

CenTran[®] Version 5 Agent Programming Reference Manual

dormakaba

dormakaba Welcome

dormakaba is the world's leading manufacturer and supplier of high security, electronic safe locks. Its complete line of self powered, battery, and mechanical locks meets virtually every safe lock requirement. dormakaba is dedicated to satisfying end-user needs for security, safety, and convenience. We welcome you to the world of dormakaba security and Cencon[®].

[Manual Conventions](#)

[Trademarks](#)

Table Of Contents

[Chapter 1 - Introduction](#)

- 1.1 [Cencon System Product Overview](#)
- 1.2 [CenTran Agent Overview](#)
- 1.3 [Generic format of input transactions with xml content](#)
- 1.4 [Generic format of output transactions with xml content](#)
- 1.5 [Version Example](#)
- 1.6 [Lock Status Example](#)
- 1.7 [Lock Event Status Example](#)

Chapter 2 - Status XML Transactions

- 2.1 [Agent Version](#)
- 2.2 [Agent Status](#)
- 2.3 [ATM Lock Status](#)

Chapter 3 - Agent Configuration XML Transactions

- 3.1 [Agent Configuration Setting](#)

Chapter 4 - Event Monitoring XML Transactions

- 4.1 [Lock Event Status Changed](#)

Manual Conventions

This CenTran Agent Programming Reference Manual contains information on the installation and set up of the software required to monitor a lock. Some conventions that are followed are:

- Input to the PC will be shown in **bold type**.
- The screen samples may not look exactly as they do on your system depending on which version of the Windows operating system you are using. Some samples in this document are from Windows XP, some are from Windows Vista, and some are from Windows 7.
- Highlighted items:

Note

Indicates a point to consider.

Helpful Hint

Indicates a technique or method that works well.

Caution

Indicates the need for care and caution to be observed during a procedure or process.

Warning

Indicates the possibility of loss of data or system integrity.

Before reading the CenTran Agent Programming Reference Manual, you should be familiar with basic Windows concepts and techniques. For detailed information, refer to your Windows documentation.

Trademarks

The following items are trademarks or registered trademarks of dormakaba in the United States and/or other countries.

- Cencon System 2000
- Cencon
- PowerStar Technology
- Smart Key

Adobe and Adobe Reader are registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

iButton is a registered trademark of Dallas Semiconductor.

Microsoft, Windows, and Windows Server are registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Notice: The information in this manual is subject to change without notice and does not represent a commitment on the part of the dormakaba. dormakaba shall not be liable for technical or editorial errors or omissions contained herein; nor for incidental or consequential damages resulting from the furnishing, performance or use of this material.

CenTran 5 Agent XML Transaction Format

Chapter 1 - Introducing CenTran® Version 5 Agent XML Programming

1.1 Cencon ATM Cash Vault Security System Product Overview

The Challenge

While the front of ATMs are accessed by customers with individual cards and PINs, the back of the ATM is often accessed by service personnel from multiple companies all using the same lock combination. Bank employees, cash replenishers, technical service reps, repair personnel all using the same combination to open an ATM safe poses a significant security risk.

The Solution

The Cencon ATM Cash Vault Security System from Kaba Mas is designed to combat insider theft from ATMs through the combined use of lock hardware, systems software and Smart Keys. The Cencon system offers total access control and accountability with its One Time Combination feature. The One Time Combination is dispatched from a central location and cannot be reused at a later date, thus eliminating temptation.

The Cencon ATM Cash Vault Security System allows locks located anywhere in the world to be controlled and monitored from one central location.

The central location consists of

- A database describing locks, lock administration personnel, lock service personnel, lock access details and history.
- Kaba Mas software that
 - Manages the creation and distribution of One Time Combinations(OTC).
 - Tracks the status of each lock and user.
 - Provides audit capability.
 - Can be used interactively by a human Dispatcher or programmatically by customer applications such as IVR or Web base interfaces.

The software provided by Kaba Mas consists of

- Cencon Interactive Application: Interactive Central Point of Control
 - Used to create, distribute, manage One Time Combinations (OTC).
- CenTran Service: Background Central Point of Control
 - Used by customer, 3rd party applications to create, distribute and manage OTCs.
- CenTran Web Services: Used by customer web applications to communicate with CenTran Service.
- CenTran Agent
 - Remote lock management: Allows CenTran Service to communicate with ATM lock, streamlining lock administration.
- Key Box Agent
 - Remote key management: Allows CenTran Service to communicate with remote key boxes, supporting decentralized lock personnel.

The Cencon ATM Cash Vault Security System supports three Modes of Operation:

FLM Mode

Designed for First Line Maintenance (FLM) personnel such as cash replenishers, technical service reps, repair personnel, the FLM mode of the Cencon lock permits the dispatch center to give a One Time Combination (OTC) for ATM lock access to a technician "on the fly." FLM personnel are each issued a Smart Key and call their dispatch center to request a One Time Combination. Once used, this combination will not function again, thus ensuring absolute access control. Once the Cencon unit is locked, a Close Seal code is displayed on the lock's LCD. Entering this Close Seal into the dispatch center's computer documents that the Cencon lock is locked and secure.

