PRIME Automation P Commands User Guide

Version 5.3

January 2026

SINCE DAY ONE.

Chyron PRIME Automation P Commands User Guide • 5.3 • January 2026 • This document is distributed by Chyron in online (electronic) form only, and is not available for purchase in printed form.

Table of Contents

Intelligent Interface P Commands for Graphics Output Channels Version 5.3

This document describes the default P commands that may affect output channels. User customizations made to the rule engine for an Intelligent Interface connection may alter the handling or appearance of these commands, however the descriptions of parameters and their behaviors should be unaffected.

Syntax

The standard P command syntax introduced for PRIME takes the form:

P\<COMMAND>:<Buffer>\<Channel>:<Layer>\<SceneName>\<NameValuePairs>\\<CR><LF>

The <COMMAND> parameter generally describes the type of operation that will occur when the command is handled. This is typically a human readable statement as opposed to the older Intelligent Interface command structure. For example, P\PLAY instead of V\5\3.

Not all of the default P commands utilize this exact syntax because some commands do not rely on all of the parameters described. Nonetheless, this is a general template for the majority of the P commands described on the pages that follow.

Parameters

<Buffer> is an optional parameter that may be used to specify whether Preview, Program or both buffers should be affected by the command. If this parameter is not provided, then the preceding colon is also excluded.

<Buffer> Value	Buffers Affected
0	Preview
Preview	Preview
1	Program
Program	Program
*****	Preview, Program
<Excluded>	Preview, Program

If the <Buffer> parameter is not provided, then the preceding colon is also excluded.

Includes <Buffer>	Expected Syntax
Yes	P\<COMMAND>**:<Buffer>**\<Channel>:<Layer>\...\\
No	P\<COMMAND>\<Channel>:<Layer>\...\\

<Channel> is a parameter which may be used to specify the numeric index of the channel that should be affected by the command. This argument is 1-based, meaning that the first channel has index 1, the second channel has index 2, and so on.

A specific channel number is required by all command behaviors that use the <Channel> parameter currently, except for the PlayAll behavior which supports a wildcard (*). Using the wildcard with the PlayAll command behavior will result in the command affecting all configured output channels. For further details, see page Play All.

<Layer> is an optional text parameter that may be used to specify which layers should be affected by the command. In the context of most P commands the layer parameter allows the sender to limit scenes that are already loaded or playing. For example, a command might opt to clear any scene that is playing on a designated background layer. Using the layer parameter allows the sender to avoid sending a clear command for only a specific scene.

<Layer> Expression	Potentially Affected
>3	Scenes on layer 4 or higher
>=3	Scenes on layer 3 or higher
<=3	Scenes on layer 3 or lower
<3	Scenes on layer 2 or lower
4	Scenes on layer 4
1-3	Scenes within the inclusive layer range 1 to 3
1,3,4	Scenes on layer 1, 3, or 4
*****	Scenes on any layer

If the <Layer> parameter is not provided, then the preceding colon is also excluded.

Includes <Layer>	Expected Syntax
Yes	P\<COMMAND>\<Channel>**:<Layer>**\...\\
No	P\<COMMAND>\<Channel>\...\\

<SceneName> is a text parameter used to indicate which scene should be affected by the command. Typically, this will contain only the name of the desired scene, however a full file path may also be provided. In accordance with standard Intelligent Interface syntax, a file name would only be valid if modified as follows:

<SceneName> Value	Example Scene Affected
1000	1000
I/Prime/Project 1/Images/1000.pbx	I:\Prime\Project 1\Images\1000.pbx
I:/Prime/Project 1/Images/1000.pbx *(Note that the colon is optional)*	I:\Prime\Project 1\Images\1000.pbx
*****	Currently only the Play and PlayAll behaviors support a wildcard for this parameter. When an asterisk is provided in place of a specific value, then these behaviors will play any scene currently loaded on the channel(s) affected by the command.

<NameValuePairs> can be used to reference a list of optional data values. Data parsed in this format is assumed to be in the form <Name>\<Value>; more specifically, each name and value pair is separated by a backslash.

For example: P\UPDATE\1\1000\Text 1\Example 1\Text 2\Example 2\\

In the above example, <NameValuePairs> would capture "Text 1\Example 1\Text 2\Example 2" and recognize these values:

Name	Value
Text 1	Example 1
Text 2	Example 2

Specific details on the P\UPDATE command are available on page .

<Values> can be used to reference a list of optional data values that do not inherently include a name/value pairing. This is used when capturing a list of names. For example:

P\PLAY_ALL\1\1000\1002\1003\\

In the above command, <Values> would capture "1000\1002\1003" and recognize these values:

1000, 1002 and 1003.

Specific details on the P\PLAY_ALL command are available on page .

Default Commands

- **Load Scene**: P\LOAD, page .
- **Play Scene**: P\PLAY, page .
- **Play Layer**: P\PLAY_LAYER, page .
- **Play Action**: P\PLAY_ACTION, page .
- **Play All**: P\PLAY_ALL, page .
- **Transfer Scene**: P\TRANSFER, page .
- **Transfer Layer**: P\TRANSFER_LAYER, page .
- **Update Scene**: P\UPDATE, page .
- **Clear Scene**: P\CLEAR, page .
- **Clear All**: P\CLEAR_ALL, page
- **Execute Command**: P\COMMAND, page .
- **Set Scene Parameter**: P\SCENE_PARAMETER, page .
- **Set Project Parameter**: P\PROJECT_PARAMETER, page .
- **Change Project**: P\CHANGE_PROJECT, page .
- **Set Project**: P\SET_PROJECT, page .
- **Command Sequence Execute**: P\SEQUENCE, page .

General notes:

Most of these commands support a combination of <Buffer>, <Channel>, <Layer> and <SceneName> parameters. It's important to recognize how these parameters allow a command to limit the list of affected scenes.

Consider this command:

P\UPDATE:Program\1:<5*\Name1\Value1\Name2\Value2\\

Split into parameters:

<Buffer>	Program
<Channel>	1
<Layer>	<5
<SceneName>	*
<NameValuePairs>	Name1\Value1\Name2\Value2

The combination of parameters allows a command to drill down and target a very specific subset of available scenes. In the example above, the UPDATE command is configured to affect any scene because the wildcard (*) is used in place of a specific scene name. However, not all scenes will actually be affected on account of the other parameters. The <Buffer> parameter limits the potential scene list to only those scenes that are playing (e.g. not loaded in Preview or closed). The <Channel> parameter further limits the list to only those scenes on channel 1 and lastly, the <Layer> expression trims the affected scenes to only those on layer 4 or lower.

In total, these parameters result in an UPDATE only to scenes playing on channel 1 in layers 4 or lower.

As fewer parameters are specified, the command becomes more generalized:

P\UPDATE\1\1002\Name1\Value1\Name2\Value2\\

<Buffer>	*<Excluded>*
<Channel>	1
<Layer>	*<Excluded>*
<SceneName>	1002
<NameValuePairs>	Name1\Value1\Name2\Value2

In the above command, the UPDATE is configured to affect scene 1002 regardless of whether the scene is loaded or playing; if the scene is both loaded to Preview and playing on Program, then both instances would receive the update. The only requirements are that the scene is named 1002 and that it is currently on channel 1.

Load Scene

P\LOAD\<Channel>:<Layer>\<SceneName>:<SceneStyle>\<NameValuePairs>\\

or

P\LOAD\\<SceneName>:<SceneStyle>\<NameValuePairs>\\

This command will load a specific scene to Preview on a specific channel. If the scene is already playing, then a second copy of the scene will appear in Preview once this command has executed successfully.

<Channel> must be a number.

<Layer> is optional, but must be a number if provided. If specified, this value will override the layer set within the scene.

<SceneName> must either be a scene name or absolute path to a scene file. In the latter case, the path must use forward slashes instead of back-slashes.

<NameValuePairs> is an optional back-slash separated list of parameter names and values.

