Using the View Client Protocol

View Manager 4.5

VMware View Client Protocol License 1.0

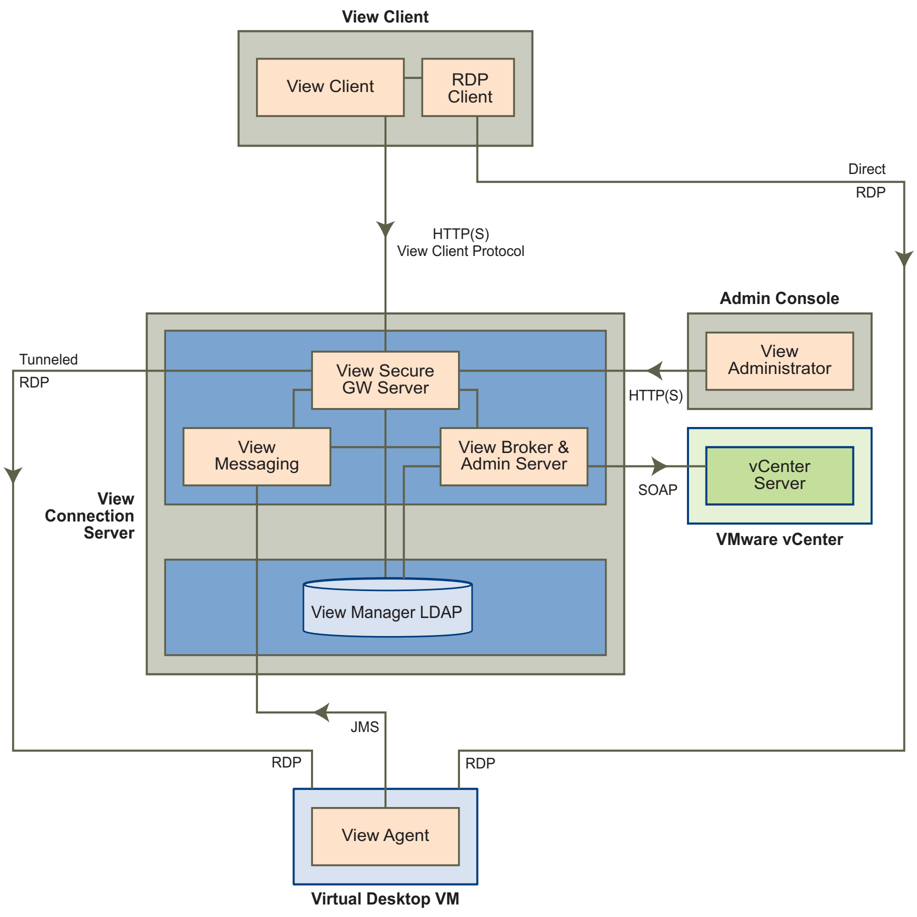
No offer is made herein by VMware to provide support or maintenance of any kind and VMware shall not be obligated to provide You with support or maintenance of any kind for the API or Your implementation thereof. The name and trademarks of VMware may not be used in any manner, including in any advertising, publicity or marketing relating to the API or in conjunction with Your implementation without the prior written permission of VMware.

The View Client Protocol

The VMware® View™ Client protocol is a set of XML instructions that client devices can use to communicate with a View Connection Server instance. This technical note describes the protocol that is supported by VMware Virtual Desktop Manager 2.x, VMware View Manager 3.x, and later versions of VMware View Manager. Third-party vendors and developers can use the protocol definitions to create software for a client device that interacts with a View Connection Server instance.

Figure 1 shows a complete View Manager component environment with connections from a View Client to a View Connection Server instance. The client sends XML requests to the View Connection Server instance over an HTTP or HTTPS connection. If the client can authenticate itself successfully using the View Client protocol, the View Connection Server instance permits the client to establish a direct (non-tunneled) connection to a virtual desktop using a protocol such as VMware PCoIP, Microsoft Remote Desktop Protocol (RDP), or HP Remote Graphics Software (RGS).

Figure 1. View Manager Components and Direct Client Connections



This document describes how to create a direct connection between a client and a View desktop virtual machine, which bypasses the View Connection Server or security server host. It does not describe how to create a tunnel connection between a client and a View desktop via a View Connection Server or security server host. Direct connections to desktops are suitable for use over LAN or VPN connections, and must be used for the PCoIP and HP RGS display protocols. To use direct client connections, select **Direct connection to desktop** in the settings for the View Connection Server instance.

To understand the message flows in the View Client protocol, install a complete View Manager environment, and disable SSL for client connections. You can then use network protocol analysis software (such as WireShark) to monitor the protocol messages that are exchanged between a client and the View Connection Server instance. To disable SSL for client connections, deselect **Require SSL for client connections and View Administrator SSL** in the global settings, and restart the View Connection Server service.

In an enterprise-level deployment, you would usually configure View Manager to use SSL for client connections over a secure HTTPS channel. If the View Connection Server or security server hosts are arranged in a load-balancing configuration, your client should use the same, persistent HTTPS connection for all requests.

Client-server communications take place using either HTTP or HTTPS where both requests and responses are properly constructed XML messages. All requests contain the root element `broker` with an accompanying `version` attribute.

You use `POST` requests to submit protocol data to the URL `/broker/xml` on a View Connection Server instance. A request can contain one or more messages. For example, the `get-desktops` request can contain just `get-desktops` or several additional requests, such as `set-locale` and `do-submit-authentication`. Responses contain a reply to each message that you sent in the request. The result for a successful request contains the value `ok`.

All versions of the View Connection Server instance require that you enable cookies. You must return cookies that the client receives from the View Connection Server instance in subsequent requests with standard cookie headers. A client should persist and expire cookies on request, according to the standard cookie mechanism.

Table 1 shows the cookies that a View Connection Server instance requires.

Table 1. Cookies Used on a Client with a View Connection Server Instance

Cookie	Description
JSESSIONID	Persists only during a session.
com.vmware.vdi.broker.location.id	Persists across sessions, and does not expire.

Sample Client Code

The VMware View Open Client allows you to connect from a Linux desktop to remote Windows desktops that are managed by View Manager. You can download the source files for the View Open Client to examine sample code that uses the View Client protocol to implement a client. The sample code includes an implementation of the tunneling connection feature, but you should note that this implementation might change in a future release. For more information, see http://code.google.com/p/vmware-view-open-client.

Using the View Client Protocol to Create a Desktop Session

Table 2 shows the generic procedure for using the View Client protocol to retrieve the information that you need to start a desktop session on a client system.

Table 2. Overview of How to Create a Desktop Session

Step	Request	Response
1.	Send a `get-configuration` request to the View Connection Server instance.	Returns the authentication mechanism that you must use.
2.	Send a `do-submit-authentication` request to authenticate the user on a View Connection Server instance and establish a session.	Shows whether the authentication succeeded.
3.	Send a `get-tunnel-connection` request for a direct connection.	Returns whether the View Connection Server instance accepts direct connections.

Table 2. Overview of How to Create a Desktop Session (Continued)

Step	Request	Response
4.	Send a `get-desktops` request to retrieve the list of desktops to which the user is entitled.	Contains the list of desktops, if successful.
5.	Send a `get-desktop-connection` request that contains the ID of one of the desktops that the previous request returned.	Returns the parameters that are required to start a desktop client for the requested desktop.