Route Mode

While permitting the same access control as FLM Mode, Route Mode additionally permits pre-defined routes for armored cash replenishers. Route personnel are each issued a Smart Key. A route of locks, along with an OTC for each lock on the route, may be assigned by a dispatch center permitting route personnel to access a series of locks without contacting their dispatch center at each individual lock. Route personnel can either call their dispatch center at the end of the shift to report the Close Seals, or the Close Seals can be read from the Smart Key by the Cencon software.

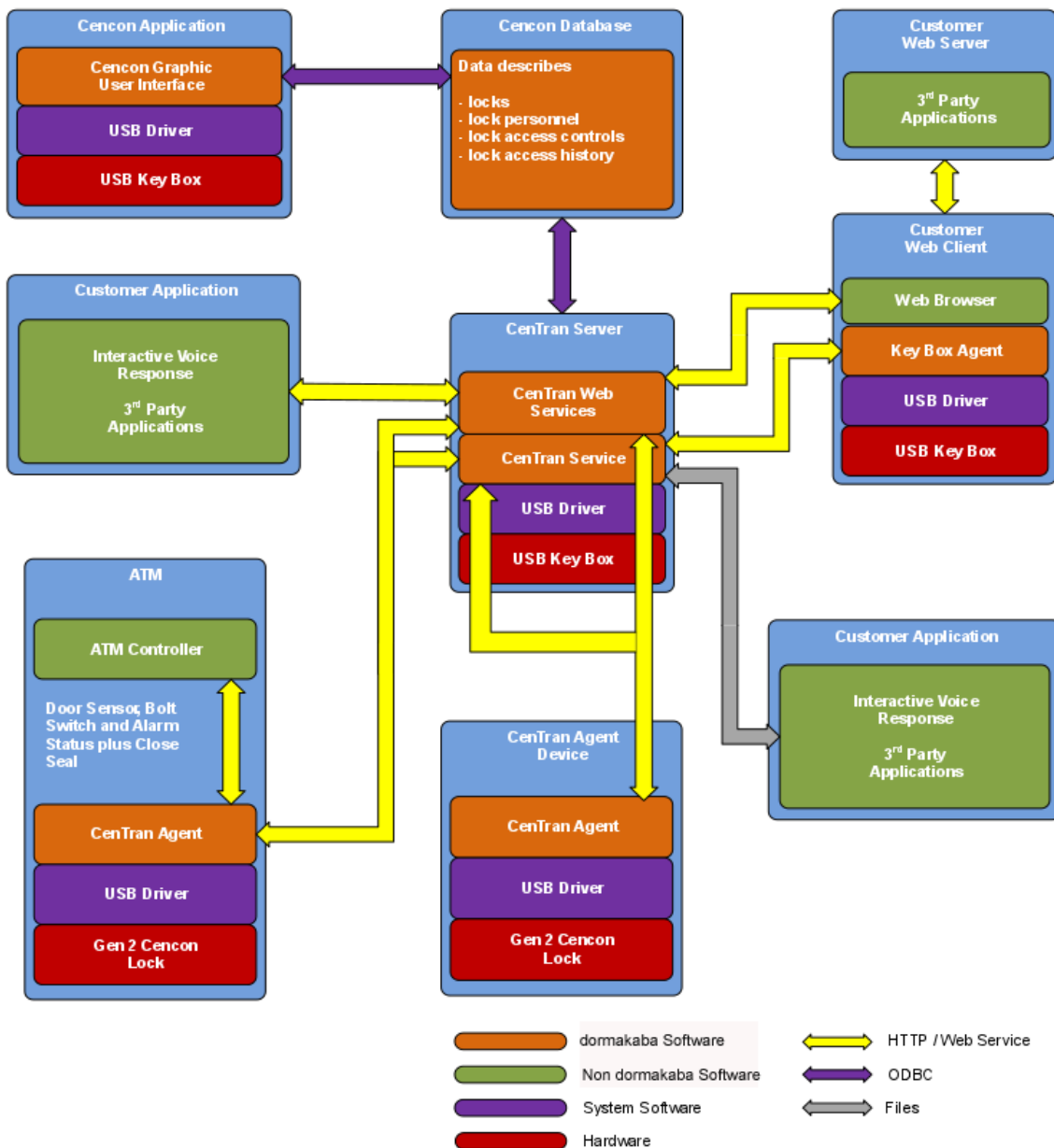
Bank Mode

Bank Mode is designed for bank personnel who may need access to the Cencon system lock but do not wish to operate through a dispatch center. Smart Keys and combinations are issued to bank personnel who may unlock the Cencon lock at any time. An internal audit trail tracks all activity of bank personnel just as it does for FLM and Route personnel. Kaba Mas Cencon software permits bank administrators to manage Bank Mode on site. Bank users may be added or deleted and audit trails may be downloaded from the lock.