Command	Effect
P\LOAD\1\1000	Loads scene 1000 on channel 1. The scene will be loaded on whatever layer is configured within the scene.
P\LOAD\1\1000\Text 1\Sample Text	Loads scene 1000 on channel 1. The scene will be loaded on whatever layer is configured within the scene. Once loaded, an object named "Text 1" will be updated with value "Sample Text"
P\LOAD\1:5\1000	Loads scene 1000 on channel 1, but overrides the scene so that it is on layer 5
P\LOAD\1:5\1000\Text 1\Sample Text	Loads scene 1000 on channel 1, but overrides the scene so that it is on layer 5. Once loaded, an object named "Text 1" will be updated with value "Sample Text"

Play Scene

P\PLAY\<Channel>:<Layer>\<SceneName>\<NameValuePairs>\\

This command will play a specific scene to Program on a specific channel. If the scene is already loaded on Preview, then that instance will be played to Program. Otherwise the scene will be played directly from disk.

<Channel> must be a number.

<Layer> is optional, but must be a number if provided. If specified, this value will override the layer set within the scene.

<SceneName> may either be a scene name or absolute path to a scene file. In the latter case, the path must use forward slashes instead of back-slashes. Alternatively, a wildcard (*) may be used to play all scenes that currently loaded (restricted further based upon the <Channel> and <Layer> parameters).

<NameValuePairs> is an optional back-slash separated list of parameter names and values.

Command	Effect
P\PLAY\1\1000	Plays scene 1000 on channel 1. The scene will be played on whatever layer was configured within the scene.
P\PLAY\1\1000\Text 1\Sample Text	Plays scene 1000 on channel 1. The scene will be played on whatever layer was configured within the scene. Once playing, an object named "Text 1" will be updated with value "Sample Text"
P\PLAY\1:5\1000	Plays scene 1000 on channel 1, but overrides the scene so that it is on layer 5
P\PLAY\1:5\1000\Text 1\Sample Text	Plays scene 1000 on channel 1, but overrides the scene so that it is on layer 5. Once loaded, an object named "Text 1" will be updated with value "Sample Text"
P\PLAY\1* *Note: The <SceneName> wildcard is not supported by all commands.*	Plays all scenes currently loaded on channel 1 in Preview.

Play Layer

P\PLAY_LAYER\<Channel>:<Layer>\\

This command will play all scenes loaded on the specified channel/layer.

<Channel> must be a number.

<Layer> must be specified using the layer expression support defined on page .

Command	Effect
P\PLAY_LAYER\1:5	Plays any scene loaded to layer 5 on channel 1.
P\PLAY_LAYER\1:<5	Plays any scene loaded to layer 4 or lower on channel 1.

Play Action

P\PLAY_ACTION:<Buffer>\<Channel>:<Layer>\<SceneName>\<Values>\\

This command will play one or more actions in a specific scene. If an appropriate scene is not found then the command will have no effect.

<Buffer> is optional, but must be included if the command should only affect Preview or Program separately. See page for more details.

<Channel> must be a number.

<Layer> is optional. If used, should be specified using the layer expression support defined on page .

<SceneName> may either be a scene name or absolute path to a scene file. In the latter case, the path must use forward slashes instead of back-slashes. Alternatively, a wildcard (*) may be used to affect all scenes that currently loaded or playing (restricted further based upon the <Buffer>, <Channel> and <Layer> parameters).

<Values> is a back-slash separated list of action names that should be played.

Command	Effect
P\PLAY_ACTION\1\1000\Action 1	Plays "Action 1" if it exists in scene 1000 on channel 1. This will affect either Preview or Program because a target buffer has not been specified.
P\PLAY_ACTION\1\1000\Action 1\Action 2	Plays both "Action 1" and "Action 2" if they exist in scene 1000 on channel 1. This will affect either Preview or Program because a target buffer has not been specified.
P\PLAY_ACTION\1*\Action 1 *Note: The <SceneName> wildcard is not supported by all commands.*	Plays "Action 1" in any scene that has such an action on channel 1. This will affect either Preview or Program because a target buffer has not been specified.
P\PLAY_ACTION:Program\1\1000\Action 1	Plays "Action 1" in scene 1000, but only if the scene is playing in Program on channel 1.

Play All

P\PLAY_ALL\<Channel>:<Layer>\<Values>\\

This command will play one or more scenes. If all of the specified scenes are already loaded, then the scenes will play in a simultaneous batch (per channel).

<Channel> must either be a number or a wildcard (*).

<Layer> is optional. If used, should be specified using the layer expression support defined on page .

<Values> is an optional back-slash separated list of scene names that should be played.

Command	Effect
P\PLAY_ALL\1	Plays all scenes currently loaded in Preview on channel 1.
P\PLAY_ALL\1:<5	Plays scenes currently loaded in Preview on channel 1, but only those on layer 4 or lower.
P\PLAY_ALL\1\1000\1002\1003	Plays scene 1000, 1002 and 1003 on channel 1. If all three scenes are already loaded, then they will be played simultaneously to air.
P\PLAY_ALL*	Plays all scenes currently loaded on all channels.
P\PLAY_ALL*\1000	Plays scene 1000 on all channels.
P\PLAY_ALL*\1000\1002\1003	Plays scene 1000, 1002, and 1003 on all channels. If all three scenes are already loaded on a single channel, then the play operation will be simultaneous on that channel.

Transfer Scene

P\TRANSFER\<Channel>:<Layer>\<SceneName>\\

This command will transfer a specific scene from Program on a specific channel back to Preview (e.g. take a playing scene and restore it to the loaded state). If the scene is not playing, it will have no effect.

<Channel> must either be a number.

<Layer> is optional, but should be specified using the layer expression support defined on page .

<SceneName> may either be a scene name or absolute path to a scene file. In the latter case, the path must use forward slashes instead of back-slashes. Alternatively, a wildcard (*) may be used to transfer all scenes that currently playing (restricted further based upon the <Channel> and <Layer> parameters).

Command	Effect
P\TRANSFER\1\1000	Transfers scene 1000 from Program on channel 1 back to Preview, regardless of which layer it is playing on.
P\TRANSFER\1:5\1000	Transfers scene 1000 from Program on channel 1 back to Preview, but only if it is playing on layer 5.
P\TRANSFER\1* *Note: The <SceneName> wildcard is not supported by all commands.*	Transfers all scenes currently playing on channel 1 back to Preview.

Transfer Layer

P\TRANSFER_LAYER\<Channel>:<Layer>\\

This command will transfer all scenes playing on the specified channel/layer.

<Channel> must be a number.

<Layer> must be specified using the layer expression support defined on page .

Command	Effect
P\TRANSFER_LAYER\1:5	Transfers any scene playing on layer 5 of channel 1.
P\TRANSFER_LAYER\1:<5	Transfers any scene playing on layer 4 or lower of channel 1.

Update Scene

P\UPDATE:<Buffer>\<Channel>:<Layer>\<SceneName>\<NameValuePairs>\\

This command will update objects within one or more scenes on a specific channel.

<Buffer> is optional, but must be included if the command should only affect Preview or Program separately. See page for more details.

<Channel> must be a number.

<Layer> is optional. If used, should be specified using the layer expression support defined on page .

<SceneName> may either be a scene name or absolute path to a scene file. In the latter case, the path must use forward slashes instead of back-slashes. Alternatively, a wildcard (*) may be used to play all scenes that currently loaded (restricted further based upon the <Buffer>, <Channel> and <Layer> parameters).

<NameValuePairs> is a back-slash separated list of parameter names and values.

Command	Effect
P\UPDATE\1\1000\Text 1\Sample Text	Updates scene 1000 on channel 1 with the attached data regardless of whether it is loaded in Preview or playing on Program. If the scene is both loaded and playing, then both instances will receive the updates. If scene 1000 is not open on either Preview or Program, then this command will have no effect.
P\UPDATE:Preview\1000\Text 1\Sample Text	Updates scene 1000 on channel 1 with the attached data, but only if the scene is loaded to Preview. If a second copy of the scene is also playing to Program, that instance will be unaffected.
P\UPDATE:Program\1000\Text 1\Sample Text	Updates scene 1000 on channel 1 with the attached data, but only if the scene is playing to Program. If a second copy of the scene is also loaded in Preview, that instance will be unaffected.
P\UPDATE\1*\Text 1\Sample Text *Note: The <SceneName> wildcard is not supported by all commands.*	Updates all scenes on channel 1 with the attached data, affecting Preview and Program.