You can start a desktop client using the parameters that the response to the `get-desktop-connection` request contains.

After the user has finished using the desktop, send the `do-logout` request to logout. To terminate the session, close the client and disconnect any remaining desktop sessions.

View Client Protocol Versions

The View Client protocol is updated when a new version of View Manager is released. Table 3 lists the View Manager version, the corresponding View Client protocol version, and a brief description of major changes in each version of the View Client protocol.

Table 3. View Client Protocol Versions

View Manager Release	View Client Protocol Version	Description of Changes
2.0	1.0	First release.
2.1	2.0	New authentication screens (disclaimer and password expiration).
3.0	3.0	Offline Virtual Desktop Infrastructure (VDI) information added to the `get-desktops` response. Certificate authentication (smart card) properties added.
3.1	3.1	Protocol selection.
4.0	4.0	Support for PCoIP desktop protocol. Triple single sign-on (SSO) and other improvements to smart card authentication.
4.5	4.5	Updated support for local (offline) desktops.

The `version` attribute of the `broker` element indicates the protocol version that the client supports. After receiving the version information, a View Connection Server instance ensures that the response is backward-compatible. If a feature is required but is not supported in an earlier client, the server responds with an error.

For example, a version 1.0 client that requests a disclaimer receives the following error response.

```
<?xml version="1.0"?>
<broker version="2.0">
  <configuration>
    <result>error</result>
    <error-code>UNSUPPORTED_VERSION</error-code>
    <error-message>Disclaimers are not supported by this client version</error-message>
    <user-message>Your View Client version does not support the features of this View Server.
     Please upgrade your View Client</user-message>
  </configuration>
</broker>
```

NOTE A client must send the same version number with every request.

Protocol Compatibility

Use the following guidelines when determining protocol compatibility between a client and a View Connection Server instance.

- For backward compatibility, a client must report its protocol version correctly, it must read the version number supplied in the response from a View Connection Server instance, and it must ensure that it sends only messages that are implemented in that version or earlier versions of the protocol.
- For forward compatibility, a client can accept a protocol version number higher than its own if it ignores any XML elements that it does not understand, instead of returning an error. For example, a version 1.0 client that receives a version 2.1 response to the `get-desktops` request ignores the unsupported `reset-allowed` element in the message.

Error Handling

A View Connection Server instance responds with an error message under the following conditions:

- The format, version, or purpose of a request is incorrect or cannot be determined.
- An unauthenticated client makes a request that requires authentication.

The following example shows the generic format of an error response.

```
<?xml version="1.0"?>
<broker version="1.0">
  <result>error</result>
  <error-code>[code]</error-code>
  <error-message>[message]</error-message>
</broker>
```

Some error responses include an additional entry in the following format:

```
<user-message>[message]</user-message>
```

This information is localized in View Manager version 2.0 and later and might be displayed to users.

Most requests from a client require that the user is authenticated. A View Connection Server instance responds with a `NOT_AUTHENTICATED` error for a message that requires an authenticated session. If a View Connection Server instance is disabled, new clients can connect but not authenticate, and they receive a `BROKER_DISABLED` error response. However, client sessions that have already been authenticated can continue.

Table 4 provides an overview of the generic error codes that a View Connection Server instance can return. Certain types of requests can return other error codes that are specific to those requests.

Table 4. Generic Error Codes

Error Code	Description
`BROKER_DISABLED`	A View Connection Server instance has been disabled from within the administrative interface and cannot process requests.
`INVALID_ROOT_ELEMENT`	The `root` element of the request is something other than `broker`.
`INVALID_VERSION`	The version attribute of the `broker` element does not contain a float or an integer.
`INVALID_XML`	The XML is not well formed. For example: `Error on line 3: The element type "broker" must be terminated by the matching end-tag`
`MISSING_CONTENT`	The request contains the correct root element and version attribute, but the envelope contains no information. This error is also used to indicate that an individual message is missing child elements, for example, `set-locale` has a missing or empty `locale` element.
`MISSING_VERSION`	The `version` attribute is missing from the `broker` element.
`NOT_AUTHENTICATED`	The client must be authenticated by a View Connection Server instance before the request can be processed.

Table 4. Generic Error Codes (Continued)

Error Code	Description
`READ_ERROR`	A communication error occurred that prevented the View Connection Server instance from reading the XML.
`UNRECOGNIZED_CONTENT`	The request contained unrecognized content. For example, attempting to send the `set-locale` instruction to a version 1.0 View Connection Server instance results in the following error: `unrecognized content: set-locale`
`UNSUPPORTED_VERSION`	The request contains data that is not compatible with the View Connection Server version.

View Client Protocol Requests

The following sections describe the request messages that can be sent and the responses that are received when using the View Client protocol. It also provides the version in which the message was first implemented, and definitions and responses for each request.

The messages are listed in alphabetical order. The following list orders the request messages in which you might invoke them in an application.

- "set-locale Request" on page 36
- "get-configuration Request" on page 21
- "do-submit-authentication Request" on page 8
- "get-tunnel-connection Request" on page 32
- "get-desktops Request" on page 27
- "get-user-global-preferences Request" on page 33
- "set-user-global-preferences Request" on page 38
- "set-user-desktop-preferences Request" on page 37
- "get-desktop-connection Request" on page 24
- "kill-session Request" on page 34
- "reset-desktop Request" on page 35
- "do-logout Request" on page 7

do-logout Request

Ends a session with a View Connection Server instance.

Request

```
<?xml version="1.0"?>
<broker version="1.0">
  <do-logout/>
</broker>
```

Response

```
<?xml version="1.0"?>
<broker version="1.0">
  <logout>
    <result>ok</result>
  </logout>
</broker>
```

Description

The `do-logout` request message ends a View Connection Server session so that a user is no longer authenticated.

Implementation

Implemented in protocol version 1.0.

Authentication

Prior authentication is required.

Errors

Table 5 shows the error code and message for `do-logout`.

Table 5. do-logout Error Codes and Messages

Code	Description
NOT_AUTHENTICATED	The client must be authenticated by a View Connection Server instance before the request can be processed. This response was first implemented in protocol version 3.1.

do-submit-authentication Request

Performs authentication of the user with the View Connection Server instance.

`do-submit-authentication` supports the following requests and responses:

- "Request: Accept Disclaimer" on page 8
- "Response: Accept Disclaimer" on page 8
- "Request: SecurID Authentication Passcode" on page 9
- "Response: SecurID Authentication Passcode" on page 9
- "Request: SecurID Authentication Token Code" on page 12
- "Response: SecurID Authentication Token Code" on page 12
- "Request: Change SecurID Authentication PIN" on page 13
- "Response: Change SecurID Authentication PIN" on page 13
- "Request: Wait for SecurID Authentication" on page 14
- "Response: Wait for SecurID Authentication" on page 14
- "Request: Password Authentication" on page 14
- "Response: Password Authentication" on page 15
- "Request: Update Password" on page 16
- "Response: Update Password" on page 16
- "Request: Certificate Authentication" on page 17
- "Response: Certificate Authentication" on page 18
- "Response: Complete Authentication Failure" on page 19
- "Response: Already Authenticated Error" on page 19

Request: Accept Disclaimer

The request for accept disclaimer was first implemented in protocol version 2.0.

```
<?xml version="1.0"?>
<broker version="2.0">
  <do-submit-authentication>
    <screen>
      <name>disclaimer</name>
      <params>
        <param>
          <name>accept</name>
          <values>
            <value>true</value>
          </values>
        </param>
      </params>
    </screen>
  </do-submit-authentication>
</broker>
```