High Level Overview of Cencon ATM Cash Vault Security System Component Architecture

The Cencon ATM Cash Vault Security System consists of several components whose method of use will vary depending on the business model of the supported enterprise. Presented here is a brief high level overview of the Cencon ATM Cash Vault Security System components.

Cencon ATM Cash Vault Security System Component Architecture



Cencon Database:

- Defines locks to be controlled, monitored and audited.
- Defines personnel allowed to access locks.
- Defines Personnel allowed to control access to locks.
- Contains access history for each lock.

Cencon Application:

- Interactive, graphical Windows application used to manage the Cencon Database contents.
- Requires a Cencon Key Box be attached to a USB port of the same computer.

CenTran Service:

- Available for use by customer applications to manage the Cencon Database contents.

- Applications communicate with the service via files or HTTP / Web Services.
- Runs in background as a Windows Service.

CenTran Web Services:

- Available for use by web applications to manage Cencon Database contents.

Customer Application Using CenTran Service Command File:

- The CenTran Service performs actions against the Cencon Database per commands imbedded in files.
- Results of those actions are reported in output files created by the CenTran Service.

Customer Application Using CenTran Web Services:

- The CenTran Service performs actions against the Cencon database per commands received via the CenTran Web Services.
- Results of those actions are reported to the customer application via the CenTran Web Services.

CenTran Agent:

- Allows CenTran Service to monitor lock in a real time manner.
- Runs on the ATM in the background as a Windows Service.

Key Box Agent:

- Allows CenTran Service access to Cencon Key Boxes on remote systems.
- Runs on remote Web Client systems in background as a Windows Service.

Customer Web Client:

- Customer developed web browser based user interface used to manage Cencon Database contents as well as Cencon Smart Keys via the CenTran Web Services.
- Can be located remotely.

Cencon ATM Cash Vault Security System Components and Business Processes

Examples of Business Processes Supported by Cencon ATM Cash Vault Security System Components:

1. FLM (First Line Maintenance) Operations with human Dispatcher:
 1. FLM personnel reaches ATM and contacts Dispatcher requesting OTC.
 2. Dispatcher generates OTC via Cencon interactive application, gives OTC to FLM personnel.
 3. FLM personnel opens and closes lock, recording the Close Seal.
 4. FLM personnel gives Close Seal to Dispatcher who enters it into Cencon database via Cencon interactive application.
2. Route Operations with human Dispatcher:
 1. Before starting route, Route personnel contacts Dispatcher and requests OTCs for each lock on route.
 2. Dispatcher generates OTCs via Cencon interactive application, gives OTCs to Route personnel.
 3. Route personnel opens and closes each lock on route, recording Close Seals.
 4. When the Route has been completed, the Close Seals are given to the Dispatcher who enters them into the Cencon database via Cencon interactive application.
3. FLM Operations with IVR (Interactive Voice Response) system:
 1. FLM personnel reaches ATM, calls IVR system and requests OTC.
 2. IVR system generates OTC via CenTran Command File interface and reports OTC to FLM person.
 3. FLM personnel opens and closes lock, recording the Close Seal.
 4. FLM personnel calls IVR system and enters Close Seal.
 5. IVR system records Close Seal in Cencon database via CenTran Command File interface.

Though business processes will vary between companies there are core lock management tasks that will be used in every enterprise such as:

- Create User and Initialize Smart Key
- Delete User
- Dispatch Lock
- Open / Close physical lock
- Close Lock in Cencon database
- Reassign Lock
- Audit Lock
- Audit Key
- Audit Cencon Database

The following lists describe for a few of the core lock management tasks one possible flow of control through the Cencon ATM Cash Vault Security System components.

Create User and Initialize Smart Key:

- Dispatcher ->
- Customer Web Client ->
- CenTran Web Services ->
- CenTran Service ->

- CenTran Web Services ->
- Key Box Agent (writes KEY) ->
- CenTran Service ->
- CenTran Web Services ->
- Customer Web Client

Note:

- Customer Web Client & Key Box Agent will probably be on same location but do not have to be
- Customer Web Client does NOT require a Keybox on the same system

Dispatch Lock(s):

- Dispatcher ->
- Customer Web Client ->
- CenTran Web Services ->
- CenTran Service ->
- Cencon Database ->
- CenTran Service ->
- CenTran Web Services ->
- Customer Web Client ->
- Dispatcher receives OTC
- Dispatcher transmits OTC to FLM/Route personnel

Lock is physically opened and closed:

