

## [MS-RDPERP]: Remote Desktop Protocol: Remote Programs Virtual Channel Extension

This topic lists the Errata found in [MS-RDPERP] since it was last published. Since this topic is updated frequently, we recommend that you subscribe to these RSS or Atom feeds to receive update notifications.



Errata are subject to the same terms as the Open Specifications documentation referenced.

Errata below are for Protocol Document Version [V22.0 - 2016/07/14](#).

Errata Published*	Description
2016/09/26	<p>In Section 1.5, Prerequisites/Preconditions, clarified that the CHANNEL_FLAG_SHOW_PROTOCOL (0x00000010) has to be set in the flags field of the Channel PDU Header in order for it to be visible for all data that flows over the RAIL virtual channel.</p> <p>Changed from:</p> <p>The Remote Programs Extensions for Remote Desktop Protocol has the assumption to operate in a fully operational RDP connection. A fully operational RDP connection is a connection that has passed the Connection Finalization phase, as specified in [MS-RDPBCGR] section 1.3.1.1.</p> <p>Changed to:</p> <p>The Remote Programs Extensions for Remote Desktop Protocol has the assumption to operate in a fully operational RDP connection. A fully operational RDP connection is a connection that has passed the Connection Finalization phase, as specified in [MS-RDPBCGR] section 1.3.1.1.</p> <p>The RAIL server endpoint expects that the Channel PDU Header ([MS-RDPBCGR] section 2.2.6.1.1) is visible for all data that flows over the RAIL virtual channel (sections 1.3.1 and 2.2.2). To ensure that this condition is met, the CHANNEL_FLAG_SHOW_PROTOCOL (0x00000010) flag has to be set in the flags field of the Channel PDU Header.</p> <p>In Section 2.2.2, Static Virtual Channel Protocol, clarified that the RAIL Static Virtual Channel is named "RAIL".</p> <p>Changed from:</p> <p>The RAIL Static Virtual Channel is responsible...</p> <p>Changed to:</p> <p>The RAIL Static Virtual Channel (named "RAIL") is responsible...</p>
2016/09/26	<p>In Section 4.3.2, TS_RAIL_ORDER_EXEC_RESULT, changed RAIL_ORDER_EXEC_RESULT to TS_RAIL_ORDER_EXEC_RESULT.</p> <p>Changed from:</p> <p>The following is a network capture of the Server Execute Result PDU</p>

Errata Published*	Description
	<p>(RAIL_ORDER_EXEC_RESULT, as specified in 2.2.2.3.2).</p> <p>...</p> <p>80 00 -&gt; TS_RAIL_PDU_HEADER::orderType = RAIL_ORDER_EXEC_RESULT(128) (2 Bytes)</p> <p>...</p> <p>Changed to:</p> <p>The following is a network capture of the Server Execute Result PDU</p> <p>(TS_RAIL_ORDER_EXEC_RESULT, as specified in 2.2.2.3.2).</p> <p>...</p> <p>80 00 -&gt; TS_RAIL_PDU_HEADER::orderType = TS_RAIL_ORDER_EXEC_RESULT(128) (2 Bytes)</p> <p>...</p>
2016/09/26	<p>In Section 3.2.5.1.6, Processing Window Information Orders, clarified how the client will process the server window events of registration, deregistration, and updates on the edge of an application desktop toolbar.</p> <p>Changed from:</p> <p>Window Information Orders (section 2.2.1.3.1) inform the client of the following types of window events on the server:</p> <p>...</p> <ul style="list-style-type: none"> <li>• Registration of a new or existing window as an application desktop toolbar.</li> <li>• Deregistration of an existing application desktop toolbar.</li> </ul> <p>...</p> <p>Upon receipt of a Window Information Order for a deleted window, as specified in section 2.2.1.3.1.2.4, the client SHOULD locate the local RAIL window that corresponds to the WindowId reported in the Hdr field and destroy it. If no such window can be found, the client SHOULD ignore the order.</p> <p>Changed to:</p> <p>Window Information Orders (section 2.2.1.3.1) inform the client of the following types of window events on the server:</p> <p>...</p> <ul style="list-style-type: none"> <li>• Registration of a new or existing window as an application desktop toolbar.</li> <li>• Deregistration of an existing application desktop toolbar.</li> <li>• Updates on the edge to which the application desktop toolbar window is anchored.</li> </ul> <p>...</p> <p>Upon receipt of a Window Information Order for a deleted window, as specified in section 2.2.1.3.1.2.4, the client SHOULD locate the local RAIL window that corresponds to the WindowId reported in the Hdr field and destroy it. If no such window can be found, the client SHOULD ignore the order.</p> <p>Upon receipt of a Window Information Order for the registration of a window as an application desktop toolbar, the client SHOULD locate the local RAIL window that corresponds to the</p>

Errata Published*	Description
	<p>WindowId reported in the Hdr field and register it as an application desktop toolbar. If no such window can be found, the client SHOULD ignore the order.</p> <p>Upon receipt of a Window Information Order for the deregistration of an application desktop toolbar window, the client SHOULD locate the local RAIL window that corresponds to the WindowId reported in the Hdr field and deregister the application desktop toolbar window. If no such window can be found, the client SHOULD ignore the order.</p> <p>Upon receipt of a Window Information Order for the edge of an application desktop toolbar window, the client SHOULD locate the local RAIL window that corresponds to the WindowId reported in the Hdr field and update the edge to which the window is anchored. If no such window can be found, the client SHOULD ignore the order.</p>
2016/09/26	<p>Added two Window Information Orders to match the client behavior in Section 3.3.5.1.6, Constructing Window Information Orders, and removed two Notification Icon Orders to match the client behavior in Section 3.3.5.1.7, Constructing Notification Icon Orders.</p> <p>In Section 3.3.5.1.6, Constructing Window Information Orders, changed from: The server generates Window Information Orders to inform the client of the following types of window events on the server:</p> <ul style="list-style-type: none"> <li>• Creation of a new window.</li> <li>• Updates on window properties for a new or existing window.</li> <li>• Updates on icons for a new or existing window.</li> <li>• Deletion of an existing window.</li> </ul> <p>...</p> <p>Changed to: The server generates Window Information Orders to inform the client of the following types of window events on the server:</p> <ul style="list-style-type: none"> <li>• Creation of a new window.</li> <li>• Updates on window properties for a new or existing window.</li> <li>• Updates on icons for a new or existing window.</li> <li>• Deletion of an existing window.</li> <li>• Registration of a new or existing window as an application desktop toolbar.</li> <li>• Deregistration of an existing application desktop toolbar.</li> </ul> <p>...</p> <p>In Section 3.3.5.1.7, Constructing Notification Icon Orders, changed from: The server generates Notification Icon Information Orders to inform the client of the following types of notification icon events on the server.</p> <ul style="list-style-type: none"> <li>• Creation of a new notification icon.</li> <li>• Updates on properties for a new or existing notification icon.</li> <li>• Deletion of an existing notification icon.</li> <li>• Registration of a new or existing window as an application desktop toolbar.</li> <li>• Deregistration of an existing application desktop toolbar.</li> </ul>

<b>Errata Published*</b>	<b>Description</b>
	<p>...</p> <p>Changed to:</p> <p>The server generates Notification Icon Information Orders to inform the client of the following types of notification icon events on the server.</p> <ul style="list-style-type: none"><li>• Creation of a new notification icon.</li><li>• Updates on properties for a new or existing notification icon.</li><li>• Deletion of an existing notification icon.</li></ul> <p>...</p>

\*Date format: YYYY/MM/DD