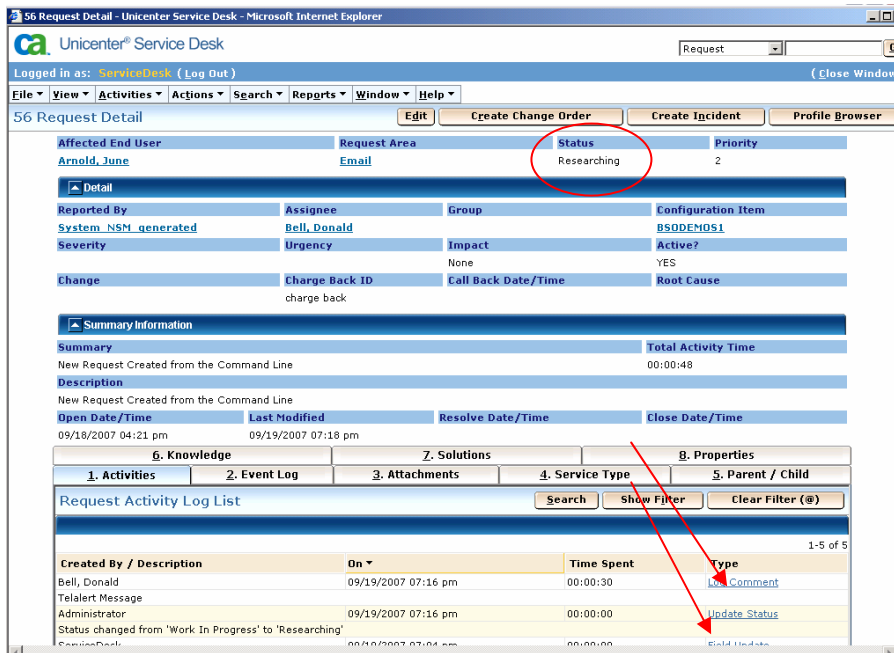
Troubleshooting

When having issues with fragment files executed by bop_cmd, you can check in the stdlogs for more information. In addition, you can use bop_logging and pdm_trace commands to debug your fragment files, as described in the following section.

Bop_logging and pdm_trace

In the CA Unicenter Service Desk architecture, most communication between processes, and within processes, is performed by exchanging BPMessages. When having issues with making your fragment files work using bop_cmd command, it is useful to trace the message flow. The following two methods allow you to trace the message flow:

■ **bop_logging** writes each BPMessage sent or received by the process being logged into a file. This method can be configured to use a single file that grows indefinitely, or as a set of files used in round robin fashion, with each file permitted to grow to a specified size before logging switches to the next file.

   Important! There is considerable overhead to bop_logging. It noticeably slows a process, and should not be used in a production environment unless absolutely necessary.

■ **pdm_trace** stores each message sent or received by the process being logged into internal memory. Only a pre-specified maximum number of messages will be kept. Messages in memory are written to a file on request, or when certain events occur, such as a process crash or a specified stdlog message.

Both forms of logging are activated with a common line command. To determine the options available for each command, invoke it with the –h argument, such as the following:

```
bop_logging -h
pdm_trace -h
```

The output from bop_logging or pdm_trace is similar. Each message captured is reported in the following format:

```
07/12 09:22:02.781 (+0.000) Received Msg [User usd; Session 1995107485]
From: 138.42.43.231.web:local.
To: 138.42.43.231.domsrvr.TOP [Top_Ob]
BPMessage
{
    method = call_attr
    arg 0 = (string) chg
    arg 1 = (string) get_new_dob
    arg 2 = nil
    arg 3 = nil
    arg 4 = gPCAAA(2references)  [Group_CO]
    reply object = 138.42.43.231.web:local.GHAAAA (not refcounted)
    reply method = get_new_dob:0
    }
```

The following information is displayed for each message:
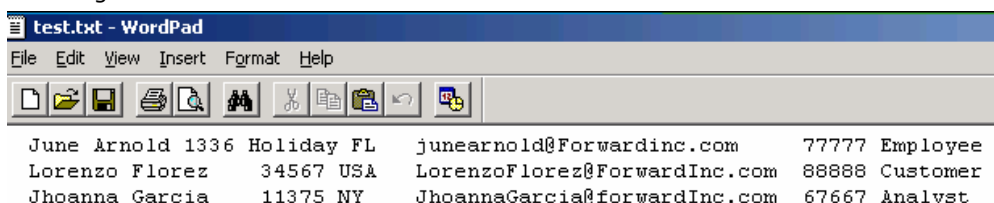
■   The date and time of the message.

■   The number of milliseconds since the previous message (parenthesized).

■   The kind of action that resulted in this message, either "Sending", "Received", or "Imported".

■   The userid and session of the user associated with the message.

■   The sending ("From") process. This is a two-part address, consisting of the IP address of the slump server, followed by a dot, followed by the name for the process supplied at slump login.

■   The receiving ("To") object. This is a three-part address. The first two parts are in the same format as the "From" address. The third part is the object name. If the object was defined by the process performing the logging, its name is followed by its C++ or Java class name in square brackets.

■   The name of the method being invoked by the message.

■   The type and value of any arguments to the message.

■   The name of the object expecting a reply to the message, and the name of the reply method.

Tip! You can export the required information to a text file, and then use C++ or a Perl script to put the parameters into a bop_cmd command line format and execute it. The following section describes an example of a Turbo C program that can do this for you.

Example

An example of a C program, *sutility*, which will read a text input file that contains the information required to fill the parameters required for a fragment file called mk_creq.frg is provided in Appendix A. Then, an output file will being generated, ready to be executed to create or update information into CA Unicenter Service Desk.
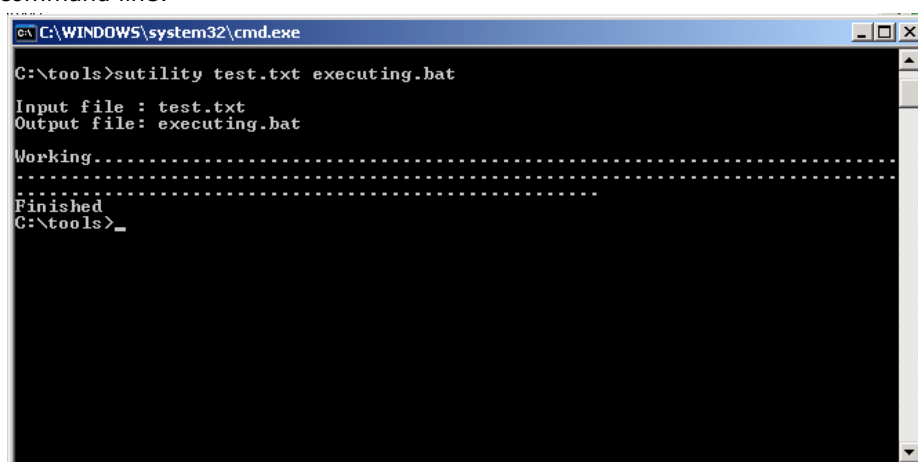
1. The sutility program is taking information from an example.txt file which has the following entries:

   sutility then formats the text into a bop_cmd command syntax.
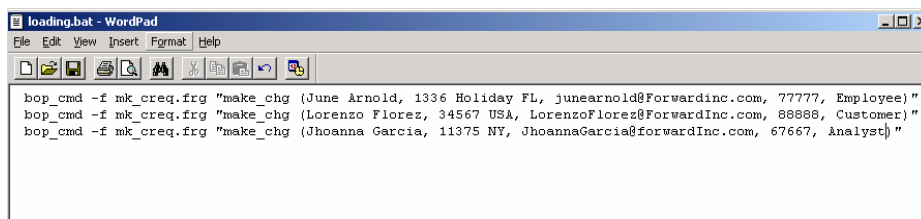
2. The following sample command-line window illustrates running sutility from the command line:

3. The output file named executing.bat is the one able to be executed to create or update information in CA Unicenter Service Desk. AT or scheduling commands could be configured to run the output file.

   The following sample window illustrates the contents of executing.bat:

bop_cmd Summary

CA Unicenter Service Desk has a command line utility which can be invoked with a file name and a set of attributes. The type of file that works in context with the bop_cmd command is called a fragment file, which contains spel code that will be interpreted. CA spel code is a proprietary interpreted language closely resembling C, and is used for specifying business logic.

The command line must have the following syntax:

```
bop_cmd –f file.frg "function ("""parameter1""", """parameter2""",
"""parameter3""")"
```

External products can put the parameter information into a text file. From the text file, a Perl script or C++ program can extract the parameters, put them into the required bop_cmd format, and execute the command.

Text API Method

The Text API is a simple common interface that allows you to create and update objects in the CA Unicenter Service Desk database such as issues, requests, contacts, and assets, using text-based input. Using the Text API, you can set most fields that are accessible from the Java client and the web interface.

Accessing the Text API

You can access the Text API using the following interfaces:

■   Command line

■   email

■   Unicenter NSM

In the following example, we will illustrate how to use the Text API from the command line. Later, we will discuss the email and NSM interfaces.

Note: You can use Web Services as an alternative to the Text API for cross application integration. For more information, see the Web Services integration method described earlier in this section, or the *CA Unicenter Service Desk Web Services User Guide*.

How the Text API Works from the Command Line

From the command line, you use the pdm_text_cmd command to activate the Text API command-line interface. You specify certain information, such as the table to process and the operation to perform, using parameters in the pdm_text_cmd command. The input to the Text API is passed to the pdm_text_cmd command in the form of an input file or directly from STDIN.

Note: For more information about the pdm_text_cmd entry, see the reference commands in the *CA Unicenter Service Desk Administrator Guide*, or type pdm_text_cmd –h from the command prompt.

Business Challenge

Michael Reed, the service desk manager for Forward, Inc., must configure CA Unicenter Service Desk to be able to create a change order from information in an in-house developed Human Resource product when a new employee is hired. The change order should be assigned to the Windows Administrators who will create all required Windows accounts for the new employee.

CA Approach

CA Technology Services advises Michael to use the pdm_text_cmd command to create the change order from the Human Resource product.

Best Practices

Review the following best practices when using the pdm_text_cmd command:

1. Validate that change order creation in CA Unicenter Service Desk supports the organization's pre-established process.

2. Clearly define when the change order needs to be created and updated.

3. Discuss the implementation with the CA Unicenter Service Desk Administrator.

4. Test in a development environment first.

5. Check that all parameters sent from the change order creation source exist in CA Unicenter Service Desk.

Configuring the Solution

Create a Change Order from the Command Prompt

Use the following steps to create a text file with the information that you want to populate into the change order.
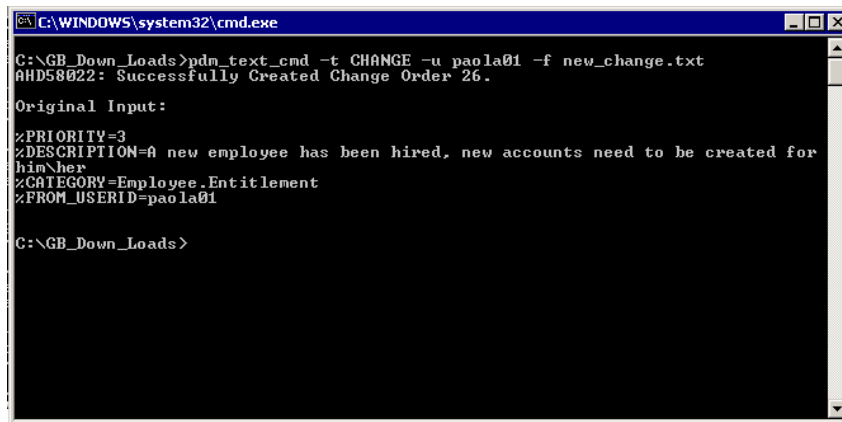
1. Using Notepad, create a text file named new_change.txt with the information illustrated in the following sample window:
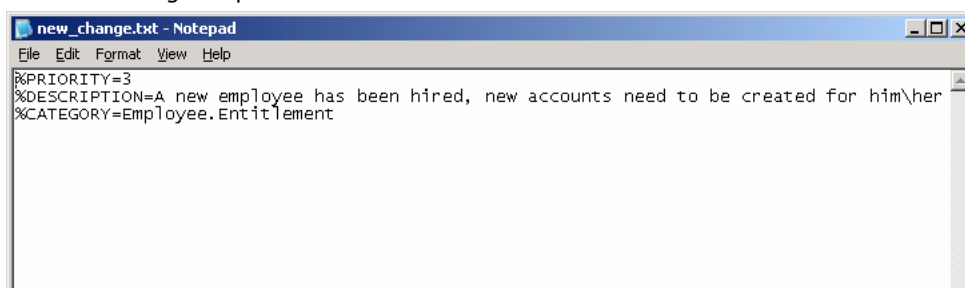
2. From the command prompt execute following command:

```
pdm_text_cmd -t CHANGE -u paola01 -f new_change.txt
```

The following sample command prompt window illustrates the command:

At this point, pdm_text_cmd is interacting with text_api.cfg. As a result, a new change order has been create in CA Unicenter Service Desk, illustrated in the following sample window:
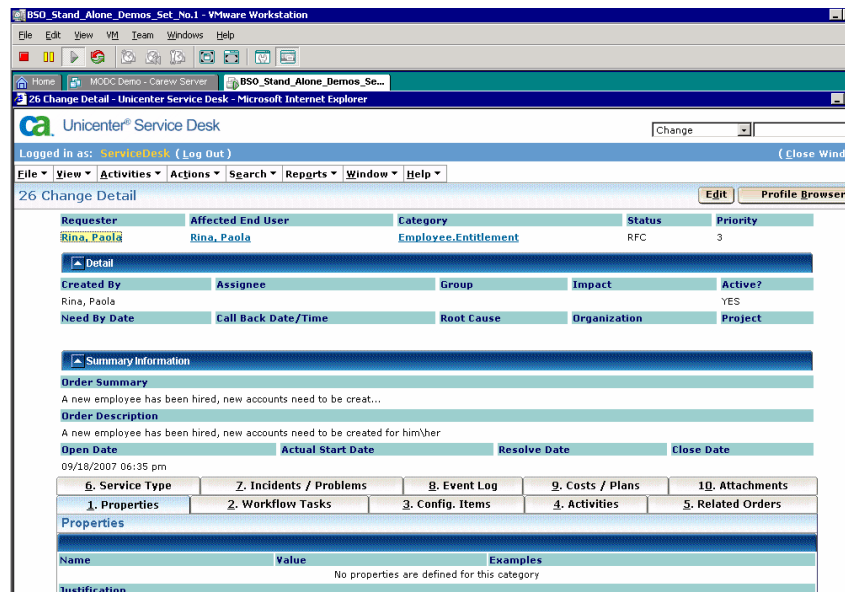
Text API Summary

The text API is a simple command interface that allows you to create and update objects in CA Unicenter Service Desk, such as issues, requests, change orders, contacts, and assets using text-based input. Using the Text API, you can set any fields that are accessible from the Java client and the web interface.

Troubleshooting

When having issues with the text_api, you should check in the stdlog for more information about what is possible wrong in your definitions.

Database-level Data Integration

All CA Unicenter Service Desk data is stored in a normalized relational database. Subject to certain restrictions and caveats, this database can be read and updated by other products. The restrictions include the following:

- CA Unicenter Service Desk keeps its own locks at the product level. Using the locks and dynamic screen updates, the user is guaranteed that the data they see is the data updated, without database lock contention. If updates are taking place outside CA Unicenter Service Desk, that guarantee is not valid.

- CA Unicenter Service Desk is designed to never deadlock and hold locks for short durations. Deadlocked rows are managed by waiting, and eventually generating errors and abandoning the operation. However, on high-volume production systems, deadlocking of critical tables can catastrophically affect performance. Table locks held for any length of time almost always have negative consequences. Therefore, any external system that works directly with CA Unicenter Service Desk tables must be carefully designed to avoid prolonged locking, table locks, and deadlocks.

Given these restrictions, data level integration can work well. The first concern, data serialization, is more theoretical than actual. In practice, occasional unserialized updates occur outside the software. For example, when a service representative closes an incident after a technician has been dispatched, but before the technician arrives on site, it is effectively an unserialized update that has nothing to do with the software. As long as integration does not increase the number of unserialized updates, the site is usually satisfied.