- Service personnel calls Close Seal in to the Dispatcher who "Closes Lock" in the Cencon Database. The component flow is the same as that described for Dispatch Lock(s).
 - OR
- CenTran Agent Service on ATM uses CenTran Web Services to report Closure to CenTran Service and "Closes Lock" in the Cencon Database programmatically:
 - Lock is closed ->
 - CenTran Agent ->
 - CenTran Web Services ->
 - CenTran Service ->
 - Cencon Database ->
 - CenTran Service ->
 - CenTran Web Services ->
 - CenTran Agent

1.2 CenTran Agent Overview

The CenTran Agent is started from an ATM containing a Cencon lock. Transactions are exchanged between the customer's program and the CenTran Agent via TCP/IP connections. The customer's program opens a TCP/IP connection to the CenTran Agent and sends the XML for the transaction over the connection. The CenTran Agent reads the incoming transaction and eventually sends the XML for the output transaction back over the same connection.

The content of each XML input and output (response) transaction varies. The format for each transaction is specified in an XML schema provided with the CenTran Agent. The XML for the input transactions must conform to the format specified in the given schema.

The different categories of transaction include Status, Configuration, and Event Monitoring transactions.

The generic format of input and output transactions is discussed in sections 1.3 and 1.4. Sections 1.5 and 1.6 have examples illustrating how the CenTran transaction processor can be used. Sections two to four have details on how each transaction works - what information is required, what information is returned, examples of actual input and output.

1.3 Generic Format of Input Transactions with XML Content

The input transactions which have XML content in them should conform to the format specified in the given schema.

- Most of the transactions have a format where the input elements (xml elements of the input transaction) are listed sequentially in the order specified in the schema.
- In the case of transactions which update the agent's configuration, only those fields of the configuration file corresponding to elements present in the input transaction are updated and the rest of the fields are not changed.

Example of an Input Transaction with XML Content

```
<?xml version="1.0" encoding="utf-8"?>
<CentranAgentXMLScript xmlns="http://www.kaba-mas.com/CenTranAgentCommands.xsd">
<COMMAND_GET_LOCK_STATUS>
  <TransmissionTimestamp>2013-04-25T15:44:52Z</TransmissionTimestamp>
</COMMAND_GET_LOCK_STATUS>
</CentranAgentXMLScript>
```

1.4 Generic Format of Output Transactions with XML Content

The output transactions which have xml content in them should also conform to the format specified in the given schema.

- If the input transaction does not conform or cannot be validated against the format specified in the schema, CenTran returns an output transaction with the details of the validation errors.
- All the output transactions have the input transaction, followed by a result record indicating success or failure. The result record has:
 - A result element - "Success" or "Failure".
 - An error code (an extensive but not exhaustive list can be found in Appendix A of the CenTran 5 Programming Guide). The error code for a completely successful transaction is always RC_OK.
 - A brief description of the result.
- After the result record, if the transaction has succeeded, there could be data from one or more records retrieved from CenTran or the lock.

For example, if a lock status request has succeeded, the status portion of the database record of the lock is retrieved and appended to the output after the result record.

- If a transaction has failed, there may or may not be extra information provided at the end of a result record, depending on the type of transaction.

Example of an Output Transaction with XML Content

```
<?xml version="1.0"?>
<CentranAgentXMLScript xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://www.kaba-mas.com/CentranAgentCommands.xsd">
  <RESPONSE_GET_LOCK_STATUS>
    <COMMAND_GET_LOCK_STATUS>
      <TransmissionTimestamp>2013-04-25T15:44:52Z</TransmissionTimestamp>
    </COMMAND_GET_LOCK_STATUS>
    <ResultRecord>
      <Result>Success</Result>
      <ErrorCode>RC_OK</ErrorCode>
      <ResponseTimestamp>2013-04-25T15:44:53Z</ResponseTimestamp>
      <Description>Lock status successfully retrieved directly from lock.</Description>
    </ResultRecord>
    <LockStatus>
      <DuressAlarmStatus>Deactivated</DuressAlarmStatus>
      <SafeDoorStatus>SafeDoorSensorNotConnected</SafeDoorStatus>
      <LockBoltStatus>Closed</LockBoltStatus>
      <BankModeActive>SingleModeActive</BankModeActive>
      <RouteModeActive>DualModeActive</RouteModeActive>
      <FLMModeActive>DualModeActive</FLMModeActive>
      <FirmwareVersion>
        <MajorVersion>2</MajorVersion>
        <MinorVersion>1</MinorVersion>
        <BuildVersion>1061</BuildVersion>
      </FirmwareVersion>
      <SerialNumber>001000112</SerialNumber>
    </LockStatus>
  </RESPONSE_GET_LOCK_STATUS>
</CentranAgentXMLScript>
```

1.5 Version Example

The following example illustrates how you would request the CenTran Agent version information:

```
<CentranAgentXMLScript xmlns="http://www.kaba-mas.com/CentranAgentCommands.xsd">
<COMMAND_GET_AGENT_VERSION>
  <TransmissionTimestamp>2013-04-25T15:34:26Z</TransmissionTimestamp>
</COMMAND_GET_AGENT_VERSION>
</CentranAgentXMLScript>
```