Response: Accept Disclaimer

```
<?xml version="1.0"?>
<broker version="2.0">
  <submit-authentication>
    <result>partial</result>
    <!-- OK so far, but authentication not yet complete -->
    <authentication>
      <screen>
```

```
        <name>securid-passcode</name>
      </screen>
    </authentication>
  </submit-authentication>
</broker>
```

Request: SecurID Authentication Passcode

The request for a SecurID authentication passcode was first implemented in protocol version 1.0.

```
<?xml version="1.0"?>
<broker version="1.0">
  <do-submit-authentication>
    <screen>
      <name>securid-passcode</name>
      <params>
        <param>
          <name>username</name>
          <values>
            <value>user1</value>
          </values>
        </param>
        <param>
          <name>passcode</name>
          <values>
            <value>271828183</value>
          </values>
        </param>
      </params>
    </screen>
  </do-submit-authentication>
</broker>
```

Response: SecurID Authentication Passcode

The following sections detail the possible responses to the `securid-passcode` request.

Authentication Partially Complete

This response is sent when authentication is partially complete and the next screen is being requested.

```
<?xml version="1.0"?>
<broker version="1.0">
  <submit-authentication>
    <result>partial</result>
    <!-- OK so far, but authentication not complete -->
    <authentication>
      <screen>
        <name>windows-password</name>
        <params>
          <param>
            <name>username</name>
            <!-- Sometimes policy requires that a value is fixed from one screen to another -->
            <readonly/>
           <values>
              <value>user1</value>
            </values>
          </param>
          <param>
            <name>domain</name>
            <values>
              <value>DOMAIN1</value>
            </values>
          </param>
        </params>
      </screen>
    </authentication>
  </submit-authentication>
</broker>
```

SecurID Token Code Required

This response is sent when the next SecureID token code is required to proceed.

```
<?xml version="1.0"?>
<broker version="1.0">
  <submit-authentication>
    <result>partial</result>
    <authentication>
      <screen>
        <name>securid-nexttokencode</name>
        <params/>
      </screen>
    </authentication>
  </submit-authentication>
</broker>
```

New PIN Required

This response is sent when the user must choose a new PIN.

```
<?xml version="1.0"?>
<broker version="1.0">
  <submit-authentication>
    <result>partial</result>
    <authentication>
      <screen>
        <name>securid-pinchange</name>
        <params>
          <!-- The user must enter a new PIN code, no PIN is pre-provided -->
          <param>
            <name>user-selectable</name>
            <values>
              <value>MUST_CHOOSE_PIN</value>
            </values></param>
          <param>
            <name>message</name>
            <values>
              <value>PIN should be between 4 and 8 numbers.</value>
            </values>
          </param>
        </params>
      </screen>
    </authentication>
  </submit-authentication>
</broker>
```

New PIN Cannot Be Chosen

This response is sent when the user must accept the assigned PIN and cannot choose a new PIN.

```
<?xml version="1.0"?>
<broker version="1.0">
  <submit-authentication>
    <result>partial</result>
    <authentication>
      <screen>
        <name>securid-pinchange</name>
        <params>
          <!--
           The user must confirm the provided PIN number, first box for PIN should be pre-filled
               and not editable
          -->
          <param>
            <name>user-selectable</name>
            <values><value>CANNOT_CHOOSE_PIN</value>
            </values>
          </param>
          <param>
            <name>pin1</name>
            <readonly/>
            <values><value>1234</value></values>
          </param>
        </params>
      </screen>
    </authentication>
  </submit-authentication>
</broker>
```

New PIN Can Be Chosen

This response is sent when the user can select a provided PIN or enter a new one.

```
<?xml version="1.0"?>
<broker version="1.0">
  <submit-authentication>
    <result>partial</result>
    <authentication>
      <screen>
        <name>securid-pinchange</name>
        <params>
          <param>
            <name>user-selectable</name>
            <values>
              <value>USER_SELECTABLE</value>
            </values>
          </param>
          <param>
            <name>message</name>
            <values><value>PIN should be between 4 and 8 numbers.</value>
            </values>
          </param>
          <param>
            <name>pin1</name>
            <readonly/>
            <values><value>1234</value></values>
          </param>
        </params>
      </screen>
    </authentication>
  </submit-authentication>
</broker>
```

SecurID Authentication Failure

This response is sent is sent in the event of a SecureID authentication failure.

```
<?xml version="1.0"?>
<broker version="1.0">
  <submit-authentication>
    <result>partial</result>
    <authentication>
      <screen>
        <name>securid-passcode</name>
        <params>
          <param>
            <name>error</name>
            <values>
              <value>Access Denied</value>
            </values>
          </param>
          <param>
            <name>username</name>
            <values>
              <value>user1</value>
            </values>
          </param>
        </params>
      </screen>
    </authentication>
  </submit-authentication>
</broker>
```

Request: SecurID Authentication Token Code

The following request was first implemented in protocol version 1.0.

```
<?xml version="1.0"?>
<broker version="1.0">
  <do-submit-authentication>
    <screen>
      <name>securid-nexttokencode</name>
      <params>
        <param>
          <name>tokencode</name>
          <values>
            <value>271828183</value>
          </values>
        </param>
      </params>
    </screen>
  </do-submit-authentication>
</broker>
```

Response: SecurID Authentication Token Code

The following sections detail the possible responses to the `securid-nexttokencode` request.

Authentication Partially Complete

This response is sent when authentication is partially complete and the next screen is being requested.

```
<?xml version="1.0"?>
<broker version="1.0">
  <submit-authentication>
    <result>partial</result> <!-- OK so far, but authentication not complete -->
    <authentication>
      <screen>
        <name>windows-password</name>
        <params>
          <param>
            <name>username</name>
            <!-- Sometimes policy requires that a value is fixed from one screen to another -->
            <readonly/>
            <values>
              <value>user1</value>
            </values>
          </param>
          <param>
            <name>domain</name>
            <values>
              <value>DOMAIN1</value>
            </values>
          </param>
        </params>
      </screen>
    </authentication>
  </submit-authentication>
</broker>
```

SecurID Authentication Failure

This response is sent in the event of a SecurID authentication failure.

```
<?xml version="1.0"?>
<broker version="1.0">
  <submit-authentication>
    <result>partial</result>
    <authentication>
      <screen>
        <name>securid-passcode</name>
        <params>
          <param>
            <name>error</name>
            <values>
              <value>Access Denied</value>
            </values>
          </param>
          <param>
            <name>username</name>
            <values>
              <value>user1</value>
            </values>
          </param>
        </params>
      </screen>
    </authentication>
  </submit-authentication>
</broker>
```

Request: Change SecurID Authentication PIN

The following request was first implemented in protocol version 1.0.

```
<?xml version="1.0"?>
<broker version="1.0">
  <do-submit-authentication>
    <screen>
      <name>securid-pinchange</name>
      <params>
        <param>
          <name>pin1</name>
          <values>
            <value>1111</value>
          </values>
        </param>
        <param>
          <name>pin2</name>
          <values>
            <value>1111</value>
          </values>
        </param>
      </params>
    </screen>
  </do-submit-authentication>
</broker>
```