In r11, the Common Registration API (CORA) and the MDB are examples of database-level integration. All Unicenter r11 products utilize the common MDB schema to store and manage their data. As the interface through which these assets are registered and as the only source for updating these tables, CORA ensures that asset data flows consistently, thereby supporting the data and referential integrity of the MDB's master asset data model.

Note: For more information about CORA and how the MDB manages and shares data across CA products, see the *Incident and Problem Management Green Book* using the Technical Support link at http://ca.com/support.

CA Technology Services can provide resources for assistance with database-level integration engagements.

pdm_load

pdm_load is a utility supplied with CA Unicenter Service Desk. It is a database-independent utility for adding, deleting and updating rows in the CA Unicenter Service Desk database. It is a convenient utility for performing batch loads and updates to CA Unicenter Service Desk. pdm_load generates numeric ids along with incident and change order numbers, and interacts correctly with CA Unicenter Service Desk application locking. Typically, pdm_load is a simple and relatively safe way to perform batch loads into CA Unicenter Service Desk.

Note: For more information about pdm_load, see the reference commands in the *CA Unicenter Service Desk Administrator Guide*.

pdm_extract

pdm_extract is a utility supplied with CA Unicenter Service Desk. It is a database-independent utility for extracting data from CA Unicenter Service Desk. The default output is in a proprietary format suitable for loading data back into CA Unicenter Service Desk with the pdm_load utility. Options are available to export data in comma-separated value (CSV) format, which can be imported into Microsoft Office products such as Excel and Project. pdm_extract accepts table names to dump the entire contents of tables, and it accepts standard basic SQL statements for joins and projections.

Note: For more information about pdm_extract, see the reference commands in the *CA Unicenter Service Desk Administrator Guide*.

pdm_deref

pdm_deref is a database-independent utility that simplifies loading normalized data. pdm_deref converts strings to their numeric key values as part of a pdm_load, making it possible to load foreign keys in a single step.

Note: For more information about pdm_deref, see the reference commands in the *CA Unicenter Service Desk Administrator Guide*.

Business Challenge

Forward, Inc.'s service desk team received a request from the security department to inactivate all customers that have not used the service desk in the last six months. The lack of use indicates that the customer is no longer using Forward, Inc's services or support contract.

CA Approach

CA Technology Services has recommended using pdm_extract, pdm_deref, and pdm_load commands to enable Forward, Inc. to identify and inactivate the appropriate customer contact records.

Configuring the Solution

Complete the following steps to configure the solution:

1. Use *pdm_extract* to find all customers (that is, contacts where the contact type equals customer) who have been inactive for more than six months. In this example, search for contacts in which the last_mod_date of the record in the contact table is greater than six months ago. Use the following command:

    ```
    pdm_extract -f "Select userid, id, last_update_date FROM ca_contact WHERE
    inactive=0 AND contact_type=2305 AND last_update_date < 1166918400 " > ex-
    customers.dat
    ```

    Note: In this command, the value 1166918400 is the UNIX format for the date that will be searched. In this example, the local time for 1166918400 is 12/24/2006. In addition, see the companion zip file posted with this Green Book for a sample Excel file (UNIX Date Format.xls) that contains a macro allowing date information to be formatted into a UNIX format. pdm_extract works with the format illustrated in the sample Excel file.

2. Based on the customers identified in the previous step, use *pdm_deref* to search in the Call_Req table to identify those customers who have never opened a call or for which no call activity has been recorded in the last six months. This condition is met if the number of calls equals zero, which means the ref_num attribute="" (NULL) in the Call_Req table, where the customer = id (from the output file ex-customers) and the last_mod_date is less than six months ago.

    Execute the following command:

    ```
    pdm_deref -s correl1.spc < ex-customers.dat > inactive.dat
    ```

    In this command, correl1.spc contains the following:

    ```
    Deref
    {
    input = id, last_update_date
    output = id, ref_num
    rule = "SELECT customer, id, ref_num, last_mod_dt FROM Call_Req WHERE customer = ?"
    AND last_mod_dt < ?"
    }
    ```

    From the command prompt, use the following command to copy the inactive.dat file to inactive2.dat as a backup activity:

    ```
    copy inactive.dat inactive2.dat
    ```

3. Use the following steps to set the identified customers to *inactive* status:

a. Edit the inactive.dat file, which looks similar to the following:

```
TABLE ca_contact
    userid id ref_num
    {"acarson", "177CABDF70BF164BADFCBE155ECE671F", ""}
    {"requester", "EB4D86B942D7F948AC3963FEC35D8602", ""}
```

b. Delete the attribute named *ref_num*, and add the attribute named *inactive* with a value of 1. The inactive.dat file will look similar to the following:

```
TABLE ca_contact
    userid id inactive
    {"acarson", "177CABDF70BF164BADFCBE155ECE671F", "1"}
    {"requester", "EB4D86B942D7F948AC3963FEC35D8602, "1"}
```

c. Execute pdm_load –f inactive.dat.

The customers are now inactive in CA Unicenter Service Desk.

Summary

CA Unicenter Service Desk r11 offers pdm_extract, pdm_load, and pdm_deref utilities to manipulate data.

■ pdm_extract extracts data from specified CA Unicenter Service Desk database tables, or the entire CA Unicenter Service Desk database, and outputs it as ASCII-formatted text.

■ pdm_load updates a CA Unicenter Service Desk database using an input file you specify, up to a maximum of 112 attributes.

■ pdm_deref processes ASCII-formatted input to exchange data found in one database table for data found in another database table. It can be used to create files compatible with pdm_userload and pdm_load from a non-CA Unicenter Service Desk database or spreadsheet. It can also be used to create reports or output files for a non-CA Unicenter Service Desk database or spreadsheet.

Note: For more information these commands, see the reference commands in the *CA Unicenter Service Desk Administrator Guide*.

Email Interface Method

The email interface can be used to create and update incidents and change orders, and to query for incident and change order data. This is often a very easy way to implement low volume integrations that do not require high speed.

How the Interface Works

You can create or update an incident or change order from an email message, formatted as follows:

To:

The email must be sent to the mailbox name assigned to the CA Unicenter Service Desk contact set up for the privileged user.

From:

Unless otherwise specified, when using the email_allow_anonymous option, the person sending the email must be defined in the ca_contact table. The email interface sends the address to the Text API using the FROM_EMAIL keyword. The Email Address field in the ca_contact record is matched against this value. It is also used as the log_agent for the ticket.

You can install the email_allow_anonymous Options Manager option to allow anonymous emails.

Note: For more information about using this option to control system behavior, see the *CA Unicenter Service Desk Administrator Guide* and CA Unicenter Service Desk online help.

You can override the From field by specifying a different email address using the FROM_EMAIL_OVERRIDE keyword in the body of the message.

Note: For more information, see the information about keywords in the *CA Unicenter Service Desk Administrator Guide*.

Attachments:

If the email has attached documents, the email interface sends them to the Text API using the ATTACHMENT keyword. For more information, see the information about keywords in the *CA Unicenter Service Desk Administrator Guide*.

Note: This feature is supported only if the email interface is running on a Windows server. For more information about the additional configuration that is required to support this feature, see the server installation (windows) information in the *CA Unicenter Service Desk Implementation Guide*.

Subject:

Enter one of the following to indicate the type of record you want to create or update:

> Issue

> Request

> Change

If the email address you are using does not match the email address in your ca_contact record, you can override the From field with another email address that you specify in the Subject field. To do this, specify -m|-M, followed by the email address that you

want to use. The email interface sends this address to the Text API using the FROM_EMAIL_OVERRIDE keyword.

Note: For more information, see the information about keywords in the *CA Unicenter Service Desk Administrator Guide*.

Body:

Format the body of the email using the Text API. If you specify the keyword ISSUE_ID, REQUEST_ID, or CHANGE_ID, depending on the type of record with which you are working, that record is updated if it exists; otherwise, a new record is created. The email interface sends the entire body of the email message to the Text API.

Note: You can simply put descriptive text into the body of the email without knowing anything about the formatting required by the Text API. If there is no initial keyword in the body of the email, the email interface adds %DESCRIPTION= to the beginning of the text. For more information, see the information about keywords in the *CA Unicenter Service Desk Administrator Guide*.

Business Challenge

Forward, Inc. has created a new service for the IT users who are located in local branches. These users must be able to request this new service, but do not have access to the CA Unicenter Service Desk web interface to enter requests for service online.

CA Approach

CA advised Forward, Inc. to configure the CA Unicenter Service Desk integration with Microsoft Outlook.

Configuring the Solution

Configure the Solution from CA Unicenter Service Desk

Complete the following steps to configure the solution for CA Unicenter Service Desk:

1. From the CA Unicenter Service Desk Administrator tab, select the Expand Options Manager Node and select the email option. This includes options to configure the incoming and outgoing mail protocols and to determine how to handle incoming email. By default, no options are installed. Determine the options that apply to your implementation, and install and set values for them accordingly.

   Note: For information about each option and how to set values, see the CA Unicenter Service Desk online help.

2. Complete the email address field with a valid email account for all CA Unicenter Service Desk contacts that are going to create or update tickets from Microsoft Outlook.

3. Complete the email address field for CA Unicenter Service Desk or the selected privileged user with the account defined for it in Microsoft Outlook.

   Note: The email protocols that are supported are SMTP for outbound email, and POP3 and IMAP4 for inbound email.

Configure the Solution from the Email Interface (Microsoft Outlook)

To configure the solution from the email interface, create an email account that can be used by the ServiceDesk user, or another CA Unicenter Service Desk privileged user.

Enabling the Solution

Complete the following steps to enable the solution:

1. After configuring and installing the CA Unicenter Service Desk options managers for email, recycle CA Unicenter Service Desk services.

2. Open Microsoft Outlook.

Testing the Solution

To test the solution, one of the users with valid contact information in CA Unicenter Service Desk must send an email, as illustrated in the following sample window:
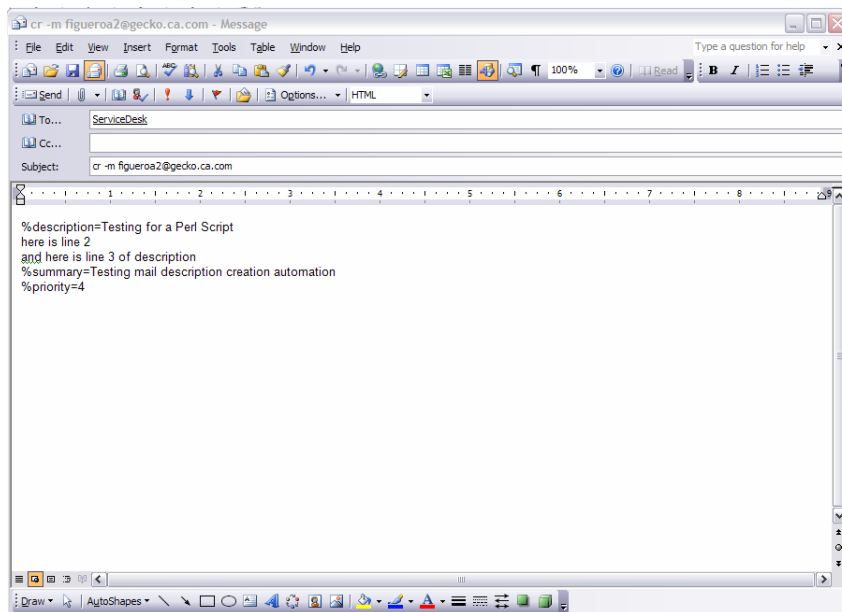
A new request will be created in CA Unicenter Service Desk, as illustrated in the following sample window:

Best Practices

To make the email interaction friendly for users to create tickets in CA Unicenter Service Desk, you can configure the email interface to provide a friendly template for users. As a result, they will not have to follow specific formatting rules to create the body of the message.

Summary

The email interface provided with CA Unicenter Service Desk lets users create and update tickets using email. The ticket could be a request, issue, or change order. There is a set of email options in the CA Unicenter Service Desk web client (Options Manager, Administration tab). These options allow the administrator to configure the incoming mail protocols and determine how to handle incoming email. Users can create or update a ticket by sending an email message formatted according to the CA Unicenter Service Desk email interface guidelines previously specified, or by using a template created by the administrator.

Incoming Event Integration Method

CA Unicenter Service Desk is designed to accept high volume event traffic from external sources for the automatic creation and update of incidents and change orders. The gateway to this interface is the CA Unicenter NSM Event Console. This interface incorporates sophisticated event filtering and event consolidation. For example, the filters convert *event*

storms into a single ticket with an event count, and identify significant event clusters and other patterns.

The interface also supports incident and change order creation policies, and can be configured to automatically create CA Unicenter Service Desk announcements. The interface can be configured to close tickets automatically when a device is restored. In addition, the CA Unicenter NSM Event Console can address the events in CA Unicenter NSM Alert Management and use policies over the alerts to provide escalation management. It then sends updates to CA Unicenter Service Desk for the incidents automatically generated from Unicenter NSM, based on the events gathered in the event console.

Note: For more information about the CA Unicenter NSM integration, see the chapter "Network Management – Integrating with CA Unicenter NSM" or the *CA Unicenter Service Desk Implementation Guide*.

Outgoing Notifications Method

CA Unicenter Service Desk notifications are typically emails, but the notification subsystem is also an integration product. These notifications are performed by exit code that is executed when a service desk object transitions from one state to another. Typically, the transitions are related to incident, problem, request, or change order status changes, but they can be configured to fire on many possible object state changes and timers.

When a notification fires, the exit code has access to the state of the system. Out-of-the-box notifications insert state information (such as the incident number, description, and assignee) into predefined text and email it to an address that is also derived from the system state (such as the email address of the person who opened the request). This mechanism provides many opportunities for outbound integration in which an activity on CA Unicenter Service Desk causes an action on an integrated system.

Note: For more information, see the information about implementing policy in the *CA Unicenter Service Desk Administrator Guide*, in addition to the information about customizing notification methods in the *CA Unicenter Service Desk Modification Guide*.

Business Challenge

Based on the number of Forward, Inc. branches and the high number of services offered by the IT department, Anita Hirsch, VP of IT Services, has created a 24 hours a day, 7 days a week service desk team. This means that the support staff must be available at any time to resolve a critical issue, escalation, or transference to avoid service level agreement (SLA) violations.

To control and implement this well-organized service support work process, and to meet her goal of delivering excellent customer service, Anita needs the service desk product to promptly notify all pertinent support staff members when critical issues occur, at any time. She does not want mistakes or missing escalations or transfers, as an SLA violation could cost thousand of dollars to the organization.

CA Approach

CA has identified the kinds of communication methods that Forward, Inc. uses, which include telephone, email, pager, Blackberry, network management event consoles, and text messages. In addition, CA Technology Services has analyzed and designed a notification process based on service types, critical events, times, people responsible for the events, and notification methods. Once this information has been identified and categorized, CA recommends that CA Unicenter Service Desk outgoing notifications are implemented to allow the system to automatically notify service desk staff using the outbound notification methods they choose. This will allow Forward, Inc. to implement the work processes required for 24 hours a day, 7 days a week support and to meet the established SLAs for service desk performance.

Best Practices

Review the following best practices when configuring the solution:

1. Based on the work process that is established, the customer must define the "who, what, when, and how" for outgoing notification actions.
2. Determine how you want the notification to be delivered (for example, sent to a Blackberry, emailed to a specified address, or printed on a particular printer).
3. Determine the contents of the notification message.
4. Specify what information from the message template to include in the notification.
5. Set up a script to transmit the notification.
6. Place the script in an executable file in the path of the CA Unicenter Service Desk server.