The following example is what the response from the CenTran Agent for version information looks like:

```
<CentranAgentXMLScript xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://www.kaba-mas.com/CentranAgentCommands.xsd">
  <RESPONSE_GET_AGENT_VERSION>
    <COMMAND_GET_AGENT_VERSION>
      <TransmissionTimestamp>2013-04-25T15:34:26Z</TransmissionTimestamp>
    </COMMAND_GET_AGENT_VERSION>
    <ResultRecord>
      <Result>Success</Result>
      <ErrorCode>RC_OK</ErrorCode>
      <ResponseTimestamp>2013-04-25T15:34:27Z</ResponseTimestamp>
      <Description>Get version of currently running CenTran Agent.</Description>
    </ResultRecord>
    <AgentVersion>4.1.5.3079</AgentVersion>
  </RESPONSE_GET_AGENT_VERSION>
</CentranAgentXMLScript>
```

1.6 Lock Status Example

The following example illustrates how you would use the transaction processor to obtain the status of the ATM lock.

The response transaction from the CenTran Agent will have the original input transaction followed by a result record and the lock

status data retrieved by CenTran.

The example is in C++, using the .NET library.

Sample Code for Obtaining Lock Status

```
// CenTran Agent command port - set in configuration file
Int32 agent_command_port = 35006;

// Lock status transaction - XML format
System::String^ agent_request =
    "<COMMAND_GET_LOCK_STATUS>" +
    "<TransmissionTimestamp>2013-04-25T15:44:52Z</TransmissionTimestamp>" +
    "</COMMAND_GET_LOCK_STATUS>";

// Create a TcpClient connected to the agent
TcpClient^ client = gcnew TcpClient("127.0.0.1", agent_command_port);

// Adjust connection timeout for reading data so we know we'll end
client->ReceiveTimeout = 25000;

// Get stream to read/write with
NetworkStream^ stream = client->GetStream();

// Create XML Writer for the stream
System::Xml::XmlWriter^ writer = System::Xml::XmlWriter::Create(stream);

// Write XML declaration
writer->WriteStartDocument();

// First element is always CentranAgentXMLScript
String^ strTransaction = "<CentranAgentXMLScript xmlns=\"http://www.kaba-mas.com/CentranAgentCommands.xsd\">" +
    agent_request + "</CentranAgentXMLScript>";

// Load XML Document from transaction string
System::Xml::XmlDocument^ document = gcnew System::Xml::XmlDocument;
document->LoadXml(strTransaction->Trim());

// Save the document to the stream
document->Save(writer);

// XML declaration
writer->WriteEndDocument();

// Flush writer's buffer
writer->Flush();

// Create XmlReader reader to receive returned document
System::Xml::XmlReader^ reader = System::Xml::XmlReader::Create(stream);

// Load XML response to XML document
document->Load(reader);

// Close XML reader
reader->Close();

// Close the connection to the agent
client->Close();

// Get lock serial number as a string
System::Xml::XmlNodeList^ xlist1 = document->GetElementsByTagName("SerialNumber");
if (xlist1->Count > 0)
{
    Console::WriteLine("Received: {0}", xlist1->Item(0)->ChildNodes[0]->Value->Trim());
}
}
```

1.7 Lock Event Status Example

The following example illustrates how you would listen for the status of ATM lock events.

The lock event status changed transaction from the CenTran Agent will have a result record and the lock event status data retrieved by CenTran.

The example is in C++, using the .NET library.

Sample Code for Obtaining Lock Event Status

```
// CenTran Agent event reporting port - set in configuration file
Int32 event_report_port = 35007;

// Establish the UDP listener
UdpClient^ listener = gcnew UdpClient(event_report_port);
IPEndPoint^ listener_endpoint = gcnew IPEndPoint(IPAddress::Any, event_report_port);

// XML document for parsing the lock event status XML
System::Xml::XmlDocument^ document = gcnew System::Xml::XmlDocument;
System::Xml::XmlNodeList^ xlist1;

while (true)
{
    // loop through as event reports arrive, writing them to the screen
    array<Byte>^ bytes_received = listener->Receive(listener_endpoint);

    // Get string version of UDP message
    String^ event_text = Text::Encoding::ASCII->GetString(bytes_received, 0, bytes_received->Length);
}
```

```
// Load message string into XML document
document->LoadXml(event_text);

// Examine the result code
xlist1 = document->GetElementsByTagName("Result");
if (xlist1->Count > 0)
{
    Console::WriteLine("Result: {0}", xlist1->Item(0)->ChildNodes[0]->Value->Trim());
}
}
```

CenTran 5 Agent Programming Guide

Chapter 2 - Status XML Transactions

2.1 CenTran Agent Version

This transaction is used to read the CenTran Agent version.