Response: Change SecurID Authentication PIN

The response waits for the next token code.

```
<?xml version="1.0"?>
<broker version="1.0">
  <submit-authentication>
    <authentication>
      <screen>
        <name>securid-wait</name>
      </screen>
    </authentication>
  </submit-authentication>
</broker>
```

Request: Wait for SecurID Authentication

The following request was first implemented in protocol version 1.0.

```
<?xml version="1.0"?>
<broker version="1.0">
  <do-submit-authentication>
    <screen>
      <name>securid-wait</name>
      <!-- empty params avoid a NullPointerException - API 2.0 and earlier -->
      <params></params>
    </screen>
  </do-submit-authentication>
</broker>
```

Response: Wait for SecurID Authentication

This response is sent if a login attempt is permitted.

```
<?xml version="1.0"?>
<broker version="1.0">
  <submit-authentication>
    <result>partial</result>
    <authentication>
      <screen>
        <name>securid-passcode</name>
        <params>
          <param>
            <name>username</name>
            <values><value>user1</value></values>
          </param>
        </params>
      </screen>
    </authentication>
  </submit-authentication>
</broker>
```

Request: Password Authentication

The following request was first implemented in protocol version 1.0.

```
<?xml version="1.0"?>
<broker version="1.0">
  <do-submit-authentication>
    <screen>
      <name>windows-password</name>
      <params>
        <param>
          <name>username</name>
          <values>
            <value>user1</value>
          </values>
        </param>
        <param>
          <name>domain</name>
          <values>
            <value>DOMAIN1</value>
          </values>
        </param>
        <param>
          <name>password</name>
          <values>
            <value>secret</value>
          </values>
        </param>
      </params>
    </screen>
  </do-submit-authentication>
</broker>
```

Response: Password Authentication

The following sections detail the possible responses to the `windows-password` request.

Authentication Completed Successfully

This response is sent if authentication is successful and the authentication process is complete.

```
<?xml version="1.0"?>
<broker version="1.0">
  <submit-authentication>
    <result>ok</result>
    <!-- Since View 3.0 -->
    <user-sid>session ID for user</user-sid>
    <!-- Since View 3.1 -->
    <offline-sso-disabled>[true|false]</offline-sso-disabled>
    <!-- Since 4.5 -->
    <!-- If set to true, reauthentication is required when a host power event occurs -->
    <logout-on-host-suspend-enabled>[true|false]</logout-on-host-suspend-enabled>
    <!-- Authentication complete. -->
  </submit-authentication>
</broker>
```

Authentication Failed

This response is sent if password authentication failed, but another attempt is permitted.

```
<?xml version="1.0"?>
<broker version="1.0">
  <submit-authentication>
    <result>partial</result>
    <authentication>
      <screen>
        <name>windows-password</name>
        <params>
          <param>
            <name>error</name>
            <values><value>Unknown user name or bad password</value></values>
          </param>
          <param>
            <name>username</name>
            <!-- Sometimes policy requires that a value is fixed from one screen to another -->
            <readonly/>
            <values>
              <value>user1</value>
            </values>
          </param>
          <param>
            <name>domain</name>
            <values>
              <value>DOMAIN1</value>
            </values>
          </param>
        </params>
      </screen>
    </authentication>
  </submit-authentication>
</broker>
```

Password Expired

This response is sent if a password authentication request was received where the password has expired.

This response was first implemented in protocol version 2.0.

```
<?xml version="1.0"?>
<broker version="2.0">
  <submit-authentication>
    <result>partial</result>
    <authentication>
```

```
        <screen>
          <name>windows-password-expired</name>
          <params>
            <param>
              <name>username</name>
              <readonly/>
              <values><value>user1</value></values>
            </param>
          </params>
        </screen>
      </authentication>
    </submit-authentication>
</broker>
```

Request: Update Password

The old password has expired, and a new password must be set.

This request was first implemented in protocol version 2.0.

```
<?xml version="1.0"?>
<broker version="2.0">
  <do-submit-authentication>
    <screen>
      <name>windows-password-expired</name>
      <params>
        <param>
          <name>oldPassword</name>
          <values>
            <value>secret</value>
          </values>
        </param>
        <param>
          <name>newPassword1</name>
          <values>
            <value>secret1</value>
          </values>
        </param>
        <param>
          <name>newPassword2</name>
          <values>
            <value>secret1</value>
          </values>
        </param>
      </params>
    </screen>
  </do-submit-authentication>
</broker>
```

Response: Update Password

This response is sent if the new password is not accepted and the user can try again.

This response was first implemented in protocol version 2.0.

```
<?xml version="1.0"?>
<broker version="2.0">
  <submit-authentication>
    <result>partial</result>
    <authentication>
      <screen>
        <name>windows-password-expired</name>
        <params>
          <param>
            <name>error</name>
            <values>
              <value>New Password Not Accepted</value>
            </values>
          </param>
          <param>
            <name>username</name>
            <readonly/>
            <values>
              <value>user1</value>
            </values>
          </param>
        </params>
      </screen>
    </authentication>
  </submit-authentication>
</broker>
```

Request: Certificate Authentication

Certificate authentication is used when smart card authentication is enabled in a View Connection Server instance.

The certificate authentication feature was first implemented in protocol version 3.0.

Accept Certificate Authentication

The client accepts the user who is associated with the certificate that was presented during the initial SSL handshake.

The following form of this request was first implemented in protocol version 3.0.

```
<?xml version="1.0"?>
<broker version="3.0">
    <do-submit-authentication>
        <screen>
            <name>cert-auth</name>
            <params>
                <!-- accept authentication using certificate -->
                <param>
                    <name>accept</name>
                    <values>
                        <value>true</value>
                    </values>
                </param>
            </params>
        </screen>
    </do-submit-authentication>
</broker>
```

The following form of this request was first implemented in protocol version 4.0.

```
<?xml version="1.0"?>
<broker version="4.0">
    <do-submit-authentication>
        <screen>
            <name>cert-auth</name>
            <params>
                <!-- accept authentication using certificate -->
                <param>
                    <name>accept</name>
                    <values><value>true</value></values>
                </param>
                <!-- Since 4.0 -->
                <!-- smart card details for SSO -->
                <param>
                    <name>smartCardPIN</name>
                    <values><value>1234</value></values>
                </param>
                <param>
                    <name>smartCardReader</name>
                    <values><value>XXXX</value></values>
                </param>
            </params>
        </screen>
    </do-submit-authentication>
</broker>
```

Reject Certificate Authentication

The client rejects the user who is associated with the certificate that was presented during the initial SSL handshake.

```
<?xml version="1.0"?>
<broker version="3.0">
  <do-submit-authentication>
    <screen>
      <name>cert-auth</name>
      <params>
        <!-- Reject authentication using certificate -->
        <param>
          <name>accept</name>
          <values>
            <value>false</value>
          </values>
        </param>
      </params>
    </screen>
  </do-submit-authentication>
</broker>
```

Response: Certificate Authentication

The following sections detail the possible responses to the `cert-auth` request.