Clear Scene

P\CLEAR:<Buffer>\<Channel>:<Layer>\<SceneName>\\

This command will clear one or more scenes from a specific channel.

<Buffer> is optional, but must be included if the command should only affect Preview or Program separately. See page for more details.

<Channel> must be a number.

<Layer> is optional. If used, should be specified using the layer expression support defined on page .

<SceneName> may either be a scene name or absolute path to a scene file. In the latter case, the path must use forward slashes instead of back-slashes. Alternatively, a wildcard (*) may be used to transfer all scenes that currently playing (restricted further based upon the <Buffer>, <Channel> and <Layer> parameters).

Command	Effect
P\CLEAR\1\1000	Clears scene 1000 from Preview and Program on channel 1, regardless of which layer it is playing on.
P\CLEAR\1:5\1000	Clears scene 1000 from Preview and Program on channel 1 back to Preview, but only if it is currently on layer 5.
P\CLEAR\1* *Note: The <SceneName> wildcard is not supported by all commands.*	Clears all scenes currently loaded or playing on channel 1.
P\CLEAR:Preview\1* *Note: The <SceneName> wildcard is not supported by all commands.*	Clears all scenes currently loaded on channel 1.
P\CLEAR:Program\1* *Note: The <SceneName> wildcard is not supported by all commands.*	Clears all scenes currently playing on channel 1.

Clear All

P\CLEAR_ALL:<Buffer>\<Channel>:<Layer>\<Values>\\

This command will clear one or more scenes from preview, program or both on a given channel.

<Buffer> is optional, but must be included if the command should only affect Preview or Program separately. See page for more details.

<Channel> must either be a number or a wildcard (*).

<Layer> is optional. If used, should be specified using the layer expression support defined on page .

<Values> is an optional back-slash separated list of scene names that should be played.

Command	Effect
P\CLEAR_ALL\1	Clears all scenes in Preview or Program on channel 1.
P\CLEAR_ALL\1:<5	Clears scenes in Preview or Program on channel 1, but only those on layer 4 or lower.
P\CLEAR_ALL\1\1000\1002\1003	Clears scene 1000, 1002 and 1003 on channel 1 regardless of whether they are on Preview or Program.
P\CLEAR_ALL*	Clears all scenes currently on all channels.
P\CLEAR_ALL*\1000	Clears scene 1000 on all channels.
P\CLEAR_ALL*\1000\1002\1003	Clears scene 1000, 1002, and 1003 on all channels.

Execute Command

P\COMMAND:<Buffer>\<Channel>:<Layer>\<SceneName>\<Values>\\

This will execute one or more object commands in a specific scene. If an appropriate scene is not found then the command will have no effect.

<Buffer> is optional, but must be included if the command should only affect Preview or Program separately. See page for more details.

<Channel> must be a number.

<Layer> is optional. If used, should be specified using the layer expression support defined on page .

<SceneName> may either be a scene name or absolute path to a scene file. In the latter case, the path must use forward slashes instead of back-slashes. Alternatively, a wildcard (*) may be used to transfer all scenes that currently playing (restricted further based upon the <Buffer>, <Channel> and <Layer> parameters).

<Values> is a back-slash separated list of object commands that should be executed. Each value in the collection must take the form ObjectName.CommandName. For example, Data1.MoveNext or Craw1.Restart.

Command	Effect
P\COMMAND\1\1000\Data1.MoveNext	Executes the "MoveNext" command on an object named "Data 1" if it exists in scene 1000 on channel 1. This will affect either Preview or Program because a target buffer has not been specified.
P\COMMAND\1\1000\Data1.Home\Data1.MoveNext	Executes both "Home" and "MoveNext" if an object named "Data 1" exists in scene 1000 on channel 1. This will affect either Preview or Program because a target buffer has not been specified.
P\COMMAND:Program\1*\Data1.Home *Note: The <SceneName> wildcard is not supported by all commands.*	Executes "Home" on "Data 1" in any playing scene that has such an object on channel 1.

Set Scene Parameter

P\SCENE_PARAMETER:<Buffer>\<Channel>:<Layer>\<SceneName>\<NameValuePairs>\\

This command will set one or more parameter values in a scene that is currently loaded or playing. The parameters are only modified for the lifetime of that particular scene instance; the scene file on disk is not modified, so once a loaded or playing scene is cleared, the parameter changes are lost. Note that this command is functionally equivalent to the P\UPDATE command; it exists only for symmetry with the P\PROJECT_PARAMETER command.

<Buffer> is optional, but must be included if the command should only affect Preview or Program separately. See page for more details.

<Channel> must be a number.

<Layer> is optional. If used, should be specified using the layer expression support defined on page .

<SceneName> may either be a scene name or absolute path to a scene file. In the latter case, the path must use forward slashes instead of back-slashes. Alternatively, a wildcard (*) may be used to transfer all scenes that currently playing (restricted further based upon the <Buffer>, <Channel> and <Layer> parameters).

<NameValuePairs> is a back-slash separated list of parameter names and values.

Command	Effect
P\SCENE_PARAMETER\1\1000\Parameter 1\Sample Text	Updates scene 1000 on channel 1 with the attached parameter data regardless of whether it is loaded in Preview or playing on Program. If the scene is both loaded and playing, then both instances will receive the updates. If scene 1000 is not open on either Preview or Program, then this command will have no effect.
P\SCENE_PARAMETER:Preview\1000\Parameter 1\Sample Text	Updates scene 1000 on channel 1 with the attached parameter data, but only if the scene is loaded to Preview. If a second copy of the scene is also playing to Program, that instance will be unaffected.

Set Project Parameter

P\PROJECT_PARAMETER\<NameValuePairs>\\

This command will set one or more parameter values in the currently selected project within the user interface. These changes are automatically saved with the project as they are applied.

Command	Effect
P\PROJECT_PARAMETER\Parameter 1\Sample Text\Parameter 2\More Text	Updates the current project with two parameters, Parameter 1 and Parameter 2. Named parameters that do not currently exist will be created.

Change Project

P\CHANGE_PROJECT\<ProjectPath>\\

This command will change the current project within the PRIME application. The user interface will update to reflect this change and any commands that are not project aware will default to using the designated project.

<ProjectPath> must either be a fully-qualified project directory or the name of a project already within the project list. In an absolute path is provided, then the path must use forward slashes instead of back-slashes.

Command	Effect
P\CHANGE_PROJECT\Samples	Changes the current project to one named "Samples" provided that a project with that name has previously been opened.
P\CHANGE_PROJECT\I:/Prime/Projects/Samples	Changes the current project to the "Samples" project within I:\Prime\Projects\Samples.

Set Project

P\SET_PROJECT\<Channel>\<ProjectName>\\

This command assigns a specific project to a channel. The user interface and current project displayed will not be changed. Intelligent Interface commands that affect that channel will behave as though the designated project is active. This change persists and will be re-applied automatically whenever the PRIME application is restarted.

<Channel> must be a number.

<ProjectName> must by the name of a project. A full project path is not supported. If unspecified, the channel will revert back to observing the current project selected within the user interface.

For example:

- P\SET_PROJECT\1\Project A\\
- P\SET_PROJECT\2\Project B\\
- P\LOAD\1\1000\\

The above command will attempt to load scene 1000 from Project A.

- P\LOAD\2\1000\\

The above command will instead attempt to load scene 1000 from Project B.