Transaction information required:

- No information required.

Transaction information returned:

Success:

- CenTran Agent version

Failure:

There should not be a failed return of data unless an internal error or connectivity error with the CenTran Agent has occurred.

Example

Example Input Transaction:

```
<COMMAND_GET_AGENT_VERSION>
  <TransmissionTimestamp>2013-04-25T15:34:26Z</TransmissionTimestamp>
</COMMAND_GET_AGENT_VERSION>
```

Example Output Transaction:

```
<RESPONSE_GET_AGENT_VERSION>
  <COMMAND_GET_AGENT_VERSION>
    <TransmissionTimestamp>2013-04-25T15:34:26Z</TransmissionTimestamp>
  </COMMAND_GET_AGENT_VERSION>
  <ResultRecord>
    <Result>Success</Result>
    <ErrorCode>RC_OK</ErrorCode>
    <ResponseTimestamp>2013-04-25T15:34:27Z</ResponseTimestamp>
    <Description>Get version of currently running CenTran Agent.</Description>
  </ResultRecord>
  <AgentVersion>4.1.5.3079</AgentVersion>
</RESPONSE_GET_AGENT_VERSION>
```

CenTran 5 Agent Programming Guide

Chapter 2 - Status XML Transactions

2.2 CenTran Agent Status

This transaction is used to read the CenTran Agent status.

Transaction information required:

- No information required.

Transaction information returned:

Success:

- CenTran Agent version
- CenTran Agent command port
- CenTran Agent pass-through IP address
- CenTran Agent pass-through port
- CenTran Agent lock event reporting IP address
- CenTran Agent lock event reporting port
- CenTran server IP address
- CenTran server port
- CenTran server agent web service URI
- CenTran Agent pass-through availability
- Timestamp for CenTran Agent service start time
- Number of lock events reported by the CenTran Agent
- Number of lock status requests received by the CenTran Agent
- Number of agent version requests received by the CenTran Agent
- Number of bytes delivered from the lock by the CenTran Agent
- Number of bytes delivered to the lock by the CenTran Agent
- Number of connections to the lock through the pass-through

Failure:

There should not be a failed return of data unless an internal error or connectivity error with the CenTran Agent has occurred.

Example

Example Input Transaction:

```
<COMMAND_GET_AGENT_STATUS>
  <TransmissionTimestamp>2013-04-25T20:34:53Z</TransmissionTimestamp>
</COMMAND_GET_AGENT_STATUS>
```

Example Output Transaction:

```
<RESPONSE_GET_AGENT_STATUS>
  <COMMAND_GET_AGENT_STATUS>
    <TransmissionTimestamp>2013-04-25T20:34:53Z</TransmissionTimestamp>
  </COMMAND_GET_AGENT_STATUS>
  <ResultRecord><Result>Success</Result>
    <ErrorCode>RC_OK</ErrorCode>
    <ResponseTimestamp>2013-04-25T20:34:53Z</ResponseTimestamp>
    <Description>Get status of currently running CenTran Agent.</Description>
  </ResultRecord>
  <AgentVersion>4.1.5.3046</AgentVersion>
  <CommandIPPort>35006</CommandIPPort>
  <PassThroughIPAddress>127.0.0.1</PassThroughIPAddress>
  <PassThroughIPPort>35005</PassThroughIPPort>
  <ReportEventIPAddress>127.0.0.1</ReportEventIPAddress>
  <ReportEventIPPort>35007</ReportEventIPPort>
  <CentranIPAddress />
  <CentranIPPort>0</CentranIPPort>
  <CentranServiceURI>https://centranserver/CenTranAgentService/rest/</CentranServiceURI>
  <IsPassThroughRunning>true</IsPassThroughRunning>
  <AgentStartTimestamp>2013-04-25T18:43:28Z</AgentStartTimestamp>
  <AgentStatistics>
    <LockEventsReported>0</LockEventsReported>
    <LockStatusRequests>6</LockStatusRequests>
    <AgentStatusRequests>1</AgentStatusRequests>
    <AgentVersionRequests>6</AgentVersionRequests>
  </AgentStatistics>
  <PassThroughStatistics>
    <PassThroughBytesOut>0</PassThroughBytesOut>
    <PassThroughBytesIn>0</PassThroughBytesIn>
    <PassThroughConnections>0</PassThroughConnections>
  </PassThroughStatistics>
</RESPONSE_GET_AGENT_STATUS>
```


CenTran 5 Agent Programming Guide

Chapter 2 - Status XML Transactions

2.3 ATM Lock Status

This transaction is used to read the ATM Cencon lock status. The Centran server is queried for the lock status first; if this fails, the lock is queried directly. The lock is also queried directly if the Centran Agent has determined that it supports direct queries.

Transaction information required:

- No information required.

Transaction information returned:

Success:

Input transaction, followed by a result record (result, error code and description) and the lock status record.

