

TECHNOTE: USER-DEFINED COMMANDS

When a voice command is recognized, XP Remote sends a command or DataRef value to the ExtPlane plugin to remotely control the aircraft.

In the `Resources/plugins` sub-folder of your X-Plane installation are two text documents, `Commands.txt` and `DataRefs.txt`. These documents contain a list of the standard commands and DataRefs that can be used to control X-Plane. Each aircraft may also have custom commands and DataRefs which can be used to control its proprietary settings.

Lee C. Baker has developed a really useful DataRef tool plugin, see <https://github.com/leecbaker/datareftool>, which can be used to view and set commands and DataRefs within X-Plane itself. You can use this tool to work out which command or DataRef needs to be sent or written, to control a specific switch, button or lever within the cockpit. Armed with this knowledge you can now add a new voice command to XP Remote.

XP Remote's voice commands are defined in XML and the latest commands can be downloaded from our website, see <http://www.planetcoops.com/apps/xp-remote/fags/voicecommands/commands.xml>. If you copy this file to the `XPRemote` folder of your device's internal storage card, XP Remote will load commands from this file when it starts.

Commands are grouped together into profiles within the file. A profile has optional semi-colon separated lists of authors and descriptions which can be used to automatically select the profile based on the currently loaded aircraft. These attributes correspond to the `sim/aircraft/view/acf_author` and `sim/aircraft/view/acf_descrip` DataRef values respectively. If a command cannot be matched within the active profile, then the X-Plane profile is searched. Commands consist of one or more tokens and a JEXL script. When speech input is matched to a command's tokens, its JEXL script is executed. More information on JEXL can be found here, <http://commons.apache.org/proper/commons-jexl/reference/syntax.html>.

A token defines a phrase within a command using a Java regular expression to match speech input, see <https://docs.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html> for more information on regular expression syntax. The special tokens `NUMBER`, `COM_FREQUENCY`, `NAV_FREQUENCY`, `ADF_FREQUENCY`, `DEGREES`, and `IDENTIFIER` are created automatically by XP Remote using the definitions for tokens `"-"`, `"."`, `0-9`, `A-Z`, `"/`, `SPACE`, and `OVERFLY`.

The following methods and variables are available from the JEXL `"cmd"` context to help develop your command script:

To write the value of a DataRef:

```
void setDataRefValue(String dataRefName, int value);
void setDataRefValue(String dataRefName, float value);
void setDataRefValue(String dataRefName, String value);
```

To read the value of a DataRef:

```
float getDataRefValue(String dataRefName);
```

```
float getDataRefValue(String dataRefName, float valueIfNotFound);
```

To send a command:

```
void sendCommand(String command);
void sendCommandBegin(String command);
void sendCommandEnd(String command);
```

To send a key stroke (see

<http://www.xsquawkbox.net/xpsdk/mediawiki/XPLMCommandKeyID> for the list of key IDs)

```
void sendKey(String XPLMCommandKeyID);
```

To write the value of a DataRef array (array indexing starts from 1):

```
void setDataRefArrayValue(String dataRefName, int value); // Sets all indexes
void setDataRefArrayValue(String dataRefName, float value); // Sets all indexes
void setDataRefArrayValue(String dataRefName, int index, int value);
void setDataRefArrayValue(String dataRefName, int index, float value);
void setDataRefArrayValue(String dataRefName, int index, String value);
```

To read the value of a DataRef array (array indexing starts from 1):

```
float getDataRefArrayValue(String dataRefName, int index);
float getDataRefArrayValue(String dataRefName, int index, float valueIfNotFound);
```

To write the value of a DataRef using a mapped value:

```
void setDataRefMappedValue(String dataRefName, Map<Object, Object> map, Number value);
```

To send a FMS/CDU key:

```
void sendFMSKeys(String keys);
void sendFFFMSKeys(String keys); // Flight Factor variant
```

To pause up to 2000 milliseconds:

```
void pause(int milliseconds);
```

To convert between data types:

```
int toIntegerOperand(float value);
float toFloatOperand(int value);
```

Variables:

```
int cmd.integerOperand // NUMBER token as an int
float cmd.floatOperand // NUMBER token as a float
String cmd.stringOperand // NUMBER token as a String
String cmd.identifierOperand // IDENTIFIER token as a String
float cmd.barometricFactor // barometric conversion (inHg/mb from Settings)
```

Please don't hesitate to contact us at support@planetcoops.com if you need further information, help creating your own commands, or would like us to add your commands to the built-in collection.

Planet Coops
March 2018