Configuring the Solution

Configure Outbound Notification Using a TeleAlert Message Management System from the CA Unicenter Service Desk Side

Forward, Inc. has already defined the following:

1. The first notification method to define.
2. The notification message sent using the telealert message management system will contain the following information:

    > End user name

    > assigned to

    > incident number

    > summary of the incident description

3. The activity notification selected to notify is Initial, and the message template to use for the notification is illustrated in the following sample window:

Message Template

Auto Notification	Notify Level
YES	Normal
Notification Message Title	
Assigned to: @{call_req_id.assignee.combo_name} Incident:@{call_req_id.ref_num}	
Notification Message Body	
summary:@{call_req_id.summary}	

4. The script is a .bat file which contains the definitions illustrated in the following sample window:


```
@echo on End User:%NX_NTF_COMBO_NAME% tell:%NX_NTF_SUMMARY% Message:%NX_NTF_MESSAGE%
pause
```

To implement this notification

1. Copy the telealert.bat file into the path of the CA Unicenter Service Desk server, into the following directory:

   $NX_ROOT\site\mods\bin

2. From the Administration Interface, choose Notification Methods from Notifications.

   The Notification Method List appears.

3. Click the Create New button.

   The Create New Notification Method window appears.

4. Click Save.

Note: You do not have to specify the full path in the Notification Method field. For a Windows server, consider using the launchit.exe utility to invoke your script or program. For more information about the launchit utility, see the CA Unicenter Service Desk online help. In addition, since the notification method runs from the CA Unicenter Service Desk 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 may be able to check this by executing the following 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 (UNIX) or $NX_ROOT\log (Windows).

Note: You can use the same procedure to interact with an external product. Once the information is created, it can be caught and sent to another product.

5. Complete the configuration so that all analysts who require notification using the telealert message management system will be assigned to the telealert notification method.

   In this example, we have configured Donald Bell to be notified using telealert during non-business hours and when there is an emergency. The definition appears as follows, illustrated in the following sample window:

Configuring Outbound Notification Using a Telealert Message Management System from the Telealert Side

This step could be different for different tele-messaging product. Typically, the messaging product must be able to send the information generated by CA Unicenter Service Desk's notification method and stored in the .bat file specified in the notification method field. In our example, the output will look similar to the following, where the telealert product will be picking up information from the variables End User, Tell, and Message:

Testing the Notification – for Action "Initial"

Complete the following steps to test the notification, for action "Initial":

1. Create a new incident.
2. As soon as the incident has been saved, you should receive the notification in the telealert message management system. If it has not yet been configured, you should receive a Command Prompt window with the following message:

```
C:\WINDOWS\system32\cmd.exe                                    _ □ ×
off End User:Arnold, June tell:Assigned to: ServiceDeskIncident: 88  Message:Inc
ident Initial summary:hello there

C:\PROGRA~1\CA\SERVIC~1\bin>pause
Press any key to continue . . .
```

How the Method Works

Notifications (applicable to incidents, problems, issues, change orders, and requests) are processed when the ticket is saved. If the notification method is not Notification (for example, Email or FAXserve), 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 and 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.

Note: For more information, see the *CA Unicenter Service Desk Administrator Guide*.

How to Overcome Environment Problems

Most notification methods invoke an executable or shell script to read the environment variables and send the message. This works well on most UNIX servers, but there may be difficulties reading the environment variables on a Windows server. As a workaround, you can use a Perl script on Windows.

CA Unicenter Service Desk 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 continue 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 the 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, 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 will start the Perl interpreter and execute the instructions in read_env.pl script.

launchit Utility

One of the functions of the *launchit* utility is to invoke your scripts or programs in a shell environment, similar to the command prompt with the proper environment variables set.

Note: For information about the launchit.exe parameters, execute the command from the command prompt. You will see the following window that provides valuable information about the utility.


```
Usage:
        Launchit [options] <document>
- or -  Launchit [options] <executable (exe or bat)> [parameters]

Options:
-t      Override default Document type with new type.
        Allows you to force or define the default open behavior for a file.
        Example: "Launchit -t txt my.bat"   opens a batch file using notepad.
-b      (NT ONLY) Invoke batch file using cmd.exe with special quoting.
        This is required for NT Batch files which use parameters.

   The following options are mutually exclusive:
-h      Will attempt to start executable hidden (Use with Caution)
-m      Will attempt to start executable minimized
-x      Will attempt to start executable maximized

                    [ OK ]
```

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 choose 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 provide you with 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 Unicenter Service Desk 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 Record 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	System Login
NX_NTF_VOICE_PHONE	Alt. Phone Number

Summary

Outbound notification methods describe how notification messages are delivered to users. Notification methods are scripts that are invoked based on activity notifications. The scripts use information supplied as variables to notify personnel or other systems of something

that has occurred. For example, you can write a script that sends voice mail to the analyst assigned to a request to indicate that the request has just been escalated.

Notification methods can be assigned for each contact record. The system looks up the notification method to use for specific contacts.

Note: For information about contact records, see the information about setting up users in the *CA Unicenter Service Desk Administrator Guide*.

The standard notification methods for CA Unicenter Service Desk are as follows:

- *Email* sends messages by electronic mail directly to the recipient through Simple Mail Transport Protocol (SMTP) mail. Messages are also sent to the recipient's notification log.

- *Fax* sends messages to the recipient using the FAXserve product. The FAXserve product must be installed and available. FAXserve is a separate product and is not part of CA Unicenter Service Desk.

- *Pager email* sends email to an address maintained by a paging system provider. The email text will typically display on an alphanumeric pager.

You can also create your own notification methods. For example, you can send notification to a particular printer for periodic collection, or to a pager. To create a notification method, you must create a shell script that includes notification variables, and then enter the new notification method into CA Unicenter Service Desk.

Note: For information about customizing notification methods, see the *CA Unicenter Service Desk Modification Guide*.

Remote Reference Method

The Remote Reference integration method is a table-driven system for inserting calls to external products into the CA Unicenter Service Desk Java Client interface menus and forms. This functionality does not work as well in the CA Unicenter Service Desk web interface, because you typically encounter security restrictions imposed by the browser when attempting to launch products.

CA Unicenter Service Desk administrators can also leverage Remote References in Execute Remote Reference types of macros. These items can then be further leveraged by Classic Workflow tasks as well as Service Type events in managing the CA Unicenter Service Desk policy.

Note: For an example of the Remote Reference capabilities in an integration, see the Configure the CA Unicenter Service Desk and CA Unicenter Patch Management Integration from the CA Unicenter Service Desk Side topic in the chapter "Patch Management – Integrating with CA Unicenter Patch Management."

CTI Integration Method

CTI integration is possible through a number of mechanisms. It is typically implemented by using web services, in combination with opening a web browser, to launch a pre-populated CA Unicenter Service Desk URL. This allows for the display of the profile browser and incident in context. Either existing tickets or new tickets will be displayed, pre-populated with information gathered using telephonic means such as Voice Response Units (VRUs).

Note: For more information about CTI integration, contact CA Technology Services.

Integration Value

Computer Telephony Integration (CTI) is technology that allows interactions on customer contact channels (voice, email, web, fax, and so forth) and computer systems to be integrated or coordinated. In this scenario, it is primarily used to launch application screens (*screen pops*) in coordination with voice and data. Interactive Voice Response Units (VRUs) capture information from a customer, which is then used not only to forward the call to an agent's desk, but also to display information in context of the responses. The advantage is that all relevant customer information is available to an agent by the time the agent is ready to answer the call. CTI is also used for call routing. Information such as account number, request type, area, and so on, captured by the VRU, is used to route a call to an appropriate agent or a group of agents. Overall, CTI contributes to enhancing the customer experience.

How the Integration Works

The CA Unicenter Service Desk CTI integration is based on CTI Software Server and desktop client software. The name for the CTI server software will change from one telecom equipment provider to another. For example, Avaya CTI Server functionality is provided by "CCE Server" and the Avaya desktop client is named CCE Agent software.

The PABX (or PBX) is linked with the CTI server which can receive data from the PABX. Equally, the CTI Server can command and control the PABX.

The following diagram illustrates how the CTI integration works.
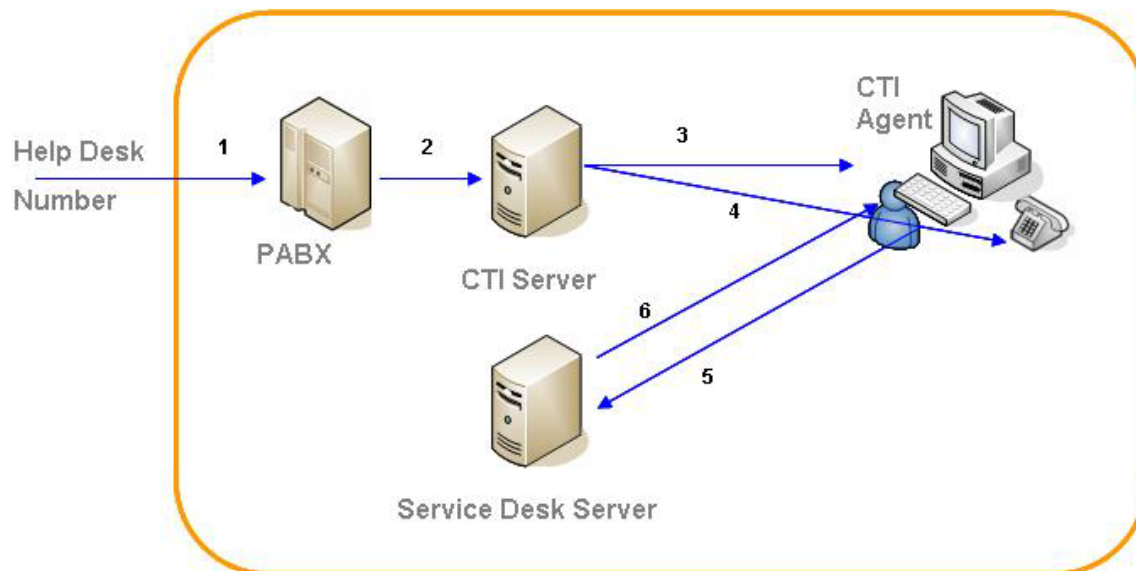
The following information applies to the previous diagram:

1. The Customer calls the "Service Desk" Number. The customer is presented with various options to respond.

2. The captured responses and the call are then passed by the PABX to the CTI Server.

3. The CTI Server finds the first available support analyst. The captured customer information is sent to the CTI agent software of the identified support analyst.

4. At the same time, the call is routed to the support analyst's phone.

5. The CTI Agent Software uses the captured information to interact with a solution such as CA Unicenter Service Desk.

6. The interaction of the CTI Agent with CA Unicenter Service Desk results in a *screen pop* on the support analyst's desktop in the context of the information passed by the CTI agent. The support analyst can now proceed with the call and has the application information relevant to that call.

It is clear from the above flow that the CTI agent software is responsible for passing information and initiation of the screen pop from CA Unicenter Service Desk. All the effort surrounding the integration of CA Unicenter Service Desk and the CTI product is focused around the CTI Agent software. Typically, no effort is required on the CA Unicenter Service Desk side.

Setting Up, Configuring, and Testing the Integration

Business Challenges

Identification and Acquisition of CTI software

Existing telephone infrastructure and a contract with a telephone service provider does not mean you have a CTI Server and the requisite CTI Agent Software. You should start discussions with your telephone vendor to acquire the software. This can include considerable expense in software and hardware.

IVRU Prompts

Thought and effort should be put in to developing Interactive Voice Response Unit prompts and call routing. Prompts should be limited, but enough to satisfy business requirement for call routing. The vendor of the application to be integrated with CTI should also be consulted in this process.

Identification and Authentication

CTI Agent software will launch an application on behalf of a support analyst. The following list of question will need to be answered in preparation for the integration:

1. How does the CTI Agent authenticate to the application?
2. How does it get the authentication information?
3. How often should it authenticate to an application?

CA Approach

The CA Unicenter Service Desk web interface makes life easier to launch contextual information. The CA Unicenter Service Desk Modification Guide describes in detail how to access CA Unicenter Service Desk information by launching a web interface URL. Here are few examples:

Launching a New Incident Screen

```
http://<host_name>:<port>/CAisd/pdmweb.exe?OP=CREATE_NEW+FACTORY=in+BOPSID=12
314432874
```

In this sample URL, BOPSID is a token that identifies a CA Unicenter Service Desk user. See the following section for an explanation on how to generate a BOPSID.

Launching a New Incident Screen and Pre-populating Fields

```
http://<host_name>:<port>/CAisd/pdmweb.exe?OP=CREATE_NEW+FACTORY=in+BOPSID=21
3123434+PRESET_REL=customer:cnt.id:contact_num:12345
```

In this sample URL, the Customer field for the new incident record is to be preset to a contact record with *employee number* = 12345

Launching and Displaying the Contact Record for the Calling Customer

```
http://<host_name>:<port>/CAisd/pdmweb.exe?OP=SEARCH+FACTORY=cnt+QBE.EQ.conta
ct_num=12345+BOPSID=34235543534535
```

This sample URL will list the Service Desk contact record for the employee with employee number 12345.

Note: For more information about arguments used in Service Desk web interface URLs, see the *CA Unicenter Service Desk Modification Guide*. In addition, all URLs in the previous examples require a BOPSID to be passed as an argument.

Getting a BOPSID

Complete the following steps to get a BOPSID:

1. Log in to CA Unicenter Service Desk using Service Desk web services. The login operation returns a Session Identifier (SID).

   Note: For more information, see the *CA Unicenter Service Desk Web Services Guide*.

2. Use the SID from the previous step to get the BOPSID. Invoke the *getBopsid* web services method. The BOPSID must be retrieved every time a URL is to be launched.

   Note: For more information, see the CA Unicenter Service Desk Web Services Guide.

Best Practice

A Support Analyst must indicate to the CTI Agent software that they are available to take phone calls. This is typically the first step. Then, the CTI agent will route incoming calls to the Support Analyst. The CTI agent software should be adapted to incorporate the CA Unicenter Service Desk login and logout in the process of the support person indicating their availability. The following diagram illustrates the sequence of events that should happen in the CTI agent software:
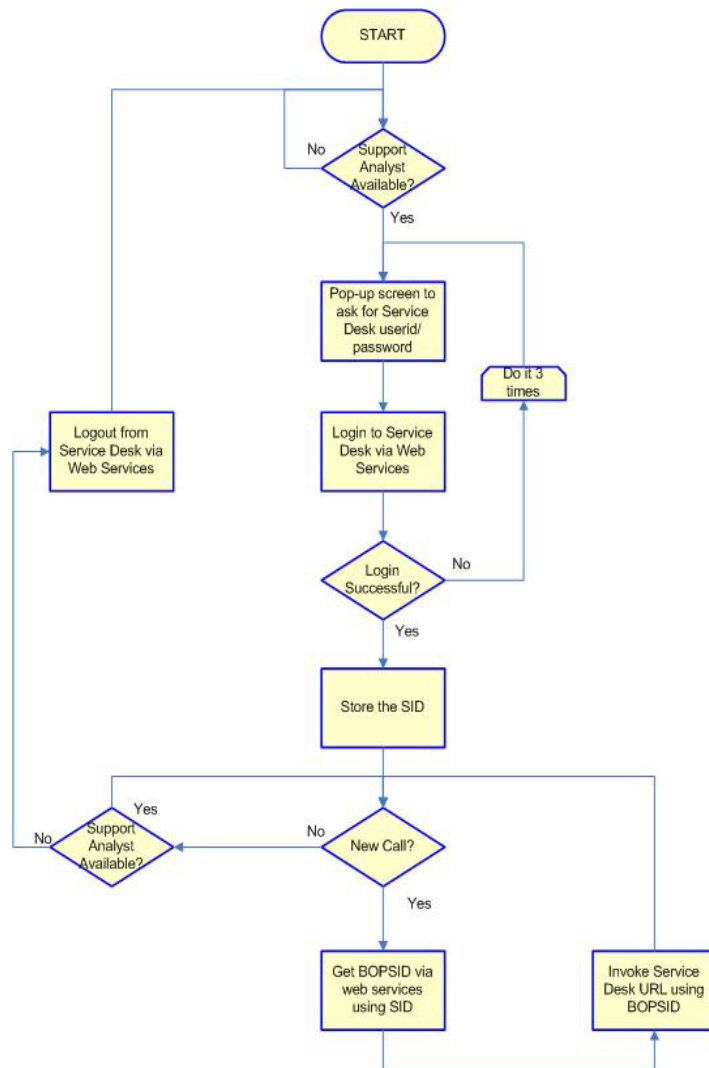
Configuring the Solution

There are no special CA Unicenter Service Desk configuration requirements. Make sure that CA Unicenter Service Desk web services have been configured, which is done when you run pdm_configure.