Failure:

Input transaction, followed by a result record (result, error code and description).

Examples of Failure Conditions (for an extensive list with error codes, refer to Appendix A):

- No lock is connected to the ATM
- The lock is currently in use

Example 1 - Success

Input Transaction:

```
<COMMAND_GET_LOCK_STATUS>
  <TransmissionTimestamp>2013-04-25T21:10:26Z</TransmissionTimestamp>
</COMMAND_GET_LOCK_STATUS>
```

Output Transaction:

```
<RESPONSE_GET_LOCK_STATUS>
  <COMMAND_GET_LOCK_STATUS>
    <TransmissionTimestamp>2013-04-25T21:10:26Z</TransmissionTimestamp>
  </COMMAND_GET_LOCK_STATUS>
  <ResultRecord>
    <Result>Success</Result>
    <ErrorCode>RC_OK</ErrorCode>
    <ResponseTimestamp>2013-04-25T21:10:27Z</ResponseTimestamp>
    <Description>Lock status successfully retrieved directly from lock.</Description>
  </ResultRecord>
  <LockStatus>
    <DuressAlarmStatus>Deactivated</DuressAlarmStatus>
    <SafeDoorStatus>SafeDoorSensorNotConnected</SafeDoorStatus>
    <LockBoltStatus>Closed</LockBoltStatus>
    <BankModeActive>SingleModeActive</BankModeActive>
    <RouteModeActive>DualModeActive</RouteModeActive>
    <FLMModeActive>DualModeActive</FLMModeActive>
    <FirmwareVersion>
      <MajorVersion>2</MajorVersion>
      <MinorVersion>1</MinorVersion>
      <BuildVersion>1061</BuildVersion>
    </FirmwareVersion>
    <SerialNumber>001000112</SerialNumber>
  </LockStatus>
</RESPONSE_GET_LOCK_STATUS>
```

Example 2 - Failure

Input Transaction:

```
<COMMAND_GET_LOCK_STATUS>
  <TransmissionTimestamp>2013-04-25T21:26:14Z</TransmissionTimestamp>
</COMMAND_GET_LOCK_STATUS>
```

Output Transaction:

```
<RESPONSE_GET_LOCK_STATUS>
  <COMMAND_GET_LOCK_STATUS>
    <TransmissionTimestamp>2013-04-25T21:26:14Z</TransmissionTimestamp>
  </COMMAND_GET_LOCK_STATUS>
  <ResultRecord>
    <Result>Failure</Result>
    <ErrorCode>RC_ERR_PCIO_NO_ACTIVE_DEVICE_CONNECTED</ErrorCode>
    <ResponseTimestamp>2013-04-25T21:26:17Z</ResponseTimestamp>
    <Description>Failed to retrieve lock status directly from lock.</Description>
    <Description>There is no active device for the Cencon USB Key Box connected to the system.
  </Description>
</ResultRecord>
```

```
</RESPONSE_GET_LOCK_STATUS>
```

© 1996-2017 Kaba Mas LLC

GenTran 5.4.0.4378 Agent XML Programming Reference Manual 28 Jun 2019

CenTran 5 Agent Programming Guide

Chapter 3 - Agent Configuration XML Transactions

3.1 Agent Configuration Update

This transaction is used to update the value read from one of the entries in the CenTran Agent configuration file. The changed value is also written out to the configuration file.

Transaction information required:

Each configuration entry being changed, and the new value for the configuration entry.

The supported configuration file entries are

- CentranAgent_PassThroughIPPort
- CentranAgent_CommandIPPort
- CentranAgent_ReportEventIPAddress
- CentranAgent_ReportEventIPPort
- CentranAgent_CentranIPAddress
- CentranAgent_CentranIPPort
- Centran_ServiceURI

Transaction information returned:

Success:

Input transaction, followed by a result record (result, error code and description) and the lock status record.

Failure:

There will not be a failed return of data unless an internal error or connectivity error with the CenTran Agent has occurred.

Note

If an unsupported configuration item is specified, it is ignored, and will not be returned in the copy of the input transaction.

Example 1 - Success

Input Transaction:

```
<COMMAND_UPDATE_AGENT_CONFIGURATION>
  <TransmissionTimestamp>2013-04-26T13:39:24Z</TransmissionTimestamp>
  <CentranAgent_ReportEventIPPort>35008</CentranAgent_ReportEventIPPort>
</COMMAND_UPDATE_AGENT_CONFIGURATION>
```

Output Transaction:

```
<RESPONSE_UPDATE_AGENT_CONFIGURATION>
  <COMMAND_UPDATE_AGENT_CONFIGURATION>
    <TransmissionTimestamp>2013-04-26T13:39:24Z</TransmissionTimestamp>
    <CentranAgent_ReportEventIPPort>35008</CentranAgent_ReportEventIPPort>
  </COMMAND_UPDATE_AGENT_CONFIGURATION>
  <ResultRecord>
    <Result>Success</Result>
    <ErrorCode>RC_OK</ErrorCode>
    <ResponseTimestamp>2013-04-26T13:39:24Z</ResponseTimestamp>
    <Description>Update configuration for currently running CenTran Agent.</Description>
  </ResultRecord>
</RESPONSE_UPDATE_AGENT_CONFIGURATION>
```