If a project has not been assigned to a channel, then the command will default to the currently selected project. This behavior can be restored by transmitting a SET_PROJECT command without specifying a project name (e.g. P\SET_PROJECT\1\\)

Application Parameters

P\APPLICATION_PARAMETER\Parameter 1\Sample Text\Parameter 2\More Text\\

Command Sequence Execute

P\SEQUENCE\<SceneName>\\
P\SEQUENCE\<Channel>\<SceneName>\\

This will execute the current item of a command sequence in a specific scene. The execution of the command sequence may move the sequence to next item, but that's entirely dependent on the behavior of the current item. For example, if the current item evaluates a condition then it'll be the responsibility of the condition to move the command sequence cursor.

<Channel> must be a number.

<SceneName> may either be a scene name or absolute path to a scene file. In the latter case, the path must use forward slashes instead of back-slashes. Alternatively, a wildcard (*) may be used to transfer all scenes that currently playing (restricted further based upon the <Buffer>, <Channel> and <Layer> parameters).

Command	Effect
P\SEQUENCE\1\1000	Executes the current item of the command sequence in scene 1000 on channel 1. The cursor may be moved depending upon the item that is executed.

Query Commands

The following are commands that can be executed asynchronously to retrieve information about the state of PRIME. These commands execute as soon as they are received and exist outside of the Intelligent Interface Rule Engine. Command and response behavior for queries **are not** customizable.

- **Project List:** P\PROJECT_LIST
- **Scene List:** P\SCENE_LIST
- **Action List:** P\ACTION_LIST
- **Clip List:** P\CLIP_LIST
- **List Assets:** P\LIST_ASSETS
- **Handle Clip Record:** P\CLIP_RECORD
- **Validate Scene Exists:** P\SCENE_EXISTS
- **Validate File Exists:** P\FILE_EXISTS
- **Scene State:** P\SCENE_STATE
- **Validate Assets:** P\VALIDATE_ASSETS
- **Save Preview:** P\CO

Get Project List

P\PROJECT_LIST\\

This command returns a list of project names currently recognized by PRIME. This list is equivalent to viewing the project drop-down in the application user interface. Projects that have not yet been opened in PRIME are still unknown and consequently will not appear in the response.

The response will take the form:

*P\PROJECT_LIST\<Project1Name>\<Project2Name>\\

For example:

*P\PROJECT_LIST\News\Sports\\

The above response identifies two projects ('News' and 'Sports')

Get Scene Name List

P\SCENE_LIST\\

or

P\ SCENE_LIST\<Channel>\\

This command returns a list of scene names within a particular project. If no channel is specified, then this command will operate within the context of the currently selected project. Otherwise the list of scene names will be drawn from the project assigned to the specified channel (SET_PROJECT, page).

If specified, <Channel> must be a number.

The response will take the form:

*P\SCENE_LIST\<Scene1Name>\<Scene2Name>\\

For example:

*P\SCENE_LIST\LowerThird\HeadShot\Scoreboard\\

Get Action List

P\ACTION_LIST\<SceneName>\\

or

P\ACTION_LIST\<Channel>\<SceneName>\\

This command returns a list of action names defined within a particular scene. If no channel is specified, then this command will operate within the context of the currently selected project. Otherwise the scene referenced must exist within the project assigned to the specified channel (SET_PROJECT, page).

If specified, <Channel> must be a number.

<SceneName> may either be a scene name or absolute path to a scene file. In the latter case, the path must use forward slashes instead of back-slashes.

The response will take the form:

*P\ACTION_LIST\<SceneName>\<Action1Name>\<Action2Name>\\

For example:

*P\ACTION_LIST\LowerThird\Default\Effect In\Effect Out\Dissolve\\

Get Clip Name List

P\CLIP_LIST\\

This command returns a list of clip names within the currently active project.

The response will take the form:

*P\CLIP_LIST\<Clip1Name>\<Clip2Name>\\

For example:

*P\ CLIP_LIST\LowerThird\HeadShot\Scoreboard\\

List Assets for Messages and Scenes

P\LIST_ASSETS\Scenes\\

P\LIST_ASSETS\Messages\\

This command returns the current project's scene or message assets. Scene name will be relative path to the Scenes folder of the current project.

The response will take the following forms:

For Scenes:

*P\LIST_ASSETS\<Scene1RelativePath>\<Scene2RelativePath>\\

For Messages:

*P\LIST_ASSETS\<Message1RelativePath>\<Message2RelativePath>\\

Response Examples:

For Scenes:

*P\LIST_ASSETS\L3.pbx\L3/123.pbx\\

For Messages:

*P\LIST_ASSETS\00001112.pbm\NBA/1233421.pbm\\

Command	Effect
P\LIST_ASSETS\Scenes	Returns list of scene assets in the current project
P\LIST_ASSETS\Messages	Returns list of message assets in the current project

Handle Clip Record

P\CLIP_RECORD\<Recorder>\<Command>\<FileName>\<Encoding>\<FrameCount>\\

Issues a record command:

- Recorder: The number of the clip recorder that will be used
- Command: The command to be sent to the recorder: Start, Stop or Still
- FileName: The file name of the clip or image to be recorded. If omitted, or "*", a file name will be automatically generated. The file will be saved to the Clips folder of the current project.
- Encoding: The encoding to be used for recording a clip: Gtc, GtcJpeg, GtcLzo, QuicktimeRle, MotionJpeg, DnxHdHq, DnxHdSq, DnxHdLb, DnxHrHq, DnxHrHqAlpha, DnxHrSq, DnxHrSqAlpha, DnxHrLb, DnxHrLbAlpha. If "*" or omitted, GtcJpeg will be used.
- FrameCount: The number of frames to record on a Start command. If omitted, then recording will occur until a Stop command is received

The response will take the form:

*

Command	Effect
P\CLIP_RECORD\1\Start\Clip_Output\ QuicktimeRle\300	Starts recording on Clip Recorder 1, saving the file to the name Clip_Output, using Quicktime RLE encoding, and saving 300 frames

Validate Scene Exists

P\SCENE_EXISTS\<SceneName>\\

P\SCENE_EXISTS\<ProjectName>\<SceneName>\\

This command verifies whether a scene with the specified name exists in the given project. If the scene exists then this command will return an asterisk to indicate success.

<ProjectName> must by the name of a project. A full project path is not supported.

<SceneName> may either be a scene name or absolute path to a scene file. In the latter case, the path must use forward slashes instead of back-slashes.

Command	Effect
P\SCENE_EXISTS\Samples\2000	Checks to see whether scene 2000 exists in the Samples project. Returns an asterisk if it exists, otherwise an error code 00004190 is returned.

Validate File Exists

P\FILE_EXISTS\<FilePath>\\

Returns "*" if the file path exists on the file system, otherwise returns 00004190. The path must be specified by using forward slashes instead of backslashes.

The response will take the form:

*

Command	Effect
P\File_Exists\Samples\2000	Checks to see whether scene 2000 exists in the Samples project. Returns an asterisk if it exists, otherwise an error code 00004190 is returned.

Get Scene State With Channel

P\SCENE_STATE\<Channel>\<SceneName>\\

This command queries the state of a particular scene on the specified channel.

<Channel> must be a number.

<SceneName> may either be a scene name or absolute path to a scene file. In the latter case, the path must use forward slashes instead of back-slashes.

If the command is malformed, then error code 00005507 will be returned. Otherwise the response will match the following pattern:

*P\SCENE_STATE\<Values>\\

<Values> is a back-slash separated list of states that describe the scene. Possible states include:

State	Description
Closed	The scene exists but is neither on Preview or Program.
Loaded	The scene exists and is currently loaded in Preview.
Playing	The scene exists and is currently playing on Program.
NonExistent	The scene does not exist.

Typically, only a single state will be returned, however two states will be returned if the same scene is used in a back-to-back scenario such that it is present on both Preview and Program at the same time.

Response	Meaning
***P\SCENE_STATE\Closed**	The scene queried exists but is currently closed.
***P\SCENE_STATE\Loaded**	The scene queried is currently loaded in Preview only.
***P\SCENE_STATE\Playing**	The scene queried is currently playing on Program only.
***P\SCENE_STATE\Loaded\Playing**	The scene queried is both loaded and playing.
***P\SCENE_STATE\NonExistent**	The scene queried does not exist.