Currently, there is no way to launch the *profile browser* with the contact details completed. The Profile browser provides a nice aggregated view of client information and serves as a launch pad to other CA Unicenter Service Desk activities. A minor adaptation to the list_cnt.htmpl file allows searching of a contact and displaying details for the contact in a profile browser.

The following htmpl code highlights the changes to be made to the list_cnt.htmpl.

Note: The code in Red is the adaptation required to the list_cnt.htmpl.

```
<PDM_MACRO NAME=lsStart>

    <PDM_MACRO NAME=lsCol hdr=Name attr=combo_name link=yes>

<pdm_macro name=lsWrite both=yes text="if (groupOnly) {">

    <PDM_MACRO NAME=lsCol hdr=Number attr=contact_num>

<pdm_macro name=lsWrite both=yes text="}">

<pdm_macro name=lsWrite both=yes text="else {">

    <pdm_macro name=lsWrite both=no text="PDM_IF \"@args.KEEP.cti\" == \"1\" &&
    \"@list.id\" != \"\" ">

    <pdm_macro name=lsWrite both=no text="profile_browser (\"@list.persistent_id\")
    ;">

    <pdm_macro name=lsWrite both=no text="/PDM_IF">

    <PDM_MACRO NAME=lsCol hdr="Contact Type" attr=type.sym>

    <PDM_MACRO NAME=lsCol hdr="Access Type" attr=access_type>

    <PDM_MACRO NAME=lsCol hdr="Contact ID" attr=contact_num>

    <PDM_MACRO NAME=lsCol hdr="System Login" attr=userid>

<pdm_macro name=lsWrite both=yes text="}">

    <PDM_MACRO NAME=lsCol hdr="Phone Number" attr=phone_number>

    <PDM_MACRO NAME=lsCol hdr=Status attr=delete_flag>

<PDM_MACRO NAME=lsEnd>
```

Contact details and the profile browser with the contact's details can now be launched using the following URL:

```
http://<host_name>:<port>/CAisd/pdmweb.exe?OP=SEARCH+FACTORY=cnt+QBE.EQ.conta
ct_num=12346+KEEP.cti=1+BOPSID=324322323423
```

In this URL, KEEP.cti=1 drives the launching of the profile browser for the searched contact.

Note: This method will only work when there is a unique contact record for the search criteria entered. In addition, the adaptation has been tested for CA Unicenter Service Desk 11.2.

Enabling the Solution

To successfully enable the solution, the adaptation to the CTI Agent software is required to do the following:

1. Prompt the support analyst for the CA Unicenter Service Desk userid and password.
2. Use CA Unicenter Service Desk web services to log in to CA Unicenter Service Desk.
3. Use CA Unicenter Service Desk web services to get the BOPSID.

4. Launch the appropriate CA Unicenter Service Desk URL with BOPSID.

Testing the Solution

CA Unicenter Service Desk URLs can be tested without BOPSID using the following URL:

```
http://<host_name>:<port>/CAisd/pdmweb.exe?OP=SEARCH+FACTORY=cnt+QBE.EQ.conta
ct_num=12346
```

This URL will display the CA Unicenter Service Desk login page. You should see the contact listed once you log in to CA Unicenter Service Desk.

Troubleshooting

Use the CA Unicenter Service Desk log files to verify any errors related to the web services call.

Integration Summary

The CA Unicenter Service Desk and CTI integration directly impacts customer perception and improves the process of call management. Therefore, it is important and critical to understand the call handling process being planned. CA Technology Services will be required to provide guidance and information to the CTI vendor to carry out the adaptations required to the CTI agent.

The CTI integration discussed in this section should serve 95 percent of the client requirements. More complicated implementations such as transfer of calls and popping of incident details with the transfer have been implemented at customer locations, but are beyond the scope of this section. Your local CA Technology Services resources are available to assist with these complex scenarios.

URL Integration Method

URLs can be used to access CA Unicenter Service Desk objects. For example, a URL can be embedded in an email to allow the email recipient to directly access the incident.

Note: For more information about the URL integration, see CTI Method, earlier in this section. For complete details about URL integration, see the *CA Unicenter Service Desk Administrator Guide*.

External Authentication Method

Using a secondary server, CA Unicenter Service Desk can be integrated with an authentication system running on a different system, or even on a different hardware platform. An example of this is working with the boplogin process, explained in the User Authentication section in the *CA Unicenter Service Desk Implementation Guide*.

Note: For more information about this integration method, see the Security section in the *Incident and Problem Management Green Book* using the Technical Support link at http://ca.com/support. Complete details are available in the *CA Unicenter Service Desk Administrator Guide*.

External Table Display Method

Tables can be added to the CA Unicenter Service Desk schema and displayed within CA Unicenter Service Desk. The data in these tables can be from an external application. CA Unicenter Service Desk specialists from CA Technology Services should be consulted when implementing this type of integration.

The following section contains two illustrations of this integration method, both comprised of integrations between CA Unicenter Service Desk, CA Unicenter NSM, and CA CMDB. The first integration supports change management and the second supports root cause analysis.

Business Challenge – Change Management

There are several risks associated with not performing any steps or tasks within CA Unicenter NSM during a planned change. Some risks are obvious, such as the inability to mature with automated processing of defects in the environment. For example, if there is a planned event on a CI and CA Unicenter NSM does not understand it, automatic remediation, escalation, and so forth will be performed unnecessarily. If CI's were put into maintenance mode to remediate the risk, without integration, extensive manual processing (that is, unmanaging a device) would have to be performed. In addition, without automatic suppression of events during planned outages, the risk of making a mistake during manual processing (for example, forgetting to remanage a device) would invariably hurt SLAs with the business in the case of missed events.

Integration Overview and Value

This integration can minimize alert traffic to CA Unicenter NSM support personnel when scheduled maintenance on managed CIs occurs. Frequently, organizations set aside a time in the week for routine maintenance (for example, early Sunday morning). During this time, it is desirable to disable CA Unicenter NSM alert monitoring to certain affected CIs so that unnecessary alarms are not sent to support personnel. This minimizes costs and optimizes the use of skilled and scarce resources.

This is accomplished by enabling CA Unicenter Service Desk and CA Unicenter NSM to have access to CA CMDB relationship data for those CIs scheduled for maintenance prior to the planned outage, and disabling automated alerts that would normally be sent by CA Unicenter NSM to support personnel.

How the Integration Works

The following diagram illustrates how the integration works:
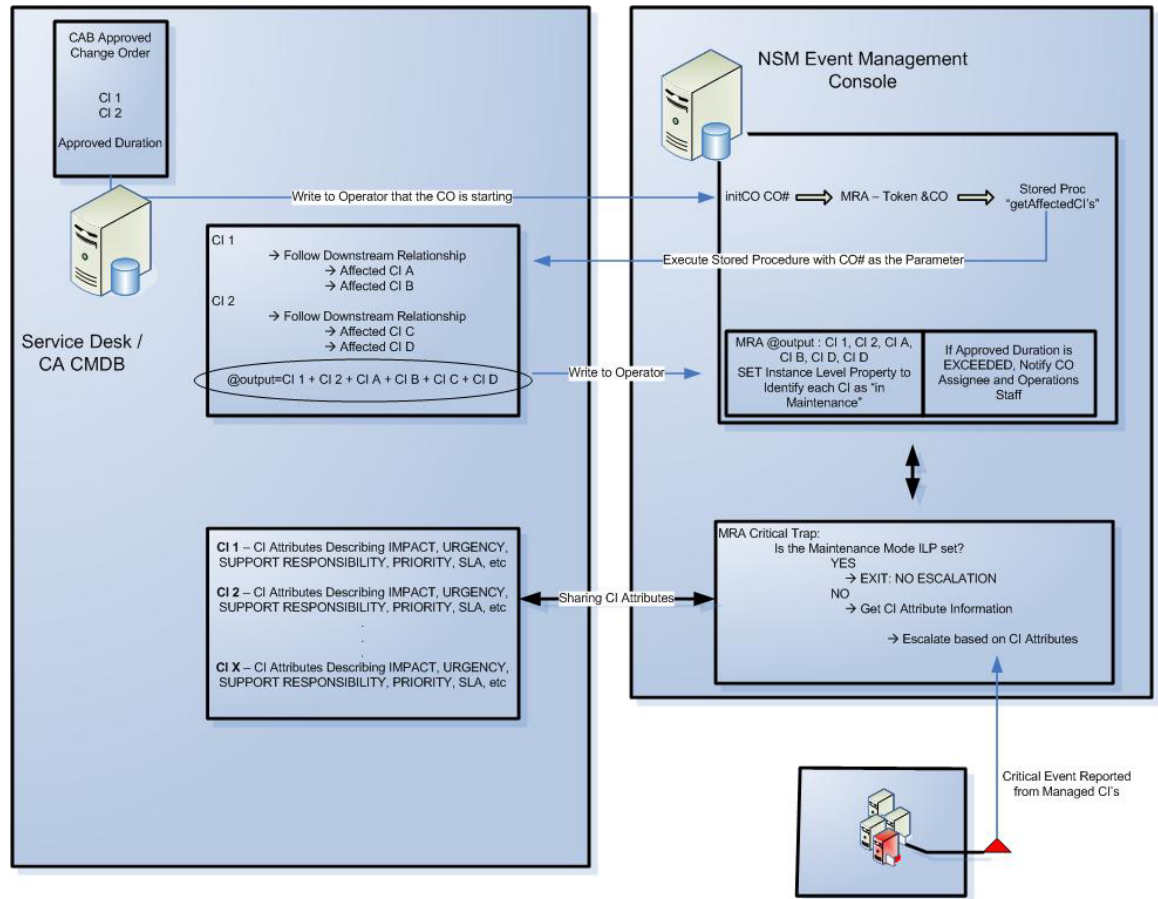
The following information applies to the previous diagram:

1. A change order (CO) is opened requesting that certain CIs undergo maintenance. The change order specifies that all events for the CIs are to be suppressed within CA Unicenter NSM during the approved window (that is, start date, duration).

   This work is completed in CA Unicenter Service Desk. A change order is opened, and CIs are associated to it. The CA CMDB Visualizer can be leveraged to see the impact of the change order, and proper risk assessment and justification can be done within CA Unicenter Service Desk in preparation of the Change Advisory Board (CAB) meeting.

2. CA CMDB and impact analysis is used to determine the impact an outage to the CIs will have on applications and service level agreements (SLAs).

3. Upon approval of the change order, and at the beginning of the *maintenance window*, a message is sent from Change Management to CA Unicenter NSM Event Management using a *write to operator* command. The message includes the change order number and the approved duration.

   For example, a vbscript with a message box is used to ask the operators to enter the change order number that is being initiated. The following sample code illustrates the script:

```
on error resume next
dim ref_num
Dim WSHShell, Args
```

```
dim CO, error1

Set Args = WScript.Arguments
Set WSHShell= WScript.CreateObject("WScript.Shell")
CI=Args(0)

CO=Msgbox("Do you have a CO number?",35,"CO?")
Set Args = WScript.Arguments
Set WSHShell= WScript.CreateObject("WScript.Shell")
if CO=6 then

    ref_num=inputBox("Please enter the Change Order Number","Initiating Change
    Order","999999")
        if ref_num="" then
                wscript.quit
                else
        ref_num=replace(Replace(ref_num,"CO",""),":","")
                WSHShell.Run("cawto -n EMCONSOLE CO# " &ref_num& " Initiated")
        end if
end if
```

4. CA Unicenter NSM Message Record Actions (MRAs) receive the message from the input message box. You can also set up events and macros in CA Unicenter Service Desk to execute a logforward/cawto on a change order status change (for example, changing from OPEN to WIP) and pass @ref_num as the CO variable) and execute a series of actions that include a "look-up" of related CIs found within the CA CMDB.

   CA CMDB returns a list of all CIs that are impacted by the change order, and all CIs that are downstream of those directly impacted CIs. This identifies which CIs (for example, URLs, data sources, routers, switches, and so forth) will, as a result of the change order, go critical in CA Unicenter NSM once the change order is executed. To prevent this from happening during the maintenance window, CA Unicenter NSM sets an instance-level property on all affected CIs as "within maintenance" (posted=1).

   The stored procedure InitCO sets the object ILP in CA Unicenter NSM to POSTED=1 Status (this attribute will be used during CA Unicenter NSM escalation. If Posted=1, then do not escalate). This stored procedure is called from an MRA triggered by the cawto within the vbscript or SD Event/Macro.

   Two stored procedures (getaffectdCI, getChildCI) are invoked to generate a temporary table that lists the change order number, affected CIs, and all "dependent on parent" CIs in two columns, along with other escalation attributes configured within CA Unicenter Service Desk (SDView_local) including SDPriority, Tier, and so forth that are used if and when escalation is performed.

   Note: Escalation may be performed on the CI when the change order is closed, but the CI is still critical. That is, the system did not fully recover from the change.

   The table name will be getAffectedCI_local or servicedesk.affectedCI, depending on the stored procedure. These two stored procedures run on scheduled jobs to generate the table. The tables are leveraged during the InitCO stored procedure.

   The following sample code provides examples of the stored procedures:

Example: Stored Procedure

```
-- =============================================
```

```
-- ALTER  procedure basic template
-- ===========================================
-- creating the store procedure


CREATE    PROCEDURE [dbo].[getaffectedCI]
AS


SET CONCAT_NULL_YIELDS_NULL OFF

DELETE FROM servicedesk.affectedCI

DECLARE @tmp AS VARCHAR(255)

DECLARE main CURSOR
FOR SELECT dbo.chg.chg_ref_num
    FROM dbo.chg WHERE  (datediff(d,dateadd(s, need_by,'January 1,1970
00:00'),getdate())) < 30 and chg_ref_num > 30000


OPEN main
FETCH main INTO @tmp
WHILE (@@Fetch_Status =0)
BEGIN
        EXEC sp_getchildCI @tmp, NULL
         FETCH NEXT FROM main INTO @tmp
End
CLOSE main
DEALLOCATE main

GO
```

Example: Stored Procedure

```
CREATE    PROCEDURE [dbo].[sp_getchildCI]
@conum nvarchar(300), @ciparent binary(16)
AS
DECLARE @tmpbin AS binary(16)

IF @ciparent is NULL
BEGIN

    DECLARE dig CURSOR LOCAL
    FOR SELECT    dbo.fn_chartobin('0x' + SUBSTRING(dbo.lrel.r_persid, 4, 60))
    FROM    dbo.chg INNER JOIN dbo.lrel ON dbo.chg.persid = dbo.lrel.l_persid
    WHERE    (dbo.chg.chg_ref_num = @conum)
```

```
    OPEN dig
    FETCH dig INTO @tmpbin
    WHILE (@@Fetch_Status =0)
    BEGIN
        IF @@NESTLEVEL < 30
        BEGIN
        EXEC sp_getchildCI @conum, @tmpbin
    END
    FETCH NEXT FROM dig INTO @tmpbin
        END
        CLOSE dig
        DEALLOCATE dig
        GOTO endsp
    END

INSERT INTO servicedesk.affectedCI (conumber, ciname)
SELECT @conum, resource_name from dbo.ca_owned_resource where own_resource_uuid =
@ciparent

IF (select count(hier_child) from busmgt where hier_parent = @ciparent) > 0
BEGIN
    DECLARE match CURSOR LOCAL
    FOR SELECT   hier_child
    FROM busmgt
    WHERE hier_parent = @ciparent

    OPEN match
    FETCH match INTO @tmpbin
    WHILE (@@Fetch_Status =0)
    BEGIN
        IF @@NESTLEVEL < 30
        BEGIN
            EXEC sp_getchildCI @conum, @tmpbin
        END
        FETCH NEXT FROM match INTO @tmpbin
    END
    CLOSE match
    DEALLOCATE match
END

endsp:

GO
```