Authentication Successful and Complete

Certificate authorization was accepted and client authentication is complete.

```
<?xml version="1.0"?>
<broker version="4.0">
    <submit-authentication>
        <!-- Authentication complete -->
        <result>ok</result>

        <!-- Since 3.0 -->
        <user-sid>S-1-5-21-12354123-123-4123</user-sid>

        <!-- Since 3.1 -->
        <offline-sso-disabled>false</offline-sso-disabled>

        <!-- Since 4.5 -->
        <!-- If set to true, reauthentication is required when a host power event occurs -->
        <logout-on-host-suspend-enabled>true</logout-on-host-suspend-enabled>

        <!-- Since 4.0 -->
        <!-- Optional, only included for certificate authentication -->
        <logout-on-cert-removal-enabled>false</logout-on-cert-removal-enabled>

    </submit-authentication>
</broker>
```

Certificate Authentication Rejected, Fall Back on Password

This response is sent if the certificate authentication failed. If this form of authentication is set as optional within a View Connection Server instance, the host falls back on password authentication.

```
<?xml version="1.0"?>
<broker version="3.0">
  <submit-authentication>
    <result>partial</result>
    <!-- OK so far, but authentication not complete -->
    <authentication>
```

```
      <screen>
        <name>windows-password</name>
        <params>
          <param>
            <name>domain</name>
            <values>
              <value>DOMAIN1</value>
            </values>
          </param>
        </params>
      </screen>
    </authentication>
  </submit-authentication>
</broker>
```

Certificate Authentication Rejected, Certificate Authentication Required by Server

This response is sent if the client rejected the certificate authentication. If this form of authentication is set as required within a View Connection Server instance, the authorization fails.

```
<?xml version="1.0"?>
<broker version="3.0">
  <submit-authentication>
    <result>error</result>
    <error-code>AUTHENTICATION_FAILED</error-code>
    <error-message>Authentication failure</error-message>
    <user-message>Smart Card or Certificate authentication is required.</user-message>
  </submit-authentication>
</broker>
```

Response: Complete Authentication Failure

If authentication fails on three consecutive attempts, an error is returned, and the authentication process must be restarted from the beginning.

This response was first implemented in protocol version 1.0.

```
<?xml version="1.0"?>
<broker version="3.0">
  <submit-authentication>
    <result>error</result>
    <error-code>AUTHENTICATION_FAILED</error-code>
    <error-message>Authentication failed</error-message>
    <user-message>Maximum login attempts exceeded</user-message>
  </submit-authentication>
</broker>
```

Response: Already Authenticated Error

An error is returned if authentication is attempted against a session that was previously authenticated. This error does not prevent the session from continuing.

This response was first implemented in protocol version 2.0.

```
<?xml version="1.0"?>
<broker version="2.0">
  <submit-authentication>
    <result>error</result>
    <error-code>ALREADY_AUTHENTICATED</error-code>
    <error-message>already authenticated</error-message>
    <user-message/>
  </submit-authentication>
</broker>
```

Description

The `do-submit-authentication` request messages submit user credentials and authenticate the user with the View Connection Server instance. The responses indicates which authentication request is required next.

Implementation

Implemented in protocol version 1.0.

Authentication

Prior authentication is not required.

Errors

Table 6 shows the error codes and messages for `do-submit-authentication`.

Table 6. do-submit-authentication Error Codes and Messages

Code	Description
`ALREADY_AUTHENTICATED`	The client session has already been authenticated, and can continue.
`AUTHENTICATION_FAILED`	Authentication failed on three consecutive attempts. The authentication process must be restarted from the beginning.
`BROKER_DISABLED`	The View Connection Server instance has been disabled from within the administrative interface and cannot process requests.
`UNSUPPORTED_VERSION`	The request contains data that is not compatible with the View Connection Server version.

get-configuration Request

Discovers the authentication method.

Request

```
<?xml version="1.0"?>
<broker version="1.0">
  <get-configuration/>
</broker>
```

Response

The following sections detail the possible responses to the `get-configuration` request.

Windows Password Authentication Required

The following response is sent when Windows password authentication is required. The response was first implemented in protocol version 1.0. The response was updated in protocol version 3.0 to introduce the `broker-guid` element for the GUID of the View Connection Server group, and in protocol version 4.0 to support Kerberos authentication.

```
<?xml version="1.0"?>
<broker version="4.0">
  <configuration>
    <result>ok</result>

    <!-- Since 4.0: optional -->
    <!-- No public interfaces are provided for this element -->
    <broker-service-principal>
        <type>kerberos</type>
        <name>machine/mybroker@mydomain.int</name>
    <broker-service-principal>

    <!-- Since 3.0 -->
    <broker-guid>6e79ced1-b474-4ce6-a48c-c40c17c935c0</broker-guid>

    <authentication>
      <screen>
        <name>windows-password</name>
        <params>
          <param>
            <name>domain</name>
            <values>
              <value>DOMAIN1</value>
            </values>
          </param>
        </params>
      </screen>
    </authentication>
  </configuration>
</broker>
```

SecurID Authentication Required

The following response is sent when SecurID authentication is required. The response was first implemented in protocol version 1.0.

```
<?xml version="1.0"?>
<broker version="1.0">
  <configuration>
    <result>ok</result>
    <authentication>
      <screen>
        <name>securid-passcode</name>
      </screen>
    </authentication>
  </configuration>
</broker>
```

Disclaimer Required

The following response is sent when a disclaimer is required. The response was first implemented in protocol version 2.0.

```
<?xml version="1.0"?>
<broker version="2.0">
  <configuration>
    <result>ok</result>
    <authentication>
      <screen>
        <name>disclaimer</name>
        <params>
          <param>
            <name>text</name>
            <values>
              <value>Disclaimer message</value>
            </values>
          </param>
        </params>
      </screen>
    </authentication>
  </configuration>
</broker>
```

Disclaimer Required for Identified Server

The following response introduces the `broker-guid` element to the message when a disclaimer is required. The response was first implemented in protocol version 3.0.

```
<?xml version="1.0"?>
<broker version="2.0">
  <configuration>
    <result>ok</result>
    <broker-guid>6e79ced1-b474-4ce6-a48c-c40c17c935c0<broker-guid>
      <authentication>
        <screen>
          <name>disclaimer</name>
          <params>
            <param>
              <name>text</name>
              <values>
                <value>Disclaimer message</value>
              </values>
            </param>
          </params>
        </screen>
      </authentication>
  </configuration>
</broker>
```

Confirmation Required for Certificate Authentication

When a client certificate is used to authenticate against a View Connection Server instance as part of the SSL handshake, the authentication chain requests confirmation that certificate authentication is to be used. Certificate authentication is used when smart card authentication is enabled in View Connection Server. The response was first implemented in protocol version 3.0.

```
<?xml version="1.0"?>
<broker version="3.0">
  <configuration>
    <result>ok</result>
    <broker-guid>6e79ced1-b474-4ce6-a48c-c40c17c935c0</broker-guid>
    <authentication>
```

```
      <screen>
        <name>cert-auth</name>
        <params>
          <param>
            <!-- name of user account that would be authenticated -->
            <name>user</name>
            <values><value>DOMAIN\user</value></values>
          </param>
        </params>
      </screen>
    </authentication>
  </configuration>
</broker>
```

Description

The `get-configuration` request message discovers which mechanism is used for the first stage of authentication, and determines which authentication screen is displayed first.

This message is used before entering the authentication flow. When used within the authentication process, it returns the next expected screen or an `ALREADY_AUTHENTICATED` error after authentication is complete.

Starting with protocol version 3.0, you can use the `broker-guid` element in the response to identify the GUID of the View Connection Server group.

Implementation

Implemented in protocol version 1.0.