Example 2 - Failure

Input Transaction:

```
<COMMAND_UPDATE_AGENT_CONFIGURATION>
  <TransmissionTimestamp>2013-04-26T15:33:45Z</TransmissionTimestamp>
  <CentranAgent_Report>35008</CentranAgent_Report>
</COMMAND_UPDATE_AGENT_CONFIGURATION>
```

Output Transaction:

```
<RESPONSE_UPDATE_AGENT_CONFIGURATION>
  <COMMAND_UPDATE_AGENT_CONFIGURATION>
    <TransmissionTimestamp>2013-04-26T15:33:45Z</TransmissionTimestamp>
  </COMMAND_UPDATE_AGENT_CONFIGURATION>
  <ResultRecord>
    <Result>Success</Result>
    <ErrorCode>RC_OK</ErrorCode>
    <ResponseTimestamp>2013-04-26T15:33:45Z</ResponseTimestamp>
    <Description>Update configuration for currently running CenTran Agent.</Description>
  </ResultRecord>
</RESPONSE_UPDATE_AGENT_CONFIGURATION>
```


CenTran 5 Agent Programming Guide

Chapter 4 - Event Monitoring XML Transactions

4.1 ATM Lock Event Status Changed

This purely output transaction is used to report the occurrence and results of the evaluation of a Cencon lock event, and is transmitted using the UDP transport protocol to the IP address and report specified in the CenTran Agent configuration file.

Transaction information required:

- Not applicable.

Transaction information returned:

Success:

Input transaction, followed by a result record (result, error code and description) and the lock event status record.

Failure:

Input transaction, followed by a result record (result, error code and description).

The lock status record for the lock will also be displayed if the lock supports the direct retrieval of lock status by the CenTran Agent and this is not a retry.

Example 1 - Success

Output Transaction:

```
<EVENT_LOCK_STATUS_CHANGED>
  <ResultRecord>
    <Result>Success</Result>
    <ErrorCode>RC_OK</ErrorCode>
    <ResponseTimestamp>2013-04-26T16:41:22Z</ResponseTimestamp>
    <Description>Lock event evaluated from lock successfully.</Description>
  </ResultRecord>
  <LockEventStatus>
    <LockStatus>
      <DuressAlarmStatus>Deactivated</DuressAlarmStatus>
      <SafeDoorStatus>SafeDoorSensorNotConnected</SafeDoorStatus>
      <LockBoltStatus>Closed</LockBoltStatus>
      <BankModeActive>SingleModeActive</BankModeActive>
      <RouteModeActive>DualModeActive</RouteModeActive>
      <FLMModeActive>DualModeActive</FLMModeActive>
      <FirmwareVersion>
        <MajorVersion>2</MajorVersion>
        <MinorVersion>1</MinorVersion>
        <BuildVersion>1061</BuildVersion>
      </FirmwareVersion>
      <SerialNumber>001000112</SerialNumber>
    </LockStatus>
    <EventType>Unspecified</EventType>
  </LockEventStatus>
</EVENT_LOCK_STATUS_CHANGED>
```

Example 2 - Failure

Output Transaction:

```
<EVENT_LOCK_STATUS_CHANGED>
  <ResultRecord>
    <Result>Failure</Result>
    <ErrorCode>RC_FAIL</ErrorCode>
    <ResponseTimestamp>2013-04-26T16:35:53Z</ResponseTimestamp>
    <Description>Exception occurred with web service request: The remote server returned an error: (407) Proxy Authentication Required.</Description>
    <Description>Lock status successfully retrieved directly from lock.</Description>
  </ResultRecord>
  <LockEventStatus>
    <LockStatus>
      <DuressAlarmStatus>Deactivated</DuressAlarmStatus>
      <SafeDoorStatus>SafeDoorSensorNotConnected</SafeDoorStatus>
      <LockBoltStatus>Closed</LockBoltStatus>
      <BankModeActive>SingleModeActive</BankModeActive>
      <RouteModeActive>DualModeActive</RouteModeActive>
      <FLMModeActive>DualModeActive</FLMModeActive>
      <FirmwareVersion>
        <MajorVersion>2</MajorVersion>
        <MinorVersion>1</MinorVersion>
        <BuildVersion>1061</BuildVersion>
      </FirmwareVersion>
      <SerialNumber>001000112</SerialNumber>
    </LockStatus>
    <EventType>UnknownNoResponseFromCentran</EventType>
  </LockEventStatus>
</EVENT_LOCK_STATUS_CHANGED>
```