Only the Loaded and Playing states can be returned together; no other combinations are possible.

Validate Assets

P\VALIDATE_ASSETS:<Buffer>\<Channel>:<Layer>\<SceneName>\\

This command verifies whether all assets referenced by a scene exist or if any are missing.

This command will prefer checking a current instance of the particular scene, using the buffer, channel and layer parameters to isolate the desired instance. However, if an appropriate scene cannot be found, then the scene file on disk will be validated instead.

<Buffer> is optional but must be included if the command should only affect Preview or Program separately. See page for more details.

<Channel> must be a number.

<Layer> is optional but should be specified using the layer expression support defined on page .

<SceneName> may either be a scene name or absolute path to a scene file. In the latter case, the path must use forward slashes instead of back-slashes.

If no assets are missing then the command will return an asterisk to indicate success; otherwise the response will match the following pattern:

*P\VALIDATE_ASSETS\<Values>\\

<Values> is a back-slash separated list of missing asset file paths referenced by the scene. These file paths have been reformatted using forward-slashes. For example:

*P\VALIDATE_ASSETS\C/Image.png\C/Movie.mov\\

The above response indicates that two assets referenced by the scene were missing from disk:

- C\Image.png
- C\Movie.mov

Command	Effect
P\VALIDATE_ASSETS\1\1000	If scene 1000 is currently loaded or playing on channel 1, then this command will verify whether all assets referenced by the scene currently exist. A single asterisk will be returned to indicate success.
P\VALIDATE_ASSETS:Preview\1\1000	If scene 1000 is currently loaded to Preview on channel 1, then this command will verify whether all assets referenced by the scene currently exist. A single asterisk will be returned to indicate success.

Capture Output Commands (Captures Still Image of Preview or Program)

P\CO:<Buffer>\<Channel>\<FileName>\\

Saves an image of the specified channel to a file:

- Buffer: 0 for Program, 1 for Preview
- Channel: Index number of the channel to capture
- FileName: Name or path of file to capture. If only name is provided, file will be saved in a Captures folder within the current project

The response will take the form:

*P\CO\\

Command	Effect
P\CO\{ChannelIndex}	Captures a still of the specified Program Channel
P\CO:{Buffer}\{ChannelIndex}\{Path}	Captures a still of the specified Buffer (0 for Program, 1 for Preview) and Channel

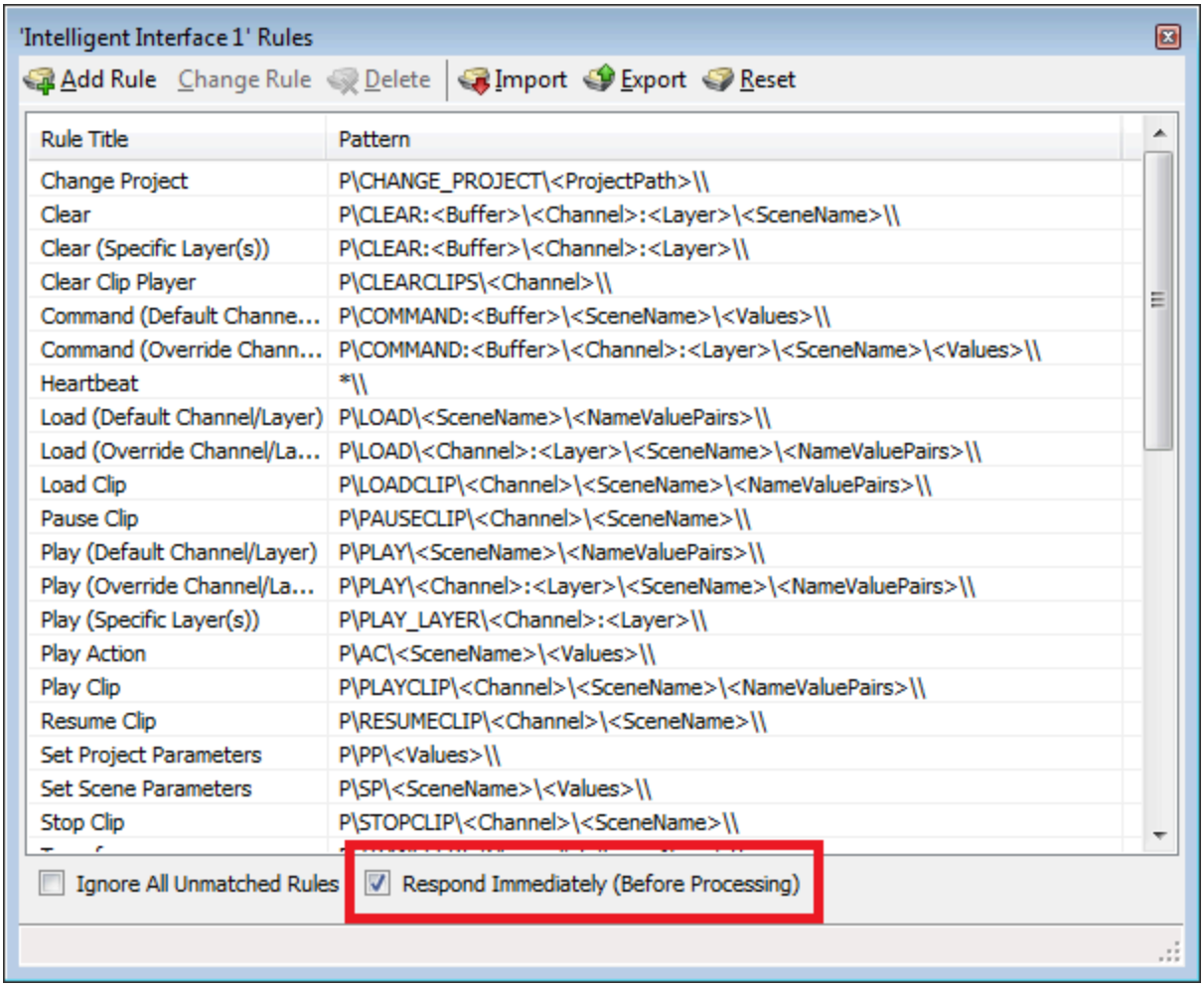
*If you do not specify a channel index {ChannelIndex}, then the active channel will be used

*If you do not specify a path {Path}, the captured file will be saved into a captures folder within the project folder