Example: Stored Procedure

```
CREATE PROCEDURE initCO (@ref_num varchar(80)) AS

--exec [servicedesk\SDInstance].mdb..sp_executesql
--      N'getaffectedCI'

set nocount on
create table #sp_co (
  CO              varchar(80),
  CI              varchar(500),)
insert into #sp_co
select * from affectedCI_local
select * from #sp_co

select distinct tng_managedobject.name as mdbname, tng_managedobject.class_name as
mdbclass, SDView_local.CI as sdname, SDView_local.CIClass as sdclass, #sp_co.CO as
COCO, SDView_local.Tier as SDTier
```

```
into #initiateCO_table from SDView_local right outer join #sp_co on
lower(replace(SDView_local.CI,'.ca.com',''))=lower(replace(#sp_co.CI,'.ca.com',''))
left outer join tng_managedobject on
lower(replace(SDView_local.CI,'.ca.com',''))=lower(replace(tng_managedobject.name,'.
ca.com',''))collate database_default
where tng_managedobject.class_name not like 'Perf%'

set concat_null_yields_null off

declare @cmd as varchar(8000), @mdbname as varchar(255), @mdbclass as varchar(8000),
@sdclass as varchar(255), @COCO as varchar(80), @output varchar(8000), @sdname
varchar(255)

declare main cursor for

select mdbname, mdbclass, sdclass, COCO, sdname from #initiateCO_table where SDTier
in ('1','2','3') and mdbname is not null and COCO = '' +@ref_num+''

open main

fetch main into @mdbname, @mdbclass, @sdclass, @COCO, @sdname

while (@@fetch_status =0)

begin
            select @output=@output + ' : ' + @mdbname
            update tng_managedobject set acknowledge='0' where name=@mdbname
and class_name=@mdbclass
            update tng_managedobject set posted='1', propagate_status='0' where
name=@mdbname   and class_name=@mdbclass

SET @cmd = 'logforward.exe -nEMCONSOLE -f"'+@mdbname+ '" -ucaunint -t"InitiateCO:
CO# ' +@COCO +' ObjName: ' + @mdbname + ' : mdbClass: ' +@mdbclass + ' : CIClass: '
+@sdclass+ ' : COCI: ' +@sdname+'"'
exec @cmd = master..xp_cmdshell @cmd

fetch main into @mdbname, @mdbclass, @sdclass, @COCO, @sdname
end

close main

deallocate main

select @output

drop table #initiateCO_table
drop table #sp_co
SET @cmd = 'logforward.exe -nEMCONSOLE -f"' +@COCO+'" -ucaunint -t"InitCO:  CO# '
+@COCO+@output+'"'
exec @cmd = master..xp_cmdshell @cmd

GO
```

Example: Stored Procedure

```
CREATE view SDView_local as

SELECT      *
FROM        OPENQUERY([servicedesk\SDInstance],
                 'select dbo.zEscalation_Table.sym as EscalationTable,
dbo.ca_resource_class.name as CIClass, dbo.ca_owned_resource.resource_name as CI,
dbo.usp_owned_resource.z_searchstring as SearchString, r2.serial_number as
Escalation_1, dbo.zEscalation_2.sym as Escalation_2, dbo.ca_contact.last_name as
SDGroupName, dbo.ztierlvl.enum as Tier, dbo.zSD_Priority.priority as SDPriority,
dbo.prob_ctg.sym as SDRequestArea, dbo.zStatus.sym as NSM_Status, dbo.srv_desc.sym
as SDServiceType, dbo.ca_owned_resource.nr_gmt as GMT, dbo.bool_tab.sym as
NSM_Escalate

from ca_owned_resource
     left outer join dbo.usp_owned_resource on
dbo.ca_owned_resource.own_resource_uuid = dbo.usp_owned_resource.owned_resource_uuid

     left outer join dbo.zEscalation_Table on dbo.zEscalation_Table.id =
dbo.usp_owned_resource.z_escalationtable

     left outer join dbo.ca_resource_class on dbo.ca_resource_class.id =
dbo.usp_owned_resource.z_class

     left outer join dbo.ca_owned_resource r2 on
dbo.usp_owned_resource.z_escalation_1 = r2.own_resource_uuid

     left outer join dbo.zEscalation_2 on dbo.zEscalation_2.id =
dbo.usp_owned_resource.z_escalation_2

     left outer join dbo.ca_contact on dbo.ca_contact.contact_uuid =
dbo.usp_owned_resource.z_sdgroup

     left outer join dbo.ztierlvl on dbo.ztierlvl.enum =
dbo.ca_owned_resource.nr_tier

     left outer join dbo.zSD_Priority on dbo.zSD_Priority.id =
dbo.usp_owned_resource.z_sdpriority

     left outer join dbo.zStatus on dbo.zStatus.id =
dbo.usp_owned_resource.z_status

     left outer join dbo.bool_tab on dbo.bool_tab.enum =
dbo.usp_owned_resource.z_nsmescalate

     left outer join dbo.srv_desc on dbo.srv_desc.code =
dbo.usp_owned_resource.nr_service_type

     left outer join dbo.prob_ctg on dbo.prob_ctg.persid =
dbo.usp_owned_resource.z_sdrequestarea
where dbo.bool_tab.sym is not null
and dbo.ca_owned_resource.inactive = 0
')
```

The tables generated by the CA CMDB stored procedures and the CA Unicenter NSM
InitCO stored procedure allow CA Unicenter NSM Escalation message record actions
(MRAs) to understand that URLs, data sources, routers, switches, and so forth may be

"within maintenance" (parent or dependent children), and that CA Unicenter NSM should not perform automated "unplanned" defect remediation (that is, CA Unicenter NSM ILP Posted=1).

5. An alert is opened within the CA Unicenter NSM Alert Management Console iterating the CIs affected, the change order number, and the approved duration.

6. All messages that come into the Event Management console that are sourced from any of the listed CIs are suppressed, and no further automation occurs. This is accomplished using a Visual Basic script within an MRA. The MRA and Visual Basic script action takes the CI as a parameter, checks to see if the CI is "posted", and if so, the MRA exists without further escalation processing.

```
Option Explicit
On Error Resume Next

Dim WSHShell, Args, cnt
Dim InventoryInfo, MDB, SQLQuery
Set Args = WScript.Arguments
Set WSHShell= WScript.CreateObject("WScript.Shell")

cnt = 1
AEC_Message = ""

While cnt < Args.Count
                    AEC_Message = AEC_Message & " " & Args(cnt)
                    cnt = cnt + 1
Wend

AEC_Message=replace(replace(replace(replace(AEC_Message,">","_"),"<","_"),"|","_"),"
?","%")

           SQLQuery="select posted from tng_managedobject where
lower(replace(name,'.ca.com',''))=lower(replace(replace('"&nodeid&"','.ca.com',''),'
_Spectrum','')) and class_name not in ('Exchange_Connector','Unreg') and class_name
not like 'Perf%'"
           MDBInfoResult=0

           Call MDBConnection(SQLQuery)

                         If MDBInfoResult = 1 Then
                                        WSHShell.Run("%COMSPEC% /c
cawto -c Orange " &nodeid & " had a critical event, but it is posted = "
&MDBInfoResult)
                                        WScript.Quit
                         End If

Function MDBConnection(SQLQuery)

                    Err.Clear
                    On Error Resume Next
                    Set TNGConn=createobject("ADODB.Connection")
                    Set MDB=CreateObject("ADODB.recordset")

      TNGConn.ConnectionString="DSN=mdb;UID=user;PWD=password;DATABASE=mdb;"
```

```
                        TNGConn.ConnectionTimeout=60
                        TNGConn.CommandTimeout = 120

                        TNGConn.Open
                        MDB.Open SQLQuery, TNGConn, 0, 1
                        MDBInfoResult = MDB(0)
                        MDB.close

End Function
```

7. If the duration of the change order is exceeded, an alert is sent to both the assignee of
    the change order and to operations. The alert is intended to inform all parties that the
    approved duration has been exceeded, and that service level agreements may possibly
    be impacted.

    A stored procedure is used to understand duration and enter into Advanced Event
    Correlation (AEC) maturity. If the maturity is exceeded, escalation occurs as
    appropriate. The following sample code illustrates a sample stored procedure:

```
CREATE PROCEDURE initCODuration (@ref_num varchar(80)) AS

declare @cmd as varchar(8000), @assignee as varchar(255), @opened_by as
varchar(8000), @affected as varchar(255), @duration as varchar(80)

declare main cursor for

select chg_ref_num, assignee, opened_by, affected, est_total_time from CODuration
where chg_ref_num = ''+@ref_num+''

open main

fetch main into @ref_num, @assignee, @opened_by, @affected, @duration

while (@@fetch_status =0)

begin
        SET @cmd = 'logforward.exe -nEMCONSOLE -fsp_Duration -ucaunint -
t"initCODuration:  CO# ' +@ref_num +' assignee: ' + @assignee + ' : opened_by: '
+@opened_by + ' : end_user: ' +@affected+ ' : duration: ' +@duration+'"'
        exec @cmd = master..xp_cmdshell @cmd

        fetch main into  @ref_num, @assignee, @opened_by, @affected, @duration
end

close main

deallocate main
GO

CREATE view CODuration as

SELECT      *
FROM        OPENQUERY([servicedesk\SDInstance],
                    'select ch.chg_ref_num, c1.userid as assignee, c2.userid as
opened_by, c3.userid as affected, ch.est_total_time
from dbo.chg ch left outer join dbo.ca_contact c1 on ch.assignee = c1.contact_uuid
            left outer join dbo.ca_contact c2 on c2.contact_uuid = ch.log_agent
```

```
                left outer join dbo.ca_contact c3 on c3.contact_uuid =
ch.affected_contact')
```

8. At the conclusion of the change order, the assignee will inform operations that the change order is complete, and that the alert in the Alert Management System (AMS) may be closed. The closure will generate another event to EMCONSOLE using Logforward and AMS Policy.

   At that time, all alerting within CA Unicenter NSM will process as usual because this stored procedure sets Posted=0 for the CIs in question. If there are any CIs within any other active change order, the specific CIs are not taken out of maintenance (that is, posted=1) until all change orders with "collision" CIs are closed.

```
CREATE PROCEDURE endCO(@ref_num varchar(800)) AS


set nocount on
create table #sp_co (
  CO                varchar(80),
  CI                 varchar(800),)
insert into #sp_co
select * from affectedCI_local

select distinct tng_managedobject.name as mdbname, tng_managedobject.class_name as
mdbclass, SDView_local.CI as sdname,
SDView_local.CIClass as sdclass, #sp_co.CO as COCO, SDView_local.Tier as SDTier,
tng_status.status_text as event
into #initiateCO_table from SDView_local right outer join #sp_co on
lower(replace(SDView_local.CI,'.ca.com',''))=lower(replace(#sp_co.CI,'.ca.com',''))
left outer join tng_managedobject on
lower(replace(SDView_local.CI,'.ca.com',''))=lower(replace(tng_managedobject.name,'.
ca.com',''))collate database_default
inner join tng_status on tng_managedobject.propagated_status_no=tng_status.status_no
where tng_managedobject.class_name not like 'Perf%'

set concat_null_yields_null off

declare @cmd as varchar(8000), @mdbname as varchar(255), @mdbclass as varchar(8000),
@sdclass as varchar(255), @COCO as varchar(80), @output varchar(8000), @sdname
varchar(255), @event varchar(255)


-----
select distinct ciname, count(*) as counter into #tmp9
from
 affectedCI_local
 group by ciname having count(*) > 1

select distinct t.ciname, w.conumber
into #tmp10
from #tmp9 t, affectedCI_local w
where w.ciname = t.ciname


order by t.ciname

select name into #ams_tmp from ams_lv1_alerts where class='ChangeOrder' and
groupid='' and active='Y'
insert #ams_tmp select @ref_num
```

```
select t.ciname, count(*) as counter into #tmp11 from #tmp10 t, #ams_tmp w
where t.conumber=w.name collate database_default
group by t.ciname having count(*) > 1

select distinct t.ciname, w.conumber into #tmp12 from #tmp11 t, #tmp10 w,
tng_managedobject x, #ams_tmp z
where t.ciname=w.ciname and lower(t.ciname)=lower(x.name) collate database_default
and w.conumber=z.name collate database_default

CREATE TABLE #tmpNoEnd
(
ciname varchar(800)
)

if @ref_num in (select conumber from #tmp12)
BEGIN
        DECLARE @name varchar(8000), @conumber varchar(8000), @cmd1 varchar(8000)
        insert #tmpNoEnd select distinct ciname from #tmp12

        DECLARE main CURSOR

        FOR

        select ciname, conumber from #tmp12

        OPEN main

        FETCH NEXT FROM main INTO @name, @conumber
        WHILE (@@fetch_status <> -1)
        BEGIN
                IF (@@fetch_status <> -2)
                BEGIN
                        select @cmd1=@cmd1 + ' ' + @name + ' within CO'+ @conumber +
' ; '
                END
                FETCH NEXT FROM main INTO @name, @conumber
        END

        CLOSE main
        DEALLOCATE main

end
select @cmd1
-----

declare main cursor for

select mdbname, mdbclass, sdclass, COCO, sdname, event  from #initiateCO_table where
SDTier in ('1','2','3') and mdbname is not null and COCO = '' +@ref_num+''

open main

fetch main into @mdbname, @mdbclass, @sdclass, @COCO, @sdname, @event

while (@@fetch_status =0)
```

```
begin

IF (select ciname from #tmpNoEnd where
lower(replace(ciname,'.ca.com',''))=lower(replace(@mdbname,'.ca.com',''))) is null
begin

        update tng_managedobject set acknowledge='0' where name=@mdbname  and
class_name=@mdbclass
        update tng_managedobject set posted='0', propagate_status='1' where
name=@mdbname  and class_name=@mdbclass

        IF (select propagated_sev from tng_managedobject where name=@mdbname  and
class_name=@mdbclass) = '5' or (select severity from tng_managedobject where
name=@mdbname  and class_name=@mdbclass) = '4' or (select severity from
tng_managedobject where name=@mdbname  and class_name=@mdbclass) = '5'
        BEGIN
                        SET @cmd = 'logforward.exe -nEMCONSOLE -f"'+@mdbname+ '" -
ucaunint -t"endCO:  CRITICAL : CO# ' +@COCO +' ObjName: ' + @mdbname + ' : mdbClass:
' +@mdbclass + ' : CIClass: ' +@sdclass+ ' : COCI: ' +@sdname+ ' : ' + @event +'"'

                        exec @cmd = master..xp_cmdshell @cmd
        END

        else
        Begin
                        SET @cmd = 'logforward.exe -nEMCONSOLE -f"'+@mdbname+ '" -
ucaunint -t"endCO:  CO# ' +@COCO +' ObjName: ' + @mdbname + ' : mdbClass: '
+@mdbclass + ' : CIClass: ' +@sdclass+ ' : COCI: ' +@sdname+'"'

                        exec @cmd = master..xp_cmdshell @cmd
        end
end
else
begin

        SET @cmd = 'logforward.exe -nEMCONSOLE -f"'+@mdbname+ '" -ucaunint -t"endCO:
CO# ' +@COCO +' ObjName: ' + @mdbname + ' ' + 'Did not come out of maintenance
because of a collision CO'
        exec @cmd = master..xp_cmdshell @cmd

end


fetch main into @mdbname, @mdbclass, @sdclass, @COCO, @sdname, @event
end

close main

deallocate main

select @output

drop table #initiateCO_table
drop table #sp_co
```

```
SET @cmd = 'logforward.exe -nEMCONSOLE -fOPERATIONS -ucaunint -t"endCO:  CO# '
+@ref_num +'"'

exec @cmd = master..xp_cmdshell @cmd
drop table #tmp9
drop table #tmp10
drop table #tmp11
drop table #tmp12
drop table #tmpNoEnd
drop table #ams_tmp
GO
```

In the event a message comes to CA Unicenter NSM Event Management and the sourced CI is not part of any change order currently under execution (posted<>1), CA CMDB is leveraged by message record actions (SQL View: SDView_local) to return SLA data, support groups, impact, urgency, and so forth. The information is passed back to CA Unicenter NSM to root cause and correlate events, and then finally alerts the appropriate support personnel. Without the CA CMDB relationship data, understanding priorities of incidents, responsible parties, and so forth would be time consuming and difficult to manage.

Business Challenge – Root Cause Analysis

Root cause analysis is critical in an IT organization to help meet service level agreements (SLAs) and other measured "time to incident restoration" metrics. Although there is extensive flexibility in how to effectively isolate defects from one another within CA Unicenter NSM, CA CMDB relationships can be leveraged to add another level of root cause analysis. On the assumption that defects in parents may be related to defects in children, an IT organization may find it valuable to understand up-stream critical events along with the actual CI defect being escalated. CA CMDB has those relationships, and this implementation will discuss how to leverage that for "management by exception".