Authentication

Prior authentication is not required.

Errors

Table 7 shows the error codes and messages for `get-configuration`.

Table 7. get-configuration Error Codes and Messages

Code	Description
`ALREADY_AUTHENTICATED`	The client session has already been authenticated and can continue.
`BROKER_DISABLED`	The View Connection Server instance has been disabled from within the administrative interface and cannot process requests.
`UNSUPPORTED_VERSION`	The request contains data that is not compatible with the View Connection Server version.

get-desktop-connection Request

Sets up and retrieves the parameters for a desktop connection.

Request

In protocol versions 1.0 and 2.0, the connection protocol is assumed to be RDP.

```
<?xml version="1.0"?>
<broker version="1.0">
  <get-desktop-connection>
    <!-- the id of a desktop from the get-desktops response -->
    <desktop-id>CN=Desktop,OU=Applications,DC=vdi,DC=vmware,DC=int</desktop-id>
  </get-desktop-connection>
</broker>
```

Starting with protocol version 3.1, the connection protocol can be specified, and local system information can be passed from a View Client to a View Agent.

```
<?xml version="1.0"?>
<broker version="4.0">
  <get-desktop-connection>
    <!-- The id of a desktop from the get-desktops response -->
    <desktop-id>CN=Desktop,OU=Applications,DC=vdi,DC=vmware,DC=int</desktop-id>
    <!--+
        |  A protocol provided in the desktop list.
        +-->
    <protocol>
      <name>RDP</name>
    </protocol>

    <!--+
        |  Local system information from View Client to be passed to View Agent.
        |  The structure shown below provides examples of parameters that could
        |  be used by View Agent to inform scripts or processes residing on the
        |  desktop.
        +-->
    <environment-information>
      <info name="IP_Address">[...]</info>
      <info name="MAC_Address">[...]</info>
      <info name="Machine_Name">[...]</info>
      <info name="Machine_Domain">[...]</info>
      <info name="LoggedOn_Username">[...]</info>
      <info name="LoggedOn_Domainname">[...]</info>
      <info name="Type">[...]</info>
    <!--+
        | Time offset is from GMT, and the format is [sign]HH:MM.
        | For example, -07:00 is 7 hours behind GMT. If the Disable Time Zone
        | Synchronization GPO is true, timezone parameters are ignored.
        +-->
      <info name="TimeOffset_GMT">-07:00</info>
    <!--+
        | Since 4.0. Olson timezone ID, as used in various Linux distros.
        | If defined, overrides TimeOffset_GMT.
        +-->
      <info name="TZID">Europe/London</info>
    <!--+
        | Since 4.0. Windows timezone ID.
        | If defined, overrides TZID and TimeOffset_GMT.
        +-->
      <info name="Windows_Timezone">GMT Standard Time</info>/info>

    </environment-information>

  </get-desktop-connection>
</broker>
```

Response

```
<?xml version="1.0"?>
<broker version="3.0">
    <desktop-connection>
        <result>ok</result>
        <id>CN=Desktop,OU=Applications,DC=vdi,DC=vmware,DC=int</id>
        <address>localhost</address>
        <port>23456</port>

        <!-- Since 2.0 -->
        <!-- List of additional listeners on the guest, e.g. MMR ports -->
        <additional-listeners>
            <additional-listener name="MMR">127.0.0.1:9427</additional-listener>
        </additional-listeners>

        <protocol>RDP</protocol>
        <user-name>user1</user-name>
        <password>MWEyNDVmODgt</password>
        <domain-name>DOMAIN1</domain-name>

        <!-- Indicates if USB forwarding should be enabled for clients supporting it -->
        <enable-usb>true</enable-usb>

        <!-- Since 3.0 -->
        <!-- Indicates if multimedia-redirection should be enabled for clients supporting it -->
        <enable-mmr>true</enable-mmr>

        <!-- Since 3.1 -->
        <!-- Used for protocol specific negotiation, such as PCOIP token -->
        <protocol-settings>
            <token>token</token>
        </protocol-settings>
    </desktop-connection>
</broker>
```

Description

The `get-desktop-connection` request message sets up and retrieves the parameters for a desktop connection. You can use the returned parameters to launch an appropriate client for the connection protocol. For example, you could launch an RDP client when `<protocol>RDP</protocol>` is returned.

If the client does not specify a specific protocol in the request, the View Connection Server instance chooses the protocol.

Implementation

Implemented in protocol version 1.0.

Authentication

Prior authentication is required.

Errors

Table 8 shows the error codes and messages for `get-desktop-connection`.

Table 8. get-desktop-connection Error Codes and Messages

Code	Description
`DESKTOP_LAUNCH_ERROR`	Desktop launch is not possible because no virtual machines are available, or an exception occurred while preparing the desktop connection.
`DESKTOP_MAINTENACE_ERROR`	Desktop launch is not possible because all available virtual machines are in maintenance mode. (Implemented in protocol version 3.0.)

Table 8. get-desktop-connection Error Codes and Messages

Code	Description
MISSING_CONTENT	The request contains the correct root element and version attribute, but the envelope contains no information. This error is also used to indicate that an individual message is missing child elements.
NOT_AUTHENTICATED	The client must be authenticated by a View Connection Server instance before the request can be processed.
NOT_ENTITLED	The user is not entitled to use the desktop.

get-desktops Request

Returns a list of desktops to which an authenticated user is entitled.

`get-desktops` supports the following requests and responses:

- "Request: Get List of Desktops" on page 27
- "Request: Get List of Desktops for a Display Protocol" on page 27
- "Request: Authenticate User and Get List Of Desktops" on page 27
- "Response" on page 28

Request: Get List of Desktops

The following request asks for the list of desktops.

```
<?xml version="1.0"?>
<broker version="1.0">
  <get-desktops/>
</broker>
```

Request: Get List of Desktops for a Display Protocol

The following request specifies the display protocols that the client supports.

This form of the request was first implemented in protocol version 3.1.

```
<?xml version="1.0"?>
<broker version="3.1">
  <get-desktops>
    <!-- Since 3.1 -->
    <supported-protocols>
      <protocol>
        <name>RDP</name>
      </protocol>
      <protocol>
        <name>RGS</name>
      </protocol>
    </supported-protocols>
  </get-desktops>
</broker>
```

Request: Authenticate User and Get List Of Desktops

The following request combines the `get-desktops` request message with the `do-submit-authentication` request message.

```
<?xml version="1.0"?>
<broker version="1.0">
  <do-submit-authentication>
    <screen>
      <name>windows-password</name>
      <params>
        <param>
          <name>username</name>
          <values>
            <value>user1</value>
          </values></param>
        <param>
          <name>domain</name>
          <values>
            <value>DOMAIN1</value>
          </values>
        </param>
        <param>
          <name>password</name>
          <values>
            <value>secret</value>
      </values>
    </param>
  </params>
</screen>
</do-submit-authentication>
  <get-desktops/>
</broker>
```