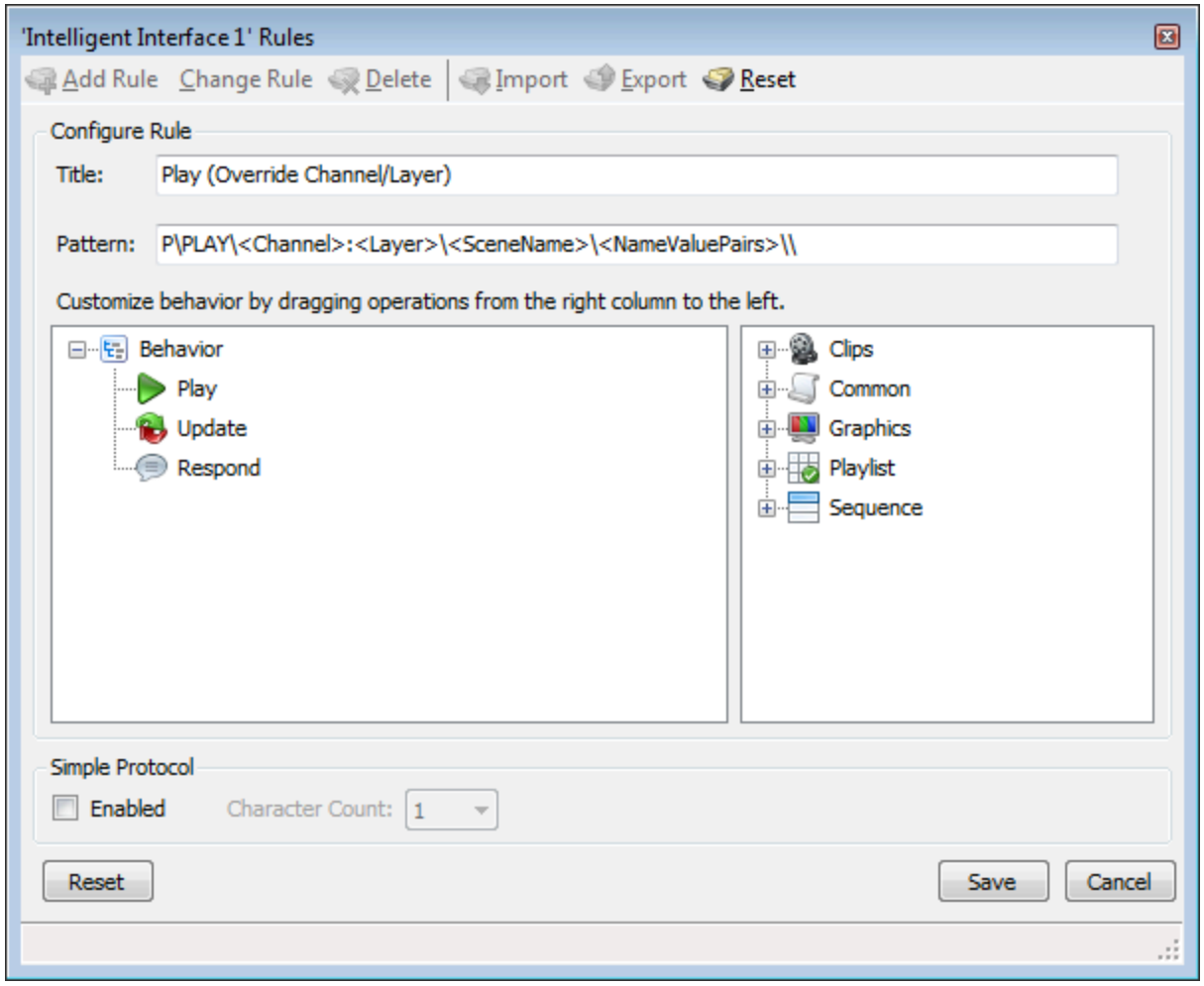
Responses

Intelligent Interface commands that appear within the Rule Engine can be customized in a variety of ways. Depending on the needs of the automation system controlling PRIME, the response may be modified to occur earlier, occur later or to return a customized message altogether.

Most commands that are received are processed sequentially in the order in which they arrive. By default, PRIME will immediately respond to each command regardless of how the individual command is configured within the rule engine. This is a consequence of the "Respond Immediately (Before Processing)" option seen below.

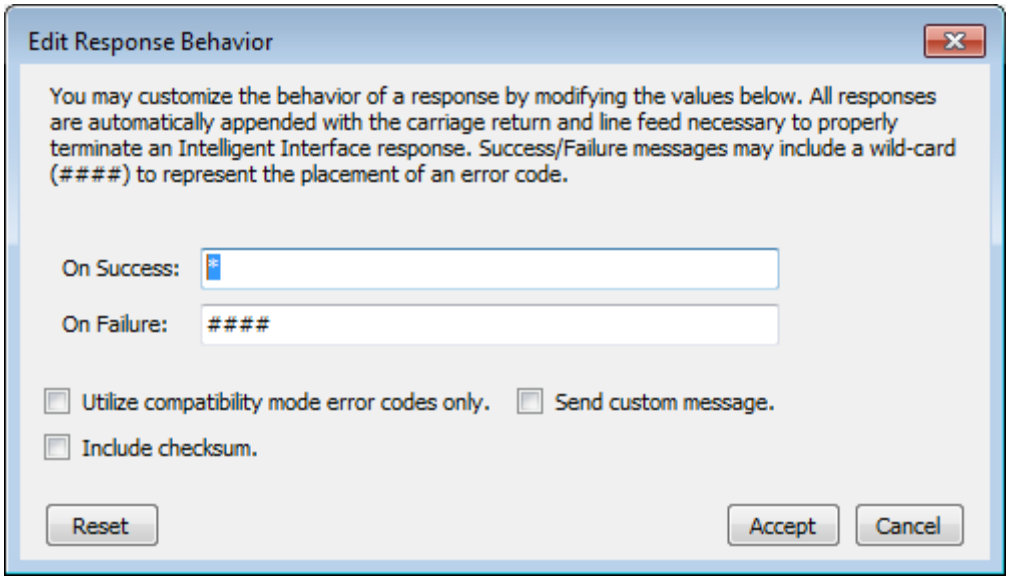


Disabling this option will force the Intelligent Interface connection to observe the order of operations defined within each command rule. For example, double-clicking one of the defined rules will display a configuration window similar to below:



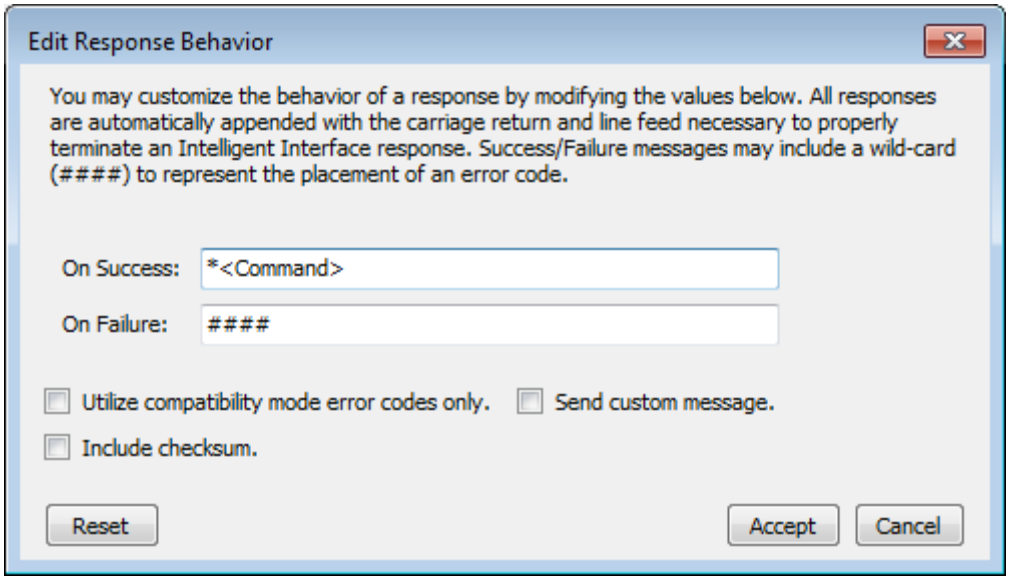
Notice that the **Respond** behavior is listed after the Play and Update. With the "Respond Immediately" option disabled, the above command will first execute the Play and Update prior to transmitting a response to the automation system. Dragging the **Respond** behavior above Play in the list would result in the response getting transmitted earlier in the process.

Furthermore, the actual content of the response can be customized as well. Double-click the Respond behavior to open its configuration.



Here the user may override the default success/failure messages. All of the parameters specified within the command rule can be referenced within these text fields. For example, the P\LOAD command being edited here specified four possible parameters: <Channel>, <Layer>, <SceneName> and <NameValuePairs>. Each of these may be referenced in the response fields above or if so desired a <Command> parameter may be referenced to include the entire incoming command text as part of the response.