Integration Overview and Value

CA Unicenter NSM and CA CMDB integration for root cause analysis is based on the need to understand how CIs impact each other when CI defects are detected by CA Unicenter NSM. Enhanced enterprise management value is derived by first attempting to perform root cause analysis within the CA Unicenter NSM engines and policies defined. Adding another level of root cause analysis by integrating with CA CMDB can been a powerful ally to meet IT's SLAs with the business.

Allowing CA Unicenter NSM to have access to CA CMDB relationship information enables CA Unicenter NSM processes to understand all the CI components that makes up a service. This includes understanding which organizations depend on the services. This provides not only root cause understanding, but who is being impacted by the outage, providing a full 360 degree view of the outage impact.

Another approach to using CA Unicenter NSM and the CA CMDB together to support root cause analysis is to start with the CA CMDB and identify the specific components that make up the business service at issue. Once the business service is identified, CA Unicenter NSM can be launched in context to the business service to display the Business Process View

(BPV). At this point, the user can use CA Unicenter NSM's root cause engine to identify the primary reason for the outage.

How the Integration Works

When CA Unicenter NSM generates an alert, it attempts to use internally defined policies and procedures to isolate the fault to a higher root cause. CA Unicenter NSM comes with many out-of-the-box rules that the user can configure. In addition, new rules can be added to the rules engine that can help further diagnose and isolate the root cause problem.

The following information provides a description of the flow for root cause analysis:

■ An alert is sent from a CA or third-party product into CA Unicenter NSM event management.

■ Advanced Event Correlation (AEC) within CA Unicenter NSM matches the event and attempts to perform root cause analysis on the defect.

■ If AEC is not able to match a root cause above the CI that delivered the defect, Event Management runs a process that examines CA CMDB relationships, examines the statuses of upstream CIs, and returns to the escalation process "Other Potential Root Cause Events".

■ Concurrent with the root cause determination is the ability to automatically send event information to CA Unicenter Service Desk's automated incident creation process.

Setting Up, Configuring, and Testing the Integration

To successfully set up, configure, and test the integration, complete the following steps:

1. Create a query that creates a list of CIs and their parent CIs. This will be a one-to-many relationship. The following is a sample query:

```
Select c.resource_name as parent, c2.resource_name as child
    From busmgt b left outer join ca_owned_Resource c on b.hier_parent =
    c.own_resource_uuid left outer join ca_owned_Resource c2
    on c2.own_resource_uuid = b.hier_child
    Where c.resource_name is not null group by c.resource_name,
    c2.resource_name having count(*) > 0
```

2. Create a stored procedure that takes the CI_Name as a variable, and asks CA CMDB to return a list of all CIs that are upstream of that CI, and to check their CA Unicenter NSM status. The following is a sample stored procedure:

```
CREATE proc CIWatcher4 (@node as varchar(255)) as
set concat_null_yields_null off
DECLARE @crit_out varchar(800), @crit_alarm varchar(800)
select t.CIParent into #tmp1 from CA_CMDB_Rel t, tng_managedobject w where
(w.alarmset_name in ('Host','Workstation','Router','LAN_Switch')  or w.class_name in
('Application'))
and
lower(replace(@node, '.ca.com',''))=lower(replace(w.name,'.ca.com',''))
collate database_default
and t.CIParent collate database_default in (select name from tng_managedobject where
```

```
propagated_sev in ('4','5','6') or (class_name in ('Application','IP_Interface') and
severity='5'))
and lower(replace(t.CI,'.ca.com',''))=lower(replace(@node,'.ca.com','')) collate
database_default
----
select tng_managedobject.name,
replace(tng_statusA.status_text,'IP_Interface:Up','')+'_'+tng_statusB.status_text as
status_text into #tmp2
from tng_managedobject, tng_status tng_statusA, tng_status tng_statusB
where tng_managedobject.status_no=tng_statusA.status_no and
tng_managedobject.propagated_status_no=tng_statusB.status_no
and
(
(
propagated_sev='5' and alarmset_name in ('Router','Switch','Workstation','Host')
)
or
(
propagated_sev='6' and alarmset_name in ('Router','Switch','Workstation','Host')
)
or
(class_name in ('Application','IP_Interface') and severity='5')
or
(severity='4' and alarmset_name in ('Router','Switch','Workstation','Host'))
)
----
DECLARE main CURSOR
READ_ONLY
FOR
select t.CIParent, w.status_text from #tmp1 t, #tmp2 w where t.CIParent = w.name
collate database_default
DECLARE @crit_parent varchar(800)
OPEN main
FETCH NEXT FROM main INTO @crit_parent, @crit_alarm
WHILE (@@fetch_status <> -1)
BEGIN
    IF (@@fetch_status <> -2)
    BEGIN
        set @crit_out = @crit_out + ' ; ' + @crit_parent+':'+@crit_alarm
    END
    FETCH NEXT FROM main INTO @crit_parent, @crit_alarm
END
CLOSE main
DEALLOCATE main
if (select @crit_out) is not null
begin
select 'CA CMDB Parents of this CI are currently Critical.  Additional Potential
Root Causes are ' + @crit_out
end
drop table #tmp1
drop table #tmp2
GO
```

3. Create a script that a Message Record can execute that runs the query. The following is
    a sample script:

```
Option Explicit
```

```
On Error Resume Next
Dim WSHShell, Args, CI, PotentialRootCause
Set Args = WScript.Arguments
Set WSHShell= WScript.CreateObject("WScript.Shell")
CI=Args(0)
SQLQuery="CIWatcher4 '"&CI&"'"
Call MDBConnection(SQLQuery)
PotentialRootCause=MDBInfoResult
wscript.quit
Function MDBConnection(SQLQuery)
    Err.Clear
    On Error Resume Next
    Set TNGConn=createobject("ADODB.Connection")
    Set MDB=CreateObject("ADODB.recordset")
    TNGConn.ConnectionString="driver={SQL
    Server};server=Server\InstanceName;database=mdb;trusted_connection=true;"
    TNGConn.ConnectionTimeout=60
    TNGConn.CommandTimeout = 120
    TNGConn.Open
    MDB.Open SQLQuery, TNGConn, 0, 1
    MDBInfoResult = MDB(0)
    Next
    MDB.close
End Function
```

Configure and Test the Solution

To configure and test the solution, you must set up a message record and action within CA Unicenter NSM to do the following:

1. Trap a critical event for a CI.

2. Set the action to execute the script with a parameter of the CI (that is, wscript.exe scriptname.vbs CriticalCIName).

3. Confirm that the script returns your CI and any other CI's that are upstream and Critical within CA Unicenter NSM.

   If you want to connect to CA Unicenter Service Desk to open an incident, you must enter information in two windows, User Options and Alert Global Definitions.

   Note: The release number of CA Unicenter NSM and CA Unicenter Service Desk do not have to be the same, just as long as the two products do not share a database. If they do not share a database, any combination of product releases will work.

Enable the Connection between CA Unicenter NSM and CA Unicenter Service Desk

Complete the following steps to enable the connection between CA Unicenter NSM and CA Unicenter Service Desk:

1. From the main menu in the Unicenter Management Command Center (MCC), choose View, Options.

   The User Options window appears with the General page open.

2. Click the Connections tab.

   The Connections page appears.

3. Enter the CA Unicenter Service Desk release number and the URL to access the CA Unicenter Service Desk web server. For example, *http://servername/CAisd/pdmweb.exe*.

    Note: For release 6.0, you also enter the URL for the Knowledge Tools. For example, *http://servername/causp/kt/KTMain.asp*.

4. Enter any additional options and close the window.

    The options are saved and take effect immediately.

    Note: Most information on the User Options window is saved immediately. The only exception is information on the Status Color Schemes tab, which has an icon to save the information.

5. Choose Enterprise Management from the drop-down list above the left pane of the Unicenter MCC.

6. Open a server and expand Alert Management.

    The left pane displays a list of the AMS profiles you can define.

7. Double-click Alert Global Definitions.

    The Alert Global Definition - Detail window opens.

8. Enter the Uniform Resource Identifier (URI) of the web service on your primary CA Unicenter Service Desk server. The default for CA Unicenter Service Desk 6.0 is http://server[:port]/usd_ws/usd_ws.asmx. The default for CA Unicenter Service Desk r11.x is http://*servername* [:*port*]/axis/services/USD_WebServiceSoap.

9. Enter a user ID and password and click Save.

    The connection to CA Unicenter Service Desk is established.

Majic Triggers

Triggers can be defined in the CA Unicenter Service Desk object definition language (Majic) to execute when the value of an object attribute changes. The trigger can execute code that is external to CA Unicenter Service Desk.

Important! Setting up this kind of trigger requires detailed knowledge of the CA Unicenter Service Desk architecture, specifically the domsrvr, and should be done only by CA Technology Services.

The Domsrvr

The Domain Object Server (domsrvr) is where the business objects "live". It is the business object repository. The objects in the domsrvr are defined and implemented in two types of files. The majic files (*.maj and *.mod) define the schema of the objects (that is, the class definitions). The spel files (*.spl) define the procedural behavior using interpreted C code.

Type	Product Files (bopcfg/majic)	Customer Files (site/mods/majic)
Majic *.maj	Initial OBJECT specification of the product.	Initial Customer specifications of local tables.
Majic modifications *.mod	Product temporary fixes to delivered *.maj files in bopcfg/majic.	Customer modifications to either product OBJECT or Customer OBJECT specifications.
Spel *.spl	Either of the following: ■ Compiled product spel libraries. ■ QOxxx.spl replacements for one or more spell methods in product spel libraries.	Either of the following: ■ Customer ASCII Spel methods. ■ Optional customer replacements for one or more spell methods in product Spel libraries.

What is Majic?

Majic is a CA proprietary language that defines the files that the domsrvr reads, as follows:

■ *objects*. An object is a rough equivalent to a database table. However, a single object can be mapped to multiple tables. For example, the CA Unicenter Service Desk cnt (contact) object maps to MDB tables ca_contact and usp_contact. There is no mapping to an external table.

■ *factories*. A factory is a manager for a class of objects. All objects have at least one factory, and may have more than one, with the additional factories encompassing a cross section of object instances. For example, the CA Unicenter Service Desk cst (customer) factory contains those contacts who are customers.

■ *attributes*. Attributes are a rough equivalent to database columns. Most object attributes are mapped to database columns. There are also local attributes held entirely within the domsrvr. Attributes can be scalar values, references to other objects (SRELs), or references to domsets (query results).

■ *triggers*. Majic triggers are a rough equivalent to a database trigger, but are considerably more flexible. Triggers can be defined for either objects or attributes and are invoked automatically by the domsrvr at key times, such as when an attribute or object is instantiated or changed. They can be written in C++, Java, or Spel.

■ *methods.* Methods are functions that can be invoked in the context of an object or factory.

What is Spel?

Spel is a procedural language closely resembling C, with some additional statements for handling the Business Object Process (BOP) messaging architecture. Spel language is interpreted by the spel_srvr server. The Spel interpreter executes Spel source code, whereas the primary purpose of the spel_srvr is to execute methods associated with business objects defined to the domsrvr. This includes object, trigger and factory methods.

Summary

By using Majic and Spel definitions, CA Unicenter Service Desk can trigger actions to add or modify CA Unicenter Service Desk object behavior to fit into business process definitions that involve interactions with other CA or third-party products.

Important! CA Unicenter Service Desk specialists from CA Technology Services should be consulted when implementing this type of integration.

Service Desk Events and Action Macros Method

CA Unicenter Service Desk uses conditions and actions defined as events to schedule, process, and track tickets and service types automatically. Conditions identify the event for which CA Unicenter Service Desk should be prepared. Actions identify the processes that occur automatically if the condition is true or false after a specified amount of time. You can assign a single action or a list of actions to occur for each monitored event.

Note: For more information about how to define events, search on defining events in the CA Unicenter Service Desk online help.

When an event occurs that matches a condition, CA Unicenter Service Desk automatically processes the actions associated with the condition. In some cases, actions automatically process the object or can result in a notification or the execution of a script. For example, there is an action macro named "run the event script", which is available for incidents, problems, requests, issues and change orders. This method can also be used to send information to an output file, which can then be utilized by other third-party products to perform any desired interaction based on information from CA Unicenter Service Desk. This method works the same for service type events or independent events defined in CA Unicenter Service Desk.

Business Challenge

Michael Reed, the Service Desk Manager at Forward, Inc., is encountering problems when trying to complete change orders for hardware upgrades associated with mission-critical products. There are situations where the hardware change order has been approved, the new hardware is ready for installation, but the old hardware device is still operating in a production mode and is not ready to be upgraded. This is creating a bottleneck for the activities assigned to the on-site technicians because they have to delay or reschedule the change activities. Forward, Inc. users are complaining to the service desk because the upgrades are not taking place on time.

CA Approach

CA Technology Services has advised Michael to create a service type event in CA Unicenter Service Desk, associated with the change orders, that triggers an event in CA Unicenter NSM. The CA Unicenter NSM event will notify the network operation center administrators when a change has been approved for a mission-critical hardware component, and will also provide the details of the involved device. Then, based on their hardware contingency plans, they can have everything ready for the scheduled day of the physical update or maintenance.

Best Practices

Consider the following best practices when configuring the solution:

1. Clearly define the specific event or condition for a CA Unicenter Service Desk object that will trigger an action.
2. Clearly define the action that will be initiated when the event or condition occurs.

Configuring the Solution

For this integration method, you will create a script based on the business process requirements. It can be a .bat file, Perl script, or in UNIX a shell script. In this example, the script is calling the CA Unicenter NSM command line to create an event log in the CA Unicenter NSM Event Console.

The following is the command line in the script named NSM.bat:

```
"C:\Program Files\CA\SharedComponents\CCS\WVEM\BIN\cawto.exe" -n BSOSVDv2 -
sev E -c Red -K "service aware application "Online Registration" has opened a
Service Desk Incident" %1
```

The following steps describe how to configure the solution:

1. Log in as Administrator or Service Desk, click the Administrator tab, and expand Events and Macros.
2. Click the Events node, and click Create New on the right side of the page.

The Event Detail page appears.

3. Complete the information on the Event Detail page, as specified in the following sample page:

4. Click the Action Information tab and associate the action macro named Run the CHG Event Script.

5. Associate the new service type event with the change order object.

6. In this example, we have created a change order Category (Change.IT.Server.Mem.Upg) associated to the same service type as the event. As a result, when Change.IT.Server.Mem.Upg is selected as the category for a change order, the event will be executed.

Testing the Solution

Complete the following steps to test the solution:

1. Create a change order with the category Change.IT.Server.Mem.Upg. Specify a valid CI/Asset in the change order, and select the status Approval in Progress.

2. Save the change order. After a few moments, an event log should be displayed on the NSM Event Console, as illustrated in the following sample window:
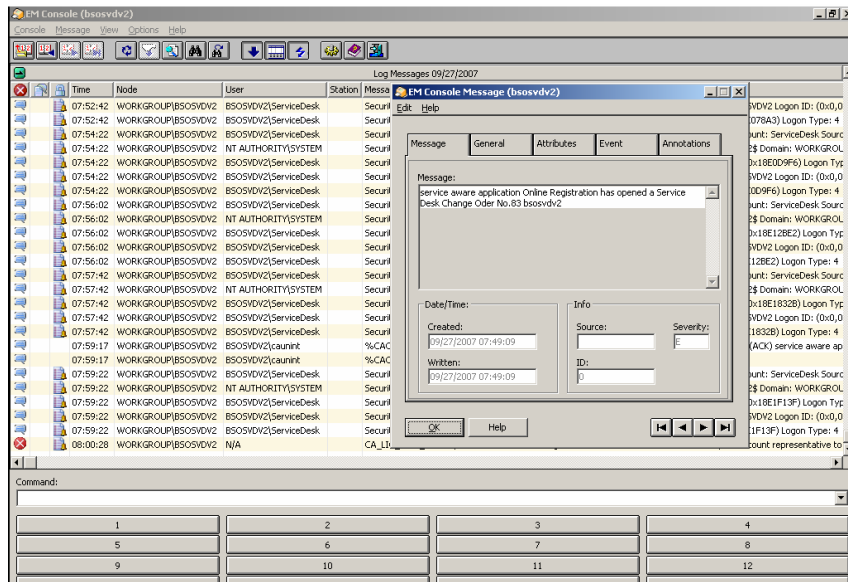
3. In the Change Order Activity Log, an "event occurred" log entry should be recorded, as illustrated in the following sample window:

Summary

CA Unicenter Service Desk uses conditions and actions defined as "events" to schedule, process, and track tickets and service types automatically. Conditions identify the event for which CA Unicenter Service Desk should be prepared. Actions identify the processes that occur automatically if the condition is true or false after a specified amount of time. You can assign a single action or a list of actions to occur for each monitored event.