Response

The following response is sent by protocol version 1.0.

```
<?xml version="1.0"?>
<broker version="1.0">
  <submit-authentication>
    <result>ok</result>
  </submit-authentication>
  <desktops>
    <result>ok</result>
    <desktop>
      <id>CN=desktop2,OU=Applications,DC=vdi,DC=vmware,DC=int</id>
      <name>desktop2</name>
      <!--
      Desktop Type - Since 1.0 values are:
      free
      sticky
      auto
      -->
      <type>free</type>
      <!-- user has no session on this desktop -->
      <state/>
      <session-id/>
      <user-preferences>
        <preference name="autoConnect">false</preference>
        <preference name="screenSize">Windowed</preference>
      </user-preferences>
    </desktop>
    <desktop>
      <id>CN=pool1,OU=Applications,DC=vdi,DC=vmware,DC=int</id>
      <name>pool1</name>
      <!-- desktop type -->
      <type>auto</type>
      <!-- user has a disconnected session on this desktop -->
      <state>disconnected</state>
      <session-id>DOMAIN1\user1(cn=...,cn=foreignsecurityprincipals,dc=...)/0@cn=...,ou=
       servers,dc=...:RDP:3389</session-id>
      <user-preferences>
        <preference name="autoConnect">false</preference>
        <preference name="screenSize">Windowed</preference>
      </user-preferences>
    </desktop>
  </desktops>
</broker>
```

The following response contains information that is sent by protocol version 2.0 and later. The version that is associated with each element or element block is indicated by an inline comment.

```
<?xml version="1.0"?>
<broker version="3.1">
    <desktops>
        <result>ok</result>
        <desktop>
            <id>CN=desktop2,OU=Applications,DC=vdi,DC=vmware,DC=int</id>
            <name>desktop2</name>

            <!--
                Desktop Type
                 Since 1.0 values are:
                  free
```

```
      sticky
      auto
     Since 3.0 values are:
      free
      sticky
      auto
      sticky-lc
      auto-lc
      free-unmanaged
      free-unmanaged-wts
      single-free
      single-free-unmanaged
      sticky-free
      sticky-free-unmanaged
-->
<type>free</type>

<user-preferences>
  <preference name="autoConnect">false</preference>
  <preference name="screenSize">Windowed</preference>
</user-preferences>

<!-- User has a disconnected session on this desktop - these entries are empty if no
   session is present. -->
<state>disconnected</state>
<session-id>
   DOMAIN1\user1(cn=...,cn=foreignsecurityprincipals,dc=...)/0@cn=...,ou=servers,
   dc=...:RDP:3389 </session-id>

<!-- Since 2.0 -->
<!--+
    | Indicates whether the user is allowed to perform a reset on this desktop's
    | virtual machine.
    | reset-allowed values are:
    | true
    | false
    +-->
<reset-allowed>true</reset-allowed>

<!--+
    |  Indicates whether the user is allowed to perform a reset after the user has
    |  established a session on the desktop.
    |  reset-allowed-on-session values are:
    |  true
    |  false
    +-->
<reset-allowed-on-session>true</reset-allowed-on-session>

<!-- Since 3.0 -->
<!--+
    |  offline-enabled values are:
    |  true
    |  false
    +-->
<offline-enabled>true</offline-enabled>

<!--+
    |  offline-state values are:
    |  checked in
    |  checked out
    |  checking in
    |  checking out
    |  [blank]
    +-->
<offline-state>checked in</offline-state>

<offline-host>[hostname]</offline-host>
<offline-checkoutTime>[time]</offline-checkoutTime>
```

```
            <offline-challenge>[GUID]</offline-challenge>
            <checkout-guid>52 64 06 88 a3 8d e7 ef-f7 50 bb 83 7a ef 7a 8d</checkout-guid>

            <!--+
                |  Maintenance mode flag (true indicates all machines for desktop are in
                |  maintenance)
                +-->
            <in-maintenance-mode>false</in-maintenance-mode>

            <!-- Since 3.1 - selectable protocols -->
            <!--+
                |  protocol-match is true if there are one or more protocols available that the
                |  client can use
                +-->
            <protocol-match>true</protocol-match>

            <protocols>
                <protocol>
                    <name>RDP</name>
                    <is-default>true</is-default>
                </protocol>
                <protocol>
                    <name>RGS</name>
                </protocol>
            </protocols>

            <!-- Since 4.5 - local (offline) desktop policies -->
            <!-- These are only included for desktops that have been checked out. -->
            <policies>
                <info name="offlineUpdateFrequency">5</info>
                <info name="offlineAllowed">true</info>
                <info name="destroyFileCheckedIn">false</info>
                <info name="disksReplicated">user</info>
                <info name="userDeferrableReplication">false</info>
                <info name="allowCopyPaste">true</info>
                <info name="endpointAllowed">true</info>
                <info name="cacheLifetime">10080</info>
                <info name="targetReplicationFrequency">86400</info>
                <info name="clientRollback">true</info>
                <info name="destroyFileNoSession">false</info>
                <info name="allowMMR">true</info>
                <info name="allowUSB">true</info>
                <info name="offlineCheckoutState">checked out</info>
                <info name="USBFiltering">-------------------</info>
                <!--+
                    |  requestedReplication values are:
                    |  policy-deferrable-request
                    |  non-deferrable-request
                    +-->>
                <info name="requestedReplication">policy-deferrable-request</info>
</policies>

        </desktop>
        <current-time-in-seconds>time on broker</current-time-in-seconds>
     </desktops>
</broker>
```

Description

The `get-desktops` request message retrieves a list of desktops to which an authenticated user is entitled. Starting with protocol version 3.1, the required protocol can be specified in the message. The message can be combined with a `do-submit-authentication` request.

Each entitled desktop has a `user-preferences` block that contains the user preferences previously supplied by a `set-user-desktop-preferences` request.

Since protocol version 3.1, each desktop that is listed in the response also contains a list of supported protocols the client can request, along with a default setting.

Desktop names are sorted numerically (0-9), then uppercase alphabetically (A-Z), and finally lowercase alphabetically (a-z).

From protocol version 4.5, `get-desktops` returns an empty list for users who have no desktop entitlements, but who are in roles with administrator privileges.

Implementation

Implemented in protocol version 1.0.

Authentication

Authentication is required. The message can be combined with a `do-submit-authentication` request.

Errors

Table 9 shows the error codes and messages for `get-desktops`.

Table 9. get-desktops Error Codes and Messages

Code	Description
`NOT_AUTHENTICATED`	The client must be authenticated by a View Connection Server instance before the request can be processed.
`DESKTOPS_ERROR`	An error occurred while retrieving the list of desktops.

get-tunnel-connection Request

Requests a direct connection rather than a tunnel session.

Request

The request for a direct connection was first implemented in protocol version 2.0.

```
<?xml version="1.0"?>
<broker version="2.0">
    <get-tunnel-connection>
        <!-- Since 2.0 - Request the session does not use tunnelling, desktop connections will be
               direct -->
        <bypass-tunnel>true</bypass-tunnel>
    </get-tunnel-connection>
</broker>
```

Response

```
<?xml version="1.0"?>
<broker version="1.0">
    <tunnel-connection>
        <result>ok</result>
        <bypass-tunnel>true</bypass-tunnel>
    </tunnel-connection>
</broker>
```

Description

You can use the `get-tunnel-connection` request message to request a direct connection between a client and a desktop if a tunnel is not required.

NOTE This document does not describe how to create a tunnel connection between a client and a desktop.

Implementation

Implemented in protocol version 1.0.

Authentication

Authentication is required. The message may be combined with a `do-submit-authentication` request.

Errors

Table 10 shows the error codes and messages for `get-tunnel-connection`.