For example:



Given the following command and the above response configuration:

P\LOAD\1\1000\Name\Value\\

If loading scene 1000 succeeded, then the response returned would have the form:

*P\LOAD\1\1000\Name\Value\\

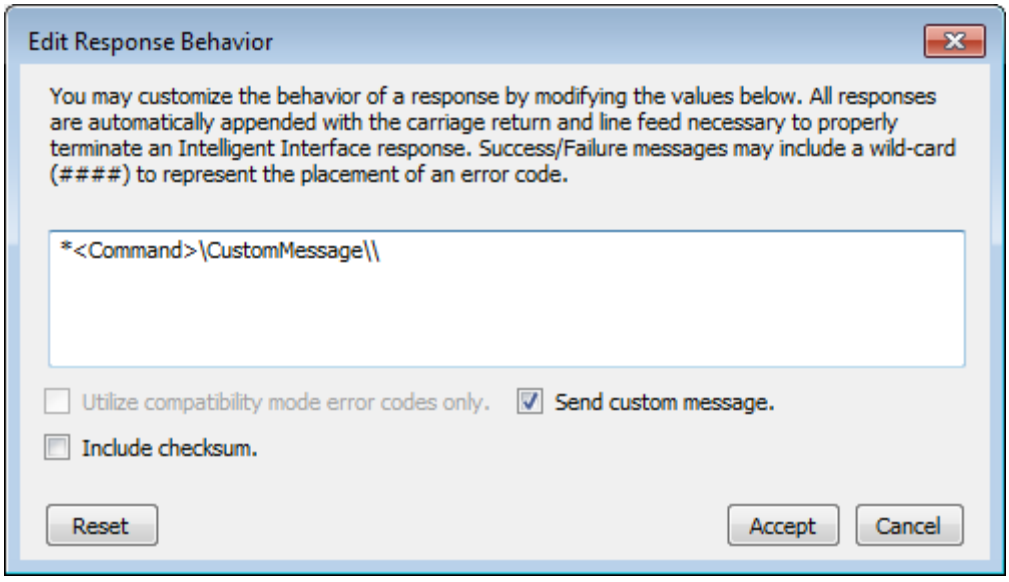
In this case, <Command> contains P\LOAD\1\1000\Name\Value\\

If only a portion of the command received should be used in the response, then individual parameter references can be utilized as seen below:

On Success: *P\LOAD\<Channel>\<SceneName>\\

Command Received	Successful Response
P\LOAD\1\1000\Field1\Value1\Field2\Value2	*P\LOAD\1\1000\\
P\LOAD\2\4000	*P\LOAD\2\4000\\

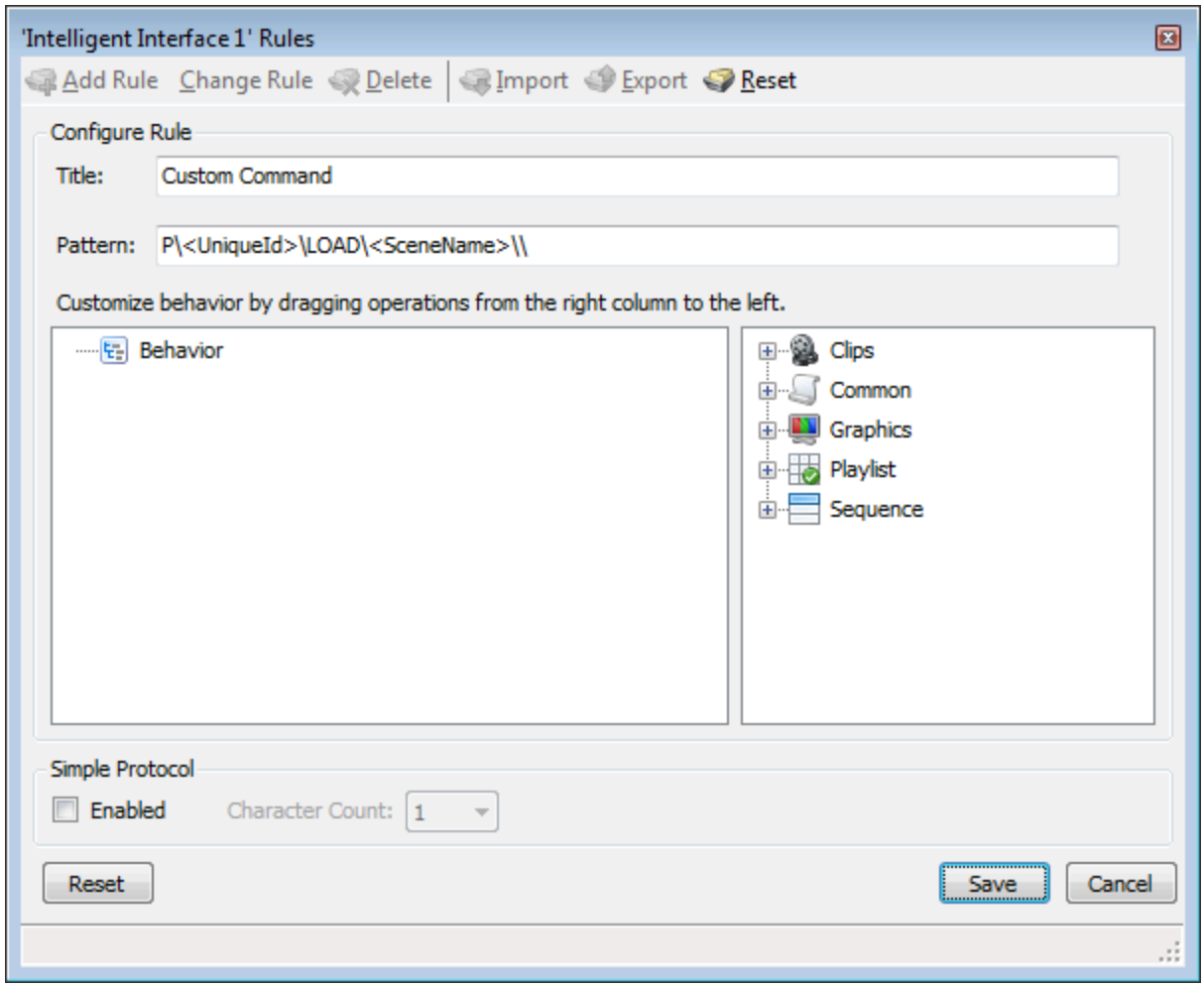
Alternatively, a custom response may be specified to always return regardless of success/failure.



Given the above configuration for the P\LOAD command addressed previously, the response would have taken the form:

*P\LOAD\1\1000\Name\Value\\\CustomMessage\\

Next imagine an entirely custom command. Click the Add Rule button to start with a blank rule. Enter a helpful title and define the pattern that reflects the type of command data that should be processed. Suppose that every automation command should contain a unique identifier that can be used to identify the command. The rule engine allows for the use of custom parameters.