Note: For more information about how to define events, search on defining events in the CA Unicenter Service Desk online help.

When an event occurs that matches a condition, CA Unicenter Service Desk automatically processes the actions associated with the condition. In some cases, actions automatically process the object or can result in a notification or the execution of a script.

Appendix A: Code Examples

gencr Method

This file is part of the out-of-the-box CA Unicenter Service Desk r11.x installation, and can be found in the NX_ROOT\samples\call_mgt directory.

```
//////////////////////////////////////////////////////////////////////////
// Copyright (c) 2000, 1995 CA.
//
// Copyright (c) 1995 Legent Software, Inc., as an
// unpublished work. This notice does not imply unrestricted or public
// access to these materials which are a trade secret of Legent
// Corporation or its subsidiaries or affiliates (together referred to
// as "LEGENT"), and which may not be reproduced, used, sold or
// transferred to any third party without LEGENT's prior written consent.
//
// All Rights Reserved.
//
// RESTRICTED RIGHTS LEGEND
// Use, duplication, or disclosure by the Government is subject to
// restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in
// Technical Data and Computer Software clause at DFARS 252.227-7013.
//////////////////////////////////////////////////////////////////////////
//
// Method:      gencr(...)
//
// Description: The fragment can be used to create a request from
//              the command line using the bop_cmd command.
//
// Input:       Description
//              Asset
//              Request Template Name
//              Priority
//              Status
//              Request Area
//              Assignee
//              Group
//              Charge Back
//              Impact
//              Urgency
//              Severity
//              Type
//
// Returns:     Request number, messages
//
// Notes:       - Description is required.
//              - Parameters are positional.  To omit, specify a null string
//                 for that parameter.  See example below.
//              - Supplied parameters will override template values
//              - If the Asset isn't found, it will be automatically added.
```

```
//                  - Assets are created with class="Discovered Hardware".  This is
declared
//                     in the beginning of the code.  Look for the comments if
//                      you want to change this.
//                  - The text that is put in the Notes field for Assets that are
//                      created is declared in the beginning of the code.
//                      Look for the comments is you want to customize it.
//                  - By default, this method creates the Requests by looking for
//                      a Customer with a user-id of "nsm" to use as "Reported By"
//                      and "Affected End User".  There is a string field in the
//                      beginning of the code where this is specified.  Look for
//                      the comments.  This must be a Custom Contact Type.
//                  - If you wish to look up the Customer/Log Agent and/or Assignee by
an
//                      attribute other than the default of userid, see the comments in
the
//                      beginning of the code about changing the search keys.
//                  - Additionally, the search key for Group defaults to Last Name.  See
//                      the comments to change this.
//                  - The message that is put in the Initial Activity Log is
//                      declared in the beginning of the code.  Look for the comments.
//                  - Many of the fields perform look-ups to other tables.  If you get
//                      a 'NOT FOUND' error message for a parameter, it may mean that the
//                      value wasn't found, or it may mean that the value wasn't unique.
//                      Check for duplicates.
//                  - Assignee must be an Analyst, or it will not be found.
//                  - Please note that bop_cmd will return a zero even if there was a
problem
//                      during execution of the spell fragment.
//                  - Please resist the temptation to modify this code.  It is not as
//                      simple as it appears.  You can very easily trash your data base
//                      without even knowing it
//                  - Special note for NT command lines:  for best results
//                      put triple double-quotes around each parameter.  For example:
//
//                          bop_cmd -f gencr.frg "gencr("""description""", """asset""",
//                              """""", """2""", """Work In Progress""",
"""Applications""",
//                              """assignee""", """group """, """charge back""",
//                              """impact""", """urgency""", """severity""",
"""type""")"
//
// The values to use as parameter are usually the values that you would normally see
in the
// Request Detail screen.  Many of the fields are keys to reference tables where the
// actual value is stored. A key is stored in the Request rather than the displayed
value.
// The exception is Assignee which searches for a userid (may be titled System Login
on
// the forms).
//
// The table below correlates the parameters to where the value will be stored and
where
// it is looked up to find the key.  Note that these are object and attribute names
not
```

```
// data base table and column names.  The "stored in" column should correspond to
the
// tag names that you see in gfe for fields on the cr_detail form.
//
// Parameter mapping
//
//      Parameter                    stored in                    look-up in
//      ---------                    ---------                    ----------
//      Description                  cr.description               text
//      Asset                        cr.affected_resource         nr.name
//      Request Template Name        look-up only
cr.template_name
//      Priority                     cr.priority                  pri.sym
//      Status                       cr.status                    crs.sym
//      Request Area                 cr.category                  pcat.sym
//      Assignee                     cr.assignee                  agt.userid
//      Group                        cr.group
grp.last_name
//      Charge Back                  cr.charge_back_id            text
//      Reported By*                 cr.log_agent                 cst.userid
//      Affected End User*           cr.customer                  cst.userid
//      Impact                       cr.impact                    imp.sym
//      Urgency                      cr.urgency                   urg.sym
//      Severity                     cr.severity                  sev.sym
//      Type**                       cr.type                      crt.sym
//
// * Not passed as a parameter
// ** Ignored unless AHD is using the ITIL configuration
//
//             Computer Associates
//             06/13/2000
//
////////////////////////////////////////////////////////////////////////////////
//////////////////


string gencr(...)
{

////////////////////////////////////////////////////////////////////////////////
//////////////////
//
// Customizable values etc.
//
////////////////////////////////////////////////////////////////////////////////
//////////////////


        //
        // Change this to the userid (System Login) of the user that you want the
        // Requests to be created with.  Note that this must be a customer type
contact.
        // This user is used for both the Customer and Reported By user.
```

```
        //
        string agt_userid;

        agt_userid="nsm";

        //
        // This is the default attribute to search by when looking for the
Customer/Log Agent
        // If you wish to search by another attribute change this to the appropriate
attribute
        // name from the cst object.  For example, if you wanted to search by
Customer ID, change
        // this value to contact_num.  Also change the value of agt_userid above.
        //
        string agt_search_key;

        agt_search_key="userid";


        //
        // This is the default attribute to search by when looking for the Assignee.
        // If you wish to search by another attribute change this to the appropriate
attribute
        // name from the agt object.  For example, if you wanted to search by
Analyst ID, change
        // this value to contact_num.
        //
        string assignee_search_key;

        assignee_search_key="userid";


        //
        // This is the default attribute to search by when looking for the Group.
        // If you wish to search by another attribute change this to the appropriate
attribute
        // name from the grp object.  For example, if you wanted to search by Group
ID, change
        // this value to contact_num.
        //
        string group_search_key;

        group_search_key="last_name";


        //
        // This is the message that is placed in the Initial Activity Log for the
Request
        //
        string activity_log;

        activity_log="Generated by gencr.frg";


        //
```

```
        // This is the message that is placed in the Notes field for all created
Assets
        //
        string asset_notes;

        asset_notes="Created by gencr.frg";


        //
        // This is the default class for Assets that are created.  This value
        // comes from the Type attribute of the nr object
        //
        string asset_class;

        asset_class="Discovered Hardware";


////////////////////////////////////////////////////////////////////////////////
//////////////////
//
// Don't change anything beyond here!
//
////////////////////////////////////////////////////////////////////////////////
//////////////////

        string desc;
        string node;
        string template;
        string priority;
        string status;
        string category;
        string assignee;
        string group;
        string charge_back;
        string impact;
        string urgency;
        string type;

        string errmsg;


        //printf("gencr\n");
        //printf("argc='%d'\n",argc);
        //int x;
        //for(x=0; x<argc; x++) {
        //      //printf("argv[%d]='%s'\n",x,argv[x]);
        //}


        // Make sure that at least 1 parameter is specified
        if(argc<1) {
                errmsg=format("Insufficient parameters specified");
                errmsg += "\n\nUsage: \tbop_cmd -d domsrvr -f gencr.frg \"gencr(";
                errmsg += " 'Description', ";
                errmsg += " ['Asset Name'], ";
```

```
            errmsg += " ['Template Name'], ";
            errmsg += " ['Priority'], \n";
            errmsg += " ['Status'], ";
            errmsg += " ['Request Area'], ";
            errmsg += " ['Assignee']";
            errmsg += " ['Group Assigned']";
            errmsg += " ['Charge Back'] ";
            errmsg += " ['Impact'] ";
            errmsg += " ['Urgency'] ";
            errmsg += " ['Severity'] ";
            errmsg += " ['Type'] ";
            errmsg += ")\n\n";
            errmsg += "\tDescription is required.  Other positional parameters
are optional.\n";

            _errmsg(errmsg);
            return;
        }


        // Parse arguments
        desc=argv[0];
        if(argc>1) node=argv[1];
        if(argc>2) template=argv[2];
        if(argc>3) priority=argv[3];
        if(argc>4) status=argv[4];
        if(argc>5) category=argv[5];
        if(argc>6) assignee=argv[6];
        if(argc>7) group=argv[7];
        if(argc>8) charge_back=argv[8];
        if(argc>9) impact=argv[9];
        if(argc>10) urgency=argv[10];
        if(argc>11) severity=argv[11];
        if(argc>12) type=argv[12];

        string the_date;
        the_date=(string) now();


        //printf("Parameters passed in:\n");
        //printf("\tdescription='%s'\n", desc);
        //printf("\tasset='%s'\n", node );
        //printf("\ttemplate='%s'\n", template);
        //printf("\tpriority='%s'\n", priority);
        //printf("\tstatus='%s'\n", status);
        //printf("\tcategory='%s'\n", category);
        //printf("\tassignee='%s'\n", assignee);
        //printf("\tgroup='%s'\n", group);
        //printf("\tcharge back='%s'\n", charge_back);
        //printf("\timpact='%s'\n", impact);
        //printf("\turgency='%s'\n", urgency);
        //printf("\tseverity='%s'\n", severity);
        //printf("\ttype='%s'\n", type);
```

```
        //
        // Before we go any farther, validate the parameters that were passed in
        //


        // Look-up Customer/Log Agent
        send_wait(0, top_object(), "call_attr", "cst", "val_by_key", agt_search_key,
                agt_userid, (int) 1, "id");
        if(msg_error()) {
                errmsg=format("Can't find Customer/Log Agent with a %s of '%s':
'%s'",
                        agt_search_key, agt_userid, msg[0]);
                _errmsg(errmsg);
                return;
        }

        uuid agt_id;
        agt_id=msg[1];

        string agt_combo;
                agt_combo=expand( format("&{'cnt:%s'=cnt.persistent_id-
>combo_name}", (string) agt_id));
        //printf("Customer/Log Agent user='%s' userid='%s' id='%s'\n", agt_combo,
agt_userid, agt_id);


        // Check if we are configured for ITIL
        string itil_env;
        int itil_flag;

        itil_env=getenv("NX_OTB_MARKET");
        //printf("ITIL? " + itil_env + "\n");
        if (itil_env=="itil") {
                itil_flag=1;
                //printf("ITIL\n");
        } else {
                itil_flag=0;
                //printf("Not ITIL\n");
        }



        // Validate the template name
        if(!is_empty(template)) {

                send_wait(0, top_object(), "call_attr", "cr", "val_by_key",
"template_name",
                        template, (int) 1, "persistent_id");
                if (msg_error()) {
                        errmsg=format("Can't find template '%s': '%s'",template,
msg[0]);
                        _errmsg(errmsg);
                        return;
                }
```

```
             string template_persid;
             template_persid=msg[1];
             //printf("template='%s' persid='%s'\n", template, template_persid);

             // Need this for later, so might as well do it now
             send_wait(0, top_object(), "call_attr", "cr", "dob_by_persid", 0,
template_persid);
             if (msg_error()) {
                     errmsg=format("Error in template dob_by_persid: '%s'",
msg[0]);
                     _errmsg(errmsg);
                     return;
             }

             object template_dob;
             template_dob=msg[0];
             //printf("got template dob\n");

     }


     // Validate the priority
     if(!is_empty(priority)) {

             send_wait(0, top_object(), "call_attr", "pri", "val_by_key", "sym",
                     priority, (int) 1, "enum");
             if (msg_error()) {
                     errmsg=format("Can't find priority '%s': '%s'",priority,
msg[0]);
                     _errmsg(errmsg);
                     return;
             }

             int pri_enum;
             pri_enum=msg[1];
             //printf("priority='%s' enum='%s'\n", priority, pri_enum);

     }


     // Validate the impact
     if(!is_empty(impact)) {

             send_wait(0, top_object(), "call_attr", "imp", "val_by_key", "sym",
                     impact, (int) 1, "enum");
             if (msg_error()) {
                     errmsg=format("Can't find impact '%s': '%s'",impact,
msg[0]);
                     _errmsg(errmsg);
                     return;
             }

             int imp_enum;
             imp_enum=msg[1];
             //printf("impact='%s' enum='%s'\n", impact, imp_enum);
```

```
        }

        // Validate the urgency
        if(!is_empty(urgency)) {

                send_wait(0, top_object(), "call_attr", "urg", "val_by_key", "sym",
                        urgency, (int) 1, "enum");
                if (msg_error()) {
                        errmsg=format("Can't find urgency '%s': '%s'",urgency,
msg[0]);

                        _errmsg(errmsg);
                        return;
                }

                int urg_enum;
                urg_enum=msg[1];
                //printf("urgency='%s' enum='%s'\n", urgency, urg_enum);

        }

        // Validate the severity
        if(!is_empty(severity)) {

                send_wait(0, top_object(), "call_attr", "sev", "val_by_key", "sym",
                        severity, (int) 1, "enum");
                if (msg_error()) {
                        errmsg=format("Can't find severity '%s': '%s'",severity,
msg[0]);

                        _errmsg(errmsg);
                        return;
                }

                int sev_enum;
                sev_enum=msg[1];
                //printf("severity='%s' enum='%s'\n", severity, sev_enum);

        }

        // Validate the status
        if(!is_empty(status)) {

                send_wait(0, top_object(), "call_attr", "crs", "val_by_key", "sym",
                        status, (int) 1, "code");
                if (msg_error()) {
                        errmsg=format("Can't find status '%s': '%s'",status,
msg[0]);

                        _errmsg(errmsg);
                        return;
                }

                string status_code;
                status_code=msg[1];
                //printf("status='%s' code='%s'\n", status, status_code);
```

```
        }


        // Validate the category
        if(!is_empty(category)) {

                send_wait(0, top_object(), "call_attr", "pcat", "val_by_key", "sym",
                        category, (int) 1, "persistent_id");
                if (msg_error()) {
                        errmsg=format("Can't find category '%s': '%s'",category,
msg[0]);

                        _errmsg(errmsg);
                        return;
                }

                string cat_persid;
                cat_persid=msg[1];
                //printf("category='%s' persid='%s'\n", category, cat_persid);

        }


        // Validate the assignee
        if(!is_empty(assignee)) {

                send_wait(0, top_object(), "call_attr", "agt", "val_by_key",
assignee_search_key,
                        assignee, (int) 1, "id");
                if (msg_error()) {
                        errmsg=format("Can't find assignee '%s': '%s'",assignee,
msg[0]);

                        _errmsg(errmsg);
                        return;
                }

                uuid assignee_id;
                assignee_id=msg[1];

                string assignee_combo;
                            assignee_combo=expand(
format("&{'cnt:%s'=cnt.persistent_id->combo_name}", assignee_id));
                //printf("assignee='%s' name='%s' id='%s'\n", assignee,
assignee_combo, assignee_id);

        }

        // Validate the group
        if(!is_empty(group)) {

                send_wait(0, top_object(), "call_attr", "grp", "val_by_key",
group_search_key,
                        group, (int) 1, "id");
                if (msg_error()) {
                        errmsg=format("Can't find group '%s': '%s'",group, msg[0]);
                        _errmsg(errmsg);
```

```
                        return;
                }

                uuid group_id;
                group_id=msg[1];

                string group_combo;
                        group_combo=expand(
format("&{'cnt:%s'=cnt.persistent_id->combo_name}", group_id));
                //printf("group='%s' name='%s' id='%s'\n", group, group_combo,
group_id);

        }


        // Validate the type
        if(!is_empty(type)) {

                if(itil_flag) {
                        send_wait(0, top_object(), "call_attr", "crt", "val_by_key",
"sym",
                                type, (int) 1, "code");
                        if (msg_error()) {
                                errmsg=format("Can't find type '%s': '%s'",type,
msg[0]);
                                _errmsg(errmsg);
                                return;
                        }

                        string type_code;
                        type_code=msg[1];
                        //printf("type='%s' code='%s'\n", type, type_code);
                } else {
                        errmsg=format("Not configured for ITIL.  Ignoring
type='%s'", type);
                        _errmsg(errmsg);
                }


        }


        // get a group leader so we can change things
        object group_leader;

        send_wait(0, top_object(), "get_co_group");
        if (msg_error()) {
                errmsg=format("Error in get_co_group: '%s'",msg[0]);
                _errmsg(errmsg);
                return;
        }

        group_leader = msg[0];
        //printf("got group leader\n");
```

```
    // Find the node
    uuid node_id;


    if (!is_empty(node)) {

            send_wait(0, top_object(), "call_attr", "nr", "val_by_key", "name",
node, (int) 1, "id");

            if (msg_error()) {
                    //printf("Asset '%s' not found: '%s' Adding new
Asset\n", node, msg[0]);

                    // Some type of error occured.  Find out what
happened.
                    if(msg[0]=="not found") {

                            // node wasn't found, so create it
                            object new_node;
                            //printf("starting logic to add Asset\n");

                            // Find the id of the asset class
                            int grc_id;
                            grc_id=(int) expand(format("&{'%s' =
grc.type->id}", asset_class));


                            if(0==grc_id || is_null(grc_id)) {
                                    errmsg=format("Can't find %s Asset
Class: '%s'", asset_class, msg[0]);

                                    _errmsg(errmsg);
                                    return;
                            }

                            //printf("using grc_id=%d\n",grc_id);

                            send_wait( 0, top_object(), "call_attr",
"nr",
                                    "get_new_dob", NULL, NULL,
group_leader );

                            if (msg_error()) {
                                    errmsg=format("Error in nr
get_new_dob: '%s'",msg[0]);

                                    _errmsg(errmsg);
                                    return;
                            }

                            new_node=msg[0];
                            //printf("got new nr dob\n");

                            // set stuff
                            new_node.name=node;
                            new_node.system_name=node;
                            new_node.description=asset_notes;
```

```
                                        // Guess why?
                                        send_wait(0, new_node, "call_attr", "class",
"set_val", grc_id);

                                        if (msg_error()) {
                                                errmsg=format("Error in nr class
set_val: '%s'",msg[0]);

                                                _errmsg(errmsg);
                                                return;
                                        }

                                        node_id=new_node.id;

                                        // Log it
                                        printf("%s Created Asset name='%s'
system_name='%s' class='%s'\n",

                                                the_date, new_node.name,
new_node.system_name, asset_class);
                                } else {
                                        // We got some other error besides "not
found"
                                        errmsg=format("Error in nr
val_by_key='%s'",msg[0]);

                                        _errmsg(errmsg);
                                        return;
                                }
                } else {
                        // The node was found.
                        node_id=msg[1];
                }

                //printf("using node='%s' id='%s'\n", node, node_id);
        }




        // Create the Request
        object cr;

        send_wait( 0, top_object(), "call_attr", "cr", "get_new_dob", NULL, NULL,
group_leader);
        if (msg_error()) {
                errmsg=format("Error in cr get_new_dob: '%s'",msg[0]);
                _errmsg(errmsg);
                return;
        }

        cr = msg[0];
        //printf("got new cr dob\n");



        // Do the template stuff here
        if (!is_null(template_dob)) {
                send_wait(0, template_dob, "make_from_template", group_leader, cr,
agt_id);
```

```
        }

        send_wait( 0, cr, "init_call_request", agt_id, agt_id, activity_log);
        if (msg_error()) {
                errmsg=format("Error in init_call_request: '%s'",msg[0]);
                _errmsg(errmsg);
                return;
        }

        //printf("init_call_request ok\n");

        cr.description=desc;

        if(!is_empty(node)) {
                cr.affected_resource=node_id;
        }

        if(!is_empty(priority)) {
                cr.priority=pri_enum;
        }

        if(!is_empty(impact)) {
                cr.impact=imp_enum;
        }

        if(!is_empty(urgency)) {
                cr.urgency=urg_enum;
        }

        if(!is_empty(severity)) {
                cr.severity=sev_enum;
        }

        if(!is_empty(status)) {
                cr.status=status_code;
        }

        if(!is_empty(category)) {
                cr.category=cat_persid;
        }

        if(!is_empty(assignee)) {
                cr.assignee=assignee_id;
        }

        if(!is_empty(group)) {
                cr.group=group_id;
        }

        if(!is_empty(charge_back)) {
                cr.charge_back_id=charge_back;
        }

        if(itil_flag) {
                if(!is_empty(type)) {
```

```
                                cr.type=type_code;
                } else {
                                cr.type="R";
                }
                //printf("type_code='%s'\n", type_code);
                if(type_code=="I") {
                                cr.incident_priority=cr.impact.value+cr.urgency.value;
                                //printf("cr.incident_priority='%d'\n",
cr.incident_priority);
                }
        }

        //SDT#10247 created_via
        cr.created_via = 3553;

        send_wait(0, group_leader, "checkin");
        if (msg_error()) {
                errmsg=format("Error in checkin: '%s'",msg[1]);
                _errmsg(errmsg);
                send_wait(0, group_leader, "uncheck");
                return;
        } else {
                //printf("checkin ok\n");
                string log_msg;
                log_msg  = format("%s Created Request # '%s' ",
                        (string) cr.open_date, cr.ref_num);
                log_msg += format("Description='%s' ", cr.description);
                log_msg += format("Asset='%s' ", cr.affected_resource.name);
                log_msg += format("Template='%s' ", cr.template_name);
                log_msg += format("Priority='%s' ", cr.priority.sym);
                log_msg += format("Status='%s' ", cr.status.sym);
                log_msg += format("Category='%s' ", cr.category.sym);
                log_msg += format("Assignee='%s' ", cr.assignee.combo_name);
                log_msg += format("Group='%s' ", cr.group.combo_name);
                log_msg += format("Charge Back='%s' ", cr.charge_back_id);
                log_msg += format("Impact='%s' ", cr.impact.sym);
                log_msg += format("Urgency='%s' ", cr.urgency.sym);
                log_msg += format("Severity='%s' ", cr.severity.sym);
                log_msg += format("Type='%s' ", cr.type.sym);
                log_msg += "\n";

                // Log it
                printf(log_msg);

                return cr.ref_num;

        }
}




/////////////////////////////////////////////////////////////////////
// Method:       _errmsg
//
```

```
// Description: Internal routine to print error message.
//
// Input      : string errmsg
//
// Returns    : Nothing
//
//              Computer Associates
//              06/13/2000
//
//////////////////////////////////////////////////////////////////////
void _errmsg(...) {
      string the_date;
      the_date=(string) now();
      printf(format("%s ERROR %s\n", the_date ,argv[0]));
}
```