Table 10. get-tunnel-connection Error Codes and Messages

Code	Description
NOT_AUTHENTICATED	The client must be authenticated by a View Connection Server instance before the request can be processed.

get-user-global-preferences Request

Returns a list of user-specific preferences.

Request

```
<?xml version="1.0"?>
<broker version="1.0">
  <get-user-global-preferences/>
</broker>
```

Response

```
<?xml version="1.0"?>
<broker version="1.0">
  <user-global-preferences>
    <result>ok</result>
    <user-preferences>
      <preference name="doautolaunch">true</preference>
    </user-preferences>
  </user-global-preferences>
</broker>
```

Description

The `get-user-global-preferences` request message returns a list of preferences that are specific to a user but not to a desktop.

Implementation

Implemented in protocol version 1.0.

Authentication

Prior authentication is required.

Errors

Table 11 shows the error codes and messages for `get-user-global-preferences`.

Table 11. get-user-global-preferences Error Codes and Messages

Code	Description
`NOT_AUTHENTICATED`	The client must be authenticated by a View Connection Server instance before the request can be processed.
`PREFERENCES_ERROR`	An error occurred while retrieving the list of preferences.

kill-session Request

Terminates a desktop session.

Request

```
<?xml version="1.0"?>
<broker version="1.0">
  <kill-session>
    <!-- uses a session-id from the get-desktops response -->
    <session-id>
     DOMAIN1\user1(cn=...,cn=foreignsecurityprincipals,dc=...)/
     0@cn=...,ou=servers,dc=...:RDP:3389</session-id>
  </kill-session>
</broker>
```

Response

```
<?xml version="1.0"?>
<broker version="1.0">
  <kill-session>
    <result>ok</result>
  </kill-session>
</broker>
```

Description

The `kill-session` request message terminates a desktop session. The user continues to have a View Connection Server session, which can be terminated by using the `do-logout` request.

Implementation

Implemented in protocol version 1.0.

Authentication

Prior authentication is required.

Errors

Table 12 shows the error codes and messages for `kill-session`.

Table 12. kill-session Error Codes and Messages

Code	Description
`KILL_SESSION_ERROR`	An error occurred while terminating the desktop session.
`MISSING_CONTENT`	The request contains the correct root element and version attribute, but the envelope contains no information. This error is also used to indicate that an individual message is missing child elements.
`NOT_AUTHENTICATED`	The client must be authenticated by a View Connection Server instance before the request can be processed.

reset-desktop Request

Requests a reset of a desktop.

Request

```
<?xml version="1.0"?>
<broker version="2.0">
  <reset-desktop>
    <!-- the id of a desktop from the get-desktops response -->
    <desktop-id>CN=Desktop,OU=Applications,DC=vdi,DC=vmware,DC=int</desktop-id>
  </reset-desktop>
</broker>
```

Response

This response is sent when the desktop can be reset.

```
<?xml version="1.0"?>
<broker version="2.0">
  <reset-desktop>
    <result>ok</result>
  </reset-desktop>
</broker>
```

Description

The `reset-desktop` request message requests the View Connection Server instance to perform a reset on the desktop of the specified virtual machine.

Implementation

Implemented in protocol version 2.0.

Authentication

Prior authentication is required.

Errors

Table 13 shows the error code and message for `reset-desktop`.

Table 13. reset-desktop Error Codes and Messages

Code	Description
`NOT_AUTHENTICATED`	The client must be authenticated by a View Connection Server instance before the request can be processed. This response was first implemented in protocol version 3.1.
`NOT_ENTITLED`	The user is not entitled or no longer allowed to access the desktop.
`RESET_DESKTOP_ERROR`	An error occurred while resetting the desktop.

set-locale Request

Specifies the locale used for messages.

Request

```
<?xml version="1.0"?>
<broker version="2.0">
  <set-locale>
    <locale>en_GB</locale>
  </set-locale>
</broker>
```

Response

```
<?xml version="1.0"?>
<broker version="2.0">
  <set-locale>
    <result>ok</result>
  </set-locale>
</broker>
```

Description

The `set-locale` request message specifies the locale. The locale determines the language that is used for `user-message` in error messages and for any user messages that are returned by the response to a `do-submit-authentication` request.

Implementation

Implemented in protocol version 2.0.

Authentication

Prior authentication is not required.

Errors

Table 14 shows the error codes and messages for `set-locale`.

Table 14. set-locale Error Codes and Messages

Code	Description
`BROKER_DISABLED`	A View Connection Server instance has been disabled from within the administrative interface and is not able to process requests.
`MISSING_CONTENT`	The request contains the correct root element and version attribute, but the envelope contains no information. This error is also used to indicate that an individual message is missing child elements.

set-user-desktop-preferences Request

Sets the preferences for a desktop.

Request

```
<?xml version="1.0"?>
<broker version="1.0">
  <set-user-desktop-preferences>
    <desktop-id>CN=Desktop,OU=Applications,DC=vdi,DC=vmware,DC=int</desktop-id>
    <user-preferences>
      <preference name="alwaysConnect">true</preference>
      <preference name="screenSize">Windowed</preference>
    </user-preferences>
  </set-user-desktop-preferences>
</broker>
```

Response

```
<?xml version="1.0"?>
<broker version="1.0">
  <set-user-desktop-preferences>
    <result>ok</result>
  </set-user-desktop-preferences>
</broker>
```

Description

The `set-user-desktop-preferences` request message sets the preferences for a specific desktop. A complete set of preferences must be specified. Preferences that are not specified are deleted. Preference names must not contain an equal sign (=).

NOTE A View Connection Server instance stores preference values on behalf of a client. It does not use the values.

Implementation

Implemented in protocol version 1.0.

Authentication

Prior authentication is required.

Errors

Table 15 shows the error codes and messages for `set-user-desktop-preferences`.

Table 15. set-user-desktop-preferences Error Codes and Messages

Code	Description
`MISSING_CONTENT`	The request contains the correct root element and version attribute, but the envelope contains no information. This error is also used to indicate that an individual message is missing child elements.
`NOT_AUTHENTICATED`	The client must be authenticated by a View Connection Server instance before the request can be processed.
`NOT_ENTITLED`	The user is not entitled to use the desktop.
`PREFERENCES_ERROR`	An error occurred while retrieving the list of preferences.

set-user-global-preferences Request

Sets the preferences for a user.

Request

```
<?xml version="1.0"?>
<broker version="1.0">
  <set-user-global-preferences>
    <user-preferences>
      <preference name="doautolaunch">true</preference>
    </user-preferences>
  </set-user-global-preferences>
</broker>
```

Response

```
<?xml version="1.0"?>
<broker version="1.0">
  <set-user-global-preferences>
    <result>ok</result>
  </set-user-global-preferences>
</broker>
```

Description

The `set-user-global-preferences` request message sets the preferences for a user. A complete set of preferences must be specified. Preferences that are not specified are deleted. Preference names must not contain an equal sign (=).

NOTE A View Connection Server instance stores preference values on behalf of a client. It does not use the values.

Implementation

Iplemented in protocol version 1.0.

Authentication

Prior authentication is required.

Errors

Table 16 shows the error codes and messages for `set-user-global-preferences`.

Table 16. set-user-global-preferences Error Codes and Messages

Code	Description
`NOT_AUTHENTICATED`	The client must be authenticated by a View Connection Server instance before the request can be processed.
`PREFERENCES_ERROR`	An error occurred while retrieving the list of preferences.