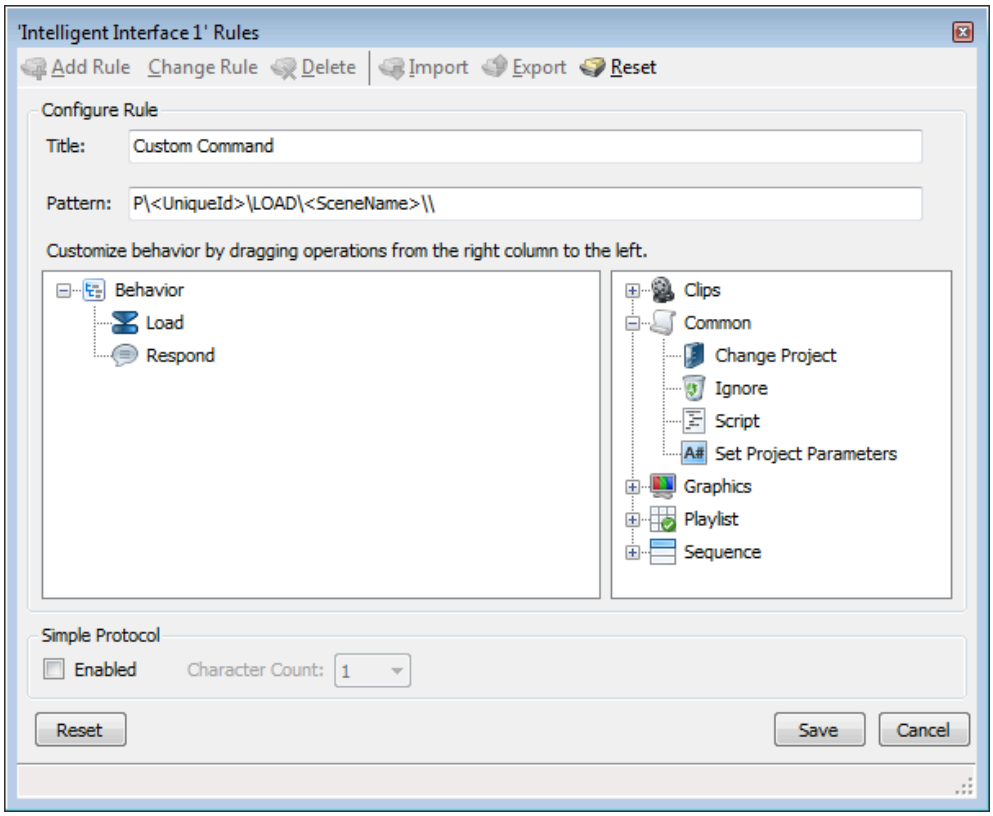
P\<UniqueId>\LOAD\<SceneName>\\

In this example, <UniqueId> is an entirely custom parameter based on user input. It will match any incoming data within a matching command string as long as that data field does not contain backslashes.

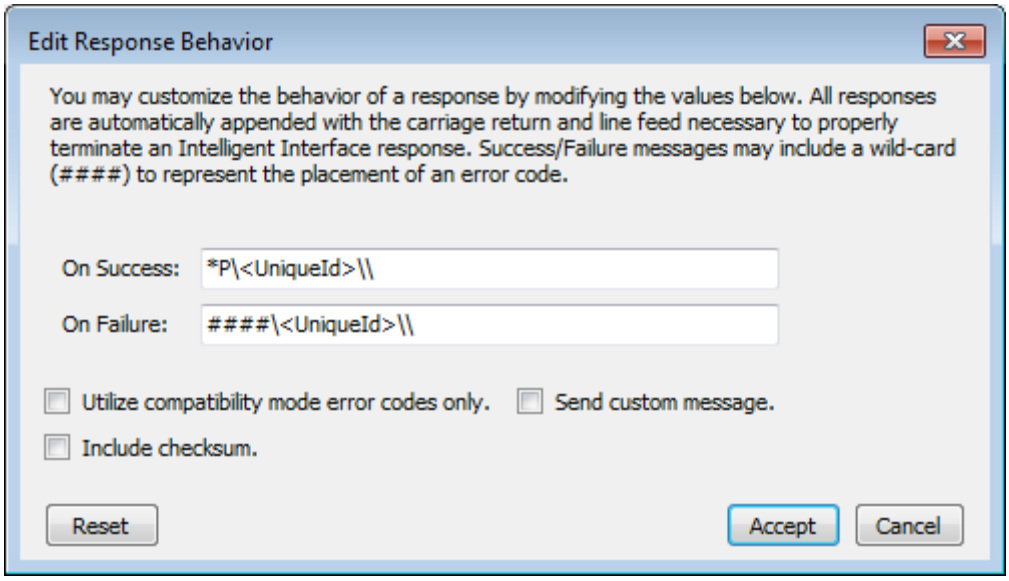
P\123ABC\LOAD\1000\\

<UniqueId> will hold value 123ABC.

Continue configuring the rule by dragging and dropping behaviors that make sense; in this example, the Load and Respond behaviors are all that are needed.



Finally, double-click the Respond behavior to customize the response. Instead of leveraging the entire <Command> parameter in the response, reference only the unique identifier parameter that was defined in the rule pattern.



Now given the command earlier (P\123ABC\LOAD\1000\\)

On success, PRIME will return:

- *P\123ABC\\

On failure, PRIME will return something similar to:

- 00004192\123ABC\\

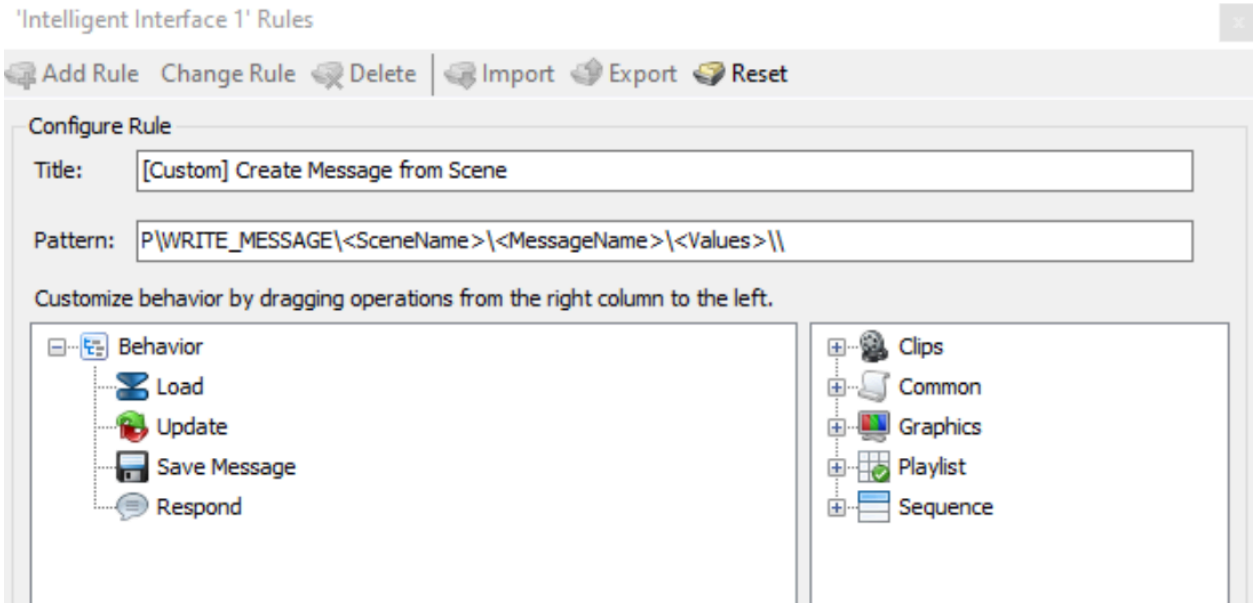
Error Codes

Most commands will return success in the form of a single asterisk; however, failure for P commands is instead denoted using one of the error codes listed below.

Error Code	Description
000040B3	The requested asset could not be found.
00004190	The command is invalid. This is typically returned when the command is properly formatted, but the request itself cannot be fulfilled (e.g. requesting a list of assets for a non-existent file path, etc.).
00004191	The command is malformed.
00004192	An unknown error has occurred. This is the catch-all error that denotes something has gone wrong in the execution of the command.

Absolute File Paths

When sending an automation P command targeting an absolute file path, it is recommended not to include the colon path separator. For example this custom automation command targets the <SceneName>



The valid automation command sent to Prime should be:

P\WRITE_MESSAGE\I/Prime/Projects/Scenes/L3.pbx\100\Test\\

Not: P\WRITE_MESSAGE**I:**/Prime/Projects/Scenes/L3.pbx\100\Test\\

ABOUT US

Chyron is ushering in the next generation of storytelling in the digital age. Founded in 1966, the company pioneered broadcast titling and graphics systems. With a strong foundation built on over 50 years of innovation and efficiency, the name Chyron is synonymous with broadcast graphics. Chyron continues that legacy as a global leader focused on customer-centric broadcast solutions. Today, the company offers production professionals the industry's most comprehensive software portfolio for designing, sharing, and playing live graphics to air with ease. Chyron products are increasingly deployed to empower OTA & OTT workflows and deliver richer, more immersive experiences for audiences and sports fans in the arena, at home, or on the go.

CONTACT SALES

EMEA • North America • Latin America • Asia/Pacific
+1.631.845.2000 • sales@chyron.com

Chyron