ZUpdateCr Method

Important! The following code is *not* provided with CA Unicenter Service Desk. It is a customization, and therefore is *not* supported by CA Support. If you want to use this code and have CA support it, you must complete a support agreement using CA Technology Services. Otherwise, you are responsible for the support of this utility and any questions or errors related to its use.

```
/////////////////////////////////////////////////////////////////////////////
/////////////////////////
//
// Method:      ZUpdateCr(...)
//
// Description: This FRAG file updates the status of a Request and add activity log
comment from
//              the command line using the bop_cmd command.
//
// Input:       RequestNo
//                  User_id
//              New Status
//                  Telealert Message
//
//              All parameters are required.
// Notes : Program validates Ref_Num, User_id and New_Status
// Usage Example:
// bop_cmd -f ZUpdateCr.frg "ZUpdateCr('Ref_Num', 'User_id', 'New_Status', 'Telalert
Message')"
//
/////////////////////////////////////////////////////////////////////////////
//////////////////
string ZUpdateCr(string RequestNo, string assignee, string status, string message)
{


      string errmsg;
      string assignee_search_key;
      assignee_search_key="userid";
```

```
 // Fetching the CR domset
 // belgl01 Comment : Validating that Request/Incident/Problem number exist in
Service Desk

        send_wait(0,top_object(),"call_attr","cr","sync_fetch","RLIST_DYNAMIC",
format("ref_num ='%s'",RequestNo), -1,0);
        if (msg_error()) {
                        errmsg=format("Cant find DOMSET '%s': '%s'\n",RequestNo,
msg[0]);

                         _errmsg(errmsg);
                }


        object get_cr_domset;
        int domset_size;
        get_cr_domset = msg[0];
        domset_size = msg[1];
        if (domset_size != 1)
         {
                printf("\n");
                printf("The Request doesnt exist \n");
                printf("Enter a valid Request No \n");
                errmsg=format("Found %d records for '%s'\n", domset_size ,
RequestNo);
                _errmsg(errmsg);
          }

  // Get the CR dob
        send_wait(0,get_cr_domset, "dob_by_index", "DEFAULT", 0,0);
        if (msg_error())
        {
                        errmsg=format("Unable to get dob for  '%s':
'%s'\n",RequestNo, msg[0]);
                         _errmsg(errmsg);
          }
         object get_cr_dob;
         get_cr_dob = msg[0];


   // get a group leader so we can change things


        send_wait(0, top_object(), "get_co_group");
        if (msg_error()) {
                errmsg=format("Error in get_co_group: '%s'\n",msg[0]);
                 _errmsg(errmsg);
        }

        object group_leader;
        group_leader = msg[0];
```

```
    //Checkout the DOB

        send_wait(0,group_leader,"checkout",get_cr_dob);
         if (msg_error())
         {
                errmsg=format("Unable to checkout: '%s'\n",msg[0]);
                _errmsg(errmsg);
         }



    // Validate the status
                send_wait(0, top_object(), "call_attr", "crs", "val_by_key", "sym",
                     status, (int) 1, "code");
                if (msg_error()) {
                       printf("Invalid status :'%s'\n",status);
                       printf("Please pass a valid Status");
                        errmsg=format("Can't find status '%s': '%s'",status,
msg[0]);

                         _errmsg(errmsg);
                         return;
                              }
                string status_code;
                status_code=msg[1];
                // printf("status='%s' code='%s'\n", status, status_code);

    //Update the Status
                if(status !='')
                {
                get_cr_dob.status=status_code;
                }



// get the agent who is making the log entry
  // belgl01 adding code to validate and define who is making the comment submitted
by Telalert.



//   Validate the assignee

                send_wait(0, top_object(), "call_attr", "agt", "val_by_key",
assignee_search_key,
                     assignee, (int) 1, "id");
                if (msg_error()) {
                       printf("Invalid assignee :'%s'\n",assignee);
                      printf("Please pass a valid Assignee");
                       errmsg=format("Can't find assignee '%s': '%s'",assignee,
msg[0]);

                         _errmsg(errmsg);
                         return;
                }

                uuid assignee_id;
```

```
                    assignee_id=msg[1];

                    string assignee_combo;
                      assignee_combo=expand( format("&{'cnt:%s'=cnt.persistent_id-
>combo_name}", assignee_id));
                      printf("assignee='%s' name='%s' id='%s'\n", assignee,
assignee_combo, assignee_id);




//////////////////////////////////////////////////////////////////////////////
        //      GET CR persid SECTION
        //////////////////////////////////////////////////////////////////////////////
/////
        string cr_persid;

        if (!is_empty(RequestNo)) {
                send_wait(0, top_object(), "call_attr", "cr", "val_by_key",
"ref_num",
                        RequestNo, (int) 1, "persistent_id");
                if ( msg_error()) {
                        errmsg=format("ERROR: Error with cr lookup on ref_num: '%s'",
msg[0]);
                        _errmsg(errmsg);
                        return -1;
                }
                cr_persid = msg[1];
        }
        printf("\nref_num\t\t->\t'%s'\t\t(CR_PERSID = %s)",ref_num, cr_persid);

        //////////////////////////////////////////////////////////////////////////////
/////
        //      ACTIVITY LOG CREATE SECTION
        //////////////////////////////////////////////////////////////////////////////
/////
        // Get new dob
        object new_actlog;
        send_wait(0, top_object(), "call_attr", "alg", "get_new_dob", NULL, NULL,
group_leader);
        if (msg_error()) {
                errmsg=format("ERROR: Error creating Activity Log dob:
'%s'",msg[0]);
                _errmsg(errmsg);
                return -1;
        }
        //printf("Got new dob\n");
        new_actlog = msg[0];

        new_actlog.description = message;
        new_actlog.action_desc = "Activity Log entered by " + assignee + " via the
ZUpdateCr.FRG file";
        new_actlog.call_req_id = cr_persid;
        new_actlog.time_stamp = now();
        new_actlog.analyst = assignee_id;
```

```
        new_actlog.type = "LOG";

        new_actlog.time_spent = 30;  // 30 secs

        send_wait(0, group_leader, "checkin");



          if (msg_error())
         {
              errmsg=format("Checkin Failed: '%s\n'",msg[0]);
               _errmsg(errmsg);
         }
         else
         {
           string log_msg;
         log_msg += format("Status='%s' ", get_cr_dob.status.sym,assigned_id);
         printf("Successfully Changed Status and Logged Comment from Telalert for
Request# '%s' ",RequestNo);

         return get_cr_dob.ref_num;
          }


   }
```

sutility

Important! The following code is *not* provided with CA Unicenter Service Desk. It is a customization, and therefore is *not* supported by CA Support. If you want to use this code and have CA support it, you must complete a support agreement using CA Technology Services. Otherwise, you are responsible for the support of this utility and any questions or errors related to its use.

```
/* Copyright (c) 2001 By Gladys Beltran */

/* This code is an example of a way of taking information from an input file (a text
file) and formatting it into bop_cmd syntax. As a result of that operation, an
output file is generated, and is available to be executed by a batch file at any
time to update information in Service Desk 6.0 or r11.x*/

#include <stdio.h>

#define COMANDO_INICIO "bop_cmd -f mk_creq.frg \"make_chg ("
#define COMANDO_FIN     ")\""
#define SEPARADOR       ", "

int main(int argc, char **argv)
{
        int i;
        FILE *in, *out;
        char entrada[255];
```

```c
        printf("\nInput file : %s\nOutput file: %s\n",argv[1], argv[2]);

        if ((in = fopen(argv[1], "rt"))        == NULL){
         fprintf(stderr, "\nCannot open input file.\n");
         return 1;
        }

        if ((out = fopen(argv[2], "wt"))       == NULL){
         fprintf(stderr, "\nCannot open output file.\n");
         return 1;
        }

        printf("\nWorking.");

        fprintf(out,"%s",COMANDO_INICIO);

        do{
                char c=fgetc(in);
                switch(c){
                        case '\t' : fprintf(out,"%s",SEPARADOR);
                                                                break;
                        case '\n' :
fprintf(out,"%s\n%s",COMANDO_FIN,COMANDO_INICIO);
                                                                break;
                        default   : fputc(c, out);
                                                                break;
                }
                printf(".");
        }while(!feof(in));

        fprintf(out,"%s",COMANDO_FIN);

        printf("\nFinished");

        fclose(in);
        fclose(out);
        return 0;
}
```

Index

U

UAPM
   integration with • 89
Unicenter NSM
   integration with • 35
Unicenter Service Catalog
   integration with • 99
Unicenter Service Delivery
   integration with • 99
Unicenter UAPM
   integration with • 89
URL integration • 263

W

Workflow IDE
   integration with